IMT Institute for Advanced Studies

Lucca, Italy

Application-Aware Image Compression and Sensing Platform for Plant Phenotyping

Ph.D. Program in Computer Science and Engineering

XXVI Cycle

By

Massimo Minervini

2015

The dissertation of Massimo Minervini is approved.

Advisor: Sotirios A. Tsaftaris, IMT Lucca, Italy

The dissertation of Massimo Minervini has been reviewed by:

Prof. Tony Pridmore, University of Nottingham, United Kingdom
Prof. Béatrice Pesquet-Popescu, Télécom ParisTech, France

IMT Institute for Advanced Studies, Lucca
2015

Table of Contents

List of Figures

List of Tables

Vita

April 16, 1984	Born, Terlizzi (BA), Italy
2002–2008	B.Sc. in Computer Science Mark: 110/110 cum laude University of Bari Bari, Italy
2008–2010	M.Sc. in Computer Science Mark: 110/110 cum laude University of Bari Bari, Italy
2014–2015	Research Collaborator Pattern Recognition and Image Analysis Research Unit IMT Institute for Advanced Studies Lucca, Italy

Publications

[1] M. Minervini, S. A. Tsaftaris, "Classification-aware distortion metric for HEVC intra coding," submitted to the *International Conference on Visual Communications and Image Processing*, 2015.

[2] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo Muntión, J.-M. Pape, G. Polder, D. Vukadinovic, X. Yin, S. A. Tsaftaris, "Leaf segmentation in plant phenotyping: A collation study," submitted for publication to *Machine Vision and Applications*, 2015.

[3] M. Minervini, A. Fischbach, H. Scharr, S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," submitted for publication to *Pattern Recognition Letters*, 2015.

[4] M. Minervini, H. Scharr, S. A. Tsaftaris, "The significance of image compression in plant phenotyping applications," *Functional Plant Biology*, 2015, to appear.

[5] M. Minervini, H. Scharr, S. A. Tsaftaris, "Image analysis: The new bottleneck in plant phenotyping," *IEEE Signal Processing Magazine*, vol. 32, 2015.

[6] M. Minervini, C. Rusu, S. A. Tsaftaris, "Computationally efficient data and application driven color transforms for the compression and enhancement of images and video," in *Color Image and Video Enhancement*. Springer, 2015, ch. 12.

[7] M. Minervini, C. Rusu, S. A. Tsaftaris, "Unsupervised and supervised approaches to color space transformation for image coding," in *21st International Conference on Image Processing*, Oct. 2014, pp. 5576–5580.

[8] M. Minervini, M. M. Abdelsamea, S. A. Tsaftaris, "Image-based plant phenotyping with incremental learning and active contours," *Ecological Informatics*, vol. 23, pp. 35–48, Sep. 2014, S. I. on Multimedia in Ecology and Environment.

[9] H. Scharr, M. Minervini, A. Fischbach, S. A. Tsaftaris, "Annotated image datasets of rosette plants," Forschungszentrum Jülich GmbH, Jülich, Germany, Tech. Rep. FZJ-2014-03837, Jul. 2014.

[10] M. Minervini, C. Rusu, M. Damiano, V. Tucci, A. Bifone, A. Gozzi, S. A. Tsaftaris, "Large-scale analysis of neuroimaging data on commercial clouds with content-aware resource allocation strategies," *International Journal of High Performance Computing Applications*, 2014.

[11] M. Minervini, C. Rusu, S. A. Tsaftaris, "Learning computationally efficient approximations of complex image segmentation metrics," in *8th International Symposium on Image and Signal Processing and Analysis*, Sep. 2013, pp. 60–65.

[12] M. Minervini, S. A. Tsaftaris, "Application-aware image compression for low cost and distributed plant phenotyping," in *18th International Conference on Digital Signal Processing*, Jul. 2013, pp 1–6.

[13] M. Minervini, M. Damiano, V. Tucci, A. Bifone, A. Gozzi, S. A. Tsaftaris, "Mouse Neuroimaging Phenotyping in the Cloud," in *3rd International Conference on Image Processing Theory, Tools and Applications*, Oct. 2012, pp. 55–60.

[14] G. Casalino, N. Del Buono, M. Minervini, "Nonnegative matrix factorizations performing object detection and localization," *Applied Computational Intelligence and Soft Computing*, vol. 2012, Jan. 2012.

[15] D. Dell'Agnello, A. M. Fanelli, C. Mencar, M. Minervini, "Serendipitous fuzzy item recommendation with ProfileMatcher," in *Fuzzy Logic and Applications*, vol. 6857, Aug. 2011, pp. 220–227.

[16] M. Minervini, "Serendipity injection in a fuzzy recommender system," in *Abstract Booklet of the First AI*IA Doctoral Consortium*, Dec. 2010.

[17] N. Del Buono, M. Minervini, "Nonnegative matrix factorizations performing object detection," Technical Report 14/10, Department of Mathematics, University of Bari, Bari, Italy, Jul. 2010.

Abstract

Plant phenotyping investigates how a plant's genome, interacting with the environment, affects the phenome (i.e. the observable traits of a plant). Quantitative assessment of phenotypes is central to our quest towards efficient and sustainable agriculture. Image-based approaches to plant phenotyping are gaining momentum and, on par with growing scientific and commercial interest, exciting computer vision problems arise. Currently available solutions for image-based plant phenotyping are either destructive and low-throughput or high-throughput and costly. We propose an affordable solution based on a distributed sensing and analysis framework. Time-lapse sequences of the scene are acquired by affordable sensors (as such, they will have limited computational power and knowledge access). The images are transmitted to the cloud, where high computational resources permit the extraction of fine-grained phenotypic information. For the automated analysis of such images, we develop a multi-channel active contour segmentation with probabilistic priors on plant appearance. To validate our approach we collect two image datasets of growing Arabidopsis plants, portions of which are manually annotated and publicly released. However, the transmission of large volumes of image data necessitates compression to meet bandwidth constraints. After demonstrating that lossy image compression does affect vision-based measurement of plant traits and can jeopardize phenotypic analyses, we investigate application-aware compression strategies on resource-constrained devices to reduce transmission and storage cost of the acquired images without compromising analysis accuracy. The possibility of sharing information between sensor and receiver is exploited: the receiver feeds back to the sensor information to optimize image compression. We inject application knowledge at different levels of the lossy encoding process. The sensor estimates regions of interest within an image and applies different levels of compression to foreground (plants) and background. We also save bits in color representation, using an orthogonal transform with class separation capabilities obtained with supervised learning. Finally, we investigate application-aware distortion metrics for pixel-level classification accuracy, and their implementation in the rate control algorithm of the High Efficiency Video Coding (HEVC) standard. We hope with such an affordable solution to increase adoption of image-based approaches to plant phenotyping by small labs and breeders, and also in developing countries, in pursuance of the democratization of science and technology.

Part I

Introduction and background

1 Introduction

1.1 The importance and evolution of plant phenotyping

Plants have always been a crucial source of food, feed, fiber, and fuel. The domestication of plants and livestock caused a revolution in human evolution: we went from being hunter-gatherers to forming pastoral, rooted communities. Farmers, since that day, plant, collect the seed (or fruit) of their farm, and retain seeds of those plants that exhibited a behavior (essentially a trait) that was better than the average plant in their crop (we use loosely the word crop here to refer to any plant of agricultural interest). This is known collectively as selective breeding, and modern-day breeders still follow the same practice to create new varieties.

Essentially, those traits are the phenotypes we seek after even to this day. A 'phenome' is the set of all phenotypes expressed by an organism (e.g., a plant), resulting from the interaction between its 'genome' (i.e. the

This chapter is partly based on:

- M. Minervini, H. Scharr, S. A. Tsaftaris, "Image analysis: The new bottleneck in plant phenotyping," *IEEE Signal Processing Magazine*, vol. 32, 2015.

genetic material) and the surrounding environment [77, 97, 135]:

$$\textsc{Phenome} = \textsc{Genome} \times \textsc{Environment}$$

Typical plant phenotypes of interest are, e.g., how fast it grows, how it reacts to water or nutrient shortage, how many seeds (or fruits) it produces, its final weight. Depending on the type of farming, different traits are important. On the other hand, typical environmental factors influencing a plant's behavior and performance include temperature, light intensity, humidity, soil, nutrients, disease, stress.

In the (not so distant) past, such collection of phenotypes was largely based on direct experience: the farmer would just observe what was 'different' with a particular plant in his crop and such visual scoring is still an important skill of breeders. As scientific means entered agriculture, and the pioneering work of Mendel on genetics, breeding took a completely different turn.

Nowadays, plant scientists are devoted to identifying how the genotype (i.e. the genetic material) affects the phenotypes of plants and how those traits can be selected and introduced to future varieties. They do this not only working on classical plants and crops (e.g., corn, rice, barley, or soybean) but also relying on model plants (e.g., *Arabidopsis thaliana* [205]) that due to small size and short growth cycle can be used to accelerate phenotype measurement ('phenomics' [135]) and genotype association. Uncovering a gene's exact properties and function ('functional genomics' [25, 132]) is of great practical interest, because important functions can be matched with agronomically important traits, of interest to breeders [88].

Working on model plants (such as Arabidopsis), combined with remarkable advances in genotyping, has revolutionized our understanding of plant biology [87]. The rate of throughput for acquiring genetic information (with sequencing and microarrays) has achieved game-changing levels [195]: modern sequencing mechanisms can produce the sequence (i.e. the genetic information) of a plant within a few days and with minimal cost, in the order of few thousands of dollars. Comparatively, 10 years ago, it took approximately 10 years to sequence the human genome, and cost $3.4B. Once a plant's genome has been fully sequenced (an approach that is already of high throughput), algorithms exist to compare sequences of unknown genes with genes whose function is already known. Following the isolation of the mutated gene, experiments are necessary to screen collections of mutant plants and quantify their phenotype. The actual phenotyping process is extremely time and effort consuming. To

discover valuable agricultural traits (e.g., growth rate, root density, grain size, drought tolerance, product quality, yield potential, etc.), replicated trials need to be carried out across multiple environments over a number of seasons, with a considerable amount of manual work for taking measurements. In addition, many phenotyping techniques are destructive for the plants, i.e. they involve removing parts of the plant or even harvesting early in the life-cycle. Precision and high throughput in genotyping has accelerated the need for precision in phenotyping [30], favoring approaches that provide quantifiable phenotyping information, which could be better used to find associations in the genotype [231].

The combination of novel technologies such as noninvasive imaging, spectroscopy, image analysis, robotics, and high-performance computing has been identified as key to address the so-called 'phenotyping bottleneck' [104]. The development of robust imaging equipment has helped in the acquisition of phenotypes and image-based approaches are gaining attention among plant researchers to measure and study visual phenotypes of plants. In the last decades, there has been a growing interest towards developing solutions for the automated analysis of visually observable traits of the plants [113]: a variety of approaches based on images have been developed to measure such traits in an automated fashion [115, 150, 318] and analyze the image data acquired during experiments [18, 71, 124, 220, 310, 318, 339]. What was previously tedious manual work, typically consisting of measuring actual plants either non-destructively (e.g., to measure leaves with a caliper) or destructively (e.g., to weigh a plant or fruit), it is becoming a matter of collecting a series of imaging data [182, 292]. And as automation advanced, this image collection became even more simplified and higher throughput. However, the task of measuring phenotypes with images remained at large a manual process: instead of working on the plant, scientists are working on its 'digital self', using image analysis tools to measure objects on images [269]. This process until recently remained satisfactory.

However, our way of life is facing significant challenges due to climate change, population increase, change in our diets and need for biofuels (cf. Figure 1.1). In fact, according to the Food and Agriculture Organization of the United Nations (FAO) [37], by 2050 the population is expected to reach 9 billion while arable land would expand by only 5%, with irrigation limited by lack of infrastructure. Crop yields are expected to increase, albeit at a lower pace. Furthermore, different trends of the above within regions and between developing and developed countries, have been

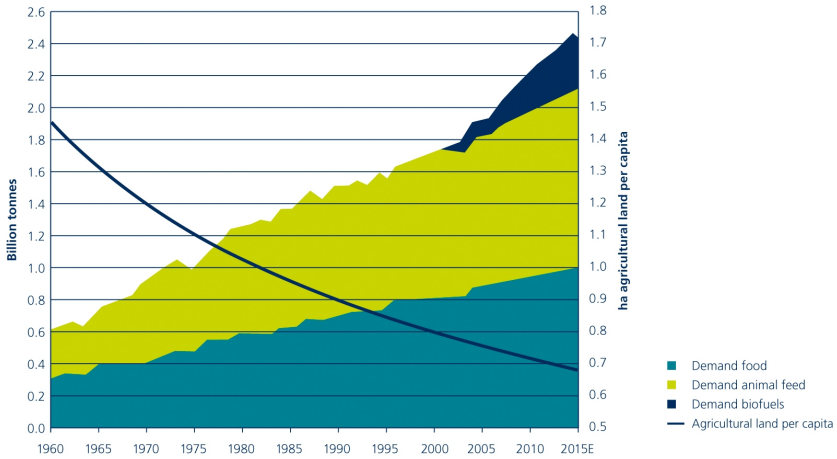


Figure 1.1: Opposing trends of increasing food demand and declining agricultural land per capita are observed between 1960 and 2015, emphasizing the importance of high-performance agriculture and knowledge based bio-economy to face loss of water resources and agricultural land mass, due to climate change, erosion, and the increasing trends of using plants for energy. Figure adapted from [24] (source: WRI, Aquila, Robeco).

observed. The report concludes that we need to increase yield, under the constraints of using less energy, water, and fertilizer, and identifies large scale phenotyping as one of the most promising approaches. Together with the breeding industry, researchers try to identify, improve, and breed key traits to satisfy the growing demands, increase resistance to parasites and diseases, reduce environmental impact (less water, less fertilizer), always striving to a more sustained agriculture [241].

Towards this goal, 1.700 seed banks, where seeds and genetic material are collected, have commenced an effort to identify the diversity of this material, with the hope of finding genotypes that could meet the above constraints tailored to local regional characteristics. Notice, however, that one bank alone may contain hundreds of thousands of different cultivars (i.e. cultivated varieties of a plant created or selected for desirable characteristics and thereafter maintained by cultivation). One possible avenue to identify those, is through large scale experiments under different environmental conditions, to identify which phenotypes exist and which gene(s)

are related, i.e. the so-called genome-wide association studies (see e.g. [17] for Arabidopsis). The complexity of genotype $\times$ environment (including cultivation) produces a tremendous amount of data. Exhaustive search, i.e. experiments covering all genotypes at all environmental conditions, is simply not possible. As experimental scale increases, thousands of plants per experiment are in reach and necessary. Furthermore, phenotyping can be either forward or reverse, where known genotypes (genes, or cultivars) are sieved for valuable traits, or the reverse of this process, were unique traits are observed and the desirable genotype (gene or genes) need to be identified. However, the rate at which phenotypes are extracted in the field or in the lab, is not matching the speed of genotyping and is creating a bottleneck [135].

While previously the bottleneck was in the equipment (the hardware) it is now also the analysis (the software). There is a need to develop accurate, robust and automated analysis algorithms that can extract phenotypic information from experiments on the small (cell) or large scale (field), in 2D or 3D (plus time for dynamics), in the lab but more importantly in the field on real crops. These algorithms should be coupled with affordable sensing platforms and should be capable of dealing with the immense amount of data typically produced in these experiments (requiring in some cases 'Big Data' approaches). The analysis of imaging data currently appears to be the the weakest, or even missing, link due to the major challenges in computer vision and image processing we are currently facing (cf. Chapter 2 for a discussion of image and signal processing challenges arising in the context of plant phenotyping).

1.2 Problem statement

Currently, the majority of phenotype collection systems rely on commercial or several custom-built solutions, ranging from the small scale of growth chambers to the very large scale of automated greenhouses or field applications. The commercial options are costly, requiring an initial investment that few laboratories can afford. Additionally, the hardware infrastructure is usually complemented by a highly integrated and proprietary analysis software; thus, the overall phenotyping solution needs to be used 'as is' when options are vendor locked. While costly phenotyping facilities may represent the best option for large research organizations and offer local high throughput, many labs (particularly also in developing countries)

and small companies (e.g., breeders) have limited financial resources, thus install base throughout the world remains limited.

As an effect of those limitations, many organizations willing to embrace the image-based approach but not appreciating the cost and vendor lock-in of commercial solutions, develop highly customized (hardware and image analysis) solutions tailored to their experimental setting and capable of addressing only specific phenotyping problems. Even when they are affordable, this variability in methods and setups creates standardization problems. Unfortunately, these solutions cannot be easily adopted by other labs with different experimental settings, since adjustments to the complex image pipeline or a redesign are necessary.

Therefore, affordable plant phenotyping solutions are desirable and are expected to increase the capability of investigating plant biology on a global scale, by reducing the entry barrier of high cost. Several laboratories have explored the potential of affordable industrial cameras for plant phenotyping, since imaging can cover a large span of phenotyping interests, in the lab or in the field when combined with affordable carriers (such as quadcopters). The experiences have shown that affordable sensors can suitably replace expensive ones, whilst maintaining adequate accuracy in phenotype measurement. The use of off-the-shelf commercial equipment could facilitate standardization across experiments, lower the entry barrier, offer affordable solutions, and help many labs adopt the image-based approach to plant phenotyping. However, to this day, most approaches remain customized, are not supported by an open community, and are typically focused on a small application range.

In this thesis, we propose a novel phenotyping solution,[1] which aims to provide a universal, turnkey, and modular platform based on distributed sensing and analysis. Figure 1.2 shows an overview of the proposed framework, on which we expand in the rest of the thesis. This distributed approach presents several key advantages. Affordable and easy-to-install sensors can be adopted in laboratories (growth chambers), the greenhouse, or the field, and they are free to move (e.g., when installed on board a tractor or a drone). Multiple affordable sensors can be readily deployed to cover areas larger than the field of view of a single camera, or to obtain higher resolution, before resorting to expensive solutions based on robotics and automation. Besides, outsourcing computation and storage to the cloud relieves the user from the cost of purchasing and maintaining

[1] `http://www.phenotiki.com`

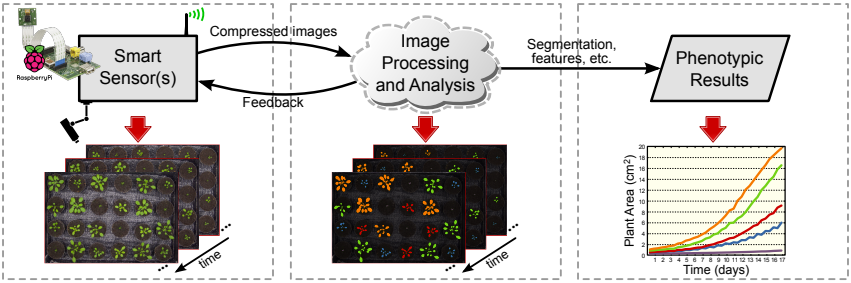


Figure 1.2: Smart and affordable camera sensors (e.g., based on the Raspberry Pi [280]) are deployed on site, and acquire time-lapse image sequences of the scene, including one or multiple plants. In the proposed distributed sensing and analysis framework (cf. Chapter 6), the images are compressed and transmitted to a centralized, remote location (e.g., the cloud) that also acts as an 'oracle'. Leveraging high computational power and a broad knowledge base, the oracle is able to run sophisticated computer vision tasks (e.g., feature extraction, leaf segmentation and tracking, optical flow analysis, 3D reconstructions) and feeds back to the sensor information to improve its ability to compress the images (or other tasks). Relying on web-based graphical user interfaces, phenotyping results are presented to the user for interpretation.

a high-performance computing infrastructure in situ. Importantly, it also permits consistency in experiments among different labs by standardizing equipment and analysis. This centralized design, particularly when combined with an open architecture, can benefit the entire community, providing a modular and expandable architecture (by changing or adding new camera sensors), favoring software reuse (e.g., user-contributed algorithms can be adopted by other labs) and knowledge sharing (e.g., a common repository of acquired data and meta-data, and also the analysis application itself learning on the user's feedback).

Overall, we believe that such a platform could represent an enabling factor to boost global high throughput in plant phenotyping. However, significant challenges are posed since data need to be transmitted over communication channels, and analysis has to occur remotely on compressed images.

1.3 Challenges and scientific contributions

Our proposed solution is based on affordable sensors and image analysis performed on the cloud. Due to the lack of publicly available imaging datasets, we devise an implementation of our sensor and setup a growth chamber to grow Arabidopsis plants and collect imaging data. To analyze the acquired images we investigate plant segmentation methodologies and to ease adoption of our solution we devise an implementation on the cloud. With sensing happening locally and analysis taking place (remotely) on the cloud, efficient transmission of the acquired images to meet bandwidth constraints becomes central, requiring compression approaches that save bits without jeopardizing accuracy of the analysis occurring on the cloud on the compressed images.

The importance of increasing adoption of image-based phenotyping is striking, and emphasizes the need for suitable solutions to the challenges that affordable phenotyping entails. The original contributions of this thesis are twofold: after introduction and background information (Part I), an affordable solution for image acquisition and automated analysis in a plant phenotyping context is presented in Part II. Subsequently, approaches to application-aware image and video compression are investigated in Part III. The thesis is divided into nine chapters, each discussing aspects of the general framework described previously. In the following paragraphs we outline organization of the material, illustrating also the scientific and engineering contributions offered by the thesis.[2]

After an overview of image-based approaches to plant phenotyping, in **Chapter 2** we outline the key role of image processing and computer vision in plant phenotyping. To motivate the involvement of the broad computer vision community in this societally important problem we published this discussion in [214].

To balance cost and reliability constraints, we propose in **Chapter 3** an affordable sensing solution to acquire time-lapse sequences of plants and transmit them to a remote location for storage and analysis (further implementation details are in **Appendix A**). Its use is showcased in two experiments including different genotypes of *Arabidopsis thaliana*. Details on plant material, environmental conditions, and imaging setup and procedures are published in [274]. To validate the approaches presented in this thesis we manually annotate the images collected during these plant

[2]To disseminate research work, parts of this thesis have been published by the author or are currently under consideration for publication.

experiments. As a benefit for the scientific community, we curate and publicly release a comprehensive collection of annotated datasets aimed at the development and evaluation of computer vision algorithms in the context of plant phenotyping, a description of which is under consideration for publication [210].

In **Chapter 4** we discuss issues related to the automated analysis of images in a plant phenotyping context. Focusing on accurate growth estimation, we propose a method (published in [209]) integrating an active contour model and prior knowledge on plant appearance for the segmentation and the analysis of time-lapse plant images from phenotyping experiments in a general laboratory setting, that can adapt to scene variability. To favor adoption of our system, a cloud-based software implementation with a graphical web interface is also released to the scientific community. Additionally, results of leaf segmentation and counting are showcased, demonstrating the current state of the art. Leaf segmentation results stem from the Leaf Segmentation Challenge (LSC) organized in 2014, a description of which is under consideration for publication [275].

Affordability and remote processing, however, pose technical challenges. The choice of optics and the fixed field of view restrict the quality (in resolution and sharpness) of the acquired images and the plants this setup can image (e.g., it may not be suitable for not co-planar plants). An affordable sensor will have limited computational power and knowledge access, thus, it requires low complexity algorithms to perform vision and information extraction tasks, and as such remote processing is necessary. Then, the transmission of (possibly) large volumes of image data from the sensor to the cloud necessitates lossy compression to meet bandwidth constraints.

However, this loss of information will affect the accuracy of vision tasks occurring at the analysis. It is demonstrated empirically in **Chapter 5** (based on several proof-of-concept experiments) that lossy image compression, albeit necessary to cope with the amount of imaging data at hand, does affect vision-based measurement fidelity, and errors introduced in plant trait extraction and analysis can be high. Hence, we identify via thorough comparisons of state-of-the-art approaches suitable compression options for a variety of typical phenotyping applications. This work is currently under consideration for publication [215].

As the images acquired in an automated phenotyping context are not meant for the human visual system, general-purpose compression standards may not guarantee best results. Therefore, in Chapters 6 to 9

we investigate application-aware compression methodologies to optimize and tailor compression of the acquired images occurring at the sensor to the specific needs of the application that will use them.

In **Chapter 6** we expand on the description of our distributed sensing and analysis framework, and discuss the concept of application-aware image compression. Next, in **Chapter 7** we propose an application-aware image compression approach (published in [216]) based on low-complexity plant detection and a JPEG 2000 encoder capable of region-of-interest coding, that exploits feedback from the receiver to the sensor. To tune compression parameters at the sensor according to plant segmentation accuracy, we explore in [211] the possibility of approximating reliable and yet computationally complex image segmentation metrics such as the Hausdorff distance, based on a linear regression framework to learn from the data a sparse model and using as features simpler to compute metrics.

Color images are central to several plant phenotyping applications. To save bits in the color space representation, in **Chapter 8** we describe two frameworks to obtain linear maps of the RGB color data that minimize the loss of information due to compression. The first is a new data-dependent color transform (termed aKLT), that achieves compression performance comparable to the Karhunen-Loève transform (KLT) at a fraction of the computational complexity and is suitable for adoption on resource-constrained devices. The second is an application-dependent color transform that aims at maximizing post-compression classifier performance, and is obtained with supervised learning methods by optimizing the Fisher discrimination criterion. This work is published in [212, 213].

A new application-aware distortion metric for pixel-level classification accuracy is defined in **Chapter 9**. The proposed metric is implemented in the rate-distortion optimization framework of the recent HEVC (High Efficiency Video Coding) standard, and optimal model parameters for the λ-domain rate control algorithm are derived by curve fitting procedures. This work is currently under consideration for publication [217].

Finally, in **Chapter 10** we summarize the findings presented throughout thesis, and offer concluding remarks and possible directions for future work.

2 Background and literature review

2.1 State-of-the-art in image-based plant phenotyping

Plant phenotyping can occur on small scale (controlled laboratory settings of a growth chamber), in the greenhouse, or in the field [96, 320, 331]. While each setup aims to address different experimental questions, the majority of early stage experiments occur in growth chambers that offer controlled laboratory settings on a small scale using model plants.

A common framework can be identified, which is the key to the interpretation of all image-based plant phenotyping solutions designed in the last decade. A typical imaging pipeline for plant phenotyping involves the following steps:

This chapter is partly based on:

- M. Minervini, M. M. Abdelsamea, S. A. Tsaftaris, "Image-based plant phenotyping with incremental learning and active contours," *Ecological Informatics*, vol. 23, pp. 35–48, Sep. 2014, Special Issue on Multimedia in Ecology and Environment.
- M. Minervini, H. Scharr, S. A. Tsaftaris, "Image analysis: The new bottleneck in plant phenotyping," *IEEE Signal Processing Magazine*, vol. 32, 2015.

1. 'sensing' the plants in order to periodically acquire images, which are stored or transmitted to a processing unit (e.g., a workstation);
2. plants are identified in an image as regions of interest (there can be more than one plant in each image);
3. single plants are then extracted, usually by means of image segmentation techniques, and each plant is identified, in order to follow its development throughout the experiment, which can last days or weeks;
4. finally, relevant measures are taken, such as plant size, color, leaf count, etc.

Performing such analysis for each plant in a period of weeks, generates a large amount of data, that can be analyzed and mined, in order to discover growth patterns and study response to treatments and external stimuli.

Among plant phenotyping techniques, a sharp distinction exists between destructive and nondestructive approaches. Destructive approaches are such, in the sense that leaves or other parts of the plant need to be removed in order to be analyzed, or the whole seedling has to be prematurely harvested. Techniques that belong to this class are usually cheap and simple to implement, since they rely on manual labor to take the samples and, for instance, use a scanner to acquire digital images to be analyzed by some software tool. On the contrary, nondestructive approaches involve much more elaborate hardware and software, as the goal is to minimize the human intervention, while injecting the highest degree of automation. The current research trends are more and more in the direction of noninvasive and fully automated systems, which are capable of both growing the plants and registering all needed phenotyping traits.

In the following paragraphs, an overview is presented of recent noteworthy solutions for growth chamber phenotype acquisition and image processing approaches towards extracting phenotypes.

2.1.1 Phenotype acquisition

Until recently, due to lack of acquisition systems, phenotype analysis was usually performed by destructive techniques that required harvesting of whole plants or plant parts at regular time points. This broke down the continuity of experiments and required large numbers of replicates.

Alternatively, on site surveys were performed by experts without destruction of the plant(s). For example, to assess plant growth users will either weight excised leaves or plants, or will use a caliper to get measurements nondestructively of plant leaves. To decrease the chance of error (due to measurement or biological noise) several repeated measurements were necessary. Nevertheless, both methods required specific measurements on each sample to be performed by a human, leading to high productivity loss and inherently low throughput. As imaging systems evolve (e.g., digital cameras, flatbed scanners) several users opt for capturing images of plants rather than relying on manual measurements obtained in nondestructive or destructive ways.

The introduction of automation and digital imaging allows the rapid collection of time-lapse images of plants in a nondestructive fashion [173]. These images are later analyzed offline mostly using expert analysis by manipulating photograph analysis software or semi-automated methods [173]. This has a particular impact on small scale experiments of model plants in growth chambers and several approaches have been proposed throughout the years. Most of the solutions are customized [15, 115, 173, 243, 306, 318], imposing strict experimental setups (e.g., uniform black background). Their cost and implementation complexity can range from a few hundred dollars [124, 165, 306] to a few thousands [115, 243, 310, 318], or to hundreds of thousands of dollars (e.g., the solutions provided by commercial entities such as LemnaTec,[1] CropDesign[2] [263], and Photon Systems Instruments[3]). While the big investment entailed by commercial solutions is warranted, e.g., if to be offered as a service (i.e. where plant material of many labs is phenotyped) or for very large experimental studies, install base throughout the world remains limited.

High resolution images can also be captured in the visible and/or infrared spectrum. Optionally, also 3D depth information is acquired using high grade [6, 310] or lower grade (e.g., using Microsoft's Kinect) hardware [56, 182]. However, their limited field of view requires that images are acquired with each plant in isolation and usually involves some robotic (or manual) placement of the plant (e.g., [124]) or the camera (e.g., [115, 318]), which lowers throughput or increases equipment cost.

On the other hand, some approaches assume static cameras with a fixed field of view [71, 124, 165, 306] and are largely used in growth-chamber

[1] http://www.lemnatec.com
[2] http://www.cropdesign.com
[3] http://www.psi.cz

experiments of small model plants such as Arabidopsis. Such systems are simpler to deploy, are more affordable (since they do not use automation), and have higher throughput since they can image many plants at a time. However, due to the fixed camera, the per plant imaging resolution can be lower, and only few imaging angles are obtained (usually only a top-view is used), thus requiring sophisticated image analysis algorithms.

Overall, current image-based plant phenotyping solutions tightly couple sensing and analysis: plant material, imaging equipment, and computational infrastructure reside in the same location. Albeit simpler in design, this approach presents major limitations: (a) it limits practical applications of image-based plant phenotyping (e.g., experiments taking place in the field or in remote locations would require an entirely different design); (b) image analysis software is integrated with the specific imaging setup and is typically proprietary and closed source; (c) cost of the hardware, accounting for a large part of the overall cost, is determined by the decision of the manufacturer (e.g., type and quality of imaging equipment, robotics). To overcome these limitations, in Chapter 6 we introduce our proposed distributed sensing and analysis framework for image-based plant phenotyping.

2.1.2 Image analysis for phenotyping

Plant segmentation

One undoubted effect of automation is the increase in the available data for analysis, which as expected called for automation also in the analysis component. In the following paragraphs we briefly describe approaches aiming at segmenting (i.e. delineating) a plant from the background, which in turn is used to measure plant growth or other color and shape traits.

To appreciate the segmentation problem at hand, Figure 2.1 shows the variability of appearance and shape: (i) among several mutants of the Arabidopsis family, (ii) through the life-cycle of the same plant, and (iii) within individuals of the same genotype. Although several other methods have been developed for the segmentation (and classification) of individual leaves [44, 70, 298] (e.g., for plant identification purposes [13, 110]), the analysis of plant vein structure [76,253], and the study of root or hypocotyl growth [102,322], here we focus on those segmenting the whole plant.

The majority of articles in plant phenotyping, since they assume strict conditions as to the composition of the scene, follow simplified approaches

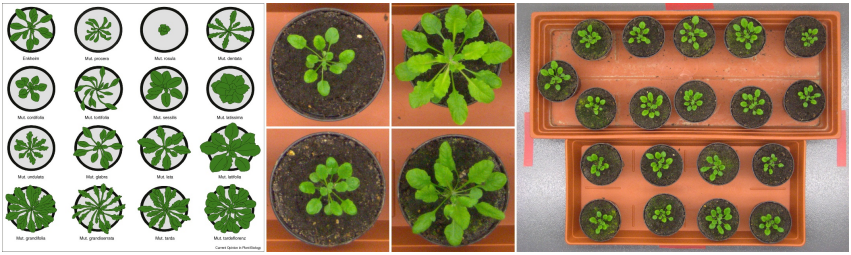


Figure 2.1: Arabidopsis images showing shape variability. Left: sketch representations of Arabidopsis mutants adapted from [208]. Center: two plants imaged at different growth stages, earlier shown left-most. Right: a top-view image showing many individual plants (all wild type, Col-0).

to the subsequent segmentation problem. They usually employ thresholding based approaches in one channel (e.g., the native green channel of an RGB acquisition or on one channel of other color spaces) or multiple channels. Those with strict restrictions on the scene rely either on calibrated thresholds [124], or on histogram based methods such as Otsu's thersholding or Gaussian mixture modeling of intensities to identify data driven thresholds [71, 170]. Some methods rely on combining more than one channel indirectly (e.g., in [71] individual segmentations from each channel obtained via thresholding are combined) or directly via unsupervised clustering methods [61, 306]. A learning approach for weed classification in field applications is proposed by Haug et al. [125], based on a random forest classifier and Markov random field smoothing. However, changes in the scene can pose a challenge. Such changes occur frequently as Figure 2.2 illustrates. For example, the soil can appear more or less dry, moss (which is green in appearance) can grow on the soil, reflections due to water and illumination distortions can be also present. While some users will remove such data from analysis, this lowers the number of instances used for analysis and consequently the statistical power of the experiment. Furthermore, approaches such as scratching the soil to remove out moss may change a plant's position, breaking the location correspondence from previous images assumed by most approaches.

From an imaging perspective, few systems utilize additional information from other imaging sources or modalities (such as depth or infrared information) to facilitate the segmentation approach [6, 56, 310]. For example, using color in combination with depth images or multiple

Figure 2.2: Examples of challenging images due to changing conditions during an experiment. Examples shown are: (left) drought where the soil appears more brown; (center) green moss growing on the soil; and (right) water residual in the tray causing reflections.

images for supervised or unsupervised plant segmentation is common practice [6,29,152,257,285,290,299,321]. However, few laboratories have the capacity to deploy such sensing technology due to its higher cost and lower (in some cases) throughput.

On the other hand, one approach towards dealing with complex backgrounds and changes in the scene is to introduce prior knowledge into the segmentation approach itself. Although naturally one would consider to introduce shape priors in deformable models and contour formulations [66,340], it would require defining and learning shape priors of all possible plant shapes. As Figure 2.1 illustrates, the complexity of shape even among mutants of the same plant or among samples of the same plant species can be significant. Thus, building appropriate plant models is rather complex and learning such shape models would require a large amount of previously labeled information; as a result, no such method exists in the literature.

Individual leaf segmentation

An alternative and more fine-grained approach towards growth estimation would be to rely instead on leaf models, and try to segment individual leaves within the scene. At first glance the problem of leaf segmentation appears similar to leaf identification and isolated leaf segmentation. Research on the latter has been motivated by several datasets showing leaves in isolation cut from plants and imaged individually, or showing leaves on the plant but with a leaf encompassing a large field of view (e.g., emulating imaging via a smart phone application). This problem has been addressed in an unsupervised [289, 328], shape-based [43], and interactive [40, 43] fashion. For example, Felzenszwalb [93] represents shapes by deformable triangulated polygons to detect precisely described objects, including maple leaves. In simplifying the shape descriptor, the authors in [44] propose a parametric active polygon model to take advantage of prior knowledge on leaf shapes to design a flexible time-efficient model, in a classification and leaf recognition problem. Although these methods are capable of segmenting individual leaves in isolation, they require prior training and large labeled datasets, usually they cannot handle occlusions, and they would require significant post processing to compose the final complete plant segmentation.

However, the problem at hand is radically different. The goal is not to identify the plant species (usually known in this context) but to segment accurately each leaf in an image showing a plant (cf. Figure 3.7 on page 46). This multi-instance segmentation problem is extremely complex in the context of this application, due to the variability in shape, pose, and appearance of leaves, but also due to lack of clearly discernible boundaries among overlapping leaves with typical imaging conditions where a top-view fixed camera is used.

Several authors have considered leaf segmentation in a tracking context, where temporal information is available. For example, Yin et al. [339] segment and track the leaves of Arabidopsis in fluorescence images using a Chamfer-derived energy functional to match available segmented leaf templates to unseen data. Dellen et al. [74] use temporal information in a graph based formulation to segment and track leaves in a high spatial and temporal resolution sequence of tobacco plants. Aksoy et al. [4] track leaves over time, merging segments derived by superparametric clustering by exploiting angular stability of leaves. De Vylder et al. [70] propose a probabilistic parametric active contour formulation that optimizes an

energy function by maximizing the probability that the contour is on the edge of a leaf, and use it to segment and track Arabidopsis leaves in time-lapse fluorescence images.

Even in the general computer vision literature, this type of similar appearance, multi-instance problem is not well explored. Although several interactive approaches exist [114, 226], user interaction inherently limits throughput. Therefore, here we discuss several interesting learning-based object segmentation approaches, which might be adaptable to leaf segmentation. Wu and Nevatia [335] present an approach that detects and segments multiple, partially occluded objects in images, relying on a learned, boosted whole-object segmentor and several part detectors. Given a new image, pixels showing part responses are extracted and a joint likelihood estimation inclusive of inter-object occlusion reasoning is maximized to obtain final segmentations. Notably, they test their approach on classical pedestrians datasets, where appearance and size variation does exist, so in leaf segmentation where neighboring leaves are somewhat similar this approach might yield less appealing results. Another interesting work [266] relies on Hough voting to jointly detect and segment objects. Interestingly, beyond pedestrian datasets they also use a dataset of house windows where appearance and scale variation is high (as is common also in leaves), but they do not overlap. Pape and Klukas [239] rely on unsupervised clustering and distance maps to segment leaves. Finally, graphical methods have also been applied to resolve and segment overlapping objects [127], and were tested on datasets of horses.

Till now, the evaluation and development of leaf segmentation algorithms using a common reference dataset of individual images without temporal information is lacking, and the datasets described in Chapter 3 aim at filling this gap.

2.2 Computer vision challenges in plant phenotyping

Noninvasive plant investigations are done on different scales and modalities using a variety of sensors [77,182]. This includes optical, hyperspectral imaging to reveal rich pixel information on plant properties, and even magnetic resonance imaging (MRI) and positron emission tomography (PET). Spatial scales vary from the microscopic subcellular level to large outdoor fields. Typical problems in measuring a plant's visible properties

comprise measuring size, shape, and other structural traits of whole plants, their organs (leaves, fruits, roots, etc.), or plant populations.

Plants are not static, but self-changing systems with complexity in shape and appearance increasing over time. They emerge below image resolution, and grow exponentially in time until, for a single leaf, growth levels off typically at several cm^2 size, i.e. several orders of magnitude change. Relevant time scales for cellular processes may be seconds or minutes, for growing leaves in the range of hours, and the status of whole plants changes over days or even months, in which the surrounding environmental (as well as measurement) conditions may also change. Environmental conditions add yet another dimension, not only to the plants, but also to imaging. In some cases images may be acquired under controlled conditions, but often are taken in challenging environments occurring in the field, in greenhouses, employing automated acquisition setups. Structural break due to resolution limitations (too small leaves) or due to spontaneous biological events (e.g., cell division) exist and algorithms need to deal with leaves (or new cells) emerging within the scene. Finally, the plant is a complex and adaptive organism: any attempt to measure phenotype must respect the complex biophysiological system of the plant and should not interfere or alter its behavior. For example, if the plant expects a dark environment (e.g., night conditions), we should not use lights to image it, because this may influence the circadian rhythm as the photo-systems of plants are highly sensitive.

2.2.1 From the microscopic to the macroscopic: Challenges and dimensions

Algorithms must deal with the above complexity and the next paragraphs describe unique challenges by illustrating typical applications. Clearly, the list of applications can never be complete, as in principle phenotypic traits can be derived for every single detail known in plant physiology, but we present some of the major themes, together with different versions of the same theme arising from using different imaging modalities.

Cells and organs: Detection, tracking, and structural breaks

One of the earliest forms of phenotyping where imaging based setups were used, is in the context of microscopy [77]. Plant tissue samples are excised from roots, leaves, etc. and imaged in a microscope to reveal the

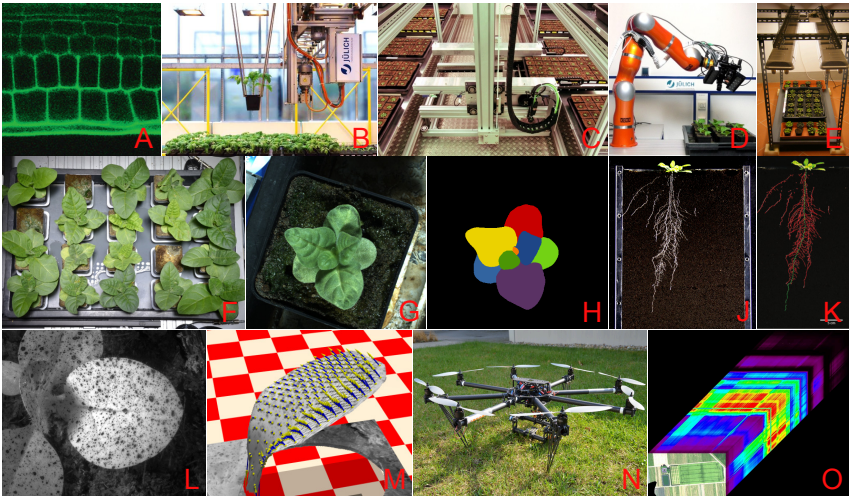


Figure 2.3: Example plant phenotyping setups and images. At the smallest scale, microscopy can image cells (A). Movable imaging setups (B, C, D) or a single overview camera setup (E) can be used to image many plants (F) or single plants (G). Roots can be imaged in rhizotrons (J), requiring delineation (K). Leaf segmentation (H) and optical flow tracking (L, M) can measure growth. Airborne vehicles, e.g. drones (N), can provide information on fields, e.g. hyperspectral images (O). Image (A) adapted from [254]; other images courtesy of: (B, C, N) Alexander Putz, (J, K) Kerstin Nagel, and (O) Uwe Rascher.

cellular structure of the underlying tissue (cf. Figure 2.3A). Typical need from an image processing perspective is the automated delineation of cell walls to establish cell morphology and cell count. Early on techniques relying on intensity or edge based segmentation approaches were used, while more recently advancements in connected snakes and level sets have been introduced to solve this problem in a more reliable and automated fashion.

However, more interesting problems arise from the use of recent techniques such as confocal microscopy, optical projection tomography and coherence microscopy (OPT and OCM), which permit the noninvasive quantification of cellular morphometry at a variety of scales and depths. These techniques enable observation of plant tissue dynamics on a short

(and long) time scale, so problems of tracking arise. These become particularly challenging when cell genesis needs to be observed and quantified, since cell division and expansion impose high spatio-temporal fidelity requirements. This problem, which occurs also in other biomedical applications, from a computer vision perspective entails the inference of time and location of when/where such events occur within the scene, a task radically different from the typical tracking of objects entering or leaving the scene.

As the resolution of these techniques increases the amount of data produced increases as well, however, it does not reach the high throughput requirements of other applications since usually few plants are studied and their preparation does require significant amount of expert handling. Until automation replaces the expert, in cell-level investigations the bottleneck will remain in the sample preparation and imaging steps and not in the image analysis.

Organ level: Detection and segmentation

Over the last decade, several controlled setups have emerged, that image top-down views of small rosette plants (referring to the circular and radial cluster of leaves they form when growing), such as e.g., Arabidopsis or tobacco, acquiring either one plant per image or several plants at once (see Figures 2.3B–2.3G), housed in so-called growth chambers where environmental conditions are controlled. Even in this very restricted imaging scenario fully automatic segmentation of single plants can be a challenge due to, e.g., background clutter due to moss growing on the soil, plant-to-plant overlap, heavy contrast changes due to self-shadowing, leaf color changes due to stress (e.g., drought), different light conditions, and pathogen infections, plant shape or size variation due to genotypic differences (cultivars or mutants) and treatments.

Segmenting single leaves is a typical multi-instance segmentation task (see Figures 2.3G, and 2.3H). However, even though all the objects share a wide range of features (e.g., they are mostly green with similar brightness distributions), they show rich variation in shape and pose due to environmental influences or genotype, even for the same species. Leaves differ in size over several orders of magnitude, introducing a structural break due to resolution limitations, and algorithms need to deal with leaves emerging in the scene.

In addition, leaves vary in shape, and while they do share a certain ba-

sic shape, they overlap, bend, and vary in pose. Even for the same species, leaves may differ substantially, as leaf shape, size, color and overall appearance of a plant depend on the genotype (e.g., there are thousands of mutants available for Arabidopsis alone), environmental factors (drought, low or high light, and temperature), and the age of each leaf. Readily apparent approaches based on learning shape from a labeled dataset reveal their limitations when having to deal with such shape diversity and different acquisition conditions. While counting and segmenting leaves from such images can be simple for a human, there is no automated algorithmic solution available yet, which comes close to human performance.

Whole plants: Anatomically correct 3D geometric modeling

For larger plants reconstruction from a single image and viewpoint is not sufficient. Most approaches aim at obtaining an as complete 3D shape reconstruction as possible, modeling geometrically the overall above-ground part of a plant, i.e. the shoot. However, also details of parts are investigated like grains on an ear of corn, berries on wine grapes, flower development, early symptoms of wilting or pest infestation, multi-plant canopy properties (e.g., for light distribution simulation within a field), etc. Imaging becomes more and more automated using conveyor-belt or robotized systems (see Figures 2.3B–2.3D), allowing high throughput with thousands of plants. Automation of image analysis is then a must.

A variety of 3D measuring strategies is currently investigated, e.g., correspondence-based triangulation methods, silhouette-based carving, time-of-flight cameras or light detection and ranging (LiDAR) laser scanning (see [250] for a comprehensive overview). Setups are usually tailored to a particular species and conditions. This is for example due to size and image resolution constraints, flexibility and deformability of a plant when moved, or self-occlusion and self-similarity hampering triangulation.

Silhouette-based methods find a plants silhouette and project it back into the 3D volume – often represented by a voxel grid – where the plant shall be reconstructed. Intersecting the volumes refines the thus derived 3D model. Correspondence-based methods try to find similar image patches or feature points in images taken from different viewpoints. This allows triangulating 3D points, resulting in 3D point clouds. When visible structure on leaves is too shallow, combining cameras with a pattern projector increasing local image contrast can help. Still, for more complex plants self-occlusion limits triangulation, as ensuring that patches are

visible from two or more viewpoints requires more and more viewpoints. Time-of-Flight cameras or LiDAR laser scanning yield point clouds independent of visual structure and without triangulation. Current drawbacks are either long acquisition times, high cost, or limited resolution.

A major challenge for all 3D measuring methods is plant motion during acquisition. Especially for outdoor measurements with wind, time delays due to scanning or sequential image acquisition lead to notable geometric distortions. Then the data cannot be described by a static model and all current approaches doing so fail one way or another.

From the 3D data, quantitative information about plant traits need to be extracted. Simple summary traits, such as covered volume or plant height, could be estimated from images alone without 3D reconstruction. But organ-wise traits, e.g., accurate leaf size or branching angle, require interpretation of 3D data and plant part models. Simple models are used today (e.g., fitting 2D surfaces to patches and merging them), but for most species new anatomically correct models are required.

Where imaging by acquiring light with cameras is not sufficient or possible, volumetric imaging techniques can be applied. Magnetic Resonance Imaging (MRI) and x-ray Computed Tomography (CT) are the most common modalities on plant shoot scale, and are sometimes combined with Positron Emission Tomography (PET) in order to investigate metabolic processes and carbon transport. Their main strength, however, is root system analysis in soil, where classical, light-based imaging techniques are extremely limited.

Whole plants below ground: Cluttered images of roots

It is not possible to look through soil by naked eyes. Thus, classical root system analysis is invasive, meaning that plants are dug out, the roots washed and photographed or scanned using flatbed scanners – a destructive and low-throughput process. Usual image analysis then applies threshold-based segmentation, connected component labeling, and skeletonization, followed by estimation of traits such as overall graph length, branching angles and others. All solutions available to date have only limited effectiveness when root systems are heavily entangled. Obviously, no time-series analysis can be performed when plants are dug out.

Noninvasive methods using standard camera imaging require other substrates than opaque soil. Well-established are aeroponics (growing in air or mist), hydroponics (water), transparent agar or gellan gum filled

Petri dishes or tubes, or paper or other flat surfaces kept wet with nutrient solution. In aeroponics or hydroponics root systems are not well formed and roots stick together. In transparent agar or gellan gum root systems are well visible and can be well segmented or even be reconstructed in 3D when imaged from multiple views. However, roots grown in soil look quite different.

In soil, roots can be imaged noninvasively using so-called rhizotrons [223], i.e. flat pots with large vertical windows, such that parts of the roots visibly grow along the window (see Figures 2.3J, and 2.3K). In dark soil and at high spatial resolution, segmentation of bright roots may be done with solutions developed, e.g., for angiograms in medicine; but under realistic conditions this is difficult: even with high-resolution cameras (in the 30 megapixel range) fine roots may be only few pixels wide, blurred and with poor contrast to the surrounding soil. Many current segmentation solutions are slow or even break down when applied to such large images. Thus, computational efficiency is an issue. In addition, windows can get scratched by frequent use and soil contains all sorts of clutter. To date, reliable segmentation of such images can only be done semi-automatically, requiring user assistance. Even learning based methodologies yield unimpressive results, which point to needs in finding (or learning) better feature representations.

Using penetrating radiation or modalities such as MRI, PET, and CT [218], roots can be imaged in soil in 3D, where different imaging techniques yield complementary contrast information and metabolic function (e.g., with PET). MRI has high contrast between root and soil for not too wet soils. Then, even fine roots with sub-resolution diameter can be measured due to the so-called 'partial volume effect'. Segmentation procedures are similar to angiogram segmentation in medical image processing. X-ray CT features much higher spatial resolution than MRI, however at the cost of lower soil-root contrast. 3D positron emission tomography (PET) can be used to visualize metabolic processes and carbon transport within plant tissue. For this, radioactive $^{11}CO_2$ is fed to a plant and radioactive decay is measured. Modeling and quantifying transport is an active research topic. Challenges are similar to medical applications including proper (co-)registration of time-series of deforming objects of potentially different modalities, disentangling objects, measuring geometric traits, etc. However, artifacts and structures are different.

Adding dynamics: Tracking, flow, and growth estimation

For many plant traits temporal dynamics are of high relevance. Growth analyses on local tissue level are typically performed on image sequences with frame rates in the range of one per minute. A long-established technique restricts the leaf of interest to a plane by pulling it flat, and images it using a single camera. Growth is then calculated as divergence of an estimated optical flow field. Unfortunately, with this simple engineering solution, gene expression analyses have shown that 'tension-stress genes' are turned on during such experiments, and thus the observed growth may be influenced on the molecular level. For non-fixed leaves moving in 3D, calculating scene flow from multi-camera 'light-field' image sequences has been investigated (see Figures 2.3L, and 2.3M). This allows precise translation and rotation field estimation. Local growth can also be estimated from divergence, however, signal-to-noise-ratio is relatively poor. To date, no reliable local growth measuring technique without fixating leaves is available.

When growth analysis in terms of summary growth over an organ is aimed for, segmentation or reconstruction techniques as described earlier are needed. For simple plant architectures, e.g., young tobacco with up to 8 leaves, or Gossypium with up to 6 leaves well separated by long stem parts, leaf-wise tracking in temporally sufficiently high-resolved datasets has been demonstrated [74]. When time intervals become larger, or plant complexity is higher, no reliable method for leaf-wise tracking has been reported in literature so far. Clearly, when leaf segmentation and tracking are solved, growth analysis in terms of size change over time becomes trivial.

Spectral image analysis: Spanning all scales

Multi-spectral and thermal imaging has become increasingly popular as sensors become affordable. Spectral resolved imaging is applied at various scales (cf. Figure 2.3O), typically ranging from small (plant parts) to large (remote sensing from unmanned aerial vehicles (UAVs) for field operations, or even from satellites), allowing for noninvasive functional analysis even at night or under low-light conditions.

A multitude of modalities intersect with a variety of applications [119]. Infrared thermography carries functional information in terms of temperature allowing to study, e.g., heat dissipation and photosynthetic efficiency,

water status (to help regulate irrigation regime or water resource management), response to drought, or water deficit. Imaging challenges originate from environmental influences as sources of light or heat, which impact thermal imaging and may affect contrast, lower time resolution (e.g., in the order of minutes) limiting applications when biological processes are faster.

Fluorescence imaging typically focuses on chlorophyll fluorescence to study photosynthetic activity (and malfunction), or to monitor plant health. Imaging challenges come from the strict control of light sources and the typically small measurable area. Sun induced fluorescence is currently investigated for applications with aerial images. Multi- and hyperspectral imaging opens a gate to spectroscopy, e.g., used to assess plant health, detect diseases, or estimate leaf chlorophyll content. However, hyperspectral equipment is still very expensive. Near-infrared (NIR) imaging is applied to plant growth and health monitoring, classification of different tissues, e.g., spots from diseases (cercospora), drought, or pests.

Signal processing challenges typically arise from the interpretation of pixel-wise data. To estimate process parameters from the spectral data, not only reliable radiometric (and geometric) calibration procedures are needed, but also dependencies on surface structure, such as bidirectional reflectance distribution function (BRDF), effects of self-shadowing, light transport and transmission in and through tissue, etc. need to be considered. Interpretation strategies as in Differential Optical Absorption Spectroscopy (DOAS), or component analysis e.g. by positive matrix factorization for time series are currently in use.

Combining information from different modalities (e.g., depth, thermal, and optical) helps gathering simultaneously anatomical and functional information for the same plant/organ. Joining such data poses typical image registration problems, needing to know accurately positioning and acquisition conditions/parameters from calibration of different imaging systems.

Greenhouse, field, and farm: More variability

While experiments in the laboratory (and with model plants) do advance our knowledge of biological systems and their functioning, ultimately phenotyping must translate the knowledge to the society and stakeholders, such as breeders and farmers [97]. Phenotyping investigations must then be conducted under 'real' (or realistic) conditions in the greenhouse or

field, on crops that carry agricultural importance, such as corn, wheat, rice, barley, etc. [284]. The complexity of this task spans multiple levels and requires significant advances in automation, imaging, computer vision, and analysis. Furthermore, higher fidelity and more precise methods tend to increase cost, thus they are adopted less.

Several laboratories and commercial entities have developed robotic systems to water and image plants in a greenhouse setting at a large scale: either by moving the plant to the imaging station or moving the imaging apparatuses to the plants, with the latter requiring less retrofitting and customization in the greenhouse design and infrastructure. Both approaches utilize sophisticated imaging technologies (at various spectra) to obtain 2D or 3D representations of plants. The amount of imaging data produced is enormous, and the analysis tasks (essentially a combination of those outlined in the paragraphs above) can be significantly complex. For example, if the imaging station is stationary, appropriate background (e.g., blue screens) helps eliminate clutter and simplifies tasks of segmentation and tracking, as opposed to approaches where the imaging sensors are moving and the plants are stationary, where background clutter is a significant factor. Independent of setting, establishing temporal correspondence for longitudinal analysis remains a significant challenge since any positioning differences either of the camera or the plant radically complicate the process of establishing correspondences between follow-up measurements.

While certain advances in robotic automation and greenhouse design help mitigate some of the problems, taking the imaging apparatus outside and in the field introduces additional challenges. Several approaches and concepts exist: placing the imaging sensors on board of specialized carriers, e.g., man-controlled tractors or alternatively on ground vehicles, or in the air with UAVs (using heli or quadcopters or airplanes [138], cf. Figure 2.3N) and balloons which can be operated either remotely or in an automated fashion to obtain imaging data. Image data differ tremendously in resolution, detail, motion blur or clutter, severely affecting subsequent analysis tasks, thus, more robust algorithms are necessary. Computational efficiency is an issue, as the amount of imaging data produced is enormous (cf. Figure 2.3O), and analysis tasks can be significantly complex.

From the imaging data, measurements of plant (leaf) growth, fruiting and seeding time, plant health (usually by means of color) are of interest with respect to different cultivars and/or environmental changes such as for example drought, climate change, response to treatment or fertil-

ization, parasites, and disease. Each approach, directly affected by the payload capability of each carrier, ultimately affects the level of resolution (not only spatial, but spectral, and temporal) of imaged plants and the ability to study the above relationships, with on the ground approaches being able to image each plant (and leaf or fruit) at unary level (with a lower throughput) while on the air approaches image canopies at a higher throughput obtaining though more aggregate image based measurements (see Figure 2.3O). From a computer vision perspective, since now the plants (or canopy) are not in a controlled environment, vibrations in imaging equipment, position ambiguity, changes in illumination, background clutter, and occlusions severely affect the subsequent analysis tasks and more robust algorithms are necessary. In fact to this date, due to lack of precise computer vision mechanisms to delineate individual leaves, only whole plant measurements (obtained by segmentation) are used.

Most approaches use these apparatuses for data collection and thus processing happens in an offline/remote manner. However, as automation and action-feedback mechanisms are investigated, some (or all) analysis should occur on the robot (sensor) itself. In some sense, the robot must detect/identify the plant as part of its planning and based on some analysis consider possible action (e.g., fertilization or weed extermination) or collect measurement data as means to provide field based analytics [125]. The latter process, is the central theme in precision agriculture, which aims at tailoring treatment and process at the individual plant level. Thus, computer vision becomes crucial in supporting the whole process and evidently there is now the additional challenge of identifying low complexity approaches to robust vision.

2.2.2 A timely and unique challenge

Quantitative description of plant phenotypes is a key ingredient for knowledge based bio-economy and this, not only literally helps in the efforts to feed the world, but is also essential for fiber and fuel production, the so-called Green Revolution 2.0. In fact, comparing the top 10 list of Emerging Technologies in 2012 according to the world economic forum, the top 1, 2, 3, and 5 are directly addressed by plant phenotyping research.[4] Recently, we have even witnessed direct investments in helping the translation of

[4] http://reports.weforum.org/global-agenda-council-2012/councils/emerging-technologies/

agricultural technology in farming. For example, Farm2050[5] includes Google and 3D Robotics among its partners, an information extraction powerhouse and a drone company.

Definitely there is not only growing interest from the application side, both scientifically and commercially, but exciting computer vision and image processing problems exist [292], which differ from other biomedical applications. While medicine focuses on the status of a single species (i.e. humans) in a diagnostic capacity, plant phenotyping addresses a large number of different plant species with hundreds to thousands of genotypes (cultivars) per species usually in group-wise experiments, on a scale not usual in medicine (with some exceptions in brain function/anatomy). It addresses the development over time in addition to static snapshots and under a wide range of environmental conditions, using various imaging setups (as opposed to medical imaging where predefined protocols are in place and equipment variability is relatively limited). Thus, even within a single application, diverse conditions need to be addressed, to ascertain a robust image-based measurement of the phenotypic trait. Plant phenotyping at a high throughput requires reliable image processing algorithms that could batch process many data accurately, and an integration with genetic databases and other frameworks. On the positive side, working on plant phenotyping is less mission critical compared to medical imaging: we can afford less reliability and it is usually possible to increase sample size (cheaply) to cope with lower accuracy. Furthermore, problems are easier to grasp for the non-initiated, e.g., it is easier to teach a non-expert to delineate a leaf. In brain MRI, it is virtually impossible: an expert is necessary to delineate anatomical regions of the brain, and intra/inter-observer variability is high.

The task of finding certain phenotyping traits and the set of genetic differences that might cause them, can be regarded as an inverse problem [161]. Since the genome (and genotype) have large degrees of freedom (billions of bases, if we care about individual mutations) compared to the number of observed phenotypes (traits), it is readily seen that the solution to finding their association may not be unique. Thus, approaches from the fields of machine learning and statistical signal processing may help here. Besides, the design and development of affordable automated systems (and the associated hardware) for robotized plant phenotyping (and its sibling, robots in agriculture), provides fertile ground for experts in hardware, signal processing, control, and robotics.

[5] `http://www.farm2050.com`

The previous sections outlined a series of challenges (e.g., dealing with structural breaks in tracking/detection), for the computer vision community to get involved with. While in this chapter we focus on extracting information from images, data mining and combing the information from genotyping, environmental, and phenotyping sources are by themselves a big undertaking as well. A joint effort of experts from different fields is therefore necessary, and by including the resources in Table 2.1 we hope to help facilitate this and lower the entry barrier. Co-operation of different disciplines to integrate expertise across the spectrum and provide biologically or agronomically meaningful and technically robust solutions [97, 254] are key to help resolve this bottleneck.

2.3 Summary

In this chapter we provided background information on image-based plant phenotyping and offered an overview of recent approaches. Relevant solutions were discussed for the collection as well as for the analysis of plant images, with particular attention to whole plant and individual leaf segmentation. Nowadays (as opposed to a few decades ago) a variety of imaging modalities are available (since there is reliable hardware) but extracting 'the biology' out of the data (i.e. automatically analyze the images to answer high-level questions of biological relevance) is still comparatively slow and underdeveloped. Several difficult image processing and computer vision problems are encountered in a plant phenotyping context, and in this chapter we provided a broad overview of challenges arising in typical phenotyping applications.

In the next chapters we will detail several aspects of our proposed solution for affordable and yet reliable plant phenotyping. In Chapter 3 we describe our affordable sensing solution for phenotype collection, while in Chapter 4 we focus on the analysis, proposing a robust plant segmentation algorithm that can adapt to several challenging and changing conditions occurring in a laboratory environment.

Table 2.1: Collection of online resources for plant phenotyping (databases, workshops, consortia, websites with software).

Name	Description	Link
Specialized Scientific Events		
CVPPP 2014/2015	The 1st and 2nd workshops for computer vision problems in plant phenotyping	http://www.plant-phenotyping.org/CVPPP2015
IAMPS	International workshop on Image Analysis Methods for the Plant Sciences	http://www.phytosystems.ulg.ac.be/iamps15/
PhenoDays	International symposium involving seed industry, breeding institutes, and academic breeding groups	http://www.phenodays.com
IPPS	International Plant Phenotyping Symposium	http://www.plant-phenotyping.org/chennai_2014
ICPA	International Conference on Precision Agriculture	https://www.ispag.org/ICPA
Image Databases		
LSC Challenge	Images and leaf-based segmentation masks as part of the 1st Leaf Segmentation Challenge (LSC)	http://www.plant-phenotyping.org/CVPPP2014-dataset
MaizeGDB	Images of maize	http://www.maizegdb.org/cgi-bin/imagebrowser.cgi
CWFID	The Crop/Weed Field Image Dataset (CWFID) contains images with crop/weed delineations for a classification task in precision agriculture	https://github.com/cwfid/dataset
Phenopsis DB	Arabidopsis Thaliana Phenotyping Database	http://bioweb.supagro.inra.fr/phenopsis/
Consortia and Organizations		
iPlant Collaborative	Connect scientists to public datasets, manage and store their data and experiments, access high-performance computing, etc.	http://www.iplantcollaborative.org
IPPN	International Plant Phenotyping Network	http://www.plant-phenotyping.org
EPPN	European Plant Phenotyping Network	http://www.plant-phenotyping-network.eu
EPSO	European Plant Science Organisation	http://www.epsoweb.org
FESPB	Federation of European Societies of Plant Biology	http://www.fespb.org
IEEE RAS	Agricultural Robotics and Automation	http://www.ieee-ras.org/agricultural-robotics
ISPA	International Society of Precision Agriculture	https://www.ispag.org
e-agriculture	ICT for sustainable agriculture	http://www.e-agriculture.org
BSA	Botanical Society of America	http://botany.org
Software Databases		
Plant Image Analysis	This database currently provides a collection of approx. 120 analysis tools	http://www.plant-image-analysis.org

Part II

Affordable sensor and image analysis on the cloud

3 An affordable image sensor

3.1 Introduction

Noninvasive plant investigations are performed using various modalities with spatial scales varying from the microscopic to large outdoor fields [27, 52, 220, 246]. Typical problems in measuring a plant's visible properties comprise measuring size, shape, color or spectral reflection, and other structural and functional traits of whole plants, their organs, or plant populations. Vision-based measurements allow recording and monitoring of relevant phenotypes noninvasively, with higher precision, accuracy, and throughput than manual measurement [282], at considerably reduced cost and human labor [104]. Biologists grow model plants, such as *Arabidosis thaliana*, in controlled environments and monitor and record behavior and appearance, i.e. the phenotype. Such experiments

This chapter is partly based on:

- H. Scharr, M. Minervini, A. Fischbach, S. A. Tsaftaris, "Annotated image datasets of rosette plants," Forschungszentrum Jülich GmbH, Jülich, Germany, Tech. Rep. FZJ-2014-03837, Jul. 2014.
- M. Minervini, A. Fischbach, H. Scharr, S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," submitted for publication to *Pattern Recognition Letters*, 2015.

are fundamental and ubiquitous, and recovering the phenotype implies that: (a) suitable imaging solutions are deployed, and (b) computer vision algorithms must deal with the complexity of the plant, the experiment, and the environmental conditions (cf. Chapter 2).

Starting from such typical experiments, we devise imaging apparatuses and setups to image plants in a phenotyping context (Section 3.2). To increase adoption of image-based approaches to plant phenotyping, we propose an affordable and flexible sensing device based on the Raspberry Pi single-board computer and the 'RaspiCam' camera module. We also implement the affordable solution originally proposed by Tsaftaris and Noutsos [306] based on a commercial camera.

Till now, despite the 20 year history of imaging plants, a comprehensive collection of benchmark datasets for image-based, nondestructive plant phenotyping is still lacking, making it difficult to compare existing methodologies. While plant related datasets exist for leaf or flower recognition [110, 140, 228, 286, 336], these datasets were obtained in an uncontrolled or destructive manner and not in a phenotyping context. In Section 3.3 we describe the details of imaging procedures, plant material, and growing conditions that we adopt to setup two plant experiments aimed at acquiring imaging data showing top-down views on different cultivars of Arabidopsis rosettes.

Based on the raw images acquired with our affordable settings, in Section 3.4 we present a collection of specially formatted datasets suited for a series of computer vision tasks, accompanied by appropriate expert annotations and metadata obtained manually (cf. Figure 3.1). We emphasize tasks such as plant/leaf detection, segmentation, and tracking, leaf counting, boundary estimation, and general regression and classification. For consistency in evaluation methodologies, we also discuss appropriate criteria.

Parts of these data have been released in the public domain, in an effort to motivate development of novel methodologies for the segmentation of plants and leaves in images from phenotyping experiments. The image data presented here are used throughout the thesis to validate the proposed approaches: in Chapter 4 we will offer exemplary use cases and results on plant and leaf segmentation and counting, while in Part III we will test image compression approaches.

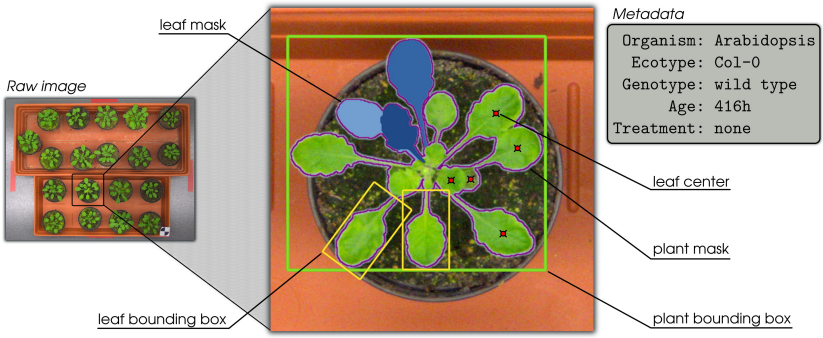


Figure 3.1: Pictorial example of metadata and annotations superimposed on an image of Arabidopsis.

3.2 Affordable sensing solutions

In this section we describe the hardware and software components of our affordable sensing solution for time-lapse photography based on the Raspberry Pi. For comparison we also implement an approach based on a commercial camera [306]. To favor adoption of our approach we publicly release implementation details and software components. Further details on how to set up and operate the imaging sensors can be found in Appendix A.

3.2.1 A smart sensor based on the Raspberry Pi

The Raspberry Pi [280, 309] is a credit-card sized single-board computer, designed and developed by the Raspberry Pi Foundation,[1] UK, as an educational tool for teaching computer science and programming [10, 67]. The Raspberry Pi is cheap (monetary cost is €35) and easy to use (it runs the Raspbian, a full-featured Linux operating system), therefore it is ideal to implement prototypes or customized systems.

As illustrated in Figure 3.2, we use a Raspberry Pi (Model B) single-board computer, shorthanded as *Rpi*, equipped with a 5 megapixel 'RaspiCam' camera module, to capture static images (width×height: 2592×1944 pixels) of the scene. The main advantage of this solution with respect to

[1] `http://www.raspberrypi.org`

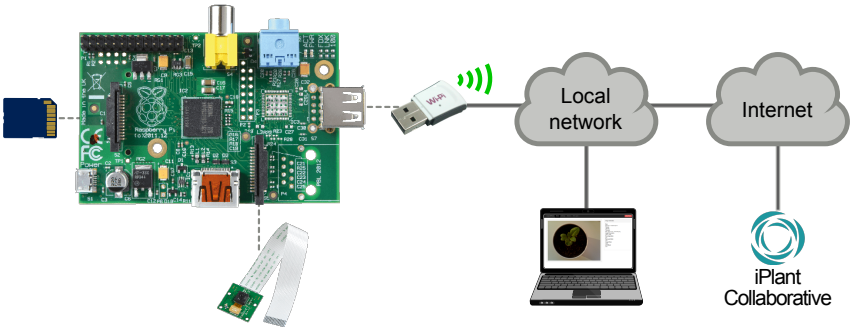


Figure 3.2: Schematic of our sensing solution based on the Raspberry Pi, showing the headless mode for image acquisition and transmission.

the setup discussed in the previous section is the greater flexibility offered by having a complete yet small computer attached to the sensor. While the RaspiCam is capable of acquiring images of good quality that can satisfy a wide range of applications (the infrared camera module 'NoIR' is also available), in contexts where superior image quality is required, the Raspberry Pi can be used in combination with an SLR camera or other imaging sensors.

To control the camera and acquire images we adopt the *raspistill* application. The images acquired by the RPi setup are also automatically transmitted to the cloud for storage and analysis. Here we rely on the scientific cloud infrastructure offered by the iPlant Collaborative project [111], to deploy our plant image analysis software solution (cf. Chapter 4).

In a distributed sensing and analysis scenario, in which the acquired data needs to be transmitted via the Internet to centralized locations (e.g., a cloud service) for analysis, it becomes necessary to compress the images effectively (cf. Chapter 6). In this context, a single-board computer such as the Raspberry Pi operating the sensor offers the possibility to: (a) perform (low-complexity) pre-processing operations on the acquired imagery (e.g., image enhancement, specialized compression, low level vision); and (b) autonomously transmit the images to a remote location.

To ease configuration and monitoring, we deploy a web-based interface to operate the sensor remotely (cf. Figure 3.2). In Appendix A, we describe how to set up the hardware and software components of our affordable sensing solution based on the Raspberry Pi.

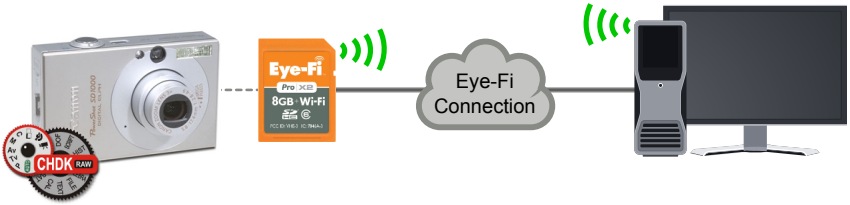


Figure 3.3: Schematic of the sensing solution based on a commercial-grade camera described in [306].

3.2.2 Imaging plants using a commercial camera

We also use a commercial-grade camera equipped with a memory card providing networking capabilities to setup an automated low-cost sensing system [306, 345], which is able to capture time-lapse images, save them both in JPEG and raw uncompressed format, and transmit them through wireless connection to a receiving PC. Figure 3.3 illustrates the architecture of the approach.

Most low-grade commercial cameras expose only a limited set of configurations and modes of use, thus restricting the user to basic imaging functionalities, while single-lens reflex (SLR) cameras generally provide the user with greater control on the device and more features but they are costly. Here we adopt a 7 megapixel Canon PowerShot SD1000 camera, shorthanded as *Canon*, equipped with an Eye-Fi Pro X2 memory card, providing 8 GB SDHC capacity for storage and 802.11n wireless networking capabilities for Wi-Fi communication with a computer. We install on the camera the open source CHDK (Canon Hack Development Kit) firmware,[2] to enable control on a richer set of camera features (e.g., saving raw images) and the ability to run scripts (e.g., software-simulated intervalometer) [278].

Flash is disabled, while other camera settings (e.g., exposure, focus distance) are obtained automatically from the camera before acquiring the first image, and are subsequently kept unchanged throughout the experiment. At each acquisition, we acquire two images of the same scene at different focus distances, e.g., to have the possibility to fuse them in a single image that is in focus everywhere [235], or to enable 3D surface estimation using depth-from-defocus techniques [294]. Using the Lua scripting

[2] Available at `http://chdk.wikia.com/wiki/CHDK`

language [141], we program the camera to capture time-lapse images of the scene, that are subsequently transmitted to a nearby workstation for storage. All acquired images (width×height: 3108×2324 pixels) are stored in both raw uncompressed (DNG) format, to avoid distortion introduced by compression, and also JPEG format, to save the EXIF (EXchangeable Image File) metadata [151].

To obtain RGB images in TIFF format, raw image files from the Canon camera are developed using the *dcraw*[3] software tool (version 8.99) and the following command line options:

`dcraw -v -w -k 31 -S 1023 -H 2 -o 1 -q 3 -T` *filename*

Finally, to reduce disk occupancy TIFF images are encoded using the lossless compression standard available in the PNG file format [315].

3.2.3 Camera sensor calibration

The camera sensors should be calibrated with respect to color and to correct lens distortion. Color calibration can be performed using a calibration target (e.g., colored stripes), to adjust the color response of the camera to a known color space [142]. This step would permit comparisons across experiments with different illumination conditions.

For accurate vision-based measurements principal distance, principal point offset, and lens distortion parameters must be known [262]. Calibration scales can also be used to assess out of focus blur. Overall, camera calibration would permit to correct lens distortion and determine exact pixel size, so as to obtain precise photogrammetric measurements.

In the plant experiments described in the next section we do not perform such calibration steps (except calculating pixel size), because we maintain illumination conditions constant. Also, we use for analysis a reduced and centered with respect to the field of view portion of the image, which is less affected by lens distortion.

3.3 Imaging Arabidopsis plants

In this section we describe two plant imaging experiments, in June 2012 and in September-October 2013, in which we deploy the affordable sensing solutions described previously. We obtain two image datasets, hereafter

[3] Available at `http://www.cybercom.net/~dcoffin/dcraw/`

Table 3.1: Summary of information of the Arabidopsis raw imaging data.

Experiment	Subjects	Wild-types	Mutants	Period	Total images	Image resolution	Plant resolution
Ara2012	19	Col-0	No	3 weeks	150	7 MPixel	0.25 MPixel
Ara2013 (Canon)	24	Col-0	Yes (4)	7 weeks	4186	7 MPixel	0.25 MPixel
Ara2013 (Rasp. Pi)	24	Col-0	Yes (4)	7 weeks	1951	5 MPixel	0.06 MPixel

named, respectively, 'Ara2012' and 'Ara2013', both consisting of top-view time-lapse images of *Arabidosis thaliana* rosettes.

The data collections are carried out at IMT Institute for Advanced Studies, Lucca, Italy, in the context of the European project "PHIDIAS: Phenotyping with a High-throughput, Intelligent, Distributed, and Interactive Analysis System".[4]

Table 3.1 summarizes relevant information regarding the datasets, which are discussed in detail in the following paragraphs. Parts of the datasets are manually annotated to provide a benchmark for analysis methods. The vision tasks required to estimate several phenotyping parameters include detection, segmentation, and tracking across time, for plants and individual leaves. Prior to discussing the datasets and the computer vision tasks for which they can be used (Section 3.4), we describe below how plants are grown and imaged.

3.3.1 Imaging setup

Following the setup proposed in [306], we devise a small and affordable laboratory setup (overall, monetary cost of the system is below €300, cf. Figure 3.4), composed of a growth shelf and an automated low-cost sensing system.[5] For the Ara2012 experiment we use the solution based on a commercial camera, while for the Ara2013 experiment we also deploy the solution based on the Raspberry Pi (using also the Rpi setting permits the evaluation of algorithms on images of the same scene but with different image quality). Example images captured with this setup are shown in Figure 3.5, illustrating plant arrangement and scene complexity.

Plants are illuminated artificially and controllably to emulate daylight imaging in a fixed day cycle, using two cool-white daylight fluorescent

[4] http://prian.lab.imtlucca.it/PROJECTS/PHIDIAS/phidias.html
[5] http://www.phenotiki.com

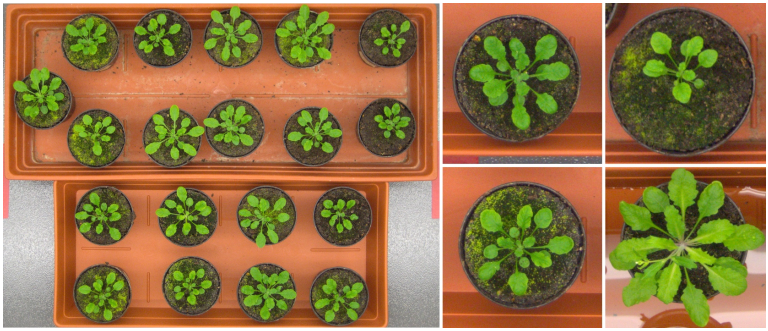
Figure 3.4: Affordable acquisition setup used to collect the 'Ara2012' and 'Ara2013' datasets.

lamps, installed 80 cm above the pots. Camera sensors are positioned between the lights, approximately 100 cm above the plants, and operated as intervalometers, i.e. acquiring images at preset times with preset imaging conditions (such as focus, exposure, field of view). Images are captured during day time only for a period of time (cf. Table 3.1). No modifications of the configuration are done after the experiments are started.

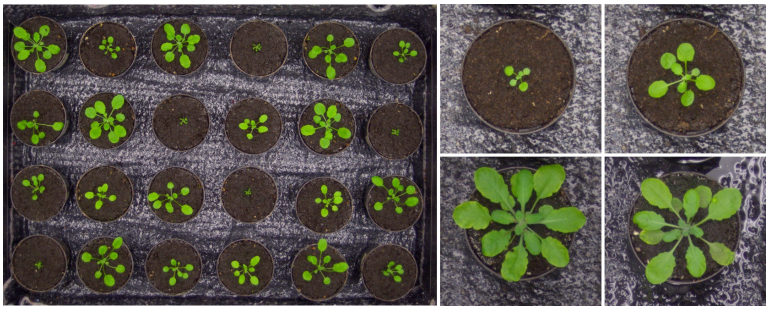
3.3.2 Plant material and growing conditions

The Ara2012 experiment involves 19 subjects, all Columbia (Col-0) wild types. On the other hand, the Ara2013 experiment involves 24 subjects, including Col-0 wild types and four different lines of mutants, all with Col-0 background. Specifically, the Ara2013 experimental setup is composed of the following genotypes:

- Col-0, 5 subjects, wild type;
- *pgm* (plastidial phosphoglucomutase), 5 subjects – impaired in starch degradation, exhibits reduction in growth, delayed in flowering;



(a) Ara2012



(b) Ara2013 (Canon)

Figure 3.5: Example raw images (left) and close-ups showing individual plant subjects (right) of: (a) Arabidopsis (Col-0, wild-type), and (b) 5 different cultivars of Arabidopsis. The plants were imaged with the setup shown in Figure 3.4.

- *ctr* (constitutive triple response), 5 subjects – exhibits dwarfism and very small rosette leaves;
- *ein2* (ethylene insensitive 2), 5 subjects – exhibits large rosettes, delayed in bolting;
- *adh1* (alcohol dehydrogenase 1), 4 subjects.

Plants are grown in individual pots with 16/8 hour light/dark cycle for Ara2012, and 12 hour light/dark cycle for Ara2013. Watering is provided two or three times per week by sub-irrigation. Images are captured during day time only, every 6 hours over a period of 3 weeks for Ara2012, and every

20 minutes over a period of 7 weeks for Ara2013. For the Ara2013 dataset, the diameter of each plant is also manually measured with a caliper and recorded on a daily basis for reference. Number of subjects is selected according to coverage area of the camera, to obtain satisfactory imaging resolution (we measure pixel size to be $\sim$0.167 mm). Pots are spaced out in the tray, to prevent adult plants from touching (from an image processing standpoint, handling this circumstance in an automated fashion is difficult, and most solutions assume that distinct subjects never touch). Arrangement of genotypes in the tray is randomized for the Ara2013 experiment, to eliminate possible bias in the results due to variations in watering or lighting conditions. No treatments are performed.

3.4 Finely-grained annotated datasets

Identifying and evaluating a plant's phenotype relies on the fine-grained categorization of a plant's properties: e.g., how many leaves, of which architecture, visual age or developmental stage, to which cultivar a plant is similar. Previously, such categorization was annotated manually by experts, but recently image-based approaches are gaining momentum (see Chapter 2 for an overview of approaches). However, several of the computer vision tasks encountered, such as leaf segmentation and counting, are particularly challenging and remain unsolved.

One of the factors that could accelerate the development of better algorithms and their consistent and systematic evaluation is the availability of benchmark data focusing on typical imaging situations and tasks in plant phenotyping. Having such data in the public domain will also introduce this societally important application to a larger audience outside the field of plant phenotyping. If it is adopted as one of the test datasets of the broad community (as the Pascal Visual Object Classes [90, 91], or the biologically focused Broad Bioimage Benchmark Collection [187]), it will help track how the constant evolution of computer vision would benefit these problems.

To provide specialized standalone datasets for a number of computer vision tasks, parts of the raw image data described in Section 3.3 have been carefully annotated by experts. Note that not all data have been annotated (thousands of raw images are available) and annotation is a continuing process. Among those that have been annotated, not all are publicly available, to permit future competitions and challenges based on

data which are blind to participants. Other computer vision competitions, such as Pascal VOC [90,91], and biologically inspired challenges[6] follow this strategy. In addition, for each specific task, the datasets are considered standalone and as such the user is unaware of other information (e.g., mutant type for leaf segmentation). This limits domain and prior knowledge and should lead to methodologies that are more robust to the changing and complex morphology of plants.

In the following paragraphs we describe first the semantic hierarchy considered, the manual annotation procedure, and then proceed in detailing available subsets of the data for each computer vision and machine learning task, defining annotations and appropriate metadata, and suitable evaluation criteria.

3.4.1 Overview of semantic hierarchy

Each experiment has generated a vast amount of imaging data. Our internal database and annotation strategy follows the hierarchy visible in Figure 3.6. The original images are higher in our semantic hierarchy. In addition, each experiment is characterized by several metadata such as an experiment identifier, camera used, and day/time of collection.

Our goal is to provide relevant annotations for a series of computer vision tasks. Gray boxes in Figure 3.6 denote annotated metadata, such as experiment type, mutant type, camera used, acquisition time, experimental treatment, segmentation difficulty, etc. Non-shaded boxes denote imaging and image level annotations. Note that an experiment may contain both tray and individual plant images. To construct each of the standalone datasets described below, we trace information in this hierarchy and provide related metadata and annotations wherever appropriate.

3.4.2 Expert segmentations

A significant number of object-based annotations (e.g., bounding boxes) can be obtained computationally on the basis of pixel-level segmentation masks of plants and leaves, respectively, which are manually annotated by experts. Here we describe how we obtain the latter and next we detail the level of annotation for each task.

[6] http://grand-challenge.org/All_Challenges/

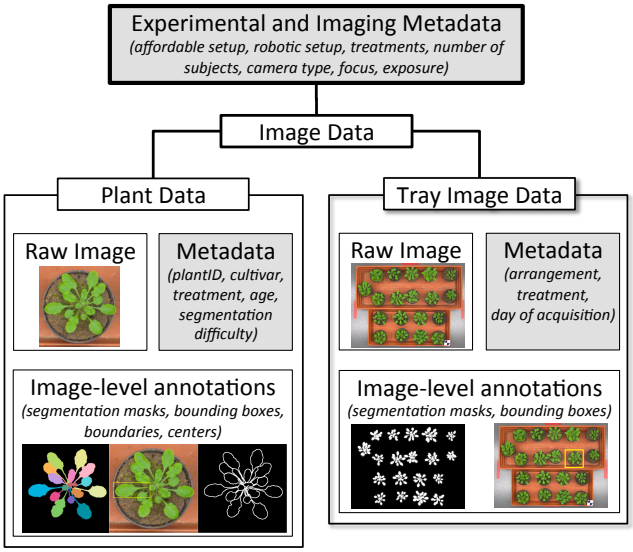


Figure 3.6: Hierarchy of relationships among data, metadata, and annotations. In parenthesis we provide examples of annotation variables, and we also provide pictorial examples of imaging data and annotations such as segmentation masks, bounding boxes, and leaf boundaries. Gray boxes denote metadata.

Annotation consists of three steps (cf. Figure 3.7). First, we obtain a binary segmentation of the plant objects in the scene in a computer-aided fashion, using the plant segmentation approach described in Chapter 4. The result of this segmentation is manually refined using raster graphics editing software (the GIMP[7]). Next, within the binary mask of each plant, we delineate individual leaves, completely manually. A pixel with black color denotes background, while all other colors are used to uniquely identify leaves of the plants in the scene. Across the frames of the time-lapse sequence, we consistently use the same color code to label occurrences of the same leaf. To reduce observer variability and increase accuracy, the labeling procedure involves always two annotators: one annotating the dataset and one inspecting the other. Figure 3.8 shows examples of plant images from the datasets and corresponding pixel level annotation masks.

[7] http://www.gimp.org/

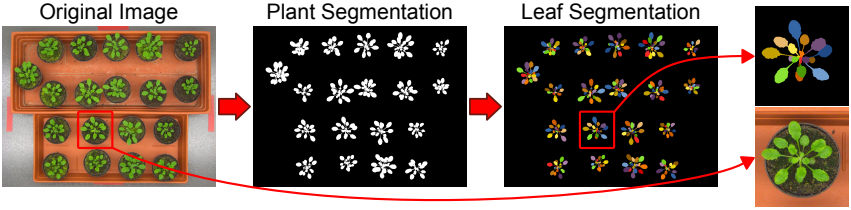


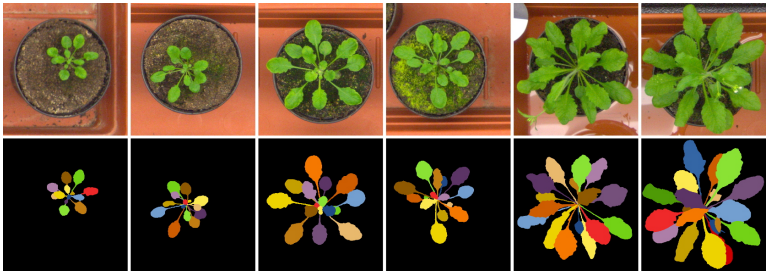
Figure 3.7: Schematic of the workflow to obtain leaf annotations. Plants are first delineated in the original image, then individual leaves are labeled.

On a secondary inspection of the data, additional categorical *qualitative annotations* are recorded by annotators such as: estimate of segmentation difficulty (in the 1[easy]-5[hard] scale), plant appears in focus, leaves appear in vertical positions (due to the so-called nastic movements), plant is occluded by another one (when pots are placed close by), and scene contains complexities (e.g., water in the background, green moss on soil, a midge or damage on leaves).

3.4.3 Computer vision tasks and datasets

Fine-grained information to be extracted from images is ubiquitous in plant phenotyping, since we do have to deal with how different mutants or treatments affect plant shape and characteristics. Even the same plant will have leaves of different shape and size according to their maturity. In some cases the phenotype of a mutant is not known, and typically researchers assign qualitative characteristics, which is simple when gross phenotype differences are evident (e.g., major differences in leaf shape or plant size). However, it is when such differences are subtle, that the ability to extract fine-grained information directly from images would make a tremendous impact: it would permit biologists to identify small traits to be explored further. Computer vision and machine learning could enable biologists to assess a new mutant's phenotypes by evaluating how similar they are to known lines of cultivars and mutants, in a quantitative fashion.

Doing this in a fully automated fashion is fertile ground for a series of very interesting vision tasks, which we outline below together with descriptions of appropriate accompanying datasets and evaluation criteria. To test and develop robust solutions for a realistic range of conditions, several challenging situations are allowed to occur by design in the plant



(a) Ara2012

(b) Ara2013 (Canon)

Figure 3.8: Examples of single plant images at different developmental stages, with corresponding ground truth leaf labeling denoted by color.

experiments, and to construct datasets we carefully select images according to our qualitative annotations (cf. Section 3.4.2).

Our annotations are initialized by computationally post-processing the expert-delineated leaf masks as discussed in Section 3.4.2. Afterwards, everything is verified (and if necessary corrected) by experts to ensure high quality and integrity. For each task we outline the size (as in number of images) of the corresponding dataset; however, note that as the annotation process is ongoing additional images may be released in the future.

Hereafter, to ease presentation we will denote as *leaf mask* the pixel-level leaf masks and as *plant mask* the pixel-level binary mask obtained by the union of all individual leaf labels within a plant. Unless otherwise noted, to reduce storage all images are lossless compressed PNG files [315].

Plant detection and localization

Task: As Figure 3.5 shows, plants can be arranged in a grid, either fixed in position in specialized trays or not. On some occasions, pots are also moved by the experimenter. Additionally, while care is taken so plants are not touching when planted, such optimal distances (with respect to maximum plant diameter) are usually not observed when plants become larger. Thus, a first vision task encountered is plant detection and localization where individual plants, usually in the form of bounding boxes, need to be outlined and occlusions and overlaps must be resolved. This problem is also encountered even outside the context of phenotyping, e.g., in precision agriculture for detecting crops [311] or weed [125]. Here learning-based object detection approaches are ideally suited to help.

Dataset: We derive bounding boxes for each plant as the smallest bounding box enclosing the plant mask, processing each plant individually in a tray image. An additional 5% of the box size is considered to include a larger aspect of the scene. The dataset consists of 100 tray images, as 30 from Ara2012, 35 from Ara2013 (Canon), and 35 from Ara2013 (Rpi). For each tray image, a comma-separated value (CSV) file is available, reporting for each plant the corner pixel coordinates of its bounding box.

Evaluation criteria: We suggest the bounding box overlap ratio criterion:

$$a_o = \frac{\text{area}(B^p \cap B^{gt})}{\text{area}(B^p \cup B^{gt})}\,, \tag{3.1}$$

between predicted B^p and ground truth B^{gt} bounding boxes [90].

Plant segmentation

Task: Plant biomass is an important plant breeding trait, because it reflects overall plant performance. In images of rosette plants usually it is measured as projected leaf area (PLA), i.e. effectively the number of plant pixels. Finding PLA translates to the segmentation of plant from background. In simple cases this can be solved by color thresholding and other unsupervised segmentation approaches [71,318], but when scene complexity is high (non-smooth background, non-uniform lighting, plant overlap, presence of moss on soil) sophisticated learning-based algorithms are necessary [209,244].

Dataset: The dataset consists of 20 tray images from Ara2012. For each image a corresponding black (background) and white (foreground) mask

encoded as an indexed image provides pixel-level information on the location of plant objects.

Evaluation criteria: Several segmentation criteria exist, and we suggest Dice coefficient, precision, and recall, since they have been used throughout image analysis and are common in plant imaging as well [209]. Among those, the Dice Similarity Coefficient (DSC):

$$\text{DSC}\ (\%) = \frac{2|P^{gt} \cap P^{ar}|}{|P^{gt}| + |P^{ar}|}\,, \tag{3.2}$$

measures the degree of overlap among ground truth P^{gt} and algorithmic result P^{ar} binary segmentation masks, where $|\cdot|$ denotes set cardinality. We also suggest the Modified Hausdorff Distance (MHD) [85]:

$$\text{MHD} = \max\left\{ \frac{1}{|P^{gt}|} \sum_{a \in P^{gt}} \min_{b \in P^{ar}} \|a - b\|,\ \frac{1}{|P^{ar}|} \sum_{b \in P^{ar}} \min_{a \in P^{gt}} \|a - b\| \right\}, \tag{3.3}$$

where $\|\cdot\|$ denotes Euclidean distance. The MHD has large discriminatory power, is robust to noise, and is easy to interpret (it can be expressed in units of length, e.g., millimeter).

Leaf segmentation

Task: In rosette plants when leaves are highly overlapping, PLA may not be an accurate estimator of plant biomass anymore and individual leaf segmentation is necessary. When individual leaves are segmented, distributions of leaf size can highlight the rate of growth of new leaves with respect to old ones. However, leaf segmentation, a multi-instance segmentation problem [127], is particularly challenging since most leaves within the same plant may share appearance and shape, but can also appear severely overlapping. To complicate matters even more, plant morphology changes radically between mutants, in response to treatment, and as plants grow. In addition, self-occlusion, shadows, leaf hairs, leaf color variations, and others make the problem even more complex. Image quality is a factor as well, so low resolution and out-of-focus (as it could occur in portions of the datasets) affect leaf segmentation accuracy.

Dataset: We use leaf masks but without temporal label consistency. 128 from Ara2012, and 31 from Ara2013 (Canon), images of single plants appearing centered are used. For each plant, annotations are provided in the

form of indexed images (a color palette is embedded for visualization) the same dimensions of the originals. We use one label per leaf, starting from '1' up to the maximum number of leaves, with '0' denoting background.

Evaluation criteria: Several segmentation criteria are available for comparing ground truth and algorithmic outcomes [207, 308]. We suggest SymmetricBestDice, the symmetric average Dice score among all objects (leaves), where for each input label the ground truth label yielding maximum Dice is used for averaging. Best Dice (BD) is defined as:

$$\mathrm{BD}(L^a, L^b) = \frac{1}{M} \sum_{i=1}^{M} \max_{1 \le j \le N} \frac{2|L_i^a \cap L_j^b|}{|L_i^a| + |L_j^b|}\,, \tag{3.4}$$

where $|\cdot|$ denotes leaf area (number of pixels), and L_i^a for $1 \le i \le M$ and L_j^b for $1 \le j \le N$ are sets of leaf object segments belonging to leaf segmentations L^a and L^b, respectively. SymmetricBestDice (SBD) is then:

$$\mathrm{SBD}(L^{ar}, L^{gt}) = \min\left\{ \mathrm{BD}(L^{ar}, L^{gt}), \mathrm{BD}(L^{gt}, L^{ar}) \right\}, \tag{3.5}$$

where L^{gt} is the ground truth and L^{ar} the algorithmic result.

Leaf detection

Task: In an image analysis pipeline, image-based leaf detection could serve to initialize other processes (segmentation or tracking). Due to size differences, shape and appearance similarities, and heavy occlusions, leaf detection is a complex task, and can benefit from approaches in computer vision of detecting overlapping objects in medicine, transportation, and surveillance [14, 333].

Dataset: On the basis of leaf masks, we extract for each individual leaf the smallest rectangular bounding box (possibly rotated with respect to the image coordinate system) enclosing the mask of that leaf. The dataset consists of individual plant images, 128 from Ara2012, and 31 from Ara2013 (Canon), and for each image, a CSV file storing per row the leaf index and the coordinates of each bounding box, with as many rows as number of leaves. Note that our annotation does include the petiole ('leaf stem') in Arabidopsis.

Evaluation criteria: Number of accurate detections and their accuracy evaluated with overlap measures (e.g., as those suggested for plant detection and localization).

Leaf counting

Task: From a phenotyping perspective the number of leaves is directly related to yield potential, drought tolerance, and flowering time [115,206]. From a vision perspective, it can also be used to constrain leaf detection or leaf segmentation algorithms [227]. To this date, user interaction is required and leaf count comes as a by product of leaf segmentation. Learning-based counting techniques could help here [94,174].

Dataset: On the basis of leaf masks we extract for each leaf, the distance transform weighted-center of mass and also the center of mass. When these disagree significantly (above a threshold), and if any of the centers lie outside the binary shape, it indicates a highly asymmetric leaf (e.g., due to heavy occlusion or orientation vertical to the imaging axis) and the annotator is prompted to select a center. The dataset consists of individual plant images, and accompanying binary images containing the centroids for each leaf as a single pixel. (This requires larger storage but we find it more appealing than storing centers in CSV files.) Overall 128 from Ara2012, and 31 from Ara2013 (Canon), raw and equally numbered annotation images are provided. A CSV file listing image names and number of leaves is also provided, for convenience of approaches that solve directly the regression problem.

Evaluation criteria: To evaluate algorithm performance in identifying the correct number of leaves we suggest:

- the difference between number of leaves in algorithm's result and ground truth DiffFGLabels $= \#L^{ar} - \#L^{gt}$, and
- AbsDiffFGLabels, the absolute value of DiffFGLabels.

Note that these criteria do not take into account good localization and while count maybe correct it may not correspond to actual leaves. Alternatively, count via detection measures can be adopted.

Leaf tracking

Task: Finding growth curves of individual leaves helps us understand how a plant (or a cultivar) is growing or the effects of treatments and stresses: for example, [60] found that drought differentially affects leaves. This growth curve usually follows an exponential relationship with time [264,317], and frequent imaging can capture small differences. This implies the precise segmentation and temporal tracking of each leaf [339].

Dataset: Building upon the leaf masks, the dataset consists of sequentially numbered PNG files of raw individual images and annotations. We provide 5 stacks of 14 images each from Ara2012 and 8 stacks of 17 images each from Ara2013 (Canon). Leaf-level segmentations are provided with leaves having the same label index throughout the sequence to ensure temporal consistency. Also, we release for each image a corresponding CSV file, where bounding box definitions are provided as previously described. Note that this dataset can also be used for leaf or plant segmentation with additional temporal information, e.g., for joint segmentation and tracking.

Evaluation criteria: We recommend the protocol of Nawaz and Cavallaro [225], which builds on overlap criteria, and the code available from the authors. When leaves are vertical in the imaging axis (due to severe nastic movements), overlap criteria may be unable to assign proper correspondences. This may cause lack of label consistency of a leaf across time in an algorithmic result, but it can be easily seen in individual leaf growth curves. Quantitatively, they can be detected by multiple local hypothesis testing to identify structural breaks on the growth parameters [74].

Boundary estimation

Task: Some approaches to multi-instance (or multi-label) segmentation rely on accurate boundary detection, which is used for example to initialize template-based models [127, 339]. When image contrast and resolution are adequate, classical edge detection works sufficiently. However, when images are partially out-of-focus and of lower resolution (due to a larger field of view) as is the case of the Arabidopsis data, learning-based methods to boundary estimation (i.e. a learned edge detector as in [198]) are expected to perform better.

Dataset: Using leaf masks, we isolate each leaf label and find its perimeter, to produce an indexed labeled image where '0' is background, '1' denotes a boundary between plant and background, and '2' denotes a boundary between overlapping leaves (which can be more than 1-pixel thick). This separation may facilitate training specialized boundary detectors. The dataset consists of pairs of plant images and these indexed images with boundary annotations, as 256 from Ara2012, and 62 from Ara2013 (Canon).

Evaluation criteria: Typical criteria such as precision and recall are suggested and those not penalizing small local misalignment, suited for evaluating performance of boundaries between leaves (e.g., the MHD and

its learning-based simplification discussed in Chapter 7).

Classification and regression

Task: While phenotyping typically occurs in forward hypothesis testing scenarios, recently reverse hypothesis and association studies have received attention and fine-grained categorization is particularly useful. In this case phenotyping traits are recorded and are correlated with genotyping information to identify relationships among them (data mining) [231]. This is particularly useful in the case of treatments, cross hybridizations, and other processes that may affect directly or indirectly (e.g., via silencing and other epigenetic functions) the genetic code of plants [304]. Therefore, given a plant image, it is of great interest to characterize the plant, i.e. plant age or development stage, other cultivars possessing similar traits, what possible treatment it has undergone. This may be done at plant level or may be even possible at leaf level. When these characteristics are distinct this is considered as a recognition and classification problem. However, if we are interested in percentile similarities, likelihood and regression frameworks can be employed. Here we consider and provide datasets for two cases: mutant recognition (classification), and age regression.

Dataset: For mutant recognition we release individual plant images and a text list denoting per each row image name and genotype. For mutant classification 100 from Ara2013 (Canon) and 100 from Ara2013 (Rpi) are currently available. We assume the only input to be image data with mutant type to be predicted values. For age regression, we release individual plant images and a text list denoting per each row image name, mutant type, and the age in hours of the plant since germination. For this task 100 images from Ara2013 (Canon), and 100 from Ara2013 (Rpi) are currently available. Notice that the age regression task when mutant or treatment information is not available or treatment is not fixed is extremely complex since each mutant has different growth rates in response to treatment, so in some sense the algorithm must be able to tell by appearance how to characterize age by estimating as a hidden variable the mutant. Hence, we assume that inputs to this learning problem are: images and mutant type information to predict age.

Evaluation criteria: Due to the diversity of problems considered in this category, for classification problems precision and recall are recommended, and for regression problems mean absolute error and mean squared error between predicted and ground-truth measures are encouraged.

3.5 Summary

This chapter described our affordable and versatile sensing solution for plant phenotyping based on the Raspberry Pi single-board computer and its camera module. For comparison, we also devise a solution based on a commercial camera. We setup a growth chamber and deploy our sensors to acquire image datasets of growing Arabidopsis plants in a phenotyping context. Subsequently, we manually annotate part of this data, which we use throughout the thesis to validate all the proposed approaches.

In the next chapter we will describe our approach for the automated analysis of plant images collected during phenotyping experiments, e.g., using the affordable sensing solution presented here. To favor adoption of our solution, for the design of the sensor we use commercial hardware and open source software.

As a benefit to the scientific community, we release in the public domain specially formatted datasets and annotations suited for a series of computer vision and learning tasks at different levels of granularity. The imaging data is accompanied by appropriate functions implementing evaluation criteria. We will explore the adoption of specialized annotation tools and algorithms [109], which when implemented in web-based environments would permit the crowd-sourcing of annotations [111, 160]. We will be also investigating additional evaluation criteria as we obtain feedback from the community.

We do hope that our publicly available datasets and future augmented versions, will be adopted by the computer vision community as well (as with the Pascal VOC [90, 91], or the BBBC [187]). Our datasets can be used to learn suitable image statistics [128], adapt and test counting algorithms with [94] and without temporal information [14, 95, 174], segmentation algorithms [21, 45, 270, 314], multi-label segmentation [57, 126, 127, 226, 227] or detection [23] approaches, and others. Additional depth information as can be computed from a pair of images with different focus [92, 294] may further facilitate segmentation [73, 316]. Using the presented datasets, image-based plant phenotyping will evolve in parallel to (and benefit from) advances in computer vision, by tracking the performance of approaches referencing these data. More importantly, it will also introduce this societally important application to a broader audience.

4

Plant image analysis with learning and active contours on the cloud

4.1 Introduction

As of now the majority of solutions for automated image-based plant phenotyping (cf. Chapter 2) pose strict experimental conditions and restrictions on experimental design to ease the complexity of the analysis task that follows.

In this chapter we propose a method for the segmentation and the automated analysis of time-lapse images from plant phenotyping experiments in a general laboratory setting, that can adapt to scene variability (Section 4.2). To accurately segment the plants from the background we propose a vector valued level set formulation that incorporates features

This chapter is partly based on:

- M. Minervini, M. M. Abdelsamea, S. A. Tsaftaris, "Image-based plant phenotyping with incremental learning and active contours," *Ecological Informatics*, vol. 23, pp. 35–48, Sep. 2014, Special Issue on Multimedia in Ecology and Environment.
- M. Minervini, A. Fischbach, H. Scharr, S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," submitted for publication to *Pattern Recognition Letters*, 2015.

of color intensity, local texture, and prior knowledge. Prior knowledge is introduced to render our solution adaptable to a variety of different settings, and abstract it from a specific setup or plant type. We incorporate the prior using a plant appearance model based on a Gaussian mixture model, which utilizes incrementally information from previously segmented instances. We validate our approach using Arabidopsis images and comparisons with state-of-the-art methods in the literature show that the proposed method is able to handle images with complicated and changing background in an automated fashion. An accuracy of 96.4% (Dice Similarity Coefficient) is observed, which is higher than other methods used for comparison.

To facilitate adoption, our solution is accompanied by an easy to use graphical user interface and the software is available to the scientific community as a web application on the iPlant Collaborative [111] cloud platform (Section 4.3). In Section 4.4 we utilize our system to analyze images including several genotypes of Arabidopsis (from the Ara2013 dataset, cf. Chapter 3), demonstrating its effectiveness as a phenotyping tool. Finally, in Section 4.5 we offer preliminary results on leaf segmentation and counting, which we envision as future directions to extend the phenotype extraction capabilities of our analysis platform.

4.2 Plant delineation from complex background

In this section we propose and test, an algorithm and a software system for the automated segmentation and analysis of time-lapse top-view plant images from phenotyping experiments of Arabidopsis rosettes. Example images of Arabidopsis rosettes are shown in Figure 3.5 on page 42. We use data from the Ara2012 dataset described in Chapter 3, to demonstrate the challenging aspects of the problem of plant segmentation and to test the robustness of our proposed solution to the complexity of the scene.

While our method focuses on the algorithm and the software solution, as an example that relates it to phenotyping experiments we measure plant growth, estimated through projected rosette area of 19 Arabidopsis Columbia (Col-0) wild-type plants. The system involves minimal user interaction (necessary to establish the statistical experiments that follow) and at every time instance (in our context a digital photograph) segments the plants in images that contain many specimens of the same species (an example input image is shown in Figure 3.5a on page 42). We rely on a

combination of level set and learning based segmentation to incrementally incorporate information from previous time instances, allowing us to adapt to changes in the scene. We learn an appearance model of the plant relying on Gaussian Mixture Models (GMM) of color and texture features collected from previously segmented instances. For each unseen instance we classify each pixel in the image to obtain a probabilistic map of pixels most likely belonging to a plant.

The probabilistic output assists the localization of a plant within an image (many plants exist in an image). Each individual plant is then segmented using a new active contour model that incorporates probabilistically weighted (using the model output) features of pixel intensity and texture. Once each plant has been segmented, several measurements relevant to the study of plant growth are extracted. Finally, the plant appearance model is updated by re-training the GMM to include the newly processed data in an online and incremental fashion.

Our inclusion of the iterative learning aspect allows us to handle images with complicated and changing background in an automated fashion, which challenge currently available solutions. Overall, the proposed approach obtains an accuracy in segmentation higher than 96.4% (Dice Similarity Coefficient) and is not affected by challenges in the scene. While here it is tested on a single plant species (*A. thaliana*), the fact that we do not employ shape driven models and we do not rely on fully supervised classification (trained on a large dataset) favors the deployment of the proposed solution for the study of different plant species in a variety of laboratory settings. The automated phenotyping solution proposed in this chapter improves upon the accuracy results obtained by the state of the art in plant phenotyping, even in an environment that is not strictly controlled, thus accommodating a broader range of experimental scenarios – handling multiple plants simultaneously without an explicit scene arrangement and with minimal user intervention.

In the following, first we describe the proposed approach and discuss the requirements imposed by the application (i.e. the segmentation of rosette plants). Next, we present experimental results on the use of our approach for the study of Arabidopsis plant growth, with respect to a reference method and a specialized software proposed in [71]. We also compare with recent innovation in image segmentation. Finally, we demonstrate how our solution can accurately calculate projected leaf area for growth estimation.

4.2.1 Image segmentation with incremental learning and active contours

As outlined in the previous section, the current state of the art in the automated analysis of images from phenotyping experiments is limited to few commercial solutions and freely available software. There is a great interest in developing an approach that:

- can accommodate most laboratory settings;
- does not rely on explicit scene arrangement;
- can handle multiple plants in the same image;
- can tolerate low spatial resolution and out of focus discrepancies, attributed to a fixed broad field of view (thus accommodating affordable sensors);
- is robust to changes in the scene;
- requires minimal user interaction (for experimental setup and training);
- is scalable to large population sizes or high sampling rates;
- offers high accuracy and consistent performance; and
- is automated and high throughput.

Such an approach, when combined with affordable sensing hardware (cf. Section 3.3.1) would provide a truly affordable and easy to deploy phenotyping collection and analysis system that can satisfy the needs of most laboratories. Here we propose a software solution that satisfies the above requirements using several innovations, which have not been previously considered in the context of plant segmentation. First of all we rely not only on color features, but also on texture information extracted from the images to effectively discern foreground (plant) from background. To increase robustness and offer consistent performance we incorporate a learning component in our solution. To favor scalability, we learn the color and texture appearance of plants using a multi-dimensional Gaussian Mixture Model (GMM), which minimizes additional computational and storage requirements. The appearance model is learned incrementally after each instance has been automatically segmented, thus reducing the need for prior labeling significantly. Finally, we propose a new vector

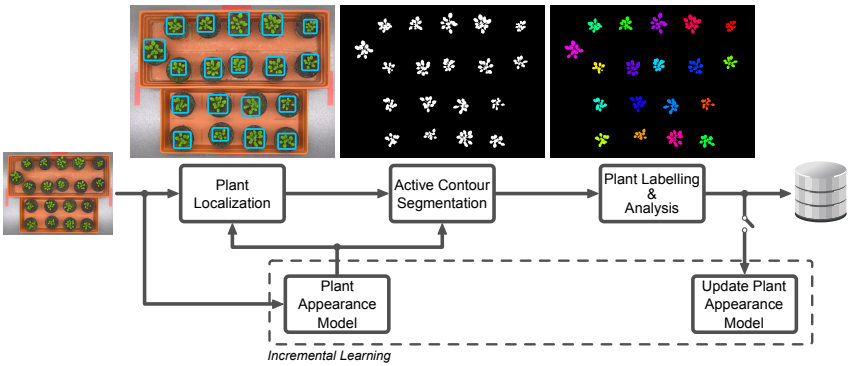


Figure 4.1: Graphical representation of the proposed approach.

valued level set formulation to segment each plant in the image by incorporating all features and prior information, in a joint energy optimization framework.

Figure 4.1 provides a general view of the proposed system. Once a new image is acquired, the first step consists in localizing plant objects in the image and obtaining an approximate plant segmentation (which is used as an initialization for the next step). Subsequently, each region containing a plant is segmented with an active contour model to accurately delineate the plant from the background. Both of these steps can take advantage of the plant appearance model employed here. Each segmented plant is labeled with an identifier coherently with previous images. This permits us to follow each plant individually across time and link all new measurements and analysis output to the corresponding plant. Finally, the processed data, the segmentations, and several indexes of interest to the plant community are added to a repository. Additionally, the features of foreground are stored and used to update the plant appearance model.

Overall, the proposed solution is designed to work automatically and minimize user interaction without compromising accuracy and generalization. The user at the beginning of the experiment informs the application about the number of plants present in the scene and their grouping if any (e.g., which mutants are present). Our only assumption regarding the composition of the scene, is that plants should not be touching (this requirement is reasonable and common among all phenotyping platforms to facilitate object separation). To use the plant appearance model, some

initial training is necessary. This initial training can be provided by the user offline or alternatively the user can operate aspects of the proposed solution without the model. This can take advantage of the fact that some scenes can be less complex (e.g., plants are smaller, or moss has not appeared yet). The user is prompted to make corrections to the segmentations (if necessary) on those (early-training) images, and subsequently the model is trained based on this output. The user could also adjust some parameters regarding the model during this training part. Afterwards the plant model is updated at each iteration and is used at each step, without any user supervision.

In the following paragraphs we present each aspect of the proposed system in detail, providing insights into both mathematical formulation and design choices.

Image features

From a computer vision perspective, a laboratory setup for plant phenotyping experiments presents several challenges such as neon light illumination, water reflection, shadows, and moss (some examples are shown in Figure 2.2 on page 17), contributing to noise and scene complexity. To eliminate issues of non uniform illumination (due to lighting distortion from neon lights and shadowing), we convert the RGB color space to the 1976 CIE $L^*a^*b^*$ color space [143, 144]. We use the a^* component which determines the color position between green and red, and the b^* component (colors between blue and yellow) as color features.

While relying only on the intensity of pixels may appear adequate, there are several conditions that would challenge this assumption. For example, the color intensity of moss can be very close to that of plant leaves. This motivates the utilization of texture features along with intensity to describe foreground and background. Several texture representations can characterize texture content at each pixel location, taking into account its neighborhood [136, 307]. Some of these approaches (e.g., co-occurrence matrices, Haralick features, and Gabor filters) result in a multi-resolution, multi-scale representation of local texture characteristics. This in turn introduces additional feature dimensions.

In this work we use a texture detection filter obtained via gray scale morphological operations as a local texture descriptor. To detect high texture regions in an image I, the response of a pillbox filter is linearly combined with a Difference of Gaussians (DoG) filter. A pillbox filter

is a circular uniform kernel H_ρ of radius ρ. The DoG filter operates by subtracting a blurred version of an intensity image from another blurred version of the same image, where the different blurring is obtained by convolving with a pair of Gaussian kernels (K_{σ_H}, K_{σ_L}) of different standard deviation (σ_H, σ_L respectively). The filtered output is defined as:

$$f\left(I; \rho, \sigma_H, \sigma_L\right) = H_\rho * I_i + \left(K_{\sigma_H} * I_j - K_{\sigma_L} * I_j\right), \tag{4.1}$$

where I_i and I_j are channels of the same multi-channel image I, and $*$ denotes the discrete 2D convolution operator. The response of the "Texture From Blurring" (TFB) filter finally is:

$$f_{\mathrm{TFB}}\left(I; \alpha, \rho, \sigma_H, \sigma_L\right) = \exp\left(-\alpha \left| f(I;\, \rho,\, H,\, L)\right|\right), \tag{4.2}$$

where α sets the decrease rate. The pillbox filter responds to smooth regions versus high texture regions, while the DoG filter responds to edges. In the context of plant images, the combination of these two operations permits the separation of high texture regions (e.g., belonging to moss or earth) from smooth regions (e.g., belonging to leaves and stems).

Our system thus relies on color information (a^*, b^*) and texture features (TFB) as a feature space to perform the image analysis tasks described in the following sections. Figure 4.2 shows examples of the features employed as extracted from an image. (For these examples TFB was found using $I_i = a^*$, $I_j = L^*$, $\sigma_H = 4$, $\sigma_L = 1$, $\rho = 3$, and $\alpha = 1/50$). Observe, their effectiveness in discerning between the object of interest (plant) and other background regions.

Plant appearance model

To safeguard our method from scene changes and increase its accuracy, we design the algorithm such that it can utilize the segmentation outcomes of previous instances. Although we will present in subsequent sections the details of our segmentation strategy, here we discuss that we can learn a plant model given the features of an image and its available segmentation into foreground and background regions. This model classifies pixels in an unseen instance according to their likelihood of belonging to a plant. Thus, with this process, we learn incrementally how a plant appears, and we feed back to the system this information.

Given a set of segmented images, there are several reasons to avoid training a sophisticated supervised classification algorithm that assigns

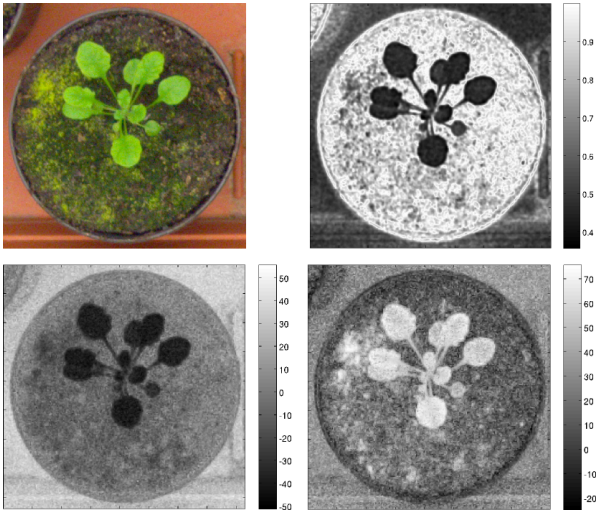


Figure 4.2: Examples of the image features used in our proposed system. Shown are: (top left) the original RGB image; (top right) texture descriptor; (bottom left) a^* and (bottom right) b^* color components.

pixels to either the foreground or the background class. (i) It should allow for fast training to permit fast updates, (ii) it should be robust towards (possibly) mislabeled data which are automatically produced by previous time instances, and (iii) it should have low data storage requirements. Classical supervised classification algorithms retain both foreground and background features and thus require additional storage and also their online (incremental learning) implementation is not straightforward. The solution we adopt here uses only foreground information and features in an appearance model.

Our solution is given an input image (and the features that can be extracted from it) and a segmentation mask identifying plants to learn the multi-dimensional distribution of the feature space using a multivariate Gaussian Mixture Model formulation. Accordingly, the density function for an observation (a pixel location with its features) with the d-dimensional feature vector $x \in \mathbb{R}^d$ is:

$$p(x|\Theta) = \sum_{j=1}^{M} \pi_j p(x|\Theta_j), \tag{4.3}$$

where $M > 1$ is the number of components of the mixture, $\Theta_j = (\mu_j, \Sigma_j)$ is the set of parameters defining the j-th component of the mixture (i.e. mean μ_j and covariance matrix Σ_j), and π_j is the prior probability of pattern x belonging to the j-th component, such that $0 \leq \pi_j \leq 1$, for $j = 1, \ldots, M$, and $\sum_{j=1}^{M} \pi_j = 1$. Each component of the mixture is characterized by a multivariate Gaussian distribution:

$$p(x|\Theta_j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} \cdot \exp\left[-\frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j)\right]. \tag{4.4}$$

Given the density function, the log-likelihood function is defined as:

$$\ell(\Theta) = \sum_{i=1}^{N} \log\left(\sum_{j=1}^{M} \pi_j p(x_i|\Theta_j)\right), \tag{4.5}$$

where N is the number of available data points.

We maximize the log-likelihood function and estimate the unknown parameters of the distribution and the pixel's prior $\Theta = (\pi_j, \mu_j, \Sigma_j)$, for $j = 1, \ldots, M$, using the Expectation-Maximization algorithm [303], to fit the GMM model to the available plant data (foreground). To eliminate the need to store the collection of all feature vectors x for all seen images, we update the GMM after each segmented image in an online fashion as in [72]. Thus, our storage requirements are limited to storing original images, their segmentation outputs, and relevant metadata (see Section 4.2.1).

Once the model is learned, for an unseen image we extract its features and obtain the probability of each pixel belonging to the plant model, by evaluating Eq. (4.3). The output of the plant model applied in a new image is a probability map P (with values in the interval $[0, 1]$), which contains an estimate of the probability of any pixel in the image belonging to a plant.

Plant localization

While some methods assume certain grid arrangement of pots [71, 306], one of our design criteria is not to impose conditions on the scene and the arrangement of the rosettes. This necessitates a process to localize plant objects in the scene. This step isolates rectangular regions of interest (of reduced size) containing plants, and estimates approximate plant segmentations which are then subsequently used as input to higher-complexity steps (e.g., the segmentation presented in the next section).

Although several approaches can be employed, for simplicity we use a K-means (with $K = 2$) clustering algorithm operating in the feature space discussed previously, to cluster a pixel as plant or background in the original image. To initialize the K-means we adopt the following schemes to get a good choice of initial cluster centroids. In the absence of a plant model or prior information we calculate initial centroids automatically from histogram thresholding in the Excess Green color space [112]. We transform the RGB to Excess Green (*ExG*) domain, using $ExG = 2G - R - B$, where R, G and B are the three components of the RGB color space. Subsequently, we use Otsu's thresholding [233] to identify a single threshold. We use pixel locations that have an *ExG* value higher than the threshold as foreground and estimate their average features (average color and texture). All other pixels are considered background and we estimate their average feature values as well. These averages are used as initial cluster centroids. Alternatively, in the presence of a plant appearance model we take advantage of it to get a good choice of initial cluster centroids. In both cases, we threshold the *ExG* or the probability map P, respectively, using a fixed threshold and consider all pixels above this threshold as foreground, while pixels below this threshold are considered as background. The initial cluster centroids are found by averaging foreground and background pixels respectively. We should note that we also allow the user to optionally calibrate and provide initial cluster centroids offline.

Subsequently, after the K-means has converged, we find a square bounding box that contains each plant object and output this region. A two cluster K-means performed on a complex image (i.e. plants surrounded by several other distinct objects) exhibits a bias towards over-segmentation; however, this behavior provides sufficient guarantees that all parts of a plant are included in its bounding box.

We also output a binary mask that serves as an approximate plant segmentation to be used as input to the active contour segmentation. Depending on the mode of operation, this binary mask can either be obtained from the K-means clustering or from the thresholded probability map. The end result of this process is a collection of rectangular regions of interest, whose union reconstructs the original scene, and a collection of rough plant segmentations.

A probabilistic vector valued active contour model

While the previous step provides a collection of rectangular regions of interest and an initial rough segmentation, the goal of this step is to obtain a highly accurate segmentation of each plant. Operating on a smaller portion of the image allows us to use more complex algorithms, which would not have been efficient and effective in the full image. Our motivation for employing an active contour method for object segmentation is its ability to model arbitrarily complex shapes and handle implicitly topological changes such as merging and splitting. Thus, the level set based segmentation method can be effectively used for extracting a foreground layer with fragmented appearance, such as leaves of the same plant.

Since we want to take advantage of the existence of multiple image features (in the following we refer to them as channels) we build upon the Chan-Sandberg-Vese active contour model for vector valued images [46]. The innovative aspects of our approach, compared to [46] and other similar works, are that: (a) we introduce the median of the foreground distribution in addition to its mean in the overall energy functional, and (b) we incorporate a probabilistic prior in our formulation.

Recently it was shown that using the median in region based level sets can adapt better to images where the object of interest maybe composed from different intensity classes [2]. Here we use the median as descriptor of non-symmetry in the distribution of the foreground for each channel thus, increasing the discriminative power between the foreground and background distributions with minimal computational overhead.

Without prior knowledge it is known that active contours may also erroneously segment regions that appear to have high statistical similarity with the object of interest. Chen et al. [53] use a non-parametric technique (namely Kernel Density Estimation) to model the shape variation, previously proposed by Cremers et al. [65] to incorporate both shape and intensity prior information. Leventon et al. [176] propose to incorporate prior information about intensity and curvature profile of the structure using a training set of images and boundaries. They model the intensity distribution as a function of signed distances from the object boundary, rather than modeling only the intensity of the object as a whole. On the other hand, Lee et al. [171] propose a supervised active contour model, which estimates a multivariate mixture density function from training samples using either parametric or non-parametric density estimation methods. This density is used to measure how likely each image pixel is to

be an element of each subset in a new probabilistic active contour formulation. We rely on a new mechanism to incorporate prior information. We use the information provided by the appearance model to weigh pixels according to the probability of belonging to the foreground or background. Thus, we do not rely on shape (which has its own complications as we discussed previously) and we decouple the prior knowledge from the active contour model increasing the flexibility of our approach.

The overall energy functional for the proposed model consists of two parts: an "Image-based Feature" term E^{IbF}, which utilizes the intensity information of each channel, and a "Prior" term E^{P} to incorporate the prior knowledge obtained based on the plant appearance model described previously.

The E^{IbF} term following the formulation in [46] is defined as:

$$\begin{aligned} E^{\mathrm{IbF}}\left(C, \boldsymbol{c}^{+}, \boldsymbol{m}^{+}, \boldsymbol{c}^{-}\right) = & \int_{in(C)} \frac{1}{N} \sum_{i=1}^{N} \lambda_i^{+} e_i^{+}(z) dz \\ & + \int_{out(C)} \frac{2}{N} \sum_{i=1}^{N} \lambda_i^{-} e_i^{-}(z) dz \,, \end{aligned} \tag{4.6}$$

$$e_i^{+}(z) = \left|I_i - c_i^{+}\right|^2 + \left|I_i - m_i^{+}\right|^2 , \tag{4.7}$$

$$e_i^{-}(z) = \left|I_i - c_i^{-}\right|^2 , \tag{4.8}$$

where z denotes a pixel location in an image channel I_i, $i = 1, \ldots, N$, λ_i^{+} and λ_i^{-} define the weight of each term (inside and outside the contour), $\boldsymbol{c}^{-}$ is the vector valued representation of the mean for each channel outside the contour, and $\boldsymbol{c}^{+}$ and $\boldsymbol{m}^{+}$ are the vector-valued representations of the mean and median respectively for each channel inside the contour. Our key difference compared to [46] is the introduction of a term corresponding to the median. The way we estimate these statistical quantities will be described shortly.

Following standard level set formulations [46] we replace the contour curve C with the level set function ϕ [346]:

$$\begin{aligned} E^{\mathrm{IbF}}\left(\phi, \boldsymbol{c}^{+}, \boldsymbol{m}^{+}, \boldsymbol{c}^{-}\right) = & \int_{\phi \geq 0} \frac{1}{N} \sum_{i=1}^{N} \lambda_i^{+} e_i^{+}(z) dz \\ & + \int_{\phi < 0} \frac{2}{N} \sum_{i=1}^{N} \lambda_i^{-} e_i^{-}(z) dz. \end{aligned} \tag{4.9}$$

The vectors $\boldsymbol{c}^+$, $\boldsymbol{m}^+$, and $\boldsymbol{c}^-$ are defined in similar fashion to other intensity driven active contour models as statistical averages and medians:

$$\begin{cases} \boldsymbol{c}^+(\phi) = \text{average}(I_i \in \phi(z) \geq 0), \\ \boldsymbol{m}^+(\phi) = \text{median}(I_i \in \phi(z) \geq 0), \\ \boldsymbol{c}^-(\phi) = \text{average}(I_i \in \phi(z) < 0), \end{cases} \tag{4.10}$$

for each channel I_i, $i = 1, \ldots, N$, inside or outside the contour.

Using the level set function ϕ to represent the contour C in the image domain Ω, the energy functional can be written as follows:

$$\begin{aligned} E^{\text{IbF}}\left(\phi, \boldsymbol{c}^+, \boldsymbol{m}^+, \boldsymbol{c}^-\right) = &\int_\Omega \frac{1}{N} \sum_{i=1}^{N} \lambda_i^+ e_i^+(z) H(\phi(z)) dz \\ &+ \int_\Omega \frac{2}{N} \sum_{i=1}^{N} \lambda_i^- e_i^-(z)(1 - H(\phi(z))) dz, \end{aligned} \tag{4.11}$$

where H is the Heaviside function.

By keeping $\boldsymbol{c}^+$, $\boldsymbol{m}^+$, and $\boldsymbol{c}^-$ fixed, we minimize the energy function $E^{\text{IbF}}\left(\phi, \boldsymbol{c}^+, \boldsymbol{m}^+, \boldsymbol{c}^-\right)$ with respect to ϕ to obtain the gradient descent flow:

$$\frac{\partial \phi}{\partial t} = \zeta^{\text{IbF}} = \delta(\phi) \left[-\frac{1}{N} \sum_{i=1}^{N} \lambda_i^+ e_i^+(z) + \frac{2}{N} \sum_{i=1}^{N} \lambda_i^- e_i^-(z) \right], \tag{4.12}$$

where δ is the Dirac delta function.

After Eq. (4.12) converges, the evolving curve C will separate the object from the background based on the non-symmetric property of foreground distribution. However, when the background and foreground are not easily separable without prior knowledge the level set may converge to a wrong boundary.

To introduce prior knowledge we require matrices P_{in}, P_{out} (the size of the image) where $P_{\text{in}}(z) \equiv p(z \in \Omega_1)$, i.e. the probability of pixel at location z belonging to the foreground class Ω_1, and naturally $P_{\text{out}}(z) \equiv p(z \in \Omega_2)$, i.e. the probability of pixel at location z belonging to the background class Ω_2. Notice that $P_{\text{in}}(z) + P_{\text{out}}(z) = 1$.

In the proposed active contour formulation we utilize both P_{in} and P_{out} to weigh each channel individually, hence the prior energy term can

be described as follows:

$$E^{\mathrm{P}}\left(\phi, \boldsymbol{c}^{\oplus}, \boldsymbol{m}^{\oplus}, \boldsymbol{c}^{\ominus}\right) = \int_{\Omega} \frac{1}{N} \sum_{i=1}^{N} \lambda_i^{\oplus} e_i^{\oplus}(z) dz + \int_{\Omega} \frac{2}{N} \sum_{i=1}^{N} \lambda_i^{\ominus} e_i^{\ominus}(z) dz \,, \tag{4.13}$$

$$e_i^{\oplus}(z) = \left|I_i \cdot P_{\mathrm{in}} - c_i^{\oplus}\right|^2 + \left|I_i \cdot P_{\mathrm{in}} - m_i^{\oplus}\right|^2, \tag{4.14}$$

$$e_i^{\ominus}(z) = \left|I_i \cdot P_{\mathrm{out}} - c_i^{\ominus}\right|^2, \tag{4.15}$$

where now $\lambda_i^{\oplus}$ and $\lambda_i^{\ominus}$ define the weights of each term, $\cdot$ denotes pointwise multiplication, $\boldsymbol{c}^{\oplus}$, $\boldsymbol{m}^{\oplus}$, and $\boldsymbol{c}^{\ominus}$ are defined as follows:

$$\begin{cases} \boldsymbol{c}^{\oplus} = \mathrm{average}(I_i \cdot P_{\mathrm{in}} \in \phi(z) \geq 0), \\ \boldsymbol{m}^{\oplus} = \mathrm{median}(I_i \cdot P_{\mathrm{in}} \in \phi(z) \geq 0), \\ \boldsymbol{c}^{\ominus} = \mathrm{average}(I_i \cdot P_{\mathrm{out}} \in \phi(z) < 0), \end{cases} \tag{4.16}$$

for $i = 1, \ldots, N$.

The level set formula based on the prior term is defined as follows:

$$\frac{\partial \phi}{\partial t} = \zeta^{\mathrm{P}} = \delta\left(\phi\right) \left[-\frac{1}{N} \sum_{i=1}^{N} \lambda_i^{\oplus} e_i^{\oplus}(z) + \frac{2}{N} \sum_{i=1}^{N} \lambda_i^{\ominus} e_i^{\ominus}(z) \right]. \tag{4.17}$$

To derive our final joint level set functional form, we follow the approach in [46] and replace $\delta\left(\phi\right)$ in Eqs. (4.12) and (4.17) by $|\nabla\phi|$. The former has a small effective range while the latter has the effective range of the whole image. Furthermore, in order to efficiently regularize the level set evolution, we convolve the level set function with a Gaussian kernel [342]. Finally, the level set formulation of our model combining Eqs. (4.12) and (4.17) becomes:

$$\frac{\partial \phi}{\partial t} = |\nabla\phi| \left[(1-\lambda)\zeta^{\mathrm{IbF}} + \lambda\zeta^{\mathrm{P}}\right], \tag{4.18}$$

where λ balances the influence of E^{IbF} and E^{P}. A larger value of λ emphasizes the effect of the E^{P} term versus the E^{IbF} term, whereas a smaller λ reduces the effect of the prior energy term. In particular, $\lambda = 0$ implies that the model is utilizing image-based features only, without relying on prior knowledge.

Implementation The main steps of our segmentation model can be summarized as follows. For each region of interest obtained from the localization step:

1. initialize the level set function ϕ to be binary;
2. evolve the level set function according to Eq. (4.18);
3. smooth the level set function with a Gaussian kernel;
4. if the curve evolution has converged, output a binary mask of the segmentation, otherwise return to step 1.

While the previous steps are general, we outline below details of our implementation for this particular application. We use as I_1, I_2 respectively the a^* and b^* components of the $L^*a^*b^*$ color space, which is regarded as representing global information. Texture features are encoded in the channels $I_3, \ldots, I_N$ as local information since they describe local neighborhood structure [324]. Furthermore, since the prior knowledge used here is learned only from foreground objects we use as P_{in} the output of the plant appearance model, and as $P_{\text{out}} = 1 - P_{\text{in}}$. To accommodate that the prior knowledge of the background is less reliable (since we did not train on the background) we use a smaller weight $\lambda_i^{\ominus}$. Alternatively, if we did not want to provide any prior knowledge for the background we could model P_{out} to follow a uniform distribution as suggested in [171]. Finally, to initialize the level set we use the output of plant localization.

Plant labeling and analysis

After we have obtained all segmentations of foreground from the active contour model, we recompose the original scene (containing all plants), and we obtain a binary representation of all plant objects in the original scene. In phenotyping experiments, plants usually belong to distinct groups, e.g., mutants of the same species or specimens undergoing different treatments. Therefore, plants need to be labeled and followed individually across time in order to maintain correspondence of individual measurements for each plant. The goal of this step is to assign a unique label to pixels of the same plant (intra-frame accuracy) and to consistently assign that label to the same plant across time (inter-frame accuracy).

The binary mask representing foreground objects is composed by a number of connected components. Due to possible errors in the segmentation (e.g., under-segmentation of the stems due to lack of resolution),

portions of the same plant may result in disconnected objects (e.g., a leaf may not be connected to its originating plant).

To address this problem we take advantage of the radial shape of Arabidopsis rosettes, and label disconnected objects in groups that minimize the Euclidean distance from a centroid. Let N_{subj} be the number of plants in the scene (recall that this parameter is fixed by the user). We perform the task of intra-frame labeling by finding the centroids of the N_{subj} largest connected components. These are then used to initialize a K-means clustering on the pixel coordinates, with $K = N_{\mathrm{subj}}$. Thus, each pixel in the foreground is labeled according to which centroid is closer in the Euclidean sense. To obtain the final label for all pixels in a given component, a connected component having different labels (e.g., a leaf of one plant close to another plant) is assigned a single label with a majority vote strategy.

Having available plant centroids and labels from the previous image, coherency in inter-frame labeling is maintained by assigning a plant in the current image the label of the closest plant (in the Euclidean sense) in the previous image. This approach tolerates small shifts of pots (which can occur when staff are handling the experiment or when plants are watered), but not significant movements or shuffling. In our setting we assume that plants do not touch and it is the user's responsibility to arrange pots in the scene with enough distance between each other (an assumption common among many phenotyping analysis platforms). To accommodate touching plants a plant shape model is necessary, which can be rather complex and specific to a particular plant species.

After a successful segmentation, several visual phenotypes can be extracted. For example, plant growth is estimated with several indexes [71] that reflect plant area, roundness, and overall color intensity. Measurements, indexes,and plant identifiers are written in a tabulated format that can be imported by several statistical analysis and plotting software.

4.2.2 Results and discussion

Experimental setup

We implement our system in Matlab (release 2011b), on a machine equipped with Intel Core 2 Duo CPU E8200 2.66 GHz and 4 GB memory, running 64-bit Linux. We test our approach on images from the Ara2012 dataset (cf. Chapter 3), covering a good range of challenging situations, such as water

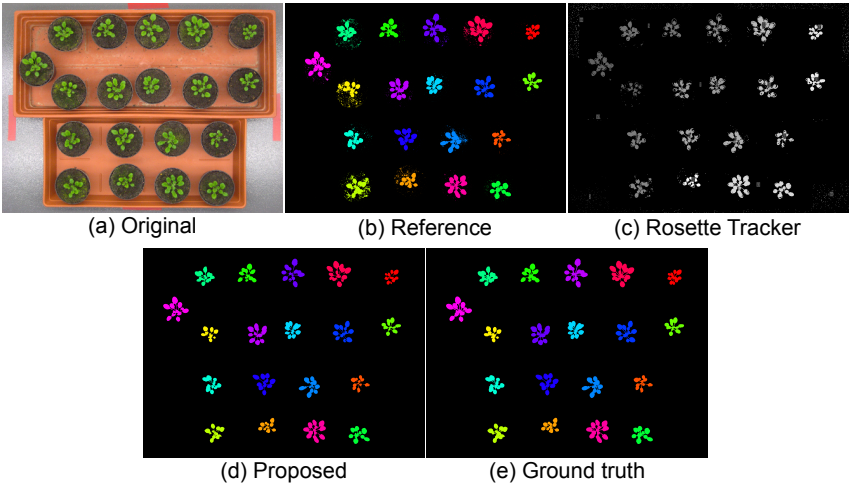


Figure 4.3: (a) A sample image from the dataset, (e) its ground truth segmentation obtained manually, and (b)-(d) the results of the systems being compared, where plant labeling is denoted by color.

reflection and moss. The scene consists of top-view images of $N_{\text{subj}} = 19$ *Arabidopsis thaliana* (Col-0) wild-type rosettes, acquired over a period of 12 days. Figure 4.3a shows an example image from the dataset, illustrating the arrangement of the plants and the complexity of the scene.

As described in Chapter 3, all plants are manually annotated to obtain ground-truth segmentations. To quantify the accuracy of the segmentation algorithms, we adopt the following metrics (cf. Section 3.4.3):

$$\text{Precision } (\%) = \frac{TP}{TP + FP} \tag{4.19}$$

$$\text{Recall } (\%) = \frac{TP}{TP + FN} \tag{4.20}$$

$$\text{Jaccard } (\%) = \frac{TP}{FP + TP + FN} \tag{4.21}$$

$$\text{Dice } (\%) = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{4.22}$$

where TP, FP, and FN represent the number of true positive, false positive, and false negative pixels, respectively, calculated by comparing algorithmic result and ground-truth segmentation masks. Precision is the

fraction of pixels in the segmentation mask that matches the ground truth, whereas recall is the fraction of ground-truth pixels contained in the segmentation mask. The Jaccard and Dice similarity coefficients are used to measure the spatial overlap between algorithmic result and ground truth. We express these metrics in percentage, with larger values denoting higher agreement between ground truth and algorithmic result.

We compare the proposed approach with state-of-the-art methods in plant phenotyping and recent image segmentation approaches. The first plant phenotyping method (referred thereafter as Reference) is an approach based on K-means segmentation, due to its widespread adoption in intensity-based plant segmentation. For fair comparison, the K-means is applied on the same intensity and texture features as the proposed approach, and we carefully select initial centroids for each image in the dataset. We use the same plant labeling and analysis procedure as the one proposed here (Section 4.2.1). We also adopt the Rosette Tracker software proposed in [71], which is made available as an ImageJ plugin by the authors. It is operated by performing proper color calibration and by enabling the options for removing moss and clutter, according to the characteristics of our dataset. We provide as input to Rosette Tracker the number of plants in the scene (as required by the software). However since Rosette Tracker assumes that plants are arranged in rows and the imaging axis is parallel to these rows, plants are often mislabeled. This requires manual post-processing to correctly assign labels to plant parts, in order to include Rosette Tracker in the evaluation.

To demonstrate that even state-of-the-art methods in color image segmentation are challenged, we select four algorithms covering a span of recent innovations in computer vision and use their reference software implementations. In particular, we consider the following methods: *gPb*-owt-ucm [12], a segmentation method that relies on contour detection and spectral clustering, providing a hierarchical representation of the image and a final segmentation after user annotation; MSRM [229], an interactive segmentation approach that relies on a new region merging framework to fuse a super-pixel segmentation obtained using [177]; the CoSand [162] method which relies on the temperature maximization of the anisotropic heat diffusion formulation on a graph representation of the image (an initial super-pixel segmentation is obtained using [177]); and finally, although it is not a true segmentation method, we also use SDSP [344], a saliency detection algorithm, because it uses similar features as the proposed approach: it combines priors related to frequency (implemented

using log-Gabor features), color (CIE *L*a*b** color space), and location of the object. For all methods standard parameters as recommended by their respective authors are used; and for the methods relying on interaction, unbiased and automated foreground and background annotations are obtained by skeletonizing and dilating the ground truth mask, as previously done in [194].

Since Arabidopsis plants do not have very pronounced veins and given the imaging resolution of our setup, for the results that follow we adopt the following parameters (which are kept constant throughout the experiments). We use $M = 2$ mixtures for the GMM implementing the plant appearance model, and only one TFB texture image, obtained using $\sigma_H = 4$, $\sigma_L = 1$, and $\rho = 3$. Thus, we use 3 image features including the *a** and *b** color components and the TFB texture feature. To initialize the contour we use the probability map, by applying a fixed threshold (0.5). The parameters of the active contour formulation are set as follows: $\lambda = 0.6$ (i.e. we rely almost equally on each of the two terms of Eq. (4.18)), $\lambda_i^+ = \lambda_i^- = 1, \forall i$ (in other words each channel I_i in Eq. (4.6) has the same weight), and since we learn only on the foreground, by choosing $\lambda_i^{\oplus} = 1$ (i.e. the channel weights of the foreground), and $\lambda_i^{\ominus} = 0.01$ (i.e. the weights of the background channels), we rely mostly on the foreground prior to drive the curve evolution. We recall that parameters λ_i^+, λ_i^- refer to the image-based feature term, while $\lambda_i^{\oplus}$, $\lambda_i^{\ominus}$ refer to the prior term in the energy functional.

We should note that no morphological operations are performed on the segmentation output of any method, thus the results presented in the following reflect the true output of the algorithms used.

Results

In this section we present the results obtained by evaluating the proposed system on the dataset described previously, in terms of both segmentation accuracy and validity as a tool for plant phenotyping applications. The performance of the proposed system is compared with the Reference method and Rosette Tracker. Then, we also compare the proposed approach with more recent segmentation methods [12, 162, 229, 344]. Finally, to illustrate the novelty of our active contour formulation and the importance of using prior knowledge and texture features, we describe and compare variants of the proposed system.

To illustrate the output of our algorithm, Figure 4.3 shows an example

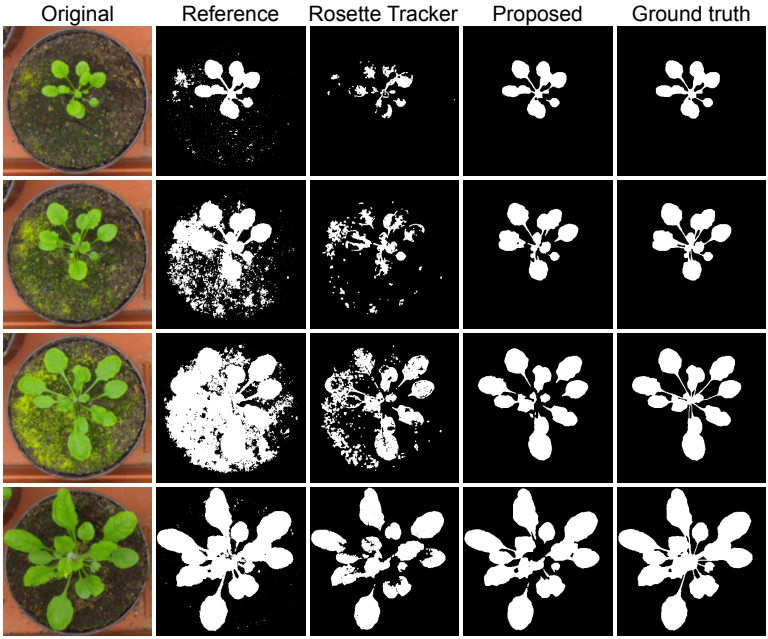


Figure 4.4: Segmentation results of the different systems for the same plant at different time points. Rows: the same plant in different time instances. Columns starting from left to right: the original plant, then the segmentation results of Reference, Rosette Tracker, and Proposed methods respectively, and Ground truth (right most).

image from the dataset. For visual comparison we also include the ground truth segmentation and labeling and include outputs from the Reference and Rosette Tracker approaches. Overall, visually a significant agreement can be observed between the Proposed method and ground truth, whereas Reference appears noisy (e.g., moss and earth in the pots are also included in the foreground). On the other hand, Rosette Tracker reports plant objects in regions belonging to the background, likely attributed to the fact that it does not include a plant localization step. We should note that in our quantitative analysis that follows we manually post-process the results of Rosette Tracker to assign labels correctly to foreground objects and to eliminate wrongly detected regions.

To better appreciate visually the differences in segmentation accuracy

and illustrate the complexity introduced as plants grow or as other objects appear in the background, Figure 4.4 shows segmentation outcomes for a single plant observed across time. It is evident that Reference has a clear bias towards over-segmentation (in other words it cannot separate moss from plant), while Rosette Tracker deals with moss by being conservative, thus under-segmenting, but without completely eliminating moss from the segmentation. Overall, Rosette Tracker shows substantial loss of leaf and stem portions (and thus it will affect growth trend measurements and other phenotyping markers). On the other hand, the Proposed is unaffected by moss or any other noise in the background, while still preserving the entirety of the plant structure with smooth boundaries.

While the previous examples demonstrated visually the accuracy of our method, Figure 4.5 shows segmentation accuracy over time, measured quantitatively using the Dice Similarity Coefficient. It is readily seen, that the performance of Reference starts decreasing very soon as plant (and moss) grow and the scene's complexity increases. After the seventh day its performance seems to improve, but this outcome is misleading. It occurs only because as plants grow they cover most of the pot and consequently most of the moss. Rosette Tracker as well is not robust to changes in the scene. It exhibits an oscillating level of accuracy, depending on events that occur in the scene, e.g., some plants experienced drought during the first days, moss incrementally appeared below some plants, the trays got shifted around the sixth day, and water was present in the trays while acquiring the last image of the dataset (a cut out of which is shown in the bottom of Figure 2.2 on page 17). Such challenging images are included in the dataset to investigate the behavior of the algorithms under different conditions than the ideal case, that are still likely to occur in practice. The proposed system adapts to these changes and accurately segments the plant. Thanks to the plant appearance model that is learned over time and is integrated in the active contour segmentation, the proposed system responds to the aforementioned challenges appearing in the scene, significantly better than the other methods; it maintains a very high accuracy (above 90%) throughout the whole cycle.

From a phenotyping perspective, segmentation accuracy is important, because it reflects the ability of the system to test phenotyping hypotheses. Figure 4.6 shows the growth pattern of a plant, comparing its estimate obtained by automatic segmentations with the ground truth. Growth is estimated from the projected rosette area and the reported results are normalized by plant area at the beginning of the experiment. The actual

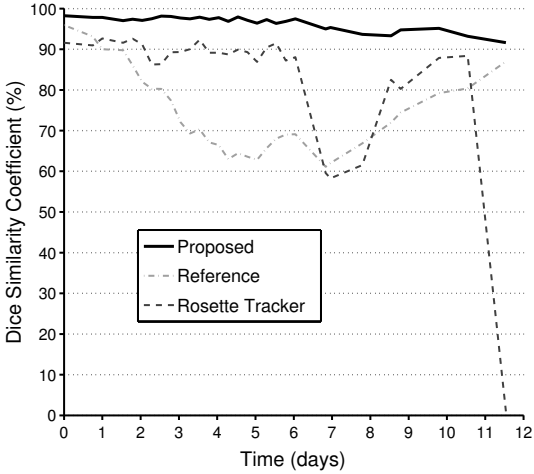


Figure 4.5: Segmentation accuracy over time estimated using the Dice Similarity Coefficient for the Proposed, Reference and Rosette Tracker methods.

growth of the plant appears steady over time and it follows a linear trend. The growth pattern is characterized by periodic saddles, which reflect the natural circadian rhythm of the plant, which is regulated by the light / dark cycles (ambient temperature did not deviate much in our climate controlled office). During the first two days all of the systems provide satisfactory results (i.e. they follow the ground truth closely); however, as the conditions in the scene change, different outcomes can be observed that are not natural and would jeopardize subsequent statistical analysis. Reference reports an unnatural exponential growth of the plant after the second day, due to moss in the pots that is erroneously segmented along with plants.

On the other hand, while Rosette Tracker is less susceptible to moss compared to Reference (see Figure 4.4), there is a tendency to overestimate plant area. After the sixth day Rosette Tracker exhibits an oscillating behavior that over-amplifies any changes in the scene. In particular, the last day, a layer of water in the tray causes Rosette Tracker to severely under-segment the plants.

Most of the approaches for image-based plant phenotyping (as it is the

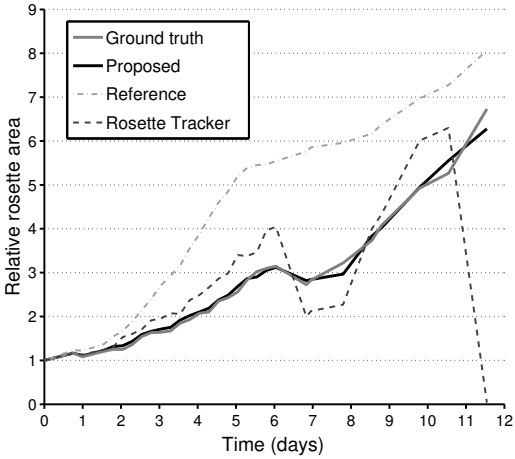


Figure 4.6: A plant's growth pattern obtained using the evaluated systems is compared against the ground truth. Relative area is reported as projected area normalized by the area of the plant in the first day.

case for Reference and Rosette Tracker) can operate appropriately only under strict assumptions on the composition and stability of the scene. When such assumptions are violated, one must discard these images (or the results) which affects the validity and the statistical power of the phenotyping experiment. As it is shown in Figure 4.6, our proposed system closely approximates the actual growth pattern of the plant, proving robust to any changing and challenging conditions in the dataset (e.g., moss, tray shifts, water). The fidelity in recovering the growth pattern is brought to such an extent that the circadian rhythm of the plant can be readily observed.

This level of accuracy is observed across all our dataset and using additional accuracy metrics. Table 4.1 reports averaged results of segmentation accuracy over the whole dataset. The Reference method shows poor accuracy in terms of precision, Jaccard and Dice, and a very high recall value due to the constant over-segmentation (i.e. the plant is fully contained in the segmentation mask, along with large portions of earth and moss from the background). Rosette Tracker appears more balanced in the overall results, although it leaves substantial room for improvement.

Table 4.1: Segmentation accuracy reported for Proposed, Rosette Tracker, and Reference methods, shown as mean (standard deviation).

System	Accuracy (%)			
	Precision	Recall	Jaccard	Dice
Proposed	97.08 (1.83)	95.86 (2.96)	93.17 (3.22)	96.44 (1.76)
Rosette Tracker	88.86 (6.49)	78.83 (24.37)	71.20 (22.29)	80.37 (22.57)
Reference	60.82 (14.55)	99.87 (0.21)	60.74 (14.43)	74.65 (10.62)

Our proposed system achieves very high accuracy values (above 90%) for all of the employed metrics.

Such capability of accurately delineating plant objects in images enables researchers to test phenotyping hypothesis in experiments with subjects from different species or undergoing different treatments, using automated phenotype collection solutions. Primarily, our approach is a tool to study phenotypes related to size, shape, and growth pattern; however, also quantification of any other visual phenotypes observable through digital images (e.g., color variations, flowering time) assumes a segmentation of the plant, which our software provides.

While previously we compared our approach with state-of-the-art methods in plant phenotyping, it is critical to showcase the challenge posed by the problem compared to recent innovations in the field of image segmentation (as outlined in the experimental setup) [12, 162, 229, 344]. Figure 4.7 shows the results of this comparison. Observe that, while on the first image all algorithms produce comparable results, as the images become more challenging (moss or water presence) the segmentation results deteriorate, while our method demonstrates greater accuracy and closely estimates the ground truth. This behavior is observed across all images in the dataset. Using the same performance metrics (shown as average and standard deviation in parenthesis) as previously, the best performing method, *gPb*-owt-ucm [12], has Precision 89.04(6.40)%, Recall 96.20(2.78)%, Jaccard 85.86(5.49)%, and Dice 92.29(3.36)%. This method requires human interaction and our testing shows that its final result depends largely on the detail and precision of such user annotation. Thus, to eliminate any user bias all interactive methods are initialized using morphological operations from the ground truth masks. This provides highly accurate annotations and even with this ideal annotations the interactive methods are unable to match the accuracy of the proposed approach (e.g., the

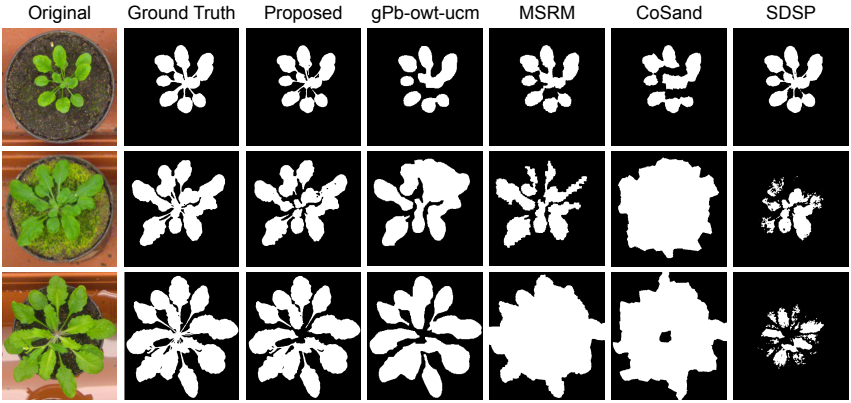


Figure 4.7: Example segmentation results of several state-of-the-art methods. Columns starting from left to right: the original image, the ground truth, then the segmentation results of Proposed method, *gPb*-owt-ucm [12], MSRM [229], CoSand [162], SDSP [344], respectively.

Dice similarity coefficient for our Proposed approach is 96.44%, according to Table 4.1), illustrating the challenges of segmenting plants in complicated background. Finally, most of these methods are computationally demanding both in terms of system memory and time, largely due to the construction of a graph representation.

The proposed approach proves superior to other methods, but it is important from an image processing viewpoint to identify the contribution to accuracy of each component of the approach. We design a series of experiments to highlight the importance of using texture, prior knowledge, and the median (in modeling the distribution of the foreground inside the contour). These scenarios can be easily tested by altering weights in the level set formulation. To this end, we use different values for the controlling parameter λ of E^{IbF} and E^{P} terms and the internal weight parameters λ_i^+, λ_i^-, $\lambda_i^{\oplus}$ and $\lambda_i^{\ominus}$. Table 4.2 summarizes conceptually the different versions used for comparison, while the last column includes the change in parameters. As shown in Table 4.2, the Proposed model refers to our proposed approach, Proposed-a is a version without considering the prior knowledge, while Proposed-b refers to a version without prior knowledge and without the median for the foreground distribution. Finally, Proposed-c is a model version where the prior knowledge, median,

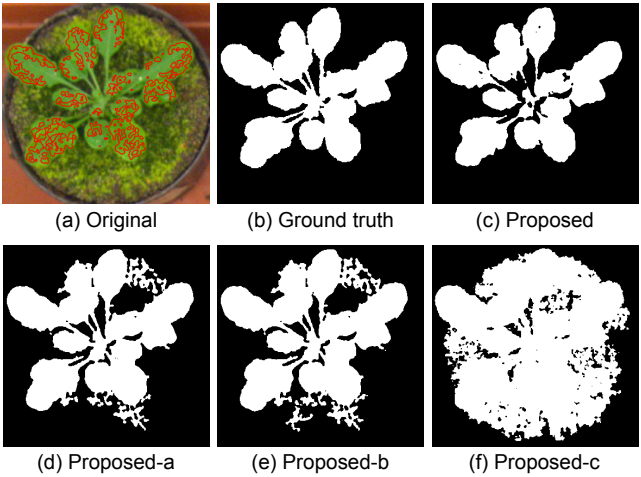


Figure 4.8: Segmentation outputs for the plant image in (a), illustrating the importance of each components of the proposed approach as described in text and summarized in Table 4.2. The same contour was used for initialization which is shown overlaid in red on the original image.

and texture components are disabled, essentially resembling the model in [46]. All versions share the same initialization of the active contour model, to highlight the individual contribution of each component.

To demonstrate the effect of each component qualitatively, Figure 4.8 shows the segmentation result of a plant image with the different versions considered here (all of them share the same contour initialization). Evidently the proposed method has the best agreement with the ground truth. Clearly the median and the texture contribute to the accuracy, incrementally but never reach the agreement of the Proposed that includes learning. Proposed-c on the other hand which relies only on color information completely over-segments.

The same quantitative conclusions can be reached also when we compare the accuracy of these modified systems using accuracy metrics standard in image segmentation. As Table 4.3 shows the Proposed model gives a substantial performance improvement across the whole dataset. Compared to the other versions, this one is not affected by the complexity of the background distribution when prior knowledge is included. The difference in performance between Proposed-b and Proposed-c highlights

Table 4.2: Variants of the proposed system introduced to show the contribution of each component.

System	Learning	Features	Median	Comments
Proposed	2-GMM	*a**, *b**, *TFB*	Yes	$\lambda = 0.6$, $\lambda_i^+ = \lambda_i^- = \lambda_i^\oplus = 1$, and $\lambda_i^\ominus = 10^{-2}\ \forall i$
Proposed-a	No	*a**, *b**, *TFB*	Yes	$\lambda = 0$, others same as Proposed
Proposed-b	No	*a**, *b**, *TFB*	No	$\boldsymbol{m}^+ = \boldsymbol{c}^+$, and $\boldsymbol{m}^\oplus = \boldsymbol{c}^\oplus$, others same as Proposed-a
Proposed-c	No	*a**, *b**	No	$\lambda_3^+ = 0$, and $\lambda_3^- = 0$, others same as Proposed-b

Table 4.3: Effects of different components (detailed in Table 4.2) on segmentation accuracy shown as mean (standard deviation).

System	Accuracy (%)			
	Precision	Recall	Jaccard	Dice
Proposed	97.08 (1.83)	95.86 (2.96)	93.17 (3.22)	96.44 (1.76)
Proposed-a	85.10 (9.37)	98.83 (1.19)	84.21 (9.08)	91.17 (5.54)
Proposed-b	83.25 (10.29)	98.99 (1.10)	82.50 (9.98)	90.08 (6.21)
Proposed-c	73.72 (17.82)	99.39 (0.92)	73.21 (17.28)	83.39 (11.92)

the importance of considering the texture component within the level set framework as local region-based information; it can differentiate between foreground and background distributions better. The effect of adding the median descriptor in the level set energy functional for the minimization is also important because it increases the accuracy (e.g., Dice is higher than Proposed-b) but also reduces the measurement deviation across the dataset. This is in agreement with the findings of [2] related to the robustness to the skewness of the distribution which we extended here to vector valued formulation.

Concluding remarks

We propose a novel approach for the segmentation of plants in image based phenotyping experiments. We propose a new vector valued active contour model which incorporates prior knowledge reflecting the likelihood of a pixel to belong to a plant. We build a plant appearance model based

on Gaussian Mixture Models and train this model based on the output of the segmentations. Thus, we use prior instances in an incremental learning fashion to accommodate changes in scenes and complexity of the background. We rely on color and texture features, but we aim to balance complexity of the approach (i.e. storage and processing requirements) with scalability to larger experiments.

We test our proposed system with several top-view images of Arabidopsis (see Chapter 3). For comparison we implement a reference unsupervised segmentation method, commonly employed in image based plant phenotyping, and use another method publicly available [71]. We also compare with several recent innovations in image segmentation [12, 162, 229, 344]. Overall, our proposed approach achieves a Dice score of 96.44%, which is significantly higher than the other two methods. Our experiments also show that the proposed approach can accurately estimate plant growth and is not susceptible to changes in the scene. Furthermore, we show that it is the integration of prior knowledge, texture features, and a new level set formulation that achieves this increase in accuracy.

Most of the currently available solutions for analysis of images for plant phenotyping are tailored to specific acquisition scenarios and as such they cannot be generalized to any laboratory environment. Our proposed solution involves minimal interaction and employs simple yet effective machine learning techniques to learn from the output (and possible feedback of the user). We use an appearance model that can accommodate several plant species since it does not require shape information.

While our approach assumes that little motion occurs between images (i.e. no shuffling) in the future we will consider to incorporate several safety checks to alert the user of possible errors. While here we use a single top-view image it is possible that acquiring images at different depth and focus, and their later fusion to a single fully focused image, could further increase segmentation accuracy.

While some approaches are free and open source (e.g., [71, 124]) the current state of the art in phenotyping analysis is represented by software solutions that accompany commercial (and costly) phenotyping infrastructures. To overcome this limitation and increase adoption, we devise a publicly available cloud implementation built on the PhytoBisque framework of the iPlant Collaborative platform [111]. Our proposed clud-based solution (see Section 4.3) is supplemented by a graphical user interface to facilitate interaction with the user.

4.3 Web application for plant image analysis on the cloud

The increasing availability of cloud computing platforms and applications is radically changing the way people work and live. Also in science, commercial and (in several cases) public cloud services are becoming a standard tool in many fields [131]. Transparent access to cloud resources via web applications has the promise to render scientific computing easy and efficient [305].

An effort towards bringing cloud solutions to plant phenomics is represented by the iPlant Collaborative[1] [111], a project funded by the U.S. National Science Foundation (NSF) whose aim is to create a (hardware and software) cyberinfrastructure to support the computational needs of the research community in plant biology. A software layer is provided by the Bisque (Bio-Image Semantic Query User Environment) platform to offer image storage, sharing, management, and high-throughput analysis capabilities.

Although our proposed distributed sensing and analysis framework (cf. Section 1.2) is agnostic to the actual infrastructure hosting the analysis software (it could be e.g. a powerful workstation or a cluster), we assume that the analysis runs on the cloud. This approach presents several advantages:

- it is easy for the user to familiarize with a cloud-based application;
- the software is always up to date;
- the user is relieved from maintaining locally a computing infrastructure; and
- standardization in equipment and analysis permits consistency in experiments among different labs.

Therefore, we devise a cloud implementation of our plant image analysis system and a web-based graphical user interface (GUI) on the scientific cloud platform offered by the iPlant and Bisque.

[1] `http://www.iplantcollaborative.org`

4.3.1 The swiss army knife for plant phenotyping

We structure our cloud solution as a suite of three web-based applications with complementary functionalities, namely `PhidiasAnnotate`, `PhidiasModel`, and `PhidiasAnalyze`. These so-called 'modules' are implemented based on the application programming interface (API) provided by Bisque. The core functionalities offered by our cloud-based system can be summarized as follows:

(a) manually annotate an image to delineate plant regions, and also visualize and store the so obtained segmentation mask;

(b) find optimized parameters for the image analysis pipeline (cf. Section 4.2), by performing a grid search (on the cloud) on a training dataset of images and corresponding plant annotations;

(c) batch analysis of a dataset of time-lapse images of plants;

(d) visualize and export analysis results (e.g., plant areas, diameters, perimeters), to permit offline processing of the phenotypes.

The images can be uploaded to the system directly by the user, or alternatively our proposed sensing solution can be configured to autonomously transmit images acquired during an experiment to the cloud storage (cf. Appendix A). A rich set of metadata can be attached to the images, to keep track of experimental settings (e.g., genotypes, treatments) and imaging conditions (e.g., camera parameters, acquisition time). Within each module interface is a help panel providing detailed instructions to operate the module and configure parameters. In the following paragraphs we provide an overview of the modules.

`PhidiasAnnotate`. An easy to use and intuitive interface (see Figure 4.9) allows the user to select an input image and perform a manual segmentation of the plant object(s) in the scene. In order to delineate an object in an image, the user clicks on the image to draw polygons, the union of which will represent a binary segmentation. The segmentation mask is stored both in the Bisque format (i.e. an XML-based description of the polylines forming the contour) and also as an indexed binary image file for offline use.

`PhidiasModel`. This module (see Figure 4.10) allows the user to perform a grid search aimed at obtaining a suitable set of parameters to run

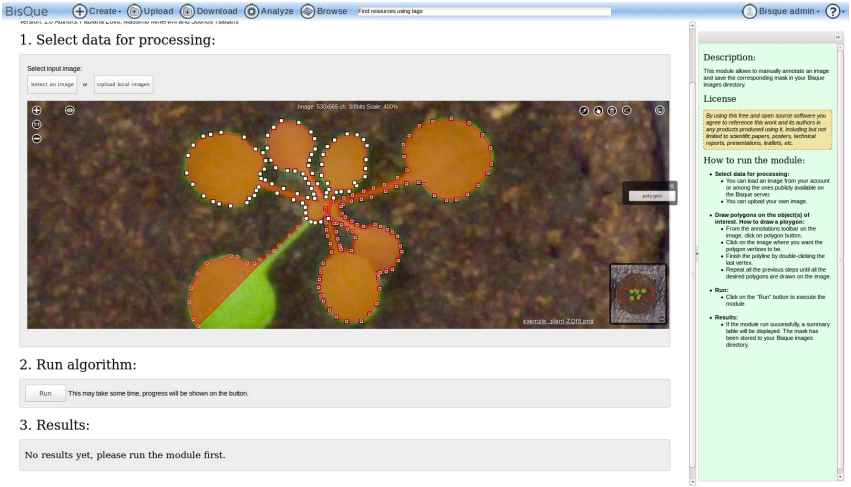


Figure 4.9: Screenshot of the `PhidiasAnnotate` module, showing an example of manual annotation of a plant in an image.

an analysis (using the `PhidiasAnalyze` module). Input to this module is a training set containing original images and the corresponding plant annotations. The ground-truth plant masks can be created with the `PhidiasAnnotate` module or they can be uploaded separately. The user can select which model parameters to optimize and either provide customized search intervals (i.e. upper and lower bounds, and step size) for each parameter or use default values. The module will then perform an exhaustive search, executing the analysis pipeline on the training set for all combinations of parameters. At each iteration, the average Dice Similarity Coefficient (DSC, see Eq. (3.2)) is computed between segmentation masks obtained algorithmically and the ground truth. The set of parameters that minimize the average DSC is eventually selected and saved in the Bisque system, and it can be used subsequently in the `PhidiasAnalyze` module. Notice that the grid search is performed on the cloud, therefore the user can launch the module execution and will be notified upon completion.

`PhidiasAnalyze`. This module allows the user to run the plant image analysis pipeline described in Section 4.2 on a time series of images. After selecting a dataset (uploaded manually by the user, or automatically by

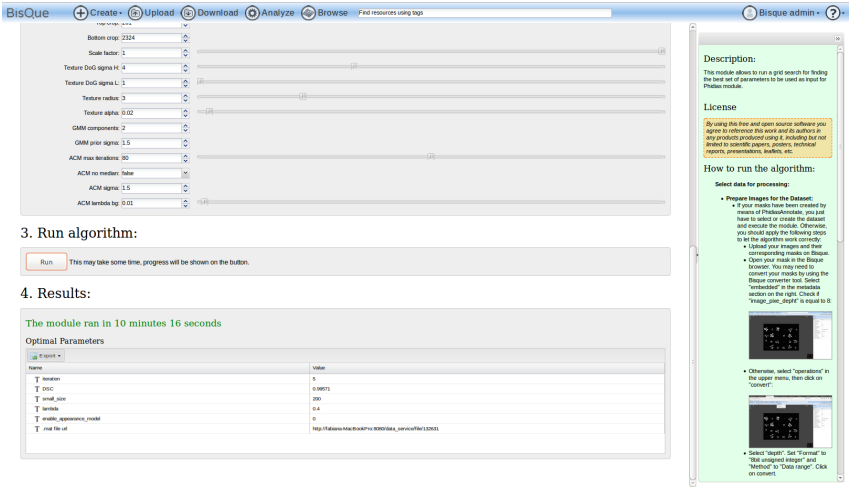


Figure 4.10: Screenshot of the `PhidiasModel` module, showing the graphical controls to configure the grid search and the optimal values reported after running the module.

the sensor) and suitable parameters (e.g., loading a configuration found with the `PhidiasModel` module), the user can launch an execution of the analysis (which will run on the cloud) and will receive notification upon completion. The module interface (see Figure 4.11) will display the results: (i) the image dataset can be navigated with the segmentation masks overlaid for visual inspection; (ii) basic plotting functionalities visualize the evolution of measured traits (e.g., projected leaf area, rosette diameter, compactness, and others) over time for each plant subject in the scene; and (iii) numerical results can be exported in tabular format to CSV files for offline use (e.g., to perform further analyses).

4.4 Phenotypic analysis of Arabidopsis

In the previous sections of this chapter, we tested our plant segmentation approach using Arabidopsis images from the Ara2012 dataset, to demonstrate the ability of the proposed computer vision method to accurately segment plants under a variety of challenging conditions. However, the Ara2012 dataset only includes wild-type subjects of the same geno-

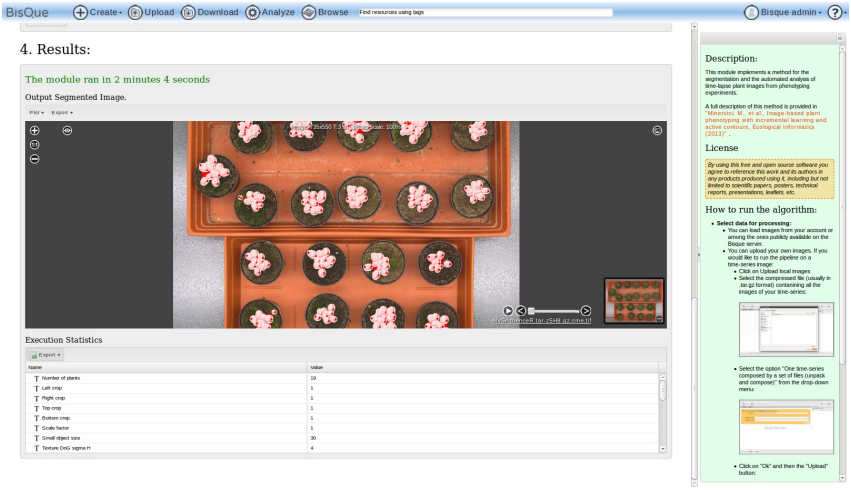


Figure 4.11: Screenshot of the `PhidiasAnalyze` module, showing an image with overlaid segmentation mask found algorithmically and execution statistics.

type and no treatment is performed, therefore it is not appealing from a phenotyping perspective.

On the other hand, the Ara2013 dataset dose include wild-type subjects and also four different lines of mutants, which are grown under controlled conditions (cf. Chapter 3). In this section, we turn our attention to the biological applications of our image analysis system, and offer a first-level phenotypic analysis of Arabidopsis based on visual traits. Furthermore, while previously we estimated growth only via plant area (cf. Figure 4.6), now we broaden our focus and adopt other widely used descriptors.

4.4.1 Visual trait descriptors of rosette plants

A variety of descriptors are used in the literature, to study quantitatively the phenotype of Arabidopsis [86] and, more in general, rosette plants. Assuming a top-view image of a rosette plant and its segmentation mask, we compute the following features.

- *Projected Leaf Area* (PLA) [15, 71, 118, 150, 302, 318, 345]: area of the plant object in a 2D projection (e.g., top view), calculated as the

number of plant pixels in the image and then converted in cm^2.

- *Diameter*: the longest distance between any two points on the boundary of the plant object, expressed in cm. Also known as Feret's diameter or caliper diameter.
- *Perimeter* [71]: number of boundary pixels, expressed in cm.
- *Compactness* (or solidity) [15,71,150]:

$$\text{Compactness} = \frac{A}{A_{\text{CVX}}}, \tag{4.23}$$

  where A denotes plant area, and A_{CVX} is the area of the smallest convex region enclosing the plant object (i.e. its convex hull). Compactness equals 1 for a perfectly solid object and is less than 1 for objects with irregular boundaries or holes.

- *Stockiness* (or form factor) [71,150]:

$$\text{Stockiness} = \frac{4\pi A}{P^2}, \tag{4.24}$$

  where A and P denote, respectively, plant area and perimeter. Stockiness ranges between 0 and 1, where 1 is achieved for a perfectly circular object.

- *Absolute Growth Rate* (AGR) [139] and *Relative Growth Rate* (RGR) [15,99,139,150,318]:

$$\text{AGR} = \frac{A_2 - A_1}{t_2 - t_1}, \tag{4.25}$$

$$\text{RGR} = \frac{\log A_2 - \log A_1}{t_2 - t_1}, \tag{4.26}$$

  where A_1 and A_2 denote plant area measured at two time instants, respectively, t_1 and t_2. AGR ($cm^2\,h^{-1}$) is the simplest index measuring plant growth between consecutive frames, whereas RGR ($\%\,h^{-1}$) is less sensitive to initial differences in size among mutants.

Several other descriptors are used in the literature to capture relevant traits related to shape, growth, and color of a plant, and can be obtained based on an accurate plant delineation from the background, which our system provides. For example, variations in the average color intensity of

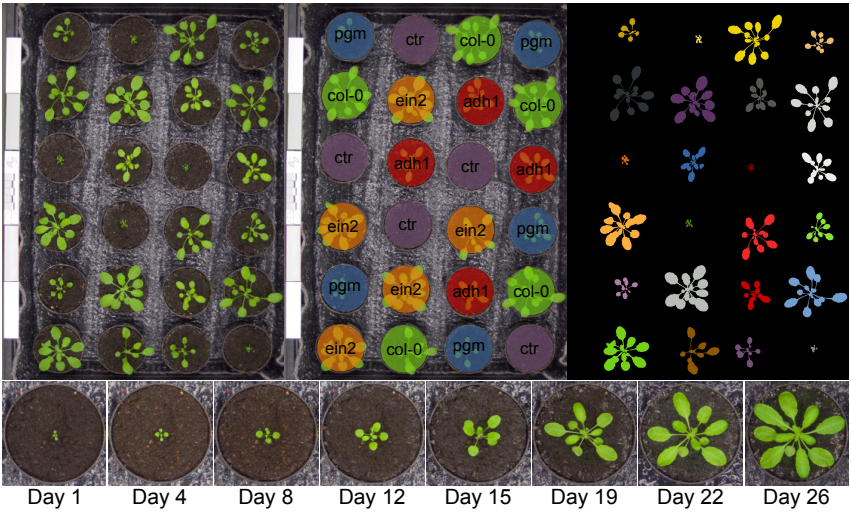


Figure 4.12: (Left) Example image from the Ara2013 dataset, (middle) arrangement of the genotypes in the scene, and (right) delineation of plant objects obtained with the segmentation method proposed in this chapter (Section 4.2). (Bottom) Images of a growing Arabidopsis plant (*adh1* mutant) at different stages.

the plant pixels [71, 302] can be used to quantify drought stress tolerance under a variety of stress conditions [26, 164, 272]. For not co-planar plants (e.g., maize), side-view images are normally preferred and traits such as plant height are measured.

Besides visual descriptors that rely on the plant mask, to obtain an even more accurate growth estimation and a fine-grained categorization of the phenotypes, traits such as the number of leaves [15, 150] or per leaf growth patterns should be sought after. However, leaf counting and segmentation are still open research problems, and some preliminary results are discussed later in Section 4.5.

4.4.2 Results and discussion

Experimental setup

We use 34 images from the Ara2013 (Canon) dataset,[2] showing top-view images of 24 growing *Arabidopsis thaliana* plants, including wild types and four different lines of mutants. Although the plants were imaged every every 20 minutes over a period of 17 days, here we use only two images per day, acquired every 12 hours (respectively, at 8 a.m. and 8 p.m.). Figure 4.12 shows an example image from the dataset, and the arrangement of genotypes in the scene. Further details regarding image acquisition, plant material, and growing conditions are discussed in Chapter 3.

Results

In this section we first present results obtained from the analysis of time-lapse images of Arabidopsis plants, and then we validate the automated approach against one based on manual measurements taken by a human.

Figure 4.13 shows average (and standard deviation as error bars) values for the visual traits described previously in Section 4.4.1. PLA, diameter, and perimeter identify three distinct groups in the genetic material: (i) *ein2* and Col-0 present the biggest size, (ii) *adh1* and *pgm* medium size, and (iii) *ctr* the smallest size. On the other hand, compactness and stockiness do not suggest any grouping. Notice that while Col-0 shares similar size with *ein-2*, the latter presents significantly higher compactness. Higher stockiness of *ctr* with respect to the others may be due to smaller size of the plant object and fixed (per plant) imaging resolution, so that dwarf plants will appear more circular. AGR shows higher growth rates for *ein-2* and Col-0, while compared to the others *ctr* appears almost entirely inhibited in growth. Removing the effect of initial size, the circadian response of the genotypes is observed in the RGR profiles, where *pgm* and *adh1* present the most significant oscillations between rosette area at day and night, respectively.

To demonstrate the accuracy of our image-based phenotyping system with respect to classical approaches based on manual measurement, we focus on rosette diameter. Figure 4.14 shows the validation results of diameter calculated algorithmically against the manual measurements

[2]Using Ara2013 (RPi) data leads to analogous findings and high positive correlation is observed with the measurements obtained from the Ara2013 (Canon) data. Thus, only results for Ara2013 (Canon) are shown here.

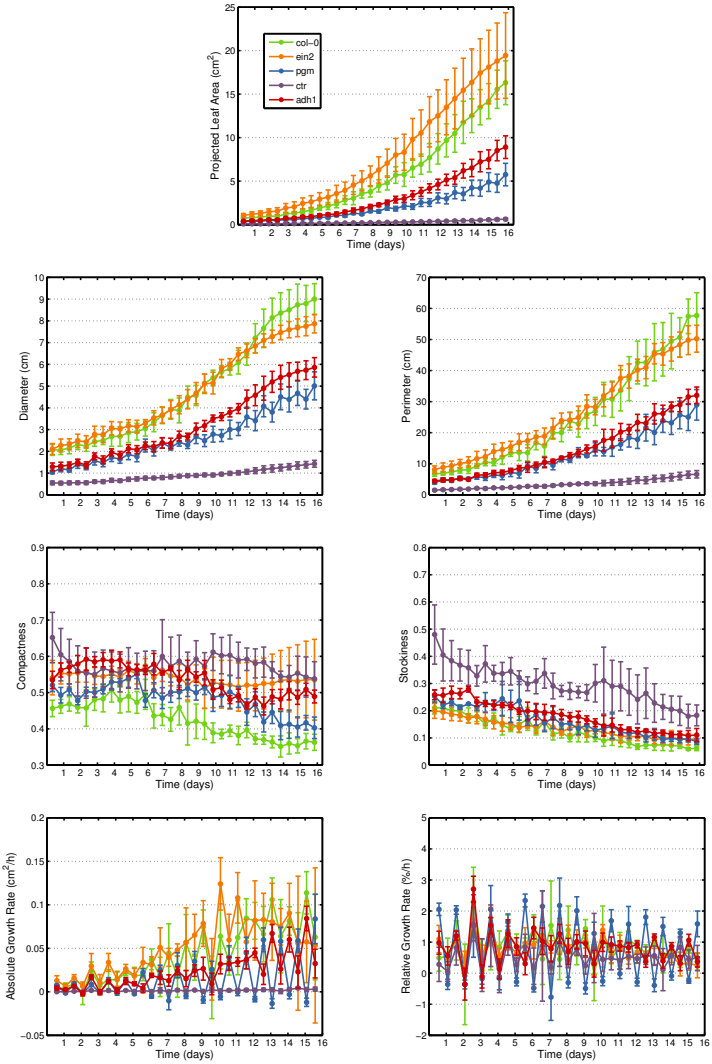


Figure 4.13: Plant trait descriptors measured every 12 hours (8 a.m., 8 p.m.) for 16 days. Plant subjects are grouped by genotype (denoted by color) and intra-group variance is shown for each data point as error bars.

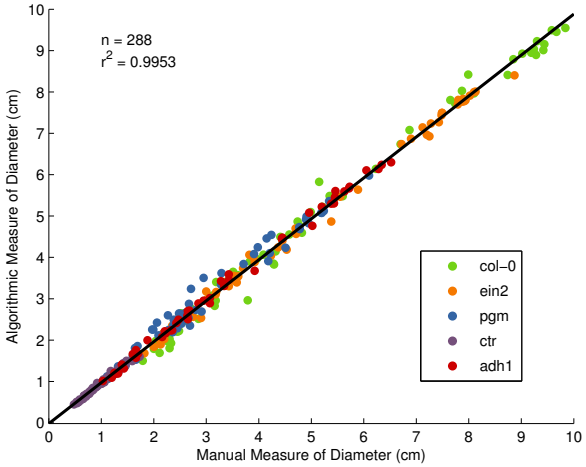


Figure 4.14: Plant diameter measured manually with a caliper plotted against diameter obtained algorithmically from the plant segmentation mask. A regression line is also shown. Colors denote the genotypes.

collected with a caliper at approximately the same time. The variables exhibit high positive correlation ($r^2 = 0.99$), showing agreement between the two approaches. While for most of the genotypes the data points follow closely the regression line, Col-0 presents a few measurements disagreeing by $\sim$0.5 cm and for *pgm* a slight tendency of the algorithmic approach to over-estimate diameter is observed. Notice however that a discrepancy between vision-based and manual measurements can be explained either by lower accuracy of the algorithm in some cases or also by inaccuracy of the human collecting the measurements. When relying on manual approach, to reduce the effect of observer variability, each data point should be obtained by averaging measurements taken repeatedly by multiple users. However, this would increase cost and render the phenotype collection process of even lower throughput. On the other hand, in a vision-based approach based on affordable sensing solutions, measurement redundancy (i.e. different instruments and algorithms) could be adopted at limited increase in cost, and an automated approach can acquire measurements with temporal resolution significantly higher than a human.

Figure 4.15: Examples of leaf segmentation and counting. (a) Original images, (b) ground truth leaf masks, (c) leaf segmentations obtained by [239], and (d) leaf count density maps obtained with the approach in [174].

4.5 Leaf segmentation and counting

While growth estimation based on whole plant area is widely used in the literature, more fine-grained phenotypic analyses (e.g., individual leaf growth, leaf angle, leaf count) rely on counting or segmenting the individual leaves of a plant.

In this section we offer preliminary results of leaf segmentation and counting. We show early findings to illustrate the complexity of the problems at hand, which inspired us to organize specialized computer vision challenges in 2014 and 2015 (see Section 3.5).

Experimental settings. We use a specially formatted part of the image data described in Chapter 3, together with a dataset of high-resolution top-view images of tobacco plants (see [274] for a detailed description). Two Arabidopsis datasets and the tobacco dataset have been released to support the Leaf Segmentation Challenge,[3] of the Computer Vision Problems in Plant Phenotyping (CVPPP) workshop, held in conjunction with the European Conference on Computer Vision (ECCV), in Zürich, Switzerland, in September 2014. Plant images (cf. Figure 4.15a) are considered separately, ignoring temporal correspondence. Images contain

[3] `http://www.plant-phenotyping.org/CVPPP2014`

instances with well separated leaves and simple background, but also more difficult examples with many leaf occlusions, complex leaf shapes, varying backgrounds, or plant objects not well in focus. Individual plant images and ground truth segmentations are used consisting of Ara2012, Ara2013 (Canon), and Tobacco datasets. The datasets are split into training and testing sets for the challenge, and training dataset and accompanying evaluation scripts are available online.[4] Respectively: for training, 128, 31, and 27 images and corresponding annotations have been released; for testing, 33, 9, and 56 images have been released (ground truth was available only to the organizers). Within this challenge, solutions are evaluated on the basis of plant segmentation, individual leaf segmentation, and leaf counting with the criteria outlined in Section 3.4.3.

Leaf segmentation. Table 4.4 summarizes overall performance among participants that completed the challenge and the leading approach of Pape and Klukas [239], which relies on unsupervised clustering and distance maps to segment leaves. Although plant segmentation accuracy (DSC of plant mask) is somewhat acceptable, when considering leaf segmentation (SymmetricBestDice, Eq. (3.5) on page 50) and leaf count criteria poor performance is observed across the algorithms, indicating the challenging nature of the problem. Some of these issues are evident also in the examples of Figure 4.15c: compared to the ground truth, the approach of [239] tends to over-segment (when moss is present, e.g., the top image), separate petioles, and merge leaves. Both images present AbsDiffFGLabels = 2, so count is off but for different reasons.

Leaf counting. We also explore learning directly leaf count density based on leaf center annotations [14, 174]. We use a subset of 84 images from the Ara2012 dataset, with images depicting leaf occlusions, some being out of focus, and some showing moss in the pot. The overall number of leaves per plant varies from 12 to 19. We follow the approach in [174] and the implementation provided by the authors. Based on leaf center annotations (cf. Section 3.4.3), the goal is to learn via appropriate losses, density functions, the integration of which provides object counts (see Figure 4.15d). To this effect, we extract from the green color channel dense SIFT descriptors [189] using the VLFeat package of [312], in 20 of the 84 images, with 7 SIFT bins and fixed orientations. Subsequently, we

[4] http://www.plant-phenotyping.org/CVPPP2014-dataset

Table 4.4: Average performance among all participants compared to the leading approach [239] for the Leaf Segmentation Challenge (LSC) of CVPPP 2014. In parenthesis standard deviations among participants for the first column, and among the three different datasets for the second column.

	Overall	Leader [239]
SymmetricBestDice [%]	48.7 (12.8)	62.6 (19.0)
DSC [%]	83.1 (10.9)	95.3 (10.1)
AbsDiffFGLabels	6.3 (4.36)	2.4 (2.1)
DiffFGLabels	1.3 (4.4)	-1.9 (2.7)

quantize the SIFT space using k-means clustering to create a codebook of size 1500. Using this codebook, we learn a linear transformation of the feature representation on the codebook approximating the density function at each pixel on 64 training images [174].

Testing on 33 images of Ara2012 of the LSC dataset, we obtain an average AbsDiffFGLabels = 2.36(2.9). Among those, in 67% of images we observe either no counting error or the count is off by 1 or 2 leaves at most. In comparison, referring to findings of [239] on the same testing data, counting via segmentation obtains AbsDiffFGLabels = 2.2(1.3). For the example of Figure 4.15, by learning, the count is off only by 1 leaf for both test images. Notice in the density maps in Figure 4.15d that higher values are located approximately at leaf boundaries and several peaks occur at the intersection between a leaf and its stem. Thus, although leaf centers are provided for training, the algorithm eventually relies on other distinctive traits to estimate the presence of leaves: this could be due to the underlying SIFT features and the fact that Arabidopsis leaves appear almost featureless at this resolution.

While these results are preliminary, and could be improved with larger training sizes and appropriate features, they demonstrate the promise of learning-based leaf count estimation. The major limitation of existing object counting approaches is the underlying assumption that the objects to be counted do share similar shape and size (this is true, e.g., for cells, cars, or people counting applications). Future investigations of this problem will be aimed at developing algorithms and features more robust to the rich shape and size variability exhibited by plant leaves.

4.6 Summary

In this chapter we presented an analysis system of time-lapse images arising in a plant phenotyping context. Based on a new active contour formulation combining color intensity, texture, and prior on plant appearance learned incrementally, our approach can accurately delineate plant objects from complex background. Besides, the incremental learning component introduces plasticity in the system, that allows to adapt to changes in the scene.

To favor adoption, we release a cloud-based implementation of the analysis system, accompanied by an easy-to-use graphical user interface. The software solution is offered as a suite of web applications offering complementary functionalities (e.g., annotation, analysis, parameter estimation).

We closed this chapter with preliminary results of leaf segmentation and counting, which represent the most interesting avenue for future developments of our analysis system. The challenging nature of these problems has motivated a Leaf Segmentation Challenge (LSC) and a Leaf Counting Challenge (LCC) organized for the Computer Vision Problems in Plant Phenotyping (CVPPP) workshops:

CVPPP 2014
: 1st LSC, CVPPP 2014[5] held in conjunction with the 13th European Conference on Computer Vision (ECCV), in Zürich, Switzerland;

CVPPP 2015
: 2nd LSC and 1st LCC, CVPPP 2015[6] held in conjunction with the 26th British Machine Vision Conference (BMVC), in Swansea, UK.

The LSC focused on multi-label image segmentation of leaves of rosette plants, whereas the LCC focused on leaf object counting, encouraging approaches not relying on segmentation (e.g., counting via regression). A collation study describing approaches to leaf segmentation is currently under consideration for publication [275].

In the following Part III of the thesis, we will turn our attention to image compression issues arising in a distributed sensing scenario, but highly relevant also to storage, e.g., for archival purposes.

[5] http://www.plant-phenotyping.org/CVPPP2014
[6] http://www.plant-phenotyping.org/CVPPP2015

Part III

Application-aware image compression

5

Effects of image compression in plant phenotyping applications

5.1 Introduction

In our distributed sensing and analysis framework, images acquired by affordable sensors need to be compressed prior to transmission to the cloud where they are analyzed. While previous chapters focus on image acquisition and analysis, now we turn our attention to issues related to image compression occurring at the sensor. Thus, the research question addressed in this chapter is to evaluate how compression affects subsequent analysis. However, image compression is useful not only in our framework, but also e.g. for archival purposes, hence our discussion will be broader.

The adoption of image-based approaches has increased significantly the throughput of phenotyping experiments. However, the design and deployment of such approaches requires a significant multi-disciplinary

This chapter is partly based on:

- M. Minervini, H. Scharr, S. A. Tsaftaris, "The significance of image compression in plant phenotyping applications," *Functional Plant Biology*, 2015, to appear.

effort and know-how in a variety of domains such as automation hardware, image acquisition, software engineering, computer vision and image analysis, and of course plant biology. Clearly, such know-how can be found in few settings and more often than not plant biologists need to rely on (and cannot control for) choices made by other parties involved (e.g., a manufacturer or a contract provider or a collaborator). One crucial case we want to highlight here, is the choice of data compression. This choice does affect the fidelity of the available data, and in many cases its presence in the acquisition procedure is unknown to the end user. An inspection of approximately 60 well cited papers in the recent literature among those present in the plant image analysis software database [188] finds that (i) most authors do not report if imaging data were compressed, and that (ii) few authors did use compression with a lossy image format (e.g., JPEG). Both of these findings are worrisome, because in the former case it could be that it is unknown even to the authors if compression was present and in the latter case it is unknown if compression had an effect. These concerns are also shared by others, stating that care in compression choice must be undertaken [288] and that it should be reported [68].

Clearly, lossy compression (which reduces an image's file size by permanently removing certain information from the original image) must have an effect, but in some scenarios such compression choice is necessary. The constant need to increase experimental scale (e.g., more subjects, higher spatial and temporal resolution, more imaging modalities [77, 97, 104]) produces vast amounts of image data [62, 116, 247]. For example, a single small-scale experiment using the setup described in [75], i.e. 10 plants imaged per hour for 19 days, produces approximately 70 gigabytes (GB) of raw image data (equivalent to 15 DVD discs). Using color images and higher resolution camera sensors (e.g., as in [34, 164]) would increase that figure even more to 250 GB (equivalent to 53 DVD discs) for the same experiment. State-of-the-art image compression standards would compact such data in a way that it would fit in a single DVD disc. While upgrading and ameliorating the e-infrastructure is a key issue [247], it is a slowly changing factor and a costly operation, which requires sensible data management strategies and planning. Furthermore, the importance of easy and rapid access to data has been highlighted for plant phenotyping projects involving institutions and parties geographically distributed [28]. Thus, any savings in the amount of data transferred or archived have significant returns to the end user.

In this chapter, we first introduce necessary concepts and terminology,

relevant error measures allowing to evaluate compression performance in plant phenotyping experiments, as well as image compression standards used here. We then offer three proof of concept experiments to illustrate the effects lossy compression can have: (i) on a simple phenotyping experiment related to measuring growth in a population of 19 *Arabidopsis thaliana* Col-0 individuals, (ii) on estimation of local growth rate of an Arabidopsis root tip from a video, and (iii) on visual perceptibility of fine roots in high-resolution rhizotron images. For systematic evaluation, we proceed by offering a series of experiments that show, how different choices of compression standards and quality settings affect the extraction of phenotypic information from images and image sequences (of roots, shoots, or leaves) obtained by plant phenotyping experiments.[1]

5.2 Materials and methods

We start by introducing fundamental concepts of digital image and video compression [273] in order to establish usual terminology. Then, we briefly review the coding standards adopted in the experiments and case studies that follow. Finally, we define the quality and error measures used to compare compression performance in phenotyping experiments.

5.2.1 Images and image compression

Typically, raw image data is highly redundant, e.g. in a spatially homogeneous region like a uniform background, pixel values do not change when stepping from one pixel to its neighbors. In such cases it is sufficient to store the pixels value once and in addition store for how long this value stays constant, when stepping from pixel to pixel. This is called 'run-length encoding', a base mechanism often used in compression, e.g., in JPEG. Raw image data cannot only be highly redundant in space, but also in time, e.g. when the background remains constant over time; or in color, e.g. when only a fraction of the available color space is used.

Data compression aims to reduce such unwanted redundancy to obtain an as compact digital representation as possible, i.e. a small image file. The smaller the file, the higher is the compression efficiency. It is expressed in

[1] Some of which are carried out and routinely used at the Institute of Bio- and Geosciences: Plant Sciences (IBG-2) of Forschungszentrum Jülich, Germany (`http://www.fz-juelich.de/ibg/ibg-2/EN`).

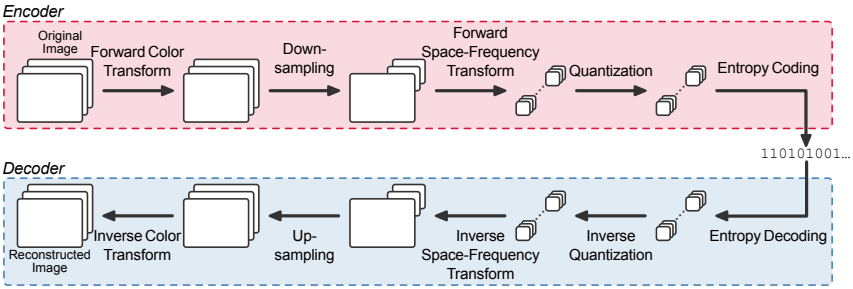


Figure 5.1: Schematic of a typical encoding and decoding process in lossy image compression, such as JPEG [146]. The input image is first converted from the original color space (e.g., RGB) to a representation reducing correlation between color bands (e.g., YC_bC_r [147]). Each color component, possibly after some down-sampling, is split into independent coding units (e.g., blocks of 8×8 pixels). Space-frequency transformation permits not only spatial decorrelation but identifies information to be selectively discarded through quantization. The transformed and quantized coefficients are further compressed by an entropy coding stage using lossless approaches. This results in a bit-stream arranged according to a predetermined syntax, that can be decoded performing all previous operations in reverse order, to obtain an approximation of the original image.

terms of bit rate (BR), measured in bits per pixel (bpp):

$$\mathrm{BR} = \frac{\text{image file size}}{\text{width} \times \text{height}}, \tag{5.1}$$

BR denotes the average number of bits required to represent a single image pixel in an image with given width and height.

Current compression standards use a variety of sophisticated techniques to achieve lower and lower bit rates. Such schemes consist of two parts: an encoder converting the original image into a compressed file and a decoder reversing this process, i.e. converting a compressed file into an image. A software or hardware implementation of a compression standard is thus termed 'codec' (coder/decoder). A typical workflow for encoding and decoding is depicted in Figure 5.1 for background information. There are two general categories of compression:

Lossless compression: Here no loss of information occurs and the decompressed image is a prefect copy of the original, as for example in the

general-purpose ZIP file compression. Thus, lossless compression does not compromise image quality or results of phenotyping experiments. Their relevance in practice lies in the achieved compression efficiency and the computational effort needed for coding and decoding. We investigate this in Section 5.4.1.

Lossy compression: Here some information is lost due to compression and the decompressed image is only an approximation of the original. Typically, lossy compression achieves much higher compression efficiency, i.e. smaller file sizes. It has become ubiquitous with the JPEG standard [146]. Lossy compression standards are designed to achieve the least mathematical or perceptible [172] difference between the original and reconstructed images, with the smallest possible compressed file size. Therefore, applying lossy compression always entails a trade off between smaller file size and better image quality.

5.2.2 Metrics for image quality evaluation

The core of this article is to investigate what 'better image quality' really means in a plant phenotyping context. As higher bit rate (BR, Eq. (5.1)) typically corresponds to less information loss, we evaluate different codecs at various bit rates. We encode the original image I at a given bit rate, reconstruct the image $\widetilde{\mathrm{I}}$ by decompression, and compare it to the unprocessed original I. For this comparison we use several information theoretic or plant science specific metrics, in order to investigate which codec is the best for a given plant phenotyping application.

Most codecs are developed without a specific application in mind. They are therefore usually evaluated against information theoretic measures like execution efficiency, image fidelity, or color distortion. We use these measures for reference and introduce them below, however these measures are not specific for plant experiments. For plant experiments the ultimate information of interest is the actual measure describing a plant trait. Therefore, compression performance in a phenotyping context should be evaluated against how accurately the trait of interest can be measured [216]. As quantitative traits we exemplarily investigate image segmentation-based traits and traits based on image sequence analysis. For segmentation-based analyses (i.e. involving the automatic delineation of plant objects in an image) we select two measures, namely Projected Leaf Area (PLA) and a more general segmentation accuracy measure (DSC). For image sequence analysis we use Relative Elemental Growth

Rate (REGR) as plant-related measure and the Average End-point Error (AEE) as a well-established measure for optical flow accuracy. Finally, we also look at visual fidelity when a human expert evaluates an image. For visual fidelity we show and discuss example images, other measures are defined in the following paragraphs.

Execution efficiency is evaluated in terms of runtime, i.e. the time to encode and decode image data. It is measured in seconds (see Section 5.4.4 for further details on execution times).

Image fidelity is expressed in terms of Peak Signal-to-Noise Ratio (PSNR), measured in decibel (dB):

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{255^2}{\mathrm{MSE}(\mathrm{I}, \widetilde{\mathrm{I}})}\,, \tag{5.2}$$

where $\mathrm{MSE}(\mathrm{I}, \widetilde{\mathrm{I}}) = \frac{1}{N} \sum_{i=1}^{N} (\mathrm{I}_i - \widetilde{\mathrm{I}}_i)^2$ is the mean squared error, and N is the number of image pixels. A higher PSNR indicates better image fidelity. For videos, we average PSNR values and also bit rate (BR, Eq. (5.1)) across all frames of a sequence.

Color distortion can be quantified using the information theoretic Kullback-Leibler (KL) divergence [169]:

$$\mathrm{KL} = \sum_{i=1}^{B} \mathcal{H}_i \log_2 \frac{\mathcal{H}_i}{\widetilde{\mathcal{H}}_i}\,, \tag{5.3}$$

where $\mathcal{H}$ and $\widetilde{\mathcal{H}}$ denote normalized histograms of intensity values of a single color channel, computed on the original and reconstructed images, respectively. B is the number of histogram bins. For RGB images, we estimate overall color distortion, $\mathrm{KL}^{\mathrm{RGB}} = (\mathrm{KL}^{\mathrm{R}} + \mathrm{KL}^{\mathrm{G}} + \mathrm{KL}^{\mathrm{B}})/3$, as the average between the KL divergence values obtained on the marginal histograms of the RGB color components. KL divergence is a unitless quantity, that should be as close to zero as possible for higher color fidelity.

Projected Leaf Area (PLA) is proportional to the number of plant pixels observed in an image, e.g., a top view of a rosette plant. PLA is frequently used to evaluate shoot development as it correlates well with plant biomass [115,318]. Plant pixels are found via automated segmentation. We quantify the amount of error in plant area estimation when compression is used as the relative change:

$$\text{PLA Error} = \frac{\widetilde{A} - A}{A}\,, \tag{5.4}$$

of area $\widetilde{A}$ found automatically on the reconstructed image (i.e. the image after the compressed image is decompressed) with respect to the area A found based on the original uncompressed image. We express PLA Error as a percentage, where best possible value is 0%, while positive or negative values indicate an over- or under-estimation of the plant area, respectively.

Segmentation accuracy is more sensitive to segmentation errors than PLA. Suppose a found segment has the correct size, but is distorted or shifted with respect to the ground truth segment, then PLA Error (Eq. (5.4)) would be 0 despite the segmentation error. A measure capturing such errors is the Dice Similarity Coefficient (DSC) [78]:

$$\mathrm{DSC} = \frac{2 \cdot |\mathrm{M} \cap \widetilde{\mathrm{M}}|}{|\mathrm{M}| + |\widetilde{\mathrm{M}}|}\,, \tag{5.5}$$

between the binary segmentation masks M and $\widetilde{\mathrm{M}}$, obtained by segmenting original and reconstructed images, respectively. We express DSC as a percentage, where a DSC value of 100% denotes perfectly matching segmentation masks.

Relative Elemental Growth Rate (REGR): Accurate estimation of local growth rates can be obtained using motion estimation techniques based on optical flow analysis [277,319]. We estimate the optical flow from an image sequence using the combined local-global approach in [36]. This allows to track points through the image sequence. REGR is quantified as the spatial 1D elongation rate between two points on e.g. a root [52,245]:

$$\mathrm{REGR} = \frac{1}{T} \ln \frac{l_j(T)}{l_j(0)}\,, \tag{5.6}$$

where T is the time duration over which growth is estimated, and $l_j(\cdot)$ is the distance between the points of interest at a given time. REGR is measured in $\%\,\mathrm{h}^{-1}$, and its calculation relies only on initial (at time $t = 0$) and final (at $t = T$) segment lengths.

Average End-point Error [234] is a more general performance measure for optical flow, also applicable e.g. in tracking scenarios. Here, optical flow $\widetilde{\boldsymbol{u}}$ calculated on the compressed sequence is compared to the ground-truth flow $\boldsymbol{u}$ calculated on the original sequence using the normalized Average End-point Error (AEE) :

$$\mathrm{AEE} = \frac{\sum_{i=1}^{N} \|\widetilde{\boldsymbol{u}}_i - \boldsymbol{u}_i\|_2}{\sum_{i=1}^{N} \|\boldsymbol{u}_i\|_2}\,, \tag{5.7}$$

where $\boldsymbol{u}_i$ and $\widetilde{\boldsymbol{u}}_i$ denote the displacement estimated on original and reconstructed sequence at the i^{th} pixel, and $\|\cdot\|_2$ denotes the L_2 norm yielding the length of the vector. AEE is expressed as a percentage, with 0% denoting perfectly matching flow fields. We normalize with respect to the ground truth motion vector length, in order to accommodate slow-moving test sequences.

5.2.3 Employed image and video codecs

We employ a variety of state-of-the-art lossless and lossy image and video coding standards. While these have been developed for multimedia and entertainment applications they are widely used in several other domains.

For lossless image compression we consider: *Portable Network Graphics* (PNG) [315], *JPEG-LS* [330], the lossless option of *JPEG 2000* [287], and *WebP*. For lossy image compression we consider: *JPEG* [146], *JPEG 2000* [287], and *WebP*. We also consider a variant of JPEG 2000, permitting native region-of-interest (ROI) coding [58], a feature allowing to encode foreground image regions at a higher quality than background regions. For video we consider only lossy standards, namely: the royalty free *VP9* [219], and the recent *High Efficiency Video Coding* (HEVC) [295]. In the following paragraphs we expand on the standards and outline parameters used and implementation details. For each standard we report codec software implementations adopted in the experiments and command line options used to execute the encoders.

PNG

Portable Network Graphics (PNG) [315] is a lossless compression standard, which uses a filtering function to enable spatial decorrelation and a compression algorithm (Deflate) similar to that of the ZIP file format. PNG employs a 2-stage scheme: first the image is pre-compressed in a differential coding fashion, using a filter that predicts the value of a pixel from its neighbors; then, the Deflate algorithm is used for entropy coding. PNG is fast in decoding speed and can handle both gray scale and color images (RGB). We adopt the libpng v1.6.12 implementation:[2]

- Lossless: `pnmtopng -compression=9 -comp_mem_level=9 -paeth -comp_window_bits=8 -comp_strategy=filtered`

[2] `http://www.libpng.org`

JPEG-LS

JPEG-LS [330] is a lossless and near-lossless compression standard and features low computational and memory requirements. It is based on the LOCO-I algorithm (LOw COmplexity LOssless COmpression for Images), and employs prediction, context modeling, Golomb-Rice codes to encode prediction residuals, and run length coding of smooth regions. Both gray scale and color images are natively supported by the standard. We adopt the Hewlett-Packard reference encoder v1.0:[3]

- Lossless: `locoe`

JPEG

JPEG (Joint Photographic Experts Group) is a lossy image compression algorithm, based on the discrete cosine transform [146]. JPEG is the most widely adopted compression standard for digital photography (it is default format on most commercial-grade cameras) and for image coding on the Web. The JPEG compression algorithm is based on RGB to YC_bC_r color space conversion, chroma sub-sampling, 8×8 discrete cosine transform, and Huffman (or, optionally, arithmetic) entropy coding. We adopt the libjpeg v9a implementation:[4]

- Lossy: `cjpeg -dct float -progressive -arithmetic -quality` q

where the quality factor q is an integer in the range from 0 (lowest quality, small file) to 100 (best quality, big file).

JPEG 2000

JPEG 2000 [287] is a lossless and lossy image coding standard based on the discrete wavelet transform. Notably, JPEG 2000 is capable of native region-of-interest (ROI) coding [58] using the Maxshift algorithm [16], a feature allowing to encode foreground image regions at a higher quality than background regions. Sophisticated rate control and bit stream organization techniques allow progressive decoding and quality scalability, while a flexible file format allows error resilience and fast random access to portions of the image. Certain operations, such as rotation, are also supported in the compressed domain for greater efficiency. We adopt the Kakadu v7.4 implementation:[5]

[3] `http://www.hpl.hp.com/research/info_theory/loco/`
[4] `http://www.ijg.org`
[5] `http://www.kakadusoftware.com`

- Lossless: `kdu_compress Creversible=yes`
- Lossy: `kdu_compress -no_weights -rate`r
- Lossy (ROI): `kdu_compress -no_weights Rshift=16 Rlevels=5 -roi`*roifile*`,0.5 -rate`r

where r is a float denoting the desired bit rate (bpp), and *roifile* is a PGM file containing the ROI mask.

WebP

WebP is based on the methodology adopted to compress keyframes in the VP8 video coding standard [20] for the purpose of royalty-free lossless and lossy image compression. WebP is mainly targeted to color images, which are converted to the YC_bC_r color space and sub-sampled in the chroma components. Notably, WebP employs variable block sizes, block prediction, 4×4 discrete cosine transform, and Huffman entropy coding. We adopt the libwebp v0.4.1 implementation:[6]

- Lossless: `cwebp -lossless -m 6 -q 20`
- Lossy: `cwebp -q`q

where q is a quality factor in the range from 0 (lowest quality, small file) to 100 (best quality, big file).

VP9

VP9 [219] is a new open and royalty-free library for lossless and lossy video coding. VP9 employs several modern coding tools (e.g., sub-pixel interpolation for motion compensation, combinations of discrete cosine transform and asymmetric discrete sine transform, binary arithmetic entropy coding, and loop filter to reduce blocking artifacts) and is mainly intended for high definition video and targets low decoding complexity. We adopt the libvpx v1.3 implementation:[7]

- Lossy: `vpxenc --codec=vp9 --passes=1 --tune=psnr --end-usage=cbr --target-bitrate=`r

where r is a float denoting target bitrate (kbps).

HEVC

High Efficiency Video Coding (HEVC) [295] is the latest generation video coding standard [148]. Similar to its predecessors (e.g., H.264 and

[6] `https://developers.google.com/speed/webp/`
[7] `http://www.webmproject.org/vp9/`

MPEG-4), HEVC employs sophisticated techniques for intra prediction and motion compensation, in order to address, respectively, spatial and temporal correlation in high definition video signals. The residuals are further processed with integer transforms derived from sine and cosine discrete transforms, and finally entropy coded with context-adaptive binary arithmetic coding. We adopt the HM v16.0 reference implementation,[8] and also the x265 v1.3 implementation:[9]

- Lossless (gray scale): `TAppEncoderStatic --Profile=main-RExt --InputChromaFormat=400 --TransquantBypassEnableFlag=1 --CUTransquantBypassFlagValue=1`
- Lossy: `x265 --qp`q

where q is an integer in the range from 0 (best quality, big file) to 51 (lowest quality, small file).

For JPEG-LS and JPEG 2000 we adopt the pre-compiled software libraries provided by the authors, whereas for the others we compile the libraries from source code.

Note that, although HM is the reference implementation of HEVC, x265 achieves superior time performance and is used here for lossy compression. On the other hand, HM is used for lossless compression, since to this day this feature is not supported by x265. To date, lossless compression of color images is not possible with HEVC, due to the chroma sub-sampling strategy mandated by current implementations.

Video codec implementations used in this study accept input data only in the YUV 4:2:0 format (i.e. one luminance component, Y, followed by two chrominance components, U and V, down-sampled by a factor of two both horizontally and vertically). Hence, RGB color images are converted to the YC_bC_r color space [147] and chroma sub-sampled prior to encoding with VP9 and HEVC (observe that JPEG and WebP perform analogous operations internally, as part of their coding strategy, whereas JPEG 2000 does not recommend chroma sub-sampling). Gray scale images are embedded into a YUV formatted byte stream, by augmenting the luminance component with uniform zero-valued chroma components (note that this operation does not affect coding efficiency).

[8] `http://hevc.hhi.fraunhofer.de`
[9] `http://x265.org`

5.3 Case studies: Compression affects phenotypic analysis

In the following, we investigate how lossy image and video compression techniques influence results in three typical plant phenotyping experiments. In Section 5.3.1, we consider size measurement of *Arabidopsis thaliana* using projected leaf area (PLA), i.e. a trait based on segmentation of single images. In Section 5.3.2, we select local growth rate of a root tip, i.e. a trait derived from an image sequence. In Section 5.3.3, we investigate rhizotron images showing complete root systems in soil, i.e. an example where the reference state-of-the-art evaluation still is the human eye.

5.3.1 Example 1: Size of a rosette plant evaluated by PLA

The purpose of this proof of concept experiment is to demonstrate the type of errors introduced by lossy compression in a typical phenotyping experiment measuring rosette growth over a period of time. We use imaging data of a population of 19 wild-type (Col-0) *Arabidopsis thaliana* plant subjects acquired using off-the-shelf commercial cameras in a controlled environment as described in Chapter 3. Twenty (20) observations within an imaging period of 7 consecutive days, 12 days after germination, are obtained for each replicate. The images are in color and are recorded in the raw, uncompressed camera format.

Two versions of the dataset are considered and are processed individually. One, *uncompressed*, containing the original images, and one *compressed* with JPEG at quality factor $q = 27$ (cf. Figure 5.2a-b). To highlight the subtle compression artifacts, Figure 5.2c-d shows a zoomed detail of one of the plants (in the blue bounding box in Figure 5.2a-b). Compression introduces slight discontinuities due to the so-called blocking artifacts but this does not lead to obvious loss in perceived image quality.

Images are analyzed independently to obtain rosette segmentations as described in [209]. Even these slight compression artifacts do affect analysis algorithms: as shown in Figure 5.2e compression causes changes in the segmentation. We observe leakage (indicated with red pixels) of the plant boundary to non-relevant plant material (in this case moss). Compression also tends to slightly affect the delineation of the plant (indicated with blue pixels) almost in its whole periphery and causes also the loss of some of the leaf stems. These segmentation differences directly affect

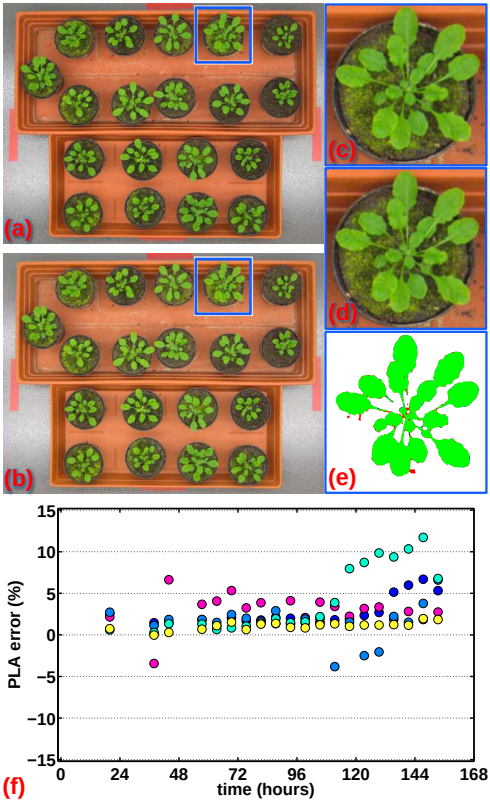


Figure 5.2: Compression affects growth observations in Arabidopsis. (a) An uncompressed image of 19 individuals of *A. thaliana* ecotype Col-0 on day 17 after germination; (b) the same image compressed with JPEG; (c)-(d) zoom in detail of the plant in the blue bounding box uncompressed and compressed, respectively; (e) color coded segmentation outcome of automatic analysis of this plant using uncompressed or compressed data: green pixels are identified as plant on both images, while red (false positives) and blue (false negatives) are identified only in the compressed or uncompressed image respectively; (f) PLA error (%) of the top 5 plants over 6 days covering the days 12-17 after germination (5 plant measurements are shown only for presentation clarity, and similar trends are observed for all 19 plants). The same colored dot is used for the same plant.

Table 5.1: One way repeated measurement ANOVA of PLA error measured on 19 Arabidopsis (Col-0) replicate plants over a period of 7 days with 20 repeated measurements of time. 'Time' is the within subjects factor and 'replicate' is the between.

Factor	Sum of Squares	Degrees of Freedom	Mean Square	F	Prob>F[C]
Time	0.022(0.109)	4.055(72.991)[A]	0.005(0.001)[B]	3.654	**0.00879**
Replicate	0.178(0.039)	1(18)	0.178(0.002)	81.272	$<$ **0.00001**

[A] Error term values in parenthesis.
[B] Greenhouse-Geisser corrections are reported to account for deviations from sphericity (Greenhouse-Geisser $\epsilon = 0.213$).
[C] Bold font indicates significance at the $p < 0.05$ level.

Projected Leaf Area (PLA). PLA Error (Eq. (5.4)) is 3% in the shown case. We estimate PLA Error for all plants and all time points (cf. Figure 5.2f). We observe that errors are diverse. PLA is mostly overestimated, up to 12%, a trend increasing with time, but sometimes also underestimated by up to -4%. Notice that the ordering of dots changes as a function of time and how as time advances larger errors are evident.

These empirical observations are statistically confirmed by an ANOVA. We conduct a one-way repeated measures ANOVA (using Stata version 11, StataCorp LP, College Station, Texas, USA) to investigate significance of effects of time (i.e. within subjects) and of replicate (i.e. between subjects) on PLA error. This indicates if compression affects replicates differently and if the error changes as plants grow. Since PLA error has been normalized by the plant's area before compression, individual growth effects of plants should be minimal, and ANOVA will be testing a linear effect of the two independent variables (IVs) time and replicate on the error introduced by compression on PLA measurements, via the PLA error dependent variable.

ANOVA results are shown in Table 5.1. Considering a significance level of < 0.05, time is a relevant factor ($F[4.05, 72.99] = 3.65$, p-value $= 0.008$) with a positive slope and between subject effects are present among the replicates ($F[1, 18] = 81.27$, p-value < 0.00001).

We conclude that in this example *visually nearly unnoticeable compression distortion affects rosette growth estimates*.

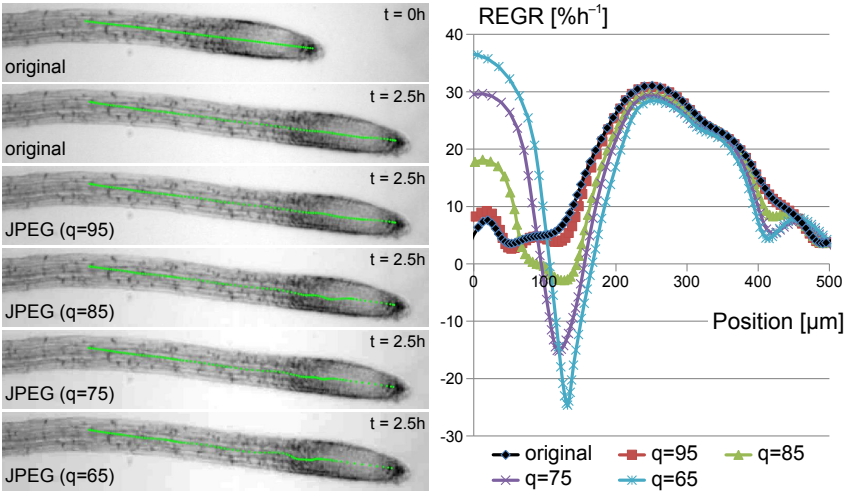


Figure 5.3: (a) Example images used for root growth analysis. Root tracking results obtained with the method described in [52] are shown as green lines, where distance between points denotes estimated local growth intensity. (b) Spatial growth (REGR) profiles obtained.

5.3.2 Example 2: Quantifying local root growth by REGR

Several image-based plant measurements rely on accurate correspondence analysis, e.g., image-based 3D reconstruction, motion and local growth rate analysis. As an example of this analysis class, we investigate local motion analysis, e.g. used to study growth-related phenotypes such as gravitropic response [52].

This experiment investigates how distortion in an optical flow field due to JPEG compression affects local growth estimation. To this end we adopt the method for REGR estimation in root tips as described in [52]. We apply it to an image sequence showing a growing Arabidopsis root tip (cf. Figure 5.3a, original) over 2.5 hours. The sequence consists of 300 images and the processed region of interest is 422×77 pixels.

The method works as follows: the mid-line C of the root (cf. the green line in Figure 5.3a, at $t = 0$ h) is given for the first image. C is represented by equidistantly spaced points C_j, one per pixel length. The positions of these points are individually tracked in time by optical flow calculated

from the image sequence (using the algorithm given in [36]). The outcome of this tracking is shown as green points in Figure 5.3a, original at t = 2.5 h. From the distance l_j between a point C_j and its neighbor C_{j+1} at t = 0 h and at t = 2.5 h Relative Elemental Growth Rate (REGR) is then calculated via Eq. (5.6).

Figure 5.3b, black line, shows the so derived growth rate mapped to the mid-line at t = 0. In agreement with [52] we observe that the growth rate between quiescence center and growth zone (position x between 0 and approx. 130 μm) is around 5 % h^{-1}. The growth zone starts around position x = 130 μm and ends at approx. x = 450 μm.

To investigate the effects of compression, we store the sequence in 4 different JPEG qualities $q \in \{95, 85, 75, 65\}$ and apply the method as before. Figure 5.3a shows the tracking results for these JPEG qualities and Figure 5.3b the corresponding REGR curves. Comparing the tracking results for the different qualities we observe, that compression affects tracking of different root zones inconsistently, mainly depending on local image contrast. For example, while the locations of growth maximum and root tip appear stable with respect to compression, width of growth zone decreases up to 20% for higher compression. However, major effects of compression occur in the zone behind the tip, whose maximum growth rate should be constant at approximately 5 % h^{-1} [51,52], while already for very high quality JPEG compression (q = 95), the observed error in REGR (Eq. (5.6)) is 21%. For higher compression ratios (i.e. lower JPEG quality factor q) the error in REGR increases dramatically up to approximately 380% (q = 65).

We conclude that in this example *compression affects the estimation of the spatio-temporal pattern of root tip growth, especially in regions with low image contrast*.

5.3.3 Example 3: Manual delineation of root images

Below-ground plant organs can be studied noninvasively using the rhizotron [223]. Figure 5.4a shows an example gray scale image (width $\times$ height: 4872$\times$3248 pixels) including the root systems of three rapeseed subjects, obtained from root phenotyping experiments at the GROWSCREEN-Rhizo [223]. The gold standard for evaluation of such images is still manual delineation of the roots by a human expert.

Lossy compression can alter the appearance of the images, introducing visual distortions or loss of details that may influence the user's capability

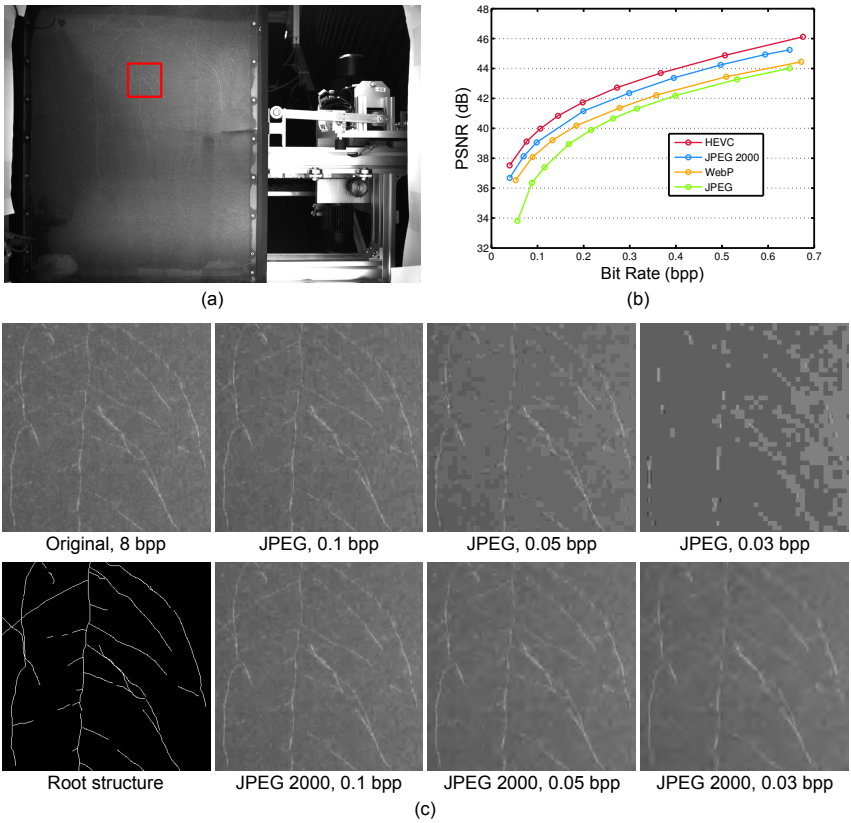


Figure 5.4: (a) Root systems of three rapeseed subjects imaged at the GROWSCREEN-Rhizo [223]. (b) Image fidelity obtained for the image in (a), after compression with different standards. (c) Detail from (a) (red box), compressed at various bit rates using the JPEG and JPEG 2000 standards.

of accurately delineating the roots. In Figure 5.4c we therefore provide a visual comparison of root images at different levels of compression with JPEG and the more advanced JPEG 2000.

At bit rate BR = 0.1 bpp (cf. Eq. (5.1)), being equivalent to space savings of 98.7%, the root structure is still clearly evident and rich in details. This can be obtained e.g. in JPEG by setting the quality factor at $q = 20$. However, when encoding at even lower bit rates, compression distortion (e.g.,

blocking artifacts in JPEG or blurring in JPEG 2000) increases substantially, rendering the thin roots increasingly difficult to recognize (even for a trained human observer). At 0.03 bpp (i.e. 99.6% space savings), the JPEG image has lost most information and is practically unusable, while in the JPEG 2000 image only the thicker roots are still distinguishable. Compression performance of different coding standards is assessed quantitatively in Figure 5.4b using PSNR (Eq. (5.2)): JPEG 2000 obtains image fidelity superior to JPEG at any bit rate, while best image quality is achieved by HEVC.

We conclude that *lossy compression is admissible even when subtle structures need to be quantified by a human expert. However, care needs to be taken that compression artifacts remain close to unnoticeable*.

5.4 Performance of compression in plant applications

The case studies above showed that lossy compression can affect results of quantitative evaluation methods. In this section we offer a richer evaluation, including more data sets, metrics, and codecs tested systematically at different compression rates. We focus on segmentation-based methods for images in Section 5.4.2 and on growth estimation from videos in Section 5.4.3. Additionally, we investigate the effectiveness of lossless codecs, which do not compromise image quality. We start with lossless in Section 5.4.1.

5.4.1 Lossless coding

Compression performance obtained using lossless compression approaches on a 16 megapixel gray scale image of rapeseed roots (Figure 5.4a) is shown in Figure 5.5. A good balance of bit rate reduction and codec efficiency (cf. Table 5.2) is achieved by JPEG-LS and JPEG 2000, while PNG obtains slightly worse results. Overall, with lossless compression it is possible to reduce file size considerably (to approximately 35% of uncompressed size) with exact reconstruction of the original image and limited computational overhead (in most cases less than one second for decoding, cf. Section 5.4.4). Analogous compression ratios are typically obtained when lossless compression is applied on images composed by more than one component (e.g., 3 for RGB color images, or in general M for hyperspectral

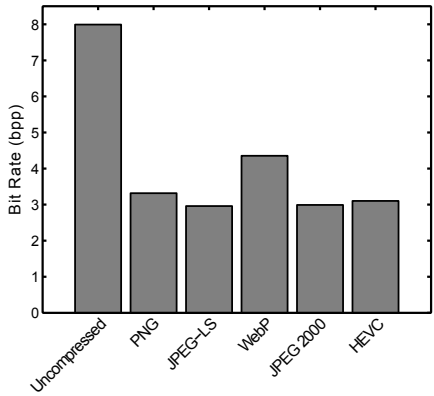


Figure 5.5: Compression performance obtained by lossless coding standards on the gray scale root image of Figure 5.4a. Baseline for the comparison is size of uncompressed image (the leftmost bar at 8 bpp).

data cubes), and scaling appropriately the results in Figure 5.5 (e.g., $\times 3$ for color, or $\times M$ for hyperspectral) would provide an estimate of expected compression performance.

To elucidate how data size may reflect to transmission times over a network, we consider an example scenario in which an 18 megapixel color image of a rapeseed shoot (Figure 5.6b) is acquired in a greenhouse and transmitted to a central processing unit at a different physical location (e.g., for processing or storage). Subsequently, the same image is downloaded from the central repository where it is stored to a user's workstation. We perform this test during working day, to ensure average network traffic conditions, using a workstation and 100 Mbit/s wired network connection. Uploading our test image in uncompressed format (53.7 MB) requires 6 seconds, and downloading the uncompressed image takes 4.9 seconds. On the other hand, encoding the image with JPEG 2000 in lossless mode and transmitting the compressed file (11.3 MB) requires overall only 2 seconds, while downloading and decoding the compressed image locally requires only 1.6 seconds. Image compression leads in this case to 79% space savings and 67% transmission time reduction.

We conclude that *lossless compression does offer significant space savings but for even more savings lossy compression is necessary*.

5.4.2 Segmentation-based shoot image analysis

Image-based investigations of above-ground plant organs often rely on color images acquired from top or side views. Plant segmentation (i.e. the delineation of the image regions containing a plant object) represents a fundamental step in most image processing pipelines for phenotyping applications , and permits to calculate a variety of morphological and color features (cf. Chapter 4).

The accuracy of plant segmentation affects all subsequent analyses, therefore we investigate compression performance with respect to automated plant segmentation from background. We adopt three color images of plant shoots (including Arabidopsis, rapeseed, and maize) acquired from different angles (top or side view), with resolutions ranging between 5 and 18 megapixels (cf. Figure 5.6a-c).

Here, plant segmentation is performed by a pixel-level classifier, which decides if a pixel is foreground (plant) or background. We use a support vector machine (SVM) operating on color values [34], trained on labeled image data, where the plant has been delineated manually. The resulting foreground/background classification is cleaned up using morphological operations to fill holes and remove small objects. Figure 5.6a-c shows example segmentation masks obtained with this method.

We quantify changes in plant segmentation due to compression using three different metrics: (i) the Dice similarity coefficient (DSC, Eq. (5.5)) as a well established segmentation measure, (ii) projected leaf area error (PLA Error, Eq. (5.4)) as a plant related segmentation measure, and (iii) Kullback-Leibler (KL) divergence (Eq. (5.3)) on the foreground to see how color information is affected (e.g., used to quantify drought stress tolerance under varying irrigation [164] or stress [26, 272] conditions).

As shown in Figure 5.6d-f, using JPEG 2000 and HEVC standards, it is possible to obtain PLA measurements very close to those obtained on the original image, even at low bit rates. The oscillating behavior observed for some codecs at very low bit rates is due to portions of the background that, due to compression artifacts, occasionally appear to the plant segmentation method as belonging to a plant object. Depending on image characteristics and segmentation method, approximation errors due to compression may lead to an over-estimation (e.g., Arabidopsis image, cf. Figure 5.6d), or an under-estimation (e.g., maize image, cf. Figure 5.6f) of the plant area.

The accuracy of the segmentation mask (based on which PLA and also several features related to plant morphology can be calculated) is

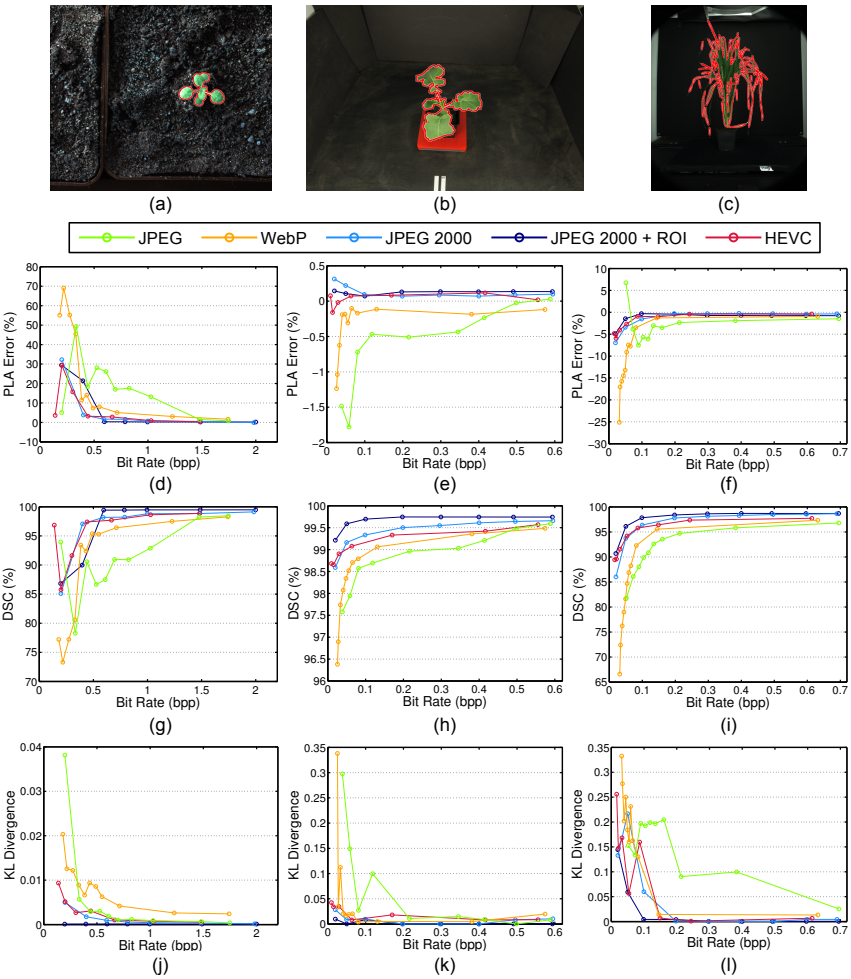


Figure 5.6: Compression performance obtained with color images of plant shoots: (a) Arabidopsis (width × height: 2448×2048 pixels), (b) rapeseed (width × height: 5184×3456 pixels), (c) maize (width × height: 2048×2448 pixels). Segmentation contour obtained with the method in [34] is overlaid in red. (d)-(f) PLA Error and (j)-(l) KL Divergence should be as close to 0 as possible. (g)-(i) Best possible DSC value is 100%.

measured by the Dice similarity coefficient (see Eq. (5.5)). As shown in Figure 5.6g-i, JPEG 2000 + ROI offers best performance, followed by plain JPEG 2000 and HEVC, whereas WebP and JPEG exhibit an erratic behavior. When the JPEG 2000 encoder is informed where to focus bit rate, JPEG 2000 + ROI clearly outperforms all other methods, achieving almost perfect segmentation even at low bit rates. Notably, on the rapeseed image, JPEG recovers at high bit rates (i.e. for JPEG quality factor above 90, cf. Figure 5.6h), encoding the large dark background regions efficiently. Comparing PLA Error to DSC, we observe that codecs obtaining comparable performance in PLA, e.g. JPEG 2000 + ROI and HEVC, differ in their performance in DSC, i.e. JPEG 2000 + ROI performs better.

Color degradation, quantified by the Kullback-Leibler divergence, is minimized by JPEG 2000 and HEVC (Figure 5.6j-l). While JPEG systematically introduces higher distortion in a plant's color, performance of WebP depends on the complexity of the image (e.g., cluttered background).

Notice the difference in bit rate ranges among test images, reflecting how much 'compression-friendly' the content is. For equivalent segmentation or color accuracy, the rapeseed and maize images can be compressed at significantly lower bit rates than Arabidopsis, due to the large uniform background regions of the former as opposed to the highly textured background of the latter, i.e. the soil, which is less efficient to encode.

Overall, the plots relative to rapeseed (Figure 5.6e, h, k) and maize (Figure 5.6f, i, l) images reveal that for JPEG 2000 + ROI bit-rates between 0.1 and 0.2 bpp (i.e. less than 1% of uncompressed 24 bpp image size) are sufficient to adequately encode such data, while for the Arabidopsis test image bit rates higher than 0.5 bpp (i.e. 2% of uncompressed image size) are recommended (cf. Figure 5.6d, g, j). As we see from this figure, allowing for even higher bit rates does not improve results with respect to the metrics employed here. In particular, using lossless compression approaches would increase bit rate even more without additional gain in performance. For example, the rapeseed test image is encoded with the lossless mode of JPEG 2000 at 5 bpp (i.e. 21% of uncompressed image size), roughly one order of magnitude more than using a lossy approach. Additionally, encoding and decoding operations are also usually considerably faster with lossy methods (e.g., with JPEG 2000, encoding the rapeseed image at 0.2 bpp is 2.4 times faster than lossless, while decoding is 4 times faster, as discussed in Section 5.4.4).

In order to give a visual impression of compression performance, Figure 5.7 shows example reconstructed images, after compression at 0.05 bpp.

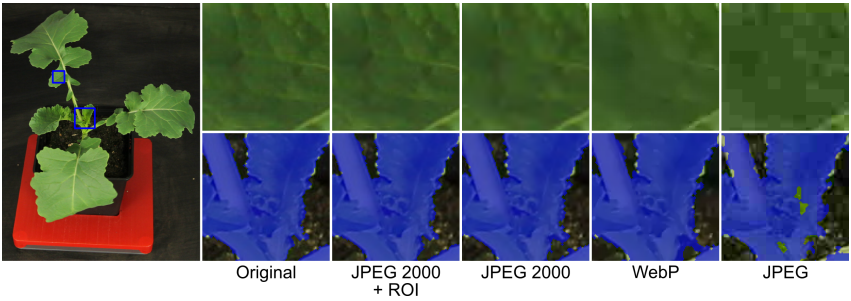


Figure 5.7: Left: a rapeseed test image; top and bottom rows: details of this rapeseed test image (in blue boxes), compressed at 0.05 bpp, using different compression standards. In the bottom row, the segmentation mask obtained with the method in [34] is overlaid in blue.

A compression factor of 1:480 is applied, to reduce uncompressed image size of 57.7 MB to approximately 112 kB. With JPEG 2000 + ROI the plant appears identical to the original, while plain JPEG 2000, without any prior knowledge on the image regions of interest, is less rich in details and the borders of the segmented plant present small errors. WebP severely over-smooths the image, thus losing the venation patterns in the leaf. Despite the low bit rate, all such images (and corresponding segmentation masks) are visually plausible, as compared to the original image. On the other hand, JPEG (using quality settings of $q = 10$ to achieve file size equivalent to other approaches) exhibits noticeable block artifacts and color degradation, introducing also larger errors (holes) in the segmentation. All of these factors may severely affect accuracy of the phenotypic analyses conducted on JPEG compressed image data.

We conclude that *newer lossy compression standards such as JPEG 2000 + ROI do offer significant benefits in bit rate reduction without degrading results significantly. However, only up to some application-dependent point, since artifacts introduced can severely affect further analysis*.

5.4.3 Local growth estimation of leaves and root tips

Measuring local growth rates in plant tissues by optical flow analysis is a widely applied method [77, 200, 236, 277, 319]. We investigate how compression affects such measurements, adopting two time-lapse sequences

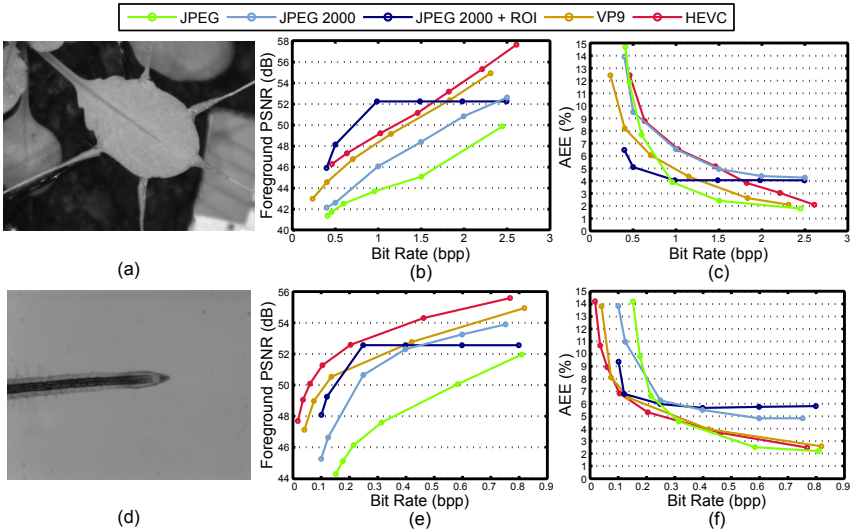


Figure 5.8: Performance of lossy coding approaches on image data used for optical flow based analysis: (a)-(c) Arabidopsis leaf sequence, (d)-(f) tobacco root tip sequence.

(videos) of gray scale images. Example image stills (frames) are shown in Figure 5.8a and d, respectively, of a growing Arabidopsis leaf (11 frames, width $\times$ height: 640$\times$480 pixels) and a growing tobacco root tip (60 frames, width $\times$ height: 740$\times$570 pixels). Effects of compression on the optical flow field can be accurately quantified using the Average End-point Error (AEE, Eq. (5.7)).

Figure 5.8 shows PSNR and AEE (see Eq. (5.7)), calculated inside a manually defined region of interest, for Arabidopsis leaf and tobacco root tip sequences, encoded at various bit rates. In general, we observe that optical flow calculations are more sensitive to image compression than other applications in previous examples. Thus, in order to keep AEE values reasonably low, we consider the highest range of bit rates (and quality) possible with lossy coding techniques.

Due to the high similarity between consecutive frames of the image sequences, video codecs (VP9, HEVC) provide considerable improvement in PSNR with respect to approaches that compress the frames independently (JPEG, JPEG 2000). Providing the JPEG 2000 encoder with region-of-

interest (ROI) information is beneficial only at lower bit rates. Above a certain bit rate (i.e. 1 bpp for the leaf and 0.25 bpp for the root tip), foreground is already encoded at the best quality possible and further bit budget is spent in background regions. Additionally, at near-lossless coding rates, the underlying effect of encoding the ROI (increased dynamic range of the values to encode, due to so-called bit plane shifting [58]) may reduce coding efficiency further.

On the other hand, image fidelity is not strictly correlated with the preservation of the optical flow fields. Surprisingly, on the shorter Arabidopsis leaf sequence (11 frames), JPEG obtains the best AEE performance at several bit rates and is always superior to HEVC (Figure 5.8c, f), despite opposite PSNR results (Figure 5.8b, e). For the longer root tip sequence (60 frames), HEVC and VP9 represent the best option for low bit rates, while JPEG is still superior at high bit rates.

We conclude that *if lossy compression is needed, JPEG at highest quality levels should be preferred, but even then additional 2% AEE due to compression should be expected.*

5.4.4 Encoding and decoding execution times

To illustrate computational complexity of the codec implementations used in this work, we outline in Tables 5.2 and 5.3, execution times for encoding/decoding of still images and video sequences, respectively. Computational experiments are conducted on a machine equipped with Intel Core 2 Duo CPU E8200 2.66 GHz and 4 GB memory, running 64-bit GNU/Linux.

For lossless compression (cf. Table 5.2), JPEG-LS and JPEG 2000 obtain encoding and decoding times below 1.5 seconds. Fastest decoding is achieved by WebP, which however presents higher encoding time than other still image compression standards. Encoding and decoding times of HEVC (using the HM implementation) are significantly higher than other codecs. Overall, shorter execution times are observed when using lossy compression (cf. Table 5.2). JPEG and JPEG 2000 achieve shortest encoding times, while average decoding times of color images remain below one second for all codecs, with fastest decoding obtained by WebP.

For image sequences (cf. Table 5.3), still image codecs are generally faster at encoding, with JPEG 2000 requiring on average less than half a second to encode the test sequences. Decoding times are in the same order of magnitude for all codecs (VP9 presents shortest decoding times). The JPEG 2000 + ROI approach results in longer execution times than

Table 5.2: Average execution times for encoding and decoding still images, expressed as mean ± standard deviation. Best results (i.e. less time) are highlighted in bold. For lossy compression, average results are shown for images compressed at a range of bit rates between 0.02 and 2 bpp.

Standard	Gray scale images		Color images	
	Encoding (s)	Decoding (s)	Encoding (s)	Decoding (s)
Lossless				
PNG	3.93(0.02)	0.63(0.03)	5.79(4.29)	0.47(0.32)
JPEG-LS	0.97(0.03)	1.10(0.01)	1.36(0.95)	1.52(1.05)
JPEG 2000	**0.74(0.02)**	0.73(0.08)	**0.78(0.40)**	0.78(0.35)
WebP	8.60(0.10)	**0.46(0.00)**	7.27(4.22)	**0.37(0.26)**
HEVC (HM)	96.24(0.14)	2.75(0.02)	–	–
Lossy				
JPEG	**0.40(0.09)**	0.30(0.09)	**0.42(0.25)**	0.34(0.21)
JPEG 2000	**0.40(0.11)**	**0.21(0.09)**	0.44(0.30)	0.27(0.20)
JPEG 2000 + ROI	0.67(0.18)	**0.21(0.07)**	1.13(0.69)	0.33(0.24)
WebP	3.46(0.27)	0.33(0.05)	2.26(1.39)	**0.20(0.12)**
HEVC (x265)	13.77(2.02)	1.36(0.24)	8.21(5.14)	0.75(0.51)

JPEG 2000, due to the ROI coding feature and no chroma sub-sampling (i.e. more data to process in the entropy coding stage of the encoder). Among video codecs, VP9 is 3 to 4 times slower than HEVC at encoding, but approximately an order of magnitude faster at decoding.

5.4.5 Discussion

Our first proof of concept experiment (Section 5.3.1) illustrates that even in the simple case of measuring rosette plant area, the most popular form of lossy image compression (i.e. images compressed with JPEG) does introduce non-negligible errors in measurements. Compression in this case does not cause visually perceptible distortion, but local loss of image fidelity does affect the outcome of segmentation: the image processing process that lies beneath the measurement of plant rosette area. More importantly, although it appears that with compression PLA is overestimated, the effect of compression on the algorithm is not constant, i.e. it is not a systematic error. Unfortunately it is not a completely random error either: it varies as plants grow (as the ANOVA experiment shows), from a time instant to another, and between plants. Although the ANOVA

Table 5.3: Average execution times obtained to encode and decode the sequences for optical flow analysis at a variety of bit rates using lossy coding standards, expressed as mean ± standard deviation. Best results (i.e. less time) are highlighted in bold.

Standard	Arabidopsis leaf		Root tip	
	Encoding (s)	Decoding (s)	Encoding (s)	Decoding (s)
JPEG	0.21(0.06)	0.17(0.07)	0.96(0.17)	0.73(0.21)
JPEG 2000	**0.14(0.03)**	0.10(0.04)	**0.41(0.07)**	0.28(0.09)
JPEG 2000 + ROI	0.54(0.07)	0.21(0.08)	2.64(0.09)	0.84(0.39)
VP9	6.82(1.39)	**0.05(0.04)**	30.29(7.36)	**0.16(0.02)**
HEVC (x265)	2.08(0.40)	0.47(0.09)	6.75(3.51)	1.61(0.64)

identifies this to be a factor, it is more readily seen in scatter plots: ordering of points changes from time to time, and even more critically the behavior changes among different plants of the same genotype. If compression was a systematic error then this would simply introduce a bias (a change in population means) which would not affect any statistical tests. If compression was a totally random and uncorrelated to the data error, then this would simply imply that larger variance attributed to compression is observed and to account for this additional variance a larger sample (more replicates) would merely be necessary, in order to match the statistical power of the data without compression. But also this is not the case.

Compression is a highly influential factor also when growth analysis relies on optical flow fields. Tasks involving the tracking of high contrast structures, e.g. root tip, generally prove robust to higher compression ratios, however, for growth analysis JPEG compression should be limited to very high quality factors ($q \geq 95$). Using more sophisticated compression standards (e.g., JPEG 2000, VP9, HEVC) may not yield better results (cf. Section 5.4.3). If the image data shows low contrast in relevant image regions, lossless compression should be adopted to avoid dramatic degradation in accuracy.

These findings are illuminating, since when compression is present without a user's knowledge, the measurements would be affected by the compression. Here, we could observe these errors because the data are acquired in original uncompressed quality. Therefore, users of image-based phenotyping platforms should first identify if compression is used in their system, report it in their papers, and analyze its effects by obtaining

some data without compression.

There are several practical reasons that do necessitate the use of compression. The rapid accumulation of data and the need to archive such data for regulatory compliance is the most common. In this case if ample storage is available then without a doubt, as our experiments show, the lossless compression options of JPEG-LS or JPEG 2000 should be considered, since they can still reduce by 60 to 80% (depending on the image characteristics) the amount of data (roughly equivalent to 2-3 bpp per each color channel), while perfectly preserving the original image content. Despite obtaining inferior compression efficiency with respect to other methods, the PNG standard is ubiquitous on the Web and its broad installed codec base eases adoption. The benefits of compression could be realized even in depositing or retrieving data from institutional repositories, where compression will maximize the utilization of the installed e-infrastructure.

If larger compression efficiency is required lossy options are necessary. Depending also on the complexity of the image content at hand (i.e. images with less complicated background), most compression algorithms offer near lossless performance in the 2-3 bpp bit rate range, with no major differences observed among algorithms.

For additional storage savings, below 2 bpp compression efficiency is required. There are several scenarios where such efficiency may be necessary. For example, when images are acquired in a greenhouse facility or even in the field, and are then transmitted to a central location for archival and analysis (e.g., as in the distributed framework proposed in this thesis, or in the gigapixel time-lapse panoramic imaging system in [35]). Another example could be the recent developments towards affordable phenotyping where users in developing countries or in rural remote areas acquire images using affordable and low computational power devices (e.g., mobile phones), and transmit them over wireless communication links (enabled in remote places by long-distance connectivity projects [221] or emerging technologies such as the Brck[10]) and the Internet to cloud services, where sophisticated analyses take place, and results are sent back in response [216, 256]. Both of these scenarios involve: a remote sensing device, which does not have the computational power to perform analysis; the use of a limited communication channel, which may not have the capacity to carry many large images; and potentially imaging

[10] `http://www.brck.com`

of plants in non-ideal settings, for example in the field [8, 38] or non-uniformly illuminated conditions, which increase the complexity of the image content. Thus, storage and transmission of the image data represent a (technical and logistic) bottleneck and may reduce overall throughput, rendering image file size a key design aspect.

For compression efficiency below 2 bpp careful evaluation of compression effects and choice of compression practice is necessary. If prior to the final deployment of the system, a set of uncompressed imaging data of a genotype and a fixed image processing pipeline are available, direct phenotyping measurements, such as PLA or model errors, can be used to evaluate compression. Statistical analyses, such as the one in Section 5.3.1, should be performed and the choice of the compression algorithm and its parameters (e.g., JPEG quality factor or bit rate) should be guided accordingly to minimize statistical effects. If group based experiments (of different genotypes) are available, changes in group level differences can be used to identify suitable compression standards and parameters.

When populations are not available but some exemplar images are available instead, then analyses such as the ones reported in Section 5.4 are recommended. If no imaging pipeline is available and human-based image annotation is adopted, visual examination and general fidelity metrics (PSNR) or psycho-visual metrics (e.g., structural similarity index, SSIM [326]) can be used to find suitable choices of compression and parameters, such that the user's capability of performing the analysis (e.g., delineating roots) is not affected.

When an image analysis pipeline is also available we recommend not only the use of application related measurements (e.g., PLA growth rates, tracking estimates in root tip, and others) but metrics such as segmentation quality and color divergence. They not only offer more sensitive evaluation (cf. DSC vs. PLA Error in Section 5.4.2) when compared to general fidelity metrics (e.g., PSNR) but can help address future changes to the analysis pipeline [216, 291]. This is necessary for example when performing new analyses to isolate new traits and explain behavior not considered during the initial experimental design and data collection.

In general, performance of coding procedures in terms of quality (or, alternatively, error) measures, suitable for the application at hand, can be visualized when plotted versus the bit rate achieved by compression. This represents a practical tool to operate lossy compression in applications. When designing a phenotyping setup, the so-called rate-distortion (R-D) curves [232] (e.g., those employing PLA Error, DSC, or KL Divergence as

distortion measure in Figure 5.6) allow to select compression approaches and parameter settings that provide the optimal trade off between compression ratio and application accuracy. For a desired level of phenotype extraction fidelity (y-axis), the compression methodology providing the lowest bit rate (x-axis) is the most efficient and should be adopted.

Several compression tools are available and the selection of an appropriate image compression strategy is not trivial. Therefore, in this chapter we compare quantitatively a variety of state-of-the-art image and video coding standards, focusing discussion on aspects of practical relevancy: (a) compression efficiency, (b) image fidelity, (c) phenotyping accuracy, and (d) encoding/decoding time efficiency. Until specialized compression algorithms tailored to the problems of plant phenotyping become ubiquitous (see the following chapters), based on our analysis we recommend the following.

- For still images, JPEG 2000 emerges as the approach achieving the best trade off among all parameters, offering noteworthy (and in several cases top) performance in all experiments but motion estimation. When regions of interests are available, for example after data analysis, JPEG 2000 + ROI offers an exceptional choice to archive data with the highest possible quality and compression efficiency. The limiting factor of JPEG 2000 is the lack of large installed codec base due to its limited popularity. This implies that appropriate software installation on workstations and other computing devices is necessary.
- JPEG should be avoided since it performs poorly in most occasions even though it is the first level of choice among users and is ubiquitous.
- When image sequences (or videos) are concerned, the new HEVC video coding standard should be used to achieve high space savings especially for long sequences (or stacks) of images with static background. Furthermore, HEVC is an excellent option for long-term storage of time-lapse sequences for growth estimation or low-resolution video streaming (e.g., transmitting a stream of low-resolution previews acquired at the sensor would allow a user to remotely check system status and adjust parameters to changing conditions, or even operate robotized solutions [5]). Its limitations are the additional computational burden introduced by video coding and not a large installed codec base, however, the latter is changing rapidly as more

software and hardware manufacturers will include such codec in their distributions.

5.5 Summary

In this chapter we showed that image compression used in image-based plant phenotyping experiments may compromise phenotyping results and needs to be taken into account. We supported this hypothesis with three illuminating proof of concept experiments which demonstrate that compression (especially in its most common lossy form of JPEG) does affect measurements of plant traits and errors introduced can be high. We further systematically explored how compression affects measurement fidelity, quantified as effects on image quality as well as errors in extracted plant visual traits (e.g., size and color of shoots, leaf and root growth, as well as root system analysis). Overall, we found that compression has a considerable effect on several types of analyses and that proper care is necessary to ensure that such choice does not affect biological findings.

The general-purpose image compression standards adopted in this chapter are tuned to the human visual system (HVS) model, and their parameters are optimized for better perception of the image by the human eye. Tuning their parameters empirically to maximize a different objective (e.g., accuracy of phenotype extraction) is difficult and may not guarantee best results, particularly for transmission purposes in a distributed sensing scenario, where a low computational power sensor must compress the acquired images to meet (tipically) tight bandwidth constraints. In the following chapter we will describe in detail our distributed sensing and analysis framework, and introduce the concept of application-aware image compression. Subsequently, we will present specialized compression techniques that tailor compression to the application that will use the image data, and are therefore well suited for adoption in an image-based plant phenotyping context.

6

Application-aware image compression for distributed plant phenotyping

6.1 Distributed sensing and analysis framework

Currently, the solutions available for plant phenotyping are either destructive (thus not repeatable) and low-throughput, or high-throughput and costly. The current approach to automated phenotyping relies on imaging sensors and processing station(s). Usually these units are tightly coupled (i.e. sensing and analysis occur in the same physical location), which limits: (a) the scalability of the system (a throughput increase requires more processing stations), (b) the ease of deployment to new facilities (e.g., moving a cluster of PCs is logistically complex), and (c) the efficiency of using the available computational resources (e.g., idle time is not utilized).

The combination of low-cost smart sensors with Internet connectivity and a cloud infrastructure can mitigate the above limitations [111]. To keep the cost of the sensor low, minimal robotics and minimal computational

This chapter is partly based on:

- M. Minervini, S. A. Tsaftaris, "Application-aware image compression for low cost and distributed plant phenotyping," in *18th International Conference on Digital Signal Processing*, Jul. 2013, pp 1–6.

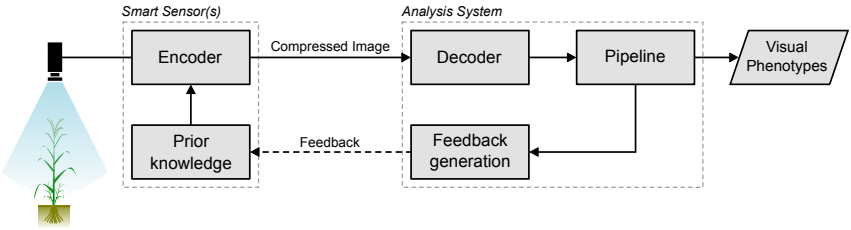


Figure 6.1: Schematic of the proposed distributed sensing and analysis framework.

power on the sensor are assumed, with the bulk of analysis occurring at remote infrastructures.

Figure 6.1 illustrates in more detail the proposed distributed architecture for plant phenotypes collection that we first introduced in Chapter 1 (cf. Figure 1.2 on page 8). Our approach is characterized by a low computational power sensor (or a grid of sensors, e.g., to cover broad areas or for multi-view imaging) that acquires time-lapse image sequences of a scene containing plants and sends them to a receiver, which we assume to have higher computational and storage capacity (e.g., a service running on a cloud computing infrastructure).

At the receiver, an analysis system processes each incoming image of the time-lapse sequence, in order to extract visual phenotypes relevant to plant scientists. Usually, plant objects are first localized, then a more sophisticated segmentation algorithm is employed to accurately delineate the plant boundaries against the background (cf. Chapter 4). The primary output of this process is a collection of masks that identify pixels belonging to each of the plants in the scene, used to measure plant visual traits (e.g., projected leaf area, average color intensity, rosette shape). Further analyses can be performed on the images, leading to even more fine-grained phenotyping information (e.g., leaf count, individual leaf area, leaf tilt angles).

However, this novel direction requires the transmission of imaging data to the now disconnected processing units. Phenotyping experiments may involve hundreds of plants, imaged several times per day, over periods of weeks, thus yielding vast amounts of image data. Therefore, transmission (but also archival) of full resolution uncompressed images becomes soon prohibitive. On the other hand, the case studies presented

in Chapter 5 demonstrate that indiscriminate compression may degrade the quality of the data and compromise its utility, thus care must be taken. Besides, general-purpose compression standards are usually designed with the human visual system in mind, and their plain adoption in our distributed sensing and analysis framework may not yield best results for the application.

Motivated by findings of Soyak et al. [291], where an application-aware approach to video compression leads to an 80% reduction in bandwidth requirements, in the following chapters we propose several approaches for application-aware compression of images, and we demonstrate their efficacy in a phenotyping context. We assume that the sensor is aware of its content (i.e. it is imaging plants) and is aware of the final task of extracting features for plant segmentation and phenotypic analysis, which occurs automatically at the cloud based on automated computer vision algorithms (cf. Chapter 4). In the next section we provide an overview of approaches in the literature to application-aware image and video compression.

Finally, we also exploit feedback from the receiver (cf. Figure 6.1), which acts as an 'oracle' with respect to the sensor: based on the output of the analysis, the receiver feeds back to the sensor information useful to improve its ability to compress the acquired images. This feedback may take on different forms depending on context, and conveys prior knowledge of the application to the encoder operating at the sensor.

6.2 Application-aware image compression

Compression schemes typically aim at maximizing image fidelity, i.e. minimizing mathematical difference between original image pixel values and the corresponding reconstructed ones, measured for example by mean squared error or PSNR. On the other hand, since the ultimate receiver of image and video data is often a human, major research effort is spent in the so-called 'psycho-visual optimization' [193,326], and some approaches have been included in recent coding standards.

On the other hand, in present days, more often than not image data are analyzed by computer vision algorithms and their transmission over channels necessitates their compression to reduce bandwidth costs (as we highlighted in the previous section for distributed plant phenotyping). Progress in vision and automation technology is leading to increasingly

many applications in which images are acquired, transmitted, and analyzed, largely without any human intervention. Prime examples of such applications are traffic video surveillance, remote sensing, medical image analysis, and distributed sensing and analysis.

In all of these contexts, the images are consumed by automated analysis algorithms and possibly never seen by a human. It was shown recently that considering the application and designing data codecs appropriately not to maximize fidelity type criteria (e.g., mean squared error [327]) or psycho-visual criteria (e.g., structural similarity [326]), but considering how would an analysis algorithm (e.g., a classifier) perform on compressed data, is beneficial from a bit rate perspective [291].

Examples of application-aware approaches to lossy compression are available in the literature, for example optimizing quantization tables [48,291,325], or focusing bits spatially in regions of interest [185,202,203] (cf. Chapter 6). Other approaches aim to preserve specific image features typically used by computer vision algorithms (e.g., texture [276], edges [224,240,276,313], SIFT and SURF descriptors [47,49,50], or binary descriptors [259]). In [129] a distortion metric specific to the lossy compression of DNA microarray images is proposed. Rate-accuracy models have also been studied for visual sensor networks, where image communication is performed on energy and resource constrained devices [260,261]. A recent example of application-aware rate-distortion optimization for the JPEG 2000 image coding standard can be found in [255], where a distortion metric based on conditional class entropy is proposed to tailor compression to target detection tasks.

In this thesis, we focus on classification as our target application, in consideration of its central role in a variety computer vision tasks (e.g., object detection, segmentation, image retrieval) and application domains (e.g., medical imaging, precision agriculture, remote sensing, robot navigation, video surveillance [155], industrial inspection [3]). Early studies on the effects of lossy compression on classification can be found in [238]. In [80], scalar quantization of synthetically generated signals is optimized for classification at the decoder, using a distortion metric based on the Kullback-Leibler divergence. Dong et al. [81] define a Lagrangian distortion metric accounting for reconstruction and classification accuracy (based on Hamming distance), and design scalar and vector quantization for joint compression and classification of underwater acoustic data. The joint design of vector quantization and classification is discussed in [181,230], combining the squared error and a Bayes risk term to mea-

sure classification accuracy.

6.3 Summary

In this chapter we discussed our proposed distributed sensing and analysis framework for image-based plant phenotyping experiments from an image transmission standpoint. Separating sensing and analysis keeps the cost of the sensor low, while still allowing to carry out sophisticated image analysis tasks on the cloud, however, it introduces the need for compressing the transmitted data. The sensor collects and compresses the images in an application-aware fashion being aware of processing that will occur in a centralized (remote) location.

In the following chapters we will demonstrate how the concept of application-aware image compression on resource-constrained devices can be incorporated in the image encoding process, by injecting prior knowledge of the application in several different ways. We will start in Chapter 7 with a low-complexity approach to estimate regions of interest in an image, which are then encoded at higher quality than background. We will continue in Chapter 8 with a data and application dependent color transform with class separation properties that efficiently combines color space transformation and ROI estimation. While such approaches rely on the 'explicit' specification of the image regions to preserve (via either an external ROI estimation module or a color transform), in Chapter 9 we will 'implicitly' inject application-awareness via a specialized distortion metric utilized in the rate-distortion optimization framework of the encoder.

Compliance to well established international standards is preferred in this context to a customized compression scheme, to permit portability of the image data and interoperability with other systems, without the need for a specialized decoder. Therefore, in the following we will adopt two state-of-the-art compression standard to demonstrate our methodologies: JPEG 2000 [287] and the recent High Efficiency Video Coding (HEVC) [295]. Our approaches will modify the behavior of the encoder in a way that the resulting bit stream is standard compliant and a standard decoder can be employed at the receiver.

7 Saving bits in space

7.1 Introduction

In this chapter we discuss an approach to application-aware image compression based on spatial prior knowledge obtained at the sensor. To mitigate the effect of lossy image compression on plant segmentation occurring at the analysis, in Section 7.2 we propose a region-of-interest (ROI) estimation algorithm that utilizes feedback information from the receiver, i.e. the 'oracle'. We compress the images using the JPEG 2000 standard [287] and its ROI coding feature. Overall, the solution has low computational requirements and the generated bit stream is standard compliant. We find that with respect to both traditional metrics (such as PSNR) and application-aware metrics, the proposed solution provides a 70% reduction of bit rate for equivalent performance.

This chapter is partly based on:

- M. Minervini, S. A. Tsaftaris, "Application-aware image compression for low cost and distributed plant phenotyping," in *18th International Conference on Digital Signal Processing*, Jul. 2013, pp 1–6.
- M. Minervini, C. Rusu, S. A. Tsaftaris, "Learning computationally efficient approximations of complex image segmentation metrics," in *8th International Symposium on Image and Signal Processing and Analysis*, Sep. 2013, pp. 60–65.

At the sensor we adopt a low-complexity ROI estimation approach based on a pixel-level classifier. Since training a classifier is usually a costly operation in terms of computation and memory, we assume that the sensor obtains from the 'oracle' model parameters necessary to perform the ROI estimation effectively. On the other hand, the 'oracle' does not have access to the original uncompressed images, therefore encoder options (e.g., target bit rate, quantization) required to maintain a desired level of plant detection accuracy need to be determined at the sensor. To enable the sensor to make informed decisions on the encoding and transmission of the acquired images in an application-aware fashion, an accuracy metric needs to be adopted at the sensor to compare segmentation outcome of images compressed using different options with the segmentation of the corresponding uncompressed images (representing in this context the 'golden standard').

Several image segmentation metrics have been proposed in the literature [268]. Some metrics are easy to compute (e.g., Dice, Jaccard), while others are more accurate (e.g., the Hausdorff distance) and may reflect local topology, but they are complex to compute. In Section 7.3 we construct approximations of the modified Hausdorff distance [85], combining a small number of computationally lightweight metrics in a linear regression model. We also consider feature selection, using sparsity inducing strategies, to restrict the number of metrics employed significantly, without penalizing the predictive power of the model. Using image data from plant phenotyping experiments, we find that a linear model can effectively approximate the Hausdorff distance using even a few features.

7.2 Saving bits in space: plant image compression based on regions of interest

In application-aware image compression, a particularly useful feature is region-of-interest (ROI) coding. An ROI is an image region relevant to the user (e.g., plants in plant phenotyping applications), thus, it should be preserved in the lossy compression process, by encoding it with better quality than background regions. ROIs (possibly composed by multiple disconnected objects) in an image I, can be represented as a binary mask M, where $M(i,j) = 1$ indicates that the pixel at that location is considered part of the foreground, whereas $M(i,j) = 0$ denotes a background pixel.

As described in Chapter 6, images are acquired by an affordable cam-

era sensor and their compression involves some prior knowledge of the application. Here, application-awareness is injected by an ROI estimation module running on the sensor, which identifies in the original uncompressed image the regions where (most likely) plants are located. Then, the raw image is encoded using the JPEG 2000 compression standard [287] (based on a discrete wavelet transform) and its ROI coding feature.

ROI coding in JPEG 2000 is described in Part I of the standard and is based on the Maxshift method [16]. By appropriately scaling the wavelet coefficients, the information related to the ROI is placed in higher bit planes than the background, thus eliminating the need to transmit the ROI shape explicitly. In the bit stream formation process, bits pertaining the ROI are placed first, and a truncation of the bit stream allows to satisfy a bit rate requirement, while encoding the regions of interest at the highest quality possible for the given target bit rate. The resulting bit stream is transmitted over a link (e.g., using a wireless or even a mobile network) to a receiver (e.g., a cloud system), where a standard JPEG 2000 decoder reconstructs the image, and automated analysis follows.

Several different approaches for ROI estimation can be considered. However, the method should provide smooth ROIs and as accurate as possible (to eliminate bits spent on non-relevant portions of the image), without being computationally intensive (it runs on an affordable sensor). In the following paragraphs, we describe our approach to identify ROIs in plant images based on plant appearance.

7.2.1 ROI estimation with feedback

As we discussed in Chapters 2 and 4, images acquired in a plant phenotyping context present several computer vision challenges (e.g., presence of moss, water reflections). While simple thresholding algorithms can provide rudimentary foreground identification, they are often far from the best solution. More sophisticated approaches are necessary, however, they require additional computational resources. Thus, to increase the robustness of the ROI estimation component, while keeping its computational complexity low, we propose to shift part of the computational burden on the receiver. This approach can conveniently drive the sensor in detecting plant objects, by sending some helpful feedback information φ. Such feedback is generated by the analysis system (e.g., based on previously processed images, or on additional information from an external knowledge base), and is transmitted to the sensor to improve its

compression capabilities in an application-oriented fashion.

On the receiver side, we assume the possibility of training a supervised binary classifier to classify pixels in an image as belonging to foreground (plant) or background. For the classifier to correctly learn the appearance of plants, an initialization step can be considered, such that the sensor sends the first acquired image uncompressed. Alternatively, a model from a previous experiment (with plants of the same species) could be employed.

We use a multivariate Gaussian Mixture Model (GMM) to learn the plant intensity distribution. Accordingly, we model the d-dimensional feature vector $x \in \mathbb{R}^d$ representing the intensity values of a pixel location, as belonging to a mixture of multivariate Gaussian distributions. Thus, each component of the mixture is defined as:

$$p(x|\Theta_j) = \frac{\exp\left(-\frac{1}{2}(x-\mu_j)^\top \Sigma_j^{-1}(x-\mu_j)\right)}{(2\pi)^{\frac{d}{2}}|\Sigma_j|^{\frac{1}{2}}}\,, \tag{7.1}$$

where $\Theta_j = (\mu_j, \Sigma_j)$ are the parameters of the j^{th} component of the mixture (i.e. mean μ_j and covariance Σ_j). An M-component mixture model is therefore characterized by the density function:

$$p(x|\Theta) = \sum_{j=1}^{M} \pi_j p(x|\Theta_j)\,, \tag{7.2}$$

where $\pi_1, \ldots, \pi_M$ are the mixing coefficients, and each π_j is the prior probability of pattern x belonging to the j-th component, such that $0 \leq \pi_j \leq 1$, for $j = 1, \ldots, M$, and $\sum_{j=1}^{M} \pi_j = 1$. The parameters of the distribution and the mixing coefficients $\Theta = (M, \pi_j, \mu_j, \Sigma_j)$, are estimated from available plant data samples, by maximizing the log-likelihood function:

$$L(\Theta) = \sum_{i=1}^{N} \log\left(\sum_{j=1}^{M} \pi_j p(x_i|\Theta_j)\right), \tag{7.3}$$

using the Expectation-Maximization algorithm [303].

We represent pixel intensities in the 1976 CIE *L*a*b** color space [143, 144], because of its capability to decorrelate luminance (encoded in the *L** component) from chrominance (encoded in the *a** and *b** components), which makes it less susceptible to lightness variations than the original

RGB color space. This process is executed at the receiver, thus learning an appearance model of the plants.

At the encoder side, ROI estimation is attained by first converting the acquired image to CIE *L*a*b**. (To reduce computational cost on limited-resource devices, the conversion is performed using a color look-up table (LUT) [64].) Then, the intensity vector of each pixel is evaluated in Eq. (7.2), to obtain the probability of that pixel belonging to a plant object. Pixels having a probability above a given threshold T are considered foreground and, thus, included in the ROI, while pixels with probability below the threshold T are assigned to the background. Finally, as the thresholding operation may result in noisy and fragmented regions, which increase the complexity of the ROI (thus penalizing compression efficiency), the obtained binary mask is post-processed by small objects removal (a fixed threshold for the area is set to A_{max} pixels), morphological dilation, and hole filling.

Although the encoder can use fixed thresholds T, A_{max}, here we rely on the receiver's feedback to provide optimal thresholds to the sensor. In essence the receiver is aware of the ROI estimation algorithm on the encoder. The thresholds T, A_{max} are estimated, based on the previously observed image of the time-lapse sequence and its segmentation mask. Optimal values are found that maximize the spatial overlap, measured by Dice Similarity Coefficient, $DSC = 2 \cdot |S \cap S'|/(|S| + |S'|)$, between the segmentation mask S of the plant segmentation algorithms, and the binary classification S' obtained with the GMM.

A further improvement in the efficiency of the plant detection process could be achieved, by pre-computing on the analysis system (e.g., in the form of a look-up table) the region in the pixel intensity domain that corresponds to values above the threshold T in the probability density domain, thus removing the need of evaluating Eq. (7.2) on the sensor.

Both learning the GMM and identifying the optimal thresholds T and A_{max} are computationally expensive tasks, hence unfeasible on limited-resource devices. In order to overcome this limitation, we propose to run these tasks at the receiver, which sends $\varphi = \{\Theta, T, A_{max}\}$ as feedback information to the sensor. We should note that the feedback vector φ is composed by few double precision numbers, hence the network overhead for their transmission is limited to approximately only a hundred bytes.

Application-oriented evaluation metrics

Lossy image compression algorithms are usually evaluated by measuring quality of reconstruction of the original signal, and typically Peak Signal-to-Noise Ratio (PSNR) is employed as a metric (cf. Eq. (5.2) on page 103). However, PSNR alone may not be sufficient in particular contexts, such as when evaluating application-specific systems, and in fact using application aware metrics may result in bit rate savings [291].

As part of our methodology, along with traditional PSNR, we propose to evaluate compression using specialized metrics, in order to take into account several different aspects of the problem at hand. In particular, bearing in mind the final application and prior information on the content (e.g., top view of plants), we are interested in designing a compression scheme that does not affect the accuracy of the segmentation algorithm, and preserves the low-level features used by image analysis algorithms to extract visual phenotypes.

Let I, $\hat{I}$ be original and decoded images, respectively, and S, $\hat{S}$ the corresponding segmentation masks obtained at the receiver side. In our context of application-oriented compression, we consider the segmentation mask S as ground truth for evaluating the segmentation accuracy. This represents the best-case input for a given algorithm [291]. Accordingly, we measure the quality of the content, by comparing original and reconstructed images, only for pixel locations belonging to the segmentation mask S. Therefore, in our attempt to assess the performance of our system without explicitly performing actual phenotype analysis, we adopt the following set of image based metrics, that quantify the fidelity of the images.

1. *Precision*, is the fraction of pixels in the segmentation mask $\hat{S}$ that matches the ground truth S;
2. *Object-level Consistency Error (OCE)* [248], is based on Jaccard Similarity coefficient (i.e. it measures the spatial overlap between binary objects), but is more sensitive to over- and under-segmentation;
3. *Structural SIMilarity (SSIM)* index [326], measures loss of correlation, contrast distortion, and luminance distortion in the reconstructed image $\hat{I}$;
4. *ExG Normalised Mean Squared Error (NMSE)* between Excess Green (*ExG*) transforms of I and $\hat{I}$;

5. *Gradient NMSE* between image color gradient maps of I and $\hat{I}$.

All of these metrics take on values between 0 and 1, with larger values indicating higher agreement between algorithmic result and ground truth. *Precision* and *OCE* address the accuracy of the segmentation mask, while the remaining metrics measure the preservation of the content. *SSIM* takes into account structural information, the *ExG* domain is often used in plant localization tasks, and the gradients are a low-level feature utilized by several computer vision algorithms (e.g., methods aiming at segmenting individual leaves may rely on edges to distinguish overlapping leaves).

In order to obtain a single number representing the overall accuracy of the system at a given bit rate, we linearly combine the aforementioned metrics, as $Accuracy = \sum_{i=1}^{5} \alpha_i m_i$, where m_i are the employed metrics. The α_i parameters allow to increase or reduce the effect of each metric, depending on its relevance for the application (e.g., if rosette area is the only trait of interest, SSIM and gradient accuracy may be assigned a lower weight).

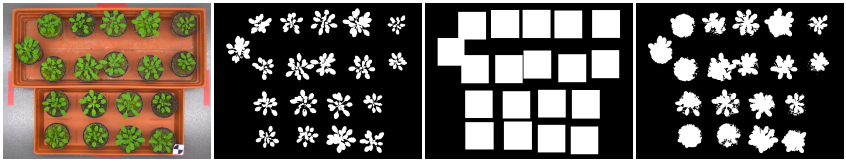
7.2.2 Results and discussion

In this section we first describe our experimental setup, which includes the data sources and computational environment, as well as the process used to segment plants on the receiver. We also describe other computationally efficient ROI estimation methods which are used for comparison.

Experimental setup

We evaluate our proposed system on time-lapse images of Arabidopsis, from the Ara2012 dataset described in Chapter 3. Figure 7.1a shows an example image from the dataset. We implement our system using Matlab R2011b, on a machine equipped with Intel Core 2 Duo CPU E8200 2.66 GHz and 4 GB memory, running 64-bit Linux. We adopt the JJ2000 software implementation[1] of the JPEG 2000 standard, to compress the original images at various bit rates with and without the ROI. For evaluation purposes, we also include in the comparison the traditional JPEG standard, using the codec implementation available in Matlab. The α_i weights in *Accuracy* are set to 0.125 for OCE and gradients, and for all other metrics are set to 0.25. For the proposed approach, the receiver estimates the

[1] Available at: `http://code.google.com/p/jj2000/`



(a) (b) (c) (d)

Figure 7.1: (a) Original image, and examples of ROI masks obtained using: (b) proposed method, (c) bounding box around fixed pots, and (d) Otsu's thresholding.

GMM and feedback based on another image (and its segmentation) which is not used in our experiments to eliminate bias.

Plant segmentation. Although our framework is generic, to segment plants in the images transmitted to the cloud we adopt the approach based on incremental learning and active contour model described in Chapter 4. For a fair comparison of the different compression algorithms, and to appreciate the true effect of compression on segmentation accuracy, the initial contour for the active contour model is fixed for each image and kept constant across the experiment. The active contour model is executed for each image (compressed and uncompressed) providing the S, $\hat{S}$ segmentation masks needed for performance evaluation.

Baseline ROI approaches. To demonstrate the accuracy of our method, and the complexity of finding a good ROI without computationally intense processes, we implement two baseline ROI extraction approaches. One that relies on fixed placement of the objects in the scene, and one that estimates automatically a foreground mask based on intensity thresholds.

Figure 7.1c shows an ROI mask where pots are assumed to be in fixed positions, and is provided to the encoder. This approach imposes strict constraints on the user, because the positions of the objects have to be manually coded into the ROI detection module and must be preserved throughout the whole duration of the experiment. Any deviation (e.g., plants may shift when watered) would result in the loss of accuracy for portions of plants outside the ROI, which will affect the validity of the phenotyping analysis.

To implement the second baseline approach, we transform the original

(a) (b) (c) (d) (e)

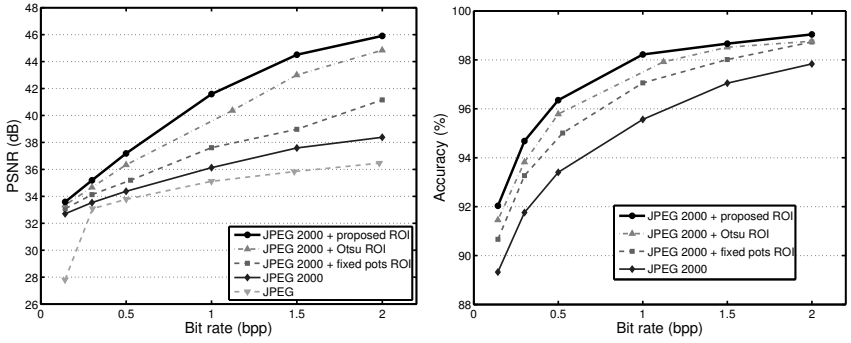
Figure 7.2: (a) Original image and (b) a detail, reconstructed after compression at 0.2 bpp with different algorithms: (c) proposed method, (d) plain JPEG 2000, and (e) JPEG.

RGB image to the Excess Green (*ExG*) domain, with $ExG = 2G - R - B$, where R, G and B are red, green, and blue channels of the RGB color space, respectively [112]. Then, we use Otsu's method [233] to identify an optimum threshold. Pixel locations having an *ExG* value higher than the threshold are included in the ROI mask, while the remaining pixels are considered background. Similarly to the proposed method, the obtained binary mask undergoes a post-processing: small objects removal (a fixed threshold for the area is set to $A_{max} = 20$ pixels), morphological dilation, and hole filling.

Although the latter approach introduces flexibility in the positioning of plants in the scene, more complex scenes (e.g., varying neon light illumination, water reflection, or moss presence) lead to an erroneous estimation of foreground (usually an over-segmentation) as Figure 7.1d illustrates.

Results

Figure 7.1 shows examples of the ROI masks used by the encoder based on the proposed and baseline approaches. It is readily apparent that the proposed method provides more accurate ROI masks without actually increasing the complexity at the sensor. The other methods lead to over-segmentation either by design (fixed squares) or due to complexity in the scene (e.g., when moss is present in the scene). This results in bits spent encoding information not related to our true objects of interest, as shown in Figure 7.2. Traditional JPEG exhibits color distortion and blocking artifacts (due to its block-based discrete cosine transform). On the other hand, JPEG 2000 introduces blurring artifacts that over-smooth textured regions



(a) (b)

Figure 7.3: (a) PSNR, and (b) application-specific accuracy, at various bit rates.

(some algorithms rely on texture information, that would in this case be lost). When introducing ROI coding with the proposed system, JPEG 2000 retains as much information as possible from the foreground and preserves well edges and texture, while roughly encoding the background.

To quantify the effects of using ROI coding and feedback to the sensor across bit rates, we evaluate application-specific accuracy (as defined in Section 7.2.1) and PSNR for different compression schemes: (a) JPEG, (b) JPEG 2000, (c) JPEG 2000 with rectangular ROI assuming fixed pot positioning, (d) JPEG 2000 with ROI obtained with Otsu's thresholding, and (e) proposed method. PSNR results (Figure 7.3a) confirm the superiority of our method already appreciated in visual findings: JPEG 2000 shows an increase of up to 2 dB with respect to JPEG, while up to 8 dB can be gained by using ROI coding. The curves in between illustrate the benefit of a more accurate ROI mask, as less bits are spent on the background, in favor of the foreground. Application-specific accuracy (Figure 7.3b) reflects an analogous ranking of the systems (curve for JPEG is omitted for clarity). The proposed approach for estimating the ROI with feedback from the analysis system provides higher accuracy in the segmentation, and preserves better the content, in terms of fidelity for image features that can be extracted for subsequent analysis.

Finally, we should note that the benefits of a more accurate ROI mask may be diminished, if the ROI mask itself becomes exceedingly complex. While this is not the case with rectangular or circular ROIs, arbitrary ROI

masks should be kept to a low complexity of their shape (e.g., by means of the morphological operations discussed in Section 7.2.1).

7.3 Computationally efficient image segmentation metrics

In the previous section, we detailed our approach to compress images in the proposed distributed sensing and analysis framework for plant phenotyping. The automated analysis occurring at the cloud on the reconstructed images is based on a first step of plant delineation from the background. Thus, it is important to find at the sensor appropriate encoding parameters to maximize segmentation accuracy while meeting bandwidth constraints. Assuming that a low-complexity, approximate segmentation method is available at the sensor (e.g., the ROI estimation module defined previously), suitable compression parameters can be found by evaluating segmentations of images compressed with different parameters against the segmentation of the corresponding uncompressed images, maximizing a segmentation accuracy criterion. Thus, accurate and yet low-complexity evaluation metrics are needed at the sensor.

Several simple metrics, such as Dice or precision/recall, are widely used in the literature, however they often prove inadequate at distinguishing 'good' segmentations from 'bad' ones in specialized applications and according to human intuition of the problem. To overcome this limitation, more sophisticated metrics have been proposed that emphasize particular types of error (e.g., over- and under-segmentation [199], topological disagreements such as region splitting and merging [149]), can be robust to noise (e.g., small boundary displacement [258], white noise [85]), or correlate well with human intuition [32, 85, 149].

The Hausdorff distance [85] has been used in a variety of applications, including the validation of image segmentation results [19, 105]. However, it is characterized by high computational complexity. Thus, considerable research effort has been put in designing efficient algorithms for its calculation (either precise or approximate) in relation to specific contexts (e.g., polyhedral objects represented as polygonal meshes), while its exact calculation in the most general case of point sets (e.g., when comparing segmentation results) still lacks an efficient solution [134].

Here we address this problem from a different viewpoint, considering a data driven approach. We propose to learn from the data a (linear)

model of an accurate (although computationally complex) metric, using as features simpler to compute metrics. To demonstrate our approach we use the Modified Hausdorff Distance (MHD) [85], for which we learn efficient linear approximations. We test our methodology on a dataset of plant image segmentations obtained from the Ara2012 dataset described in Chapter 3. We find that a linear model is effective at approximating the MHD, and feature selection strategies allow to reduce the number of metrics employed, without penalizing prediction accuracy.

In general, this framework can be utilized on sensing devices with limited computational and storage capacity, which may lack the necessary resources to compute complex metrics, when operating on high-resolution images (e.g., the low-cost sensor for plant phenotyping discussed in this thesis) or under real-time constraints (e.g., autonomous vehicle navigation via object matching [85]).

In the following, after describing the MHD and the surrogate metrics, we learn approximations of the complex metric. Next, we use selection algorithms to reduce the number of metrics employed. Finally, experimental results are shown.

7.3.1 Approximate metrics via learning

Let I be an image, S its ground-truth segmentation, and $\hat{S}$ the segmentation obtained by an algorithm, where both S and $\hat{S}$ are binary maps of the same size of I, with pixel values set to 1 to denote foreground (e.g., plant), and pixel values set to 0 to denote background (e.g., tray, pot, earth, moss). Our goal is to demonstrate that certain complex segmentation evaluation metrics (such as the MHD) can be approximated using linear combinations of simple to compute metrics.

While the framework is generic and can accommodate any complex metric, in the following we use the MHD as a motivating example of a complex metric. This metric has proved useful when developing plant image segmentation algorithms. We observe that MHD relates well with the human intuition of the problem at hand, and is effective at penalizing segmentations containing errors that would jeopardize the accurate extraction of visual phenotypes (e.g., missing leaves, cut stems, holes inside leaves). Notably, MHD can be expressed in units of length (e.g., millimeter), easing the interpretation.

However, MHD is computationally expensive, and not feasible on sensing devices with limited storage and computational capacity. Hence,

we propose to learn a (linear) model of the MHD from the data, based on (computationally) easier metrics as features. In the following sections we begin with defining the MHD, the easier metrics, and proceed by presenting the linear model and feature selection algorithms.

Modified Hausdorff distance

The Hausdorff distance measures the distance between two point sets as the greatest of all the distances from a point in one set to the closest point in the other set. It is defined as:

$$f_{\mathrm{H}}(S, \hat{S}) = \max \left\{ \sup_{a \in S} \inf_{b \in \hat{S}} d(a, b), \sup_{b \in \hat{S}} \inf_{a \in S} d(a, b) \right\}, \tag{7.4}$$

where S and $\hat{S}$ are two non-empty subsets of a metric space and $\sup$ and $\inf$ denote supremum and infimum, respectively. Dubuisson et al. [85] proposed a Modified Hausdorff Distance and demonstrated its efficacy for real-world applications. Let $S = \{a_1, \ldots, a_N\}$ and $\hat{S} = \{b_1, \ldots, b_{\hat{N}}\}$ be sets of points. A directed distance between S and $\hat{S}$ can be defined as:

$$d(S, \hat{S}) = \frac{1}{N} \sum_{a \in S} d(a, \hat{S}), \tag{7.5}$$

where $d(a, \hat{S}) = \min_{b \in \hat{S}} \|a - b\|_2$ is the distance between a point $a \in S$ and the set of points $\hat{S}$, and $\|a - b\|_2$ denotes the Euclidean distance between a and b. The two directed distances $d(S, \hat{S})$ and $d(\hat{S}, S)$ are then combined to define an undirected distance measure:

$$f_{\mathrm{MHD}}(d(S, \hat{S}), d(\hat{S}, S)) = \max \Big\{ d(S, \hat{S}), d(\hat{S}, S) \Big\}. \tag{7.6}$$

The MHD is characterized by a large discriminatory power, and robustness to noise. However, the computational complexity of its exact calculation can be challenging to a low-computational power device operating on high-resolution images, when the segmentation masks (the point sets) contain millions of pixels.

Low-complexity segmentation accuracy metrics

Calculating the MHD involves a geometric search problem in a vast space, in order to reach a solution and output the distance between two segmentations. To lower the computational requirements, we considered

the following set of metrics to model the desirable behavior of the MHD, and quantify segmentation accuracy in a way that is affordable by a low-computational power device.

1. *Statistical Error* (StatErr), is the sum of false positive and false negative errors, i.e. the fraction of misclassified pixels locations, and is equivalent to the Hamming distance between S and $\hat{S}$;
2. *Precision*, is the fraction of pixels in the segmentation $\hat{S}$ that matches the ground truth S;
3. *Recall*, is the fraction of ground-truth pixels contained in the segmentation $\hat{S}$;
4. *Rand Index* [258], measures the similarity between two segmentations as the frequency with which S and $\hat{S}$ agree on the classification of pairs of pixels;
5. *Variation of Information* (VoI) [204], measures the distance between two segmentations as a linear expression involving entropy of S and $\hat{S}$, and their mutual information;
6. *Jaccard Similarity Coefficient* (JSC) and *Dice Similarity Coefficient* (DSC) are used to quantify the spatial overlap between S and $\hat{S}$;
7. *Object-level Consistency Error* (OCE) [248], is based on the Jaccard Similarity Coefficient (i.e. it measures the spatial overlap between binary objects), but is more sensitive to over- and under-segmentation;
8. *Global Consistency Error* (GCE) [199], measures subset relationship between S and $\hat{S}$, based on local overlap;
9. Assuming a connected component labeling of the objects in the error mask (i.e. false positive and negative pixels), their number (CC-count), the error with maximum size (CC-max), their average size (CC-mean), and the standard deviation in size (CC-std). These metrics are inspired by application specific metrics, e.g., [32].

Calculating the above metrics is significantly less complex than the MHD. They do not require any optimization procedures and most of them can be calculated in a single pass of the segmentation masks. Only the last class requires connected component labeling that is slightly more complex, although efficient implementations exist. Additionally, several groups of such metrics share common intermediate steps for their computation,

hence favoring optimized implementations. Our goal is to construct linear combinations of the presented surrogate metrics, to approximate the behavior of the MHD with a simple pixel process.

We adopt these metrics to include in the model additional information about the error, in a similar fashion to [32], where a number of terms are included in the definition of an accuracy metric for cell nuclei segmentation. Ad hoc metrics have also been designed for specific applications (e.g., cell nuclei segmentation [32]).

Modeling complex metrics

We use linear regression to learn from the data the relationship between the MHD and the set of surrogate metrics presented previously. We use a linear regression since it is computationally efficient; other approaches such as support vector regression [82] and random forest regression [33], while they have shown better performance in some applications, they can be significantly more demanding computationally and create models that are less interpretable. As we will show in the results section, linear regression adequately fits the problem at hand.

A linear regression model with p predictors can be formulated in matrix notation as:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}, \tag{7.7}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is called the response (i.e. the MHD values), $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is the design matrix (i.e. the collection of surrogate metrics), and $\boldsymbol{\beta} \in \mathbb{R}^p$ contains the regression coefficients, which are obtained from the pseudo-inverse solution $\boldsymbol{\beta} = \boldsymbol{X} \backslash \boldsymbol{y}$. In order to preserve the non-negativity of the response, we calculate the predicted value $y_{\text{new}} \in \mathbb{R}$ for a new sample $\boldsymbol{x}_{\text{new}} \in \mathbb{R}^p$, as $y_{\text{new}} = zH(z)$, where $z = \boldsymbol{x}_{\text{new}}^T \boldsymbol{\beta}$, and H is the Heaviside step function.

7.3.2 Selection algorithms

We also investigate the possibility to identify appropriate subsets of surrogate metrics without loss of performance, using feature selection strategies based on sparse approximation algorithms. The sparse solution not only eliminates features (in our context the surrogate metrics) that are unnecessary, but also creates a more interpretable model. We analyze two popular approaches originating from the sparse approximation field: convex optimization and greedy iterations. While sparse penalty terms have been

previously and successfully proposed for regression (e.g., Lasso [301]), in our approach we explicitly impose the maximum allowed representation error, with respect to the model obtained with the full support of predictors.

ℓ_1 optimization-based selection algorithm

The first algorithmic approach involves the utilization of the ℓ_1 norm optimization problem, that has been used successfully in sparse representation and model reduction applications. We propose a variation of the Iterative Reweighted ℓ_1 (IRL1) minimization [39], that is shown to produce some of the best results in terms of model reduction capabilities, in the family of convex optimization methods.

IRL1 algorithm. Given response $\boldsymbol{y} \in \mathbb{R}^n$, design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, maximum number of iterations M, and target representation error $\epsilon \in \mathbb{R}$, return the sparse solution $\boldsymbol{\beta} \in \mathbb{R}^p$ such that $\|\boldsymbol{W}\boldsymbol{\beta}\|_1$ is minimized under the error constraint.

For iterations: $k = 1, \ldots, M$

1. If $k = 1$ then $\boldsymbol{W} = \boldsymbol{I}_p$, otherwise compute the new weights in the diagonal matrix $\boldsymbol{W}$:

$$W_{ii} = 1/(|\beta_i| + c),\ 0 < c \ll 1,\ i = 1, \ldots, p. \tag{7.8}$$

2. Solve the new optimization problem:

$$\boldsymbol{\beta} = \underset{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2 \leq \epsilon}{\arg\min} ||\boldsymbol{W}\boldsymbol{\beta}||_1. \tag{7.9}$$

Notice that in the first iteration IRL1 solves the actual ℓ_1 optimization problem. The next iterations refine this result by adding weights to the problem, such that if a coefficient of the solution is large in absolute value, the weight is small. Conversely, if a coefficient is small in absolute value, the weight is large (we try to drive the coefficient exactly to zero). Due to this improvement, IRL1 consistently outperforms regular ℓ_1 optimization procedure [39]. Note that with respect to [39] here a different optimization problem is cast that incorporates the constraint on the allowed representation error.

Greedy iterations-based selection algorithm

The second optimization strategy uses optimized orthogonal greedy iterations to gradually eliminate design patterns that cause the smallest model error increase, while the representation error constraint is still satisfied.

Greedy algorithm. Given response $\boldsymbol{y} \in \mathbb{R}^n$, design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, and target representation error $\epsilon \in \mathbb{R}$, return the sparse solution $\boldsymbol{\beta} \in \mathbb{R}^p$ and its support set S optimized under the error constraint.

- Given initial support $S^1 = \{1, \ldots, p\}$,
- For iterations: $k = 1, \ldots, p$
    1. Iterations: $j = 1, \ldots, p - k + 1$
        (a) Eliminate from the support S^k the j^{th} column to get the active support $A = S^k - S^k_j$.
        (b) Solve $\boldsymbol{X}_A \boldsymbol{\beta}_A = \boldsymbol{y}$.
        (c) $e_j = \|\boldsymbol{y} - \boldsymbol{X}_A \boldsymbol{\beta}_A\|_2$.
    2. Find the minimum error increase $j_0 = \arg\min \boldsymbol{e}$.
    3. If $\epsilon > e_{j_0}$ stop iterations and return support $S = S^k$, otherwise continue with $S^{k+1} = S^k - S^k_{j_0}$.
- Get final solution $\boldsymbol{\beta} = \boldsymbol{X} \backslash \boldsymbol{y}$ on S.

The greedy method reduces the support of the solution by removing, at each iteration, the design pattern that offers the lowest increase in the representation error, while still under the target imposed representation error. The Greedy approach does not require any parameters, while IRL1 requires two parameters (M, c) to be selected a priori by the user.

Depending on the size of p, iteration (1) can lead to long running times, therefore we adopt some strategies to streamline this step. The projections $\boldsymbol{X}^T \boldsymbol{y}$ are computed only once. Notice that these iterations solve a sequence of least squares problems with a specific structure: each iteration adds to the previous set of columns a different candidate column, and solves. This type of problems has been extensively studied and an efficient algorithm is given in [11]. The improvement is substantial particularly for large values of p.

(a)	(b)	(c)	(d)	(e)
		$\mathrm{MHD} = 0.139$	$\mathrm{MHD} = 1.526$	$\mathrm{MHD} = 0.118$
		$\widehat{\mathrm{MHD}} = 0.119$	$\widehat{\mathrm{MHD}} = 1.514$	$\widehat{\mathrm{MHD}} = 0.073$

Figure 7.4: Example segmentations of the original image in (a). Shown row wise: (b) ground truth, (c) k-means clustering, (d) Rosette Tracker [71], (e) active contour model [209]. Below (c)-(e), the corresponding values of the Modified Hausdorff Distance [85], and its approximation ($\widehat{\mathrm{MHD}}$), with respect to the ground truth in (b).

7.3.3 Results and discussion

The aim of our evaluation is to demonstrate the appropriateness of a linear model for predicting the MHD, and assess the benefit of reducing the number of features employed, using selection algorithms. We first describe the plant segmentation dataset used and experimental settings, and conclude with results and their discussion.

Plant segmentation dataset

We conduct our experiments on a dataset of 222 segmentations of images (~ 0.14 megapixel) showing single Arabidopsis plant subjects (see Figure 7.4a-b). The dataset is composed of 62 algorithmic results and 160 synthetic segmentations. Each sample of the dataset is accompanied by the corresponding ground-truth segmentation obtained manually, which is used to compute the accuracy metrics described in Section 7.3.1.

The algorithmic results (see Figure 7.4 for an example) are obtained using four different image segmentation methods previously proposed to segment plants in images:

- k-means clustering on pixel intensity values (commonly used in plant image segmentation [124]);
- Rosette Tracker software [71];

- vector-valued active contour model operating on pixel color intensities (cf. Chapter 4); and
- vector-valued active contour model operating on color intensity and texture features (cf. Chapter 4).

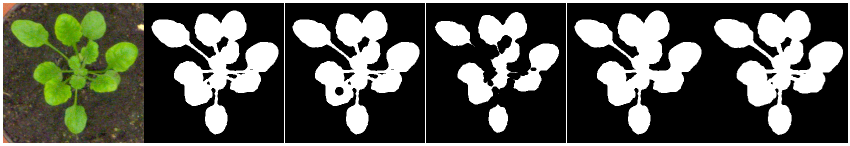
The synthetic data (see Figure 7.5) are obtained by introducing a variety of errors commonly encountered in this context (e.g., missing object parts) in the ground-truth segmentations. The synthetic data are generated in a computational manner, easing the process of obtaining training data to learn a model of a complex metric. While the algorithmic data contain mixed types of errors, the synthetic data represent specific classes (e.g., resulting from a biased segmentation), overall accounting for greater variability in both type and amount of errors.

Figures 7.4 and 7.5 show examples of plant segmentations, and the corresponding value of the MHD. The MHD penalizes more boundary displacements (e.g., cut leaf), and holes inside the object, while remaining robust to noise (e.g., scattered noise in the background, or noisy border lines). Also critical in this context, it penalizes under-segmentation that causes splits in the object (Figure 7.5). In addition, the MHD exhibits a linear behavior with the amount of error introduced, rendering it suitable for linear regression.

The metrics presented here are implemented in Matlab and their values are recorded. Prior to fitting any model, the metrics in the design matrix $\boldsymbol{X}$ are normalized to mean 0 and standard deviation 1, while the responses $\boldsymbol{y}$ are centered on the mean. Although this step is not necessary, it renders the regression coefficients comparable and eases discussion.

Experimental settings

To demonstrate the generalization capability of the linear model, a K-fold cross validation strategy where $K = 10$ is employed, with random splits of the original data into training/testing sets. As a goodness-of-fit measure, the coefficient of determination $R^2 = 1 - (SS_{\text{err}}/SS_{\text{tot}})$ is used, where SS_{err} is the sum of squared residuals, and SS_{tot} is the total sum of squares. At each round of the cross validation, the R^2 is calculated on both the training set, to evaluate the goodness of fit, and the testing set, to assess the capability of the model of generalizing to previously unknown data samples.



(a) original image	(b) ground truth	(c) hole inside leaf	(d) under-segmentation	(e) filled spaces	(f) over-segmentation
		$\text{MHD} = 0.030$	$\text{MHD} = 0.582$	$\text{MHD} = 0.065$	$\text{MHD} = 0.062$
		$\widehat{\text{MHD}} = 0.020$	$\widehat{\text{MHD}} = 0.537$	$\widehat{\text{MHD}} = 0.027$	$\widehat{\text{MHD}} = 0.065$

Figure 7.5: Examples of synthetic errors in the segmentation of the original image. Below each image, the corresponding values of exact MHD [85], and its approximation ($\widehat{\text{MHD}}$), with respect to the ground truth.

For the IRL1 algorithm $M = 5$, and $c = 10^{-7}$ in Eq. (7.8) are chosen empirically. For both Greedy and IRL1 the target error ϵ is chosen such that an increase in the representation error of up to a 4% of error of the full model is allowed. All experiments are executed on a machine equipped with Intel Core 2 Duo CPU E8200 2.10 GHz, 3 GB memory, and running 64-bit GNU/Linux.

Results

In our distributed sensing and analysis scenario, the $\boldsymbol{\beta}$ coefficients are estimated from training data on the cloud. Subsequently, they are transmitted to the sensor, where they are used to efficiently predict the response $\hat{\boldsymbol{y}}$ of the complex metric, for new, previously unseen segmentations, using the values of the surrogate metrics as input variables.

Table 7.1 shows an example of standardized regression coefficients of a linear model using all features learned on the entire dataset, to study the contribution of the surrogate metrics. The $R^2 = 0.9$ supports the validity of a linear model in representing the MHD, as 90% of the response variation in the data can be explained using a linear relationship. Precision, Recall, Rand, and JSC exhibit high coefficient magnitude, providing the largest contribution to the linear approximation of the MHD. Conversely, VoI, DSC, and OCE have small coefficients, as they convey similar information

Table 7.1: Example regression coefficients estimated on the entire dataset and corresponding R^2 statistic.

Model	β coefficients													R^2
	StatErr	Precision	Recall	Rand	VoI	JSC	DSC	OCE	GCE	CC-count	CC-max	CC-mean	CC-std	
No selection	-1.18	-2.15	-3.30	-2.63	-0.12	3.38	0.04	-0.05	-1.22	0.02	0.03	-0.02	0.01	0.902
IRL1	0	-1.85	-2.81	-0.78	0	2.81	0	-0.05	-0.77	0.01	0	0.01	0	0.895
Greedy	-0.90	-2.14	-3.27	-2.17	0	3.36	0	-0.05	-1.18	0	0	0	0.01	0.895

Table 7.2: Average selection rates for inclusion of the metrics in the model, obtained from K-fold cross validation.

Model	Selection Rate (%)												
	StatErr	Precision	Recall	Rand	VoI	JSC	DSC	OCE	GCE	CC-count	CC-max	CC-mean	CC-std
IRL1	20	100	100	80	20	100	20	100	70	80	20	40	20
Greedy	60	100	100	90	0	100	10	100	90	40	10	10	20

to JSC. In addition, the last group of metrics based on the connected component labeling (which can be less computationally efficient) has small coefficient magnitudes. Figures 7.4 and 7.5 report predicted values of MHD for example images (excluded from the training set), using the full linear model.

Similar conclusions can be obtained by optimizing which metrics are chosen using selection algorithms (Table 7.1). Allowing a 4% representation error, both IRL1 and Greedy selection strategies halves the number of predictors (by setting to 0 the coefficients of the discarded ones) with minimal reduction in R^2. These predictors (except CC-count, which is occasionally included) are underused also when considering the K-fold cross validation (Table 7.2 shows the percentage of times a feature is selected). Analogously, VoI and DSC present small coefficients in the full model, and we expect them to rarely be included by the selection algorithms.

The outcome of the K-fold cross validation with respect to R^2 is shown in Table 7.3. The full linear model obtains a cross-validated $R^2 \approx 0.9$ on the training set, showing a good fit of the regression. The capability of the linear model of generalizing to new unseen data is supported by an $R^2 = 0.835$ on the testing set. This observation holds also for the sparse

Table 7.3: Cross-validated results of the approximate linear models, represented as mean (standard deviation).

Model	R^2		Support
	Training	Testing	
No selection	0.904 (0.012)	0.835 (0.128)	15.0 (0.0)
IRL1	0.898 (0.014)	0.814 (0.133)	7.7 (1.3)
Greedy	0.897 (0.013)	0.839 (0.123)	7.3 (1.2)

approaches. The sparser solutions with a support of only (on average) 7 features obtain comparable R^2 values in the testing and training set with respect to the full linear model. IRL1 incurs slightly lower prediction accuracy than the full model, while Greedy exhibits the best performance among the three. This result demonstrates that the shrinkage operation actually improves the ability of the model to generalize, and that the Greedy strategy selects better features than IRL1.

The proposed approximations reduce execution time considerably, by several orders of magnitude. On average, the exact computation of the MHD requires 35 minutes, whereas all surrogate metrics are computed collectively in ~ 0.05 seconds.

Finally, Figure 7.6 demonstrates the relationship between support size and R^2 and the behavior of the selection algorithms, as obtained with cross validation and by varying the error ϵ. (Overall, the larger the ϵ allowed the more sparse the solution and the less features are used.) As previously, the Greedy strategy outperforms the ℓ_1 optimization in the choice of the predictors to include in the model, providing comparable or better prediction accuracy across the range of support (or equivalently the allowed error ϵ). The Greedy algorithm exhibits the global optimum with 7 predictors; i.e. discarding a subset of predictors improves the capacity of the model to generalize to new unseen data samples.

Overall, the linear approximation of the MHD proposed here are ideally suited for remote sensing applications or for a distributed sensing and analysis scenario, since a linear combination of 7 simple to compute metrics can adequately approximate the MHD ($R^2 \approx 0.9$ in most cases). The optimization of the linear coefficients and which metrics to be used can be done at the centralized unit and communicated via feedback to the sensor. Thus, the sensor can adapt to the scene at hand and always use an

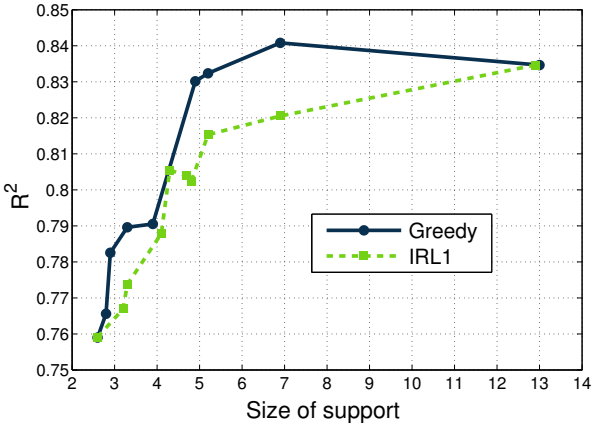


Figure 7.6: Cross-validated prediction accuracy of the linear models, when varying the degree of sparsity enforced by the selection strategies.

efficient approximation of the MHD.

7.4 Summary

In order to keep performance of the analysis as close as possible to a scenario where no compression occurs, we proposed to compress images acquired at the sensor using the JPEG 2000 standard and ROI coding, adopting a Gaussian mixture model to estimate ROIs accurately. In order to relieve the sensor from learning the GMM, we proposed to shift such computational burden to the cloud, while the few parameters describing the model are sent as a feedback to the sensor. Experimental results confirmed the efficacy of the proposed approach, both in the fidelity of the reconstructed images and in the accuracy achieved by the application (i.e. the plant segmentation).

While here we used JPEG 2000, our approach is generic and different coding standards could be employed, such as the recent High Efficiency Video Coding (HEVC) standard [295] for which an ROI coding methodology has been proposed by Meddeb et al. [202, 203].

In a distributed scenario, the sensor must decide the compression parameters with regards to segmentation accuracy, where the uncompressed

image is considered the ground truth (the sensor has the capability to obtain a roughly good segmentation using a low-complexity algorithm and feedback from a centralized unit). Identifying a segmentation metric that could be adopted efficiently at the sensor is critical for optimal performance.

Therefore, we modeled the behavior of a complex image segmentation accuracy metric (e.g., the Modified Hausdorff Distance), using easier to compute metrics. We considered several of such metrics and used sparse representation approaches for feature selection to build linear regression models that can approximate the complex one. We presented for the first time the concept of using a collection of low-complexity metrics to approximate the behavior of a complex one. Our experimental results showed that it is possible to closely estimate the MHD ($R^2 \approx 0.9$ in most cases) without necessarily calculating its exact value (a process involving a costly geometric search), using a linear combination of 7 simple to compute metrics. While we used the MHD, the data-driven framework and concept presented here can be adapted to other complex metrics as well, or even not in segmentation scenarios (e.g., in object matching) with appropriately identified features.

In the next chapter, we will continue with a different approach to application-aware image compression on resource-constrained devices, addressing this problem from the color space representation point of view.

8

Saving bits in color representation

8.1 Introduction

In plant phenotyping applications, optical imaging in the visible part of the spectrum encompasses a wide range of applications to assess plant growth, or other organ-level visual traits. In this context, color information is routinely used for robust delineation of plant objects from background (cf. Chapter 4), but it also conveys valuable information on plant function and reaction to disease or stress conditions (e.g., drought tolerance [26, 164, 272]).

Currently, the standard for acquisition and display of color digital images is the RGB (red, green, blue) color model. However, this representation is not efficient for coding, due to high correlation between color bands of natural images [252]. To reduce spectral redundancy, image

This chapter is partly based on:

- M. Minervini, C. Rusu, S. A. Tsaftaris, "Computationally efficient data and application driven color transforms for the compression and enhancement of images and video," in *Color Image and Video Enhancement*. Springer, 2015, ch. 12.
- M. Minervini, C. Rusu, S. A. Tsaftaris, "Unsupervised and supervised approaches to color space transformation for image coding," in *21st International Conference on Image Processing*, Oct. 2014, pp. 5576–5580.

and video compression algorithms operate on luminance/chrominance representations of the color information, achieved through linear transformations of the RGB color space [237, 251]. Each color band is coded independently, therein deploying a variety of techniques to address spatial and, for video, also temporal correlation. A family of such color models is the YC_bC_r, defined by the International Telecommunication Union [145] and adopted by many coding standards [281]. However, due to high variability in source image characteristics, a fixed transform may easily result in suboptimal performance, thus motivating the adoption in some contexts of a data-dependent one.

Among the linear transformations, the energy compaction and decorrelation properties of the Karhunen-Loève transform (KLT) [154] make it desirable for color image compression [106, 166, 252, 271, 334]. The KLT has been adopted in many coding schemes (e.g., for dimensionality reduction [1, 59]), and it was shown to be superior to other approaches in a variety of contexts, both for color [54, 120, 159] and hyperspectral [83] imagery, and has formed the basis for new fixed transforms [121, 163, 192]. However, the computational complexity of calculating the color covariance matrix limits its applicability in real-time video enhancement applications (e.g., denoising [343], contrast and color [107, 108] enhancement, color to gray scale conversion [79]), and sensing environments with low computational power (e.g., surveillance cameras or visual sensor networks operating under low lighting conditions).

A variety of approaches have been proposed to circumvent this bottleneck. Kountchev et al. [167] rely on covariance matrix approximations, while [79] uses numerical methods to estimate eigenvalues and eigenvectors of the covariance matrix. Subsampling strategies to limit complexity by reducing the amount of input data are discussed in [84, 242]. Du et al. [84] adopt a learning approach based on neural networks to estimate projection directions. Porikli et al. [249] propose an algorithm based on integral images for fast computation of the covariance matrix. Others focus on schemes that favor parallel implementations [9, 341] of the KLT, or implementations optimized for graphics processing units [9, 156, 191]. In this chapter, we propose a new data-dependent color transform, the aKLT, rooted in the orthogonal Procrustes problem, that preserves energy compaction and performs similar to the KLT, but is less computationally complex.

Although KLT and aKLT are designed to match the statistical properties of the image data, they are agnostic to the semantics of the scene (e.g.,

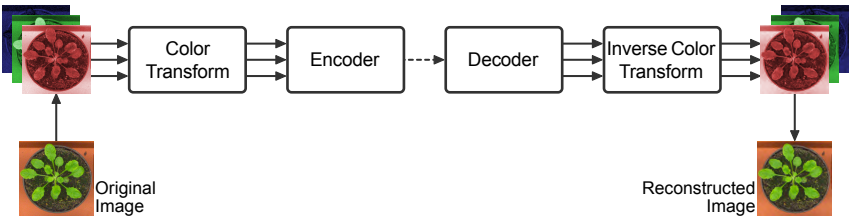


Figure 8.1: Schematic of the typical encoding/decoding process of a color image.

distinction between foreground and background regions). Thus, we propose for the first time to obtain a color transform using supervision (e.g., previously labeled image data), aiming at the preservation of the image features relevant to the application. We formulate our methodology as a supervised learning problem, envisioning two alternative approaches to find a solution, relying either on the Foley-Sammon transform (FST) [101] or on metric learning methods [168]. From an application-aware image compression perspective, it is desirable to achieve: (a) classification accuracy, pursued by enhancing separability of the transformed data, and (b) compression performance, achieved with decorrelation and energy compaction. These two requirements appear conflicting and designing a color transform that optimally accommodates both remains an open challenge. Therefore, we propose to adopt an optimization approach to obtain application-dependent color transforms that while aiming to retain energy compaction properties, also try to maximize separability of the transformed data.

The rest of the chapter is organized as follows. Section 8.2 details our methodology to learn color transforms from the data. Section 8.3 demonstrates the proposed approaches on a variety of image datasets, using the JPEG 2000 standard to compress test images.

8.2 Learning the color transform from the data

We represent an RGB image as a $3 \times n$ matrix $\mathrm{X} = \left(\boldsymbol{r}, \boldsymbol{g}, \boldsymbol{b}\right)^{\mathsf{T}}$, where $\boldsymbol{r}$, $\boldsymbol{g}$, and $\boldsymbol{b}$ are the linearized color components, and n is the number of pixels. Prior to lossy coding, X is projected into a new color space by $\mathrm{T} \in \mathbb{R}^{3\times3}$. Each pixel value $\boldsymbol{x}_i = \left(r_i, g_i, b_i\right)^{\mathsf{T}}$ in X is transformed by the linear relation

$\boldsymbol{y}_i = \mathrm{T}\boldsymbol{x}_i$. Upon reconstruction, the color transform is inverted, obtaining the approximation $\widetilde{\boldsymbol{x}}_i = \mathrm{T}^{-1}\widetilde{\boldsymbol{y}}_i$ in the RGB domain. To ensure that the output range of $\boldsymbol{y}_i$ is the same as $\boldsymbol{x}_i$ (e.g., 0 to 255, for 8-bit unsigned integer representation), we scale the directions (rows) of T with respect to the ℓ_1 norm [163].

We operate in the general framework depicted in Figure 8.1, in which color transformation is decoupled from encoding/decoding operations. Therefore, any compression scheme can be adopted (in the experiments we use the JPEG 2000 standard [287]) and the resulting bit stream will be standard-compliant.

In the following, we address the problem of obtaining data-driven color space transformations,[1] that change according to the image to be encoded or the application the images will be used for. In Section 8.2.1, based on a heuristic, we derive a new low-complexity transform (aKLT) that adapts to the content using statistical information from the image being processed. In Section 8.2.2, we propose a novel approach to obtain transforms that adapt according to the application (here a pixel-level classifier for foreground-background segmentation), relying on supervised learning methods and labeled training data. Finally, in Section 8.2.3, we combine the unsupervised and the supervised transforms using an optimization approach.

8.2.1 The aKLT: A low-complexity unsupervised data dependent transform

The KLT produces an orthogonal transformation, K, obtained from the eigendecomposition of the color covariance matrix:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^{\mathsf{T}}, \tag{8.1}$$

where $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$ is the mean color vector [252]. The eigenvectors of Σ, sorted in decreasing order of magnitude of the corresponding eigenvalues, define the directions of K. The KLT achieves complete statistical decorrelation of the color signals and energy compaction in the first channel, thus favoring efficient representation and subsampling of the other two channels [184, 271].

[1] Approaches to efficiently compute the color space conversion for a known transform are discussed, e.g., in [41, 337].

However, estimation of Σ can be computationally demanding in memory and computing power, particularly when images are large, and its application in resource-constrained sensing devices can be problematic. Thus, we seek to find a transform that is close to the KLT but less computationally complex to obtain.

Let $\overline{\mathrm{X}} \in \mathbb{R}^{3\times n}$ be the matrix obtained by normalizing each column (pixel) of X with respect to the ℓ_2 norm. We seek an orthogonal transform $\Omega \in \mathbb{R}^{3\times 3}$ that maps $\overline{\mathrm{X}}$ to a given reference matrix $\mathrm{W} \in \mathbb{R}^{3\times n}$, and formulate it as:

$$\begin{aligned} \underset{\Omega}{\text{minimize}} \quad & \|\mathrm{W} - \Omega\overline{\mathrm{X}}\|_F \\ \text{subject to} \quad & \Omega^{\mathsf{T}}\Omega = \mathrm{I}\,, \end{aligned} \tag{8.2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and I is the identity matrix. Let $\mathrm{Z} = \mathrm{W}\overline{\mathrm{X}}^{\mathsf{T}}$, and $\mathrm{Z} = \mathrm{USV}^{\mathsf{T}}$ be the singular value decomposition (SVD) of Z. The optimization problem of Eq. (8.2), known as the orthogonal Procrustes problem, admits closed-form solution UV^{T} [279]. In order to concentrate energy in the first direction, we impose structure to W:

$$\mathrm{W} = \begin{pmatrix} 1 & \dots & 1 \\ 0 & \dots & 0 \\ 0 & \dots & 0 \end{pmatrix}. \tag{8.3}$$

Notably, this leads to a simplified form of Z:

$$\mathrm{Z} = \begin{pmatrix} \sum_{i=1}^{n} r_i & \sum_{i=1}^{n} g_i & \sum_{i=1}^{n} b_i \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{8.4}$$

with only a single direction, $\boldsymbol{a}_1 = \boldsymbol{z}_1^{\mathsf{T}}/\|\boldsymbol{z}_1\|_2$, that corresponds to the principal direction, thus making the SVD computation unnecessary. We adopt the vector $\boldsymbol{a}_1$ as an approximation of the principal direction of the KLT.

In order to obtain the full transform, we proceed by constructing the 3×3 matrix $\mathrm{A} = \big(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\big)$, where $\boldsymbol{a}_2$ and $\boldsymbol{a}_3$ are initialized with random elements, e.g., uniformly distributed on the interval $[0, 1]$ (the effect of randomness on performance is explored in Section 8.3). Subsequently, we use QR factorization to decompose A into the product $\mathrm{A} = \mathrm{QR}$, where $\mathrm{Q} \in \mathbb{R}^{3\times 3}$ has orthogonal columns and $\mathrm{R} \in \mathbb{R}^{3\times 3}$ is upper triangular. The aKLT transformation matrix, $\widetilde{\mathrm{K}} = \mathrm{Q}^{\mathsf{T}}$, shares relevant properties with the regular KLT: (a) orthogonality, and (b) energy compaction capabilities.

Table 8.1: Comparison of KLT approaches as a function of input size n, where n denotes the number of image pixels.

KLT, ACKLT [167], Power method [79],	Penna et al. [242]	IPCA[A] [84]	Porikli et al. [249]	aKLT
$15n$	$\rho 15n$	$15n$	$15n$	$12n$

[A] Complexity reported by the authors of [84] only takes into account multiplications.

Although there is no guarantee on sorting and relative amount of energy of second and third channel, this is not of concern from a compression standpoint (e.g., chroma subsampling strategies would downsample the lower-energy components using the same scheme).

Complexity comparison between KLT approaches. The computation of the KLT requires $15n$ floating point operations in total, where n is the number of pixels, and is dominated by computation of mean color vector $\boldsymbol{\mu}$ ($3n$ operations) and covariance matrix Σ (due to symmetry, $12n$ operations are necessary to compute its 6 distinct entries). Notice that while most authors center the original data on the mean prior to calculating Σ (a step that would require additional $3n$ operations, since it is performed on all image pixels), the covariance matrix can also be defined as $\Sigma = \frac{1}{n}\left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}\right) - \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}}$ [7]. Thus, mean subtraction can be performed on the small 3×3 autocorrelation matrix, with fixed computational cost [100]. We also ignore cost of subsequent eigenvalue decomposition of Σ to obtain K, since this step does not depend on n.

Approaches that speed up the eigendecomposition of Σ (e.g., power method [79], or ACKLT [167]) provide negligible benefit in this context (particularly as image resolution increases). As shown in Table 8.1, the IPCA [84], based on neural networks, achieves an approximation of the principal direction using $15n$ multiplications (additions are not reported by the authors), while [242] necessitates to keep a fraction $\rho = 0.8$ of the data to match the aKLT. Porikli et al. [249] propose a method based on integral images for fast construction of feature covariance matrices of all possible rectangular regions in an image. However, when used in the full image, this approach does not provide any benefit, computing the color covariance matrix using also $15n$ operations.

On the other hand, our proposed aKLT estimates the statistical prop-

erties of the source image and computes the color transform with lower computational complexity. It requires $9n$ operations to normalize the input data, i.e. $3n$ to square the pixel values, $2n$ to add the components of each pixel, n to compute the square root for each pixel, and $3n$ to divide each channel by the so-obtained ℓ_2 norm (notice that mean subtraction is not necessary for the aKLT). Furthermore, $3n$ operations are required to calculate $\boldsymbol{z}_1$, and a small fixed cost (27 operations) for the QR decomposition of A, resulting in $12n$ operations in total to obtain $\widetilde{\mathrm{K}}$, i.e. 20% reduction in complexity compared to the regular KLT. Computational complexity of the aKLT can be further reduced if combined with the sub-sampling strategies proposed in [242].

8.2.2 A supervised approach to an application-dependent color transform

It is known that projecting to principal components is not always optimal from a pattern recognition perspective: clusters of points belonging to semantically different objects in the scene may overlap now in the projected color space [300] (cf. Figure 8.2). Introducing distortions due to lossy compression may affect this separability further. With an application-aware compression setting in mind, we seek to identify a transform that maintains (a) class separation as well as (b) decorrelation and energy compaction properties.

We assume that the computation of the supervised color transform will occur in an offline fashion and we will use a training set (pixels partitioned in two classes), thus it is supervised. Compression of newly acquired images at the sensor occurs as before, with the transform now known. The calculation of a new transform is necessary only if the scene conditions change (depending on the process being observed) and if new training data are available.

Let C_1 and C_2 be disjoint sets of pixel values (i.e. $C_1 \cap C_2 = \emptyset$) representative of distinct pattern classes (e.g., foreground and background). We seek an orthogonal transform $\mathrm{D} \in \mathbb{R}^{3\times 3}$ that projects data points belonging to distinct classes, $\boldsymbol{x}_1 \in C_1$ and $\boldsymbol{x}_2 \in C_2$, in a domain where they are maximally separated according to a measure of separability $\mathcal{C}$:

$$\underset{\mathrm{D}}{\text{maximize}} \quad \mathcal{C}\{\mathrm{D}\boldsymbol{x}_1, \mathrm{D}\boldsymbol{x}_2\} \tag{8.5a}$$

$$\text{subject to} \quad \mathrm{D}^{\mathsf{T}}\mathrm{D} = \mathrm{I}\,, \tag{8.5b}$$

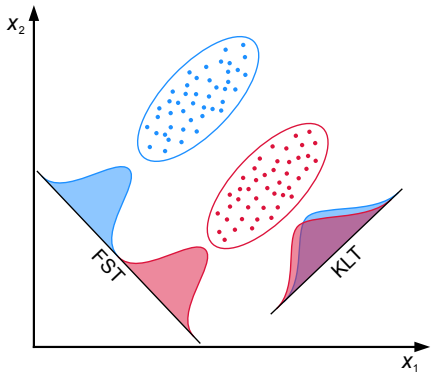


Figure 8.2: The KLT seeks directions of maximum variance in the projected data. On the other hand, the FST as other discriminative methods seeks directions that maximize class separability in a lower-dimensional subspace [300].

$$\|\mathrm{D}\Sigma\mathrm{D}^{\mathsf{T}} - \Lambda\|_F \leq \varepsilon\,, \tag{8.5c}$$

where I is the identity matrix, Σ is the color covariance matrix, Λ is a diagonal matrix whose elements are the eigenvalues of Σ, and $\varepsilon \geq 0$. The objective function of Eq. (8.5a) accounts for the class separation property of D, while the constraints of Eq. (8.5b) and (8.5c) allow for decorrelation and energy compaction. The parameter ε determines the trade-off between class separation on the one hand and decorrelation and energy compaction on the other hand. Therefore, solving the complete problem of Eq. (8.5) would lead to an orthogonal transform with the full set of the desired properties. On the other hand, imposing the orthogonality constraint (non-convex in nature) renders the optimization problem of Eq. (8.5) non-convex, possibly admitting multiple local optima. Non-convex problems are generally difficult to solve (i.e. finding the global optimum), and require to resort to global optimization methods, which are however computationally intensive. In the remainder of this section, we relax the problem by ignoring Eq. (8.5c), but we revisit the complete problem in the next section.

In the following paragraphs, we discuss two supervised learning approaches to obtaining a color transform with class separation capabilities: (a) the Foley-Sammon transform, based on the linear discriminant analysis, and (b) metric learning approaches.

Foley-Sammon transform. An effective measure of class separability is the Fisher's criterion [98, 300], defined as the difference between the means of the class data points, normalized by a measure of the within-class dispersion. This notion is formally expressed in the form of the generalized Rayleigh quotient:

$$J(\boldsymbol{d}) = \frac{\boldsymbol{d}^{\mathsf{T}} \mathrm{S}_b \boldsymbol{d}}{\boldsymbol{d}^{\mathsf{T}} \mathrm{S}_w \boldsymbol{d}}\,, \tag{8.6}$$

where $\boldsymbol{d} \in \mathbb{R}^3$, and $\mathrm{S}_b, \mathrm{S}_w \in \mathbb{R}^{3\times 3}$ are, respectively, between-class scatter matrix and within-class scatter matrix:

$$\mathrm{S}_b = \sum_{i=1}^{2} (\boldsymbol{m}_i - \boldsymbol{\mu})(\boldsymbol{m}_i - \boldsymbol{\mu})^{\mathsf{T}}\,, \tag{8.7}$$

$$\mathrm{S}_w = \sum_{i=1}^{2} \sum_{\boldsymbol{x}\in C_i} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^{\mathsf{T}}\,, \tag{8.8}$$

where $\boldsymbol{\mu} = \sum_{i=1}^{2} P_i \boldsymbol{m}_i$ is the mean sample vector, and $\boldsymbol{m}_i \in \mathbb{R}^3$ and $P_i \in \mathbb{R}$ are, respectively, mean and a priori probability of class i.

A closed-form solution to finding orthogonal discriminant vectors that maximize the Fisher's criterion, can be obtained adopting the Foley-Sammon transform (FST) [101]. The first direction, $\boldsymbol{d}_1$, termed Fisher's discriminant vector (or Fisher's linear discriminant [98]), corresponds to the projection direction that yields maximum between-class scatter and minimum within-class scatter, i.e. $\boldsymbol{d}_1 = \arg\max_{\boldsymbol{d}} J(\boldsymbol{d})$, and is obtained as the eigenvector associated with the largest eigenvalue α_1 satisfying $\mathrm{S}_b \boldsymbol{d}_1 = \alpha_1 \mathrm{S}_w \boldsymbol{d}_1$ [300]. The following directions to complete the three-dimensional transform are found recursively, by maximizing the Fisher's criterion with an orthogonality constraint. If $\mathrm{D} = \left(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_r\right)^{\mathsf{T}}$ is the set of previously obtained directions, $\boldsymbol{d}_{r+1}$ corresponds to the eigenvector associated with the largest eigenvalue α_{r+1} satisfying $\mathrm{M}\mathrm{S}_b \boldsymbol{d}_{r+1} = \alpha_{r+1} \mathrm{S}_w \boldsymbol{d}_{r+1}$, where $\mathrm{M} = \mathrm{I} - \mathrm{D}^{\mathsf{T}}(\mathrm{D}\mathrm{S}_w^{-1}\mathrm{D}^{\mathsf{T}})^{-1}\mathrm{D}\mathrm{S}_w^{-1}$ [153], and I is the identity matrix. For three-dimensional RGB data, the final color transformation matrix is defined by $\mathrm{D} = \left(\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3\right)^{\mathsf{T}}$.

In this chapter we consider only a two-class classification problem (i.e. foreground vs. background), however, the FST formulation can be easily generalized to an arbitrary number of pattern classes [103]. According to the class distribution of our test image datasets (cf. Section 8.3), we use the

standard FST formulation to find a linear separation between foreground and background. Kernel formulations of the FST [347] could be adopted to accommodate non-linearly separable classes.

Metric learning approaches. Metric learning methods (see [168] for a comprehensive survey) seek to estimate from supervised information a Mahalanobis distance function over data points:

$$\mathcal{D}_{\mathrm{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathsf{T}} \mathrm{A} (\boldsymbol{x}_i - \boldsymbol{x}_j)\,, \tag{8.9}$$

parametrized by a positive semidefinite matrix $\mathrm{A} = \mathrm{L}^{\mathsf{T}}\mathrm{L}$. Computing the distance in the input space is equivalent to applying a linear transformation L of the input space, such that data points with small distance according to $\mathcal{D}_{\mathrm{A}}$ are close (in a Euclidean sense) in the projected space, i.e. the matrix L minimizes $\mathcal{D}_{\mathrm{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\mathrm{L}\boldsymbol{x}_i - \mathrm{L}\boldsymbol{x}_j\|_2^2$, which is another view of the objective function in Eq. (8.5a). While the FST matrix D is orthogonal, the L transformation matrix obtained by a metric learning approach is positive semidefinite, and in general not orthogonal. Observe that orthogonality is beneficial for the numerical stability of the color transformation, i.e. errors introduced by compression and decompression operations are not magnified when forward and reverse color space conversions are computed. Therefore, with respect to the framework defined by Eq. (8.5), metric learning approaches optimize solely for class separability.

Recent metric learning methods include Relevant Component Analysis (RCA) [22], Large Margin Nearest Neighbors (LMNN) [329], and Information Theoretic Metric Learning (ITML) [69], which can all be used to find L.

8.2.3 Combining unsupervised and supervised approaches

Our approach for finding the supervised transform D in Section 8.2.2 relaxed the constraint of optimal decorrelation and energy compaction of Eq. (8.5), finding one that only optimizes for separation. In the previous section we also obtained orthogonality when using the FST, however, this does not guarantee energy compaction, which is achieved by the KLT (or can be approximated by the aKLT). Therefore, we consider now a different approach, removing the orthogonality constraint to obtain a convex relaxation of the problem of Eq. (8.5).

We seek to find a new transform $\mathrm{D}' \in \mathbb{R}^{3\times 3}$ that is close to D whilst trying to satisfy Eq. (8.5c), or equivalently, since we know that the KLT (or the aKLT) optimizes Eq. (8.5c), we can pose the following unconstrained optimization problem:

$$\underset{\mathrm{D}'}{\text{minimize}} \ \|\mathrm{D}' - \mathrm{D}\|_F + \lambda \|\mathrm{D}' - \widetilde{\mathrm{K}}\|_F \,, \tag{8.10}$$

thus, finding a transform that is between D (application-aware, obtained offline using labeled data) and the aKLT (obtained at the sensor and computed based on the unseen image), where the trade-off is controlled by the value of the regularization parameter λ (playing here a role similar to ε in Eq. (8.5)). In the same fashion, the L transform obtained with metric learning methods could be used in Eq. (8.10) instead of D. Although D and $\widetilde{\mathrm{K}}$ in Eq. (8.10) are orthogonal, in general D' will not be orthogonal. Approaches for finding the nearest orthonormal matrix to D' can be adopted, e.g., relying on the polar decomposition [130], or the square root matrix [133] of D'.

While this approach adapts the supervised transform to unseen data on the sensor and is expected to gain decorrelating and compacting capabilities, from a computational perspective may be less attractive. In this setting, with the FST (or the RCA) known, the encoder is required to compute the (a)KLT and then solve Eq. (8.10) to obtain the final color transform.

The approaches presented in Sections 8.2.1 and 8.2.2 admit closed-form solutions, whereas D' is found relying on iterative optimization procedures computing the solution path along the λ parameter. On the other hand, Eq. (8.10) involves only color transforms (i.e. small 3×3 matrices), rather than the original image pixels.

8.3 Results and discussion

8.3.1 Experimental settings

The proposed methodology is evaluated on color image data from a variety of classes. We demonstrate the unsupervised approach on standard test images, including natural, aerial, and retinal [293] images (Figure 8.4). We showcase the supervised transform using images of different size (up to 18 megapixel) downloaded from the Internet,[2] including horses,

[2] `http://www.flickr.com/`

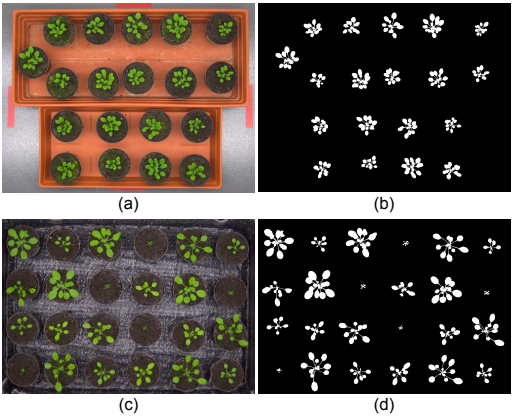


Figure 8.3: (Left) Example images of Arabidopsis plants from different experiments [274], and (right) corresponding ground truth segmentations delineated manually.

balloons, and fish (Figure 8.6). The approaches are also evaluated on a dataset of 20 images (width $\times$ height: 3108$\times$2324 pixels) from a time-lapse sequence of Arabidopsis plant subjects (Figure 8.3a), arising from plant phenotyping experiments [274]. We use images from this application since they are usually large and due to design requirements they may need to be communicated via the Internet to centralized locations for processing [216]. Thus, any bit rate savings possible are desirable.

We include in the comparison plain RGB (i.e. no color transform) and YC_bC_r (ITU-R BT.601) [145]. KLT and aKLT are computed for each image. We also adopt the Relevant Component Analysis (RCA) [22], a metric learning approach to find a supervised transform L that aims to preserve variability in the data relevant to the classification task at hand. For brevity and clarity of presentation we do not include other popular metric learning approaches, such as LMNN [329] and ITML [69], because they perform similar to the RCA in our image compression context, while being more computationally demanding (they rely on iterative optimization procedures). The supervised transforms (FST, RCA) are estimated on manually labeled training image data (excluded from testing). On the plant dataset, the supervised transforms (D and L) are estimated from the first image of the time-lapse sequence using pixel label information obtained manually. The so-obtained D and L are then applied to all subsequent images of the

same sequence and also to a test image of Arabidopsis plants with different scene conditions (Figure 8.3c). While the other transforms included in the comparison are either fixed (RGB, YC_bC_r) or present closed-form solutions (KLT, aKLT, FST, RCA), to solve the optimization problem of Eq. (8.10) we use CVX,[3] a package for specifying and solving convex programs [117].

After color space transformation, the images are compressed at various bit rates (between 0.0625 and 2 bpp) using the JJ2000 software implementation,[4] version 5.1, of the JPEG 2000 coding standard [287]. We implement the proposed methods using Matlab R2011b, and conduct all experiments on a machine with Intel Core 2 Duo CPU E8200 2.66 GHz and 4 GB memory.

The approaches are evaluated according to: (a) reconstruction accuracy, and (b) application error. Reconstruction accuracy is measured using Peak Signal-to-Noise Ratio (PSNR) in RGB image domain, either in the full image or in regions of interest (e.g., foreground regions as in Figure 8.3). To estimate application error, we adopt the task of plant segmentation for plant phenotyping applications [209, 216], therefore, we first build a rudimentary classifier. Similar to the approach described in [209], we train a Gaussian mixture model, $\mathcal{M}$, on color features (a^* and b^* components of the CIE $L^*a^*b^*$ color space [144]), using labeled foreground (plant) data from the first uncompressed image of the time-lapse sequence (excluded from testing). At each tested bit rate, we calculate the average application error:

$$E_{\mathcal{M}} = \frac{\sum_{i=1}^{n} (\mathcal{M}(\widetilde{\boldsymbol{x}}_i) - \mathcal{M}(\boldsymbol{x}_i))^2}{\sum_{i=1}^{n} (\mathcal{M}(\boldsymbol{x}_i))^2}\,, \tag{8.11}$$

between the posterior probabilities estimated by $\mathcal{M}$ on the n original, $\boldsymbol{x}_i$, and reconstructed, $\widetilde{\boldsymbol{x}}_i$, image pixels. Application error is expressed in percentage, where best possible value of $E_{\mathcal{M}}$ is 0%.

8.3.2 Results

In this section, we present rate-distortion performance of the proposed approaches. We first compare them in terms of overall reconstruction accuracy. Next, we demonstrate the supervised approach in an application-aware context.

[3] http://cvxr.com/cvx

[4] http://code.google.com/p/jj2000/

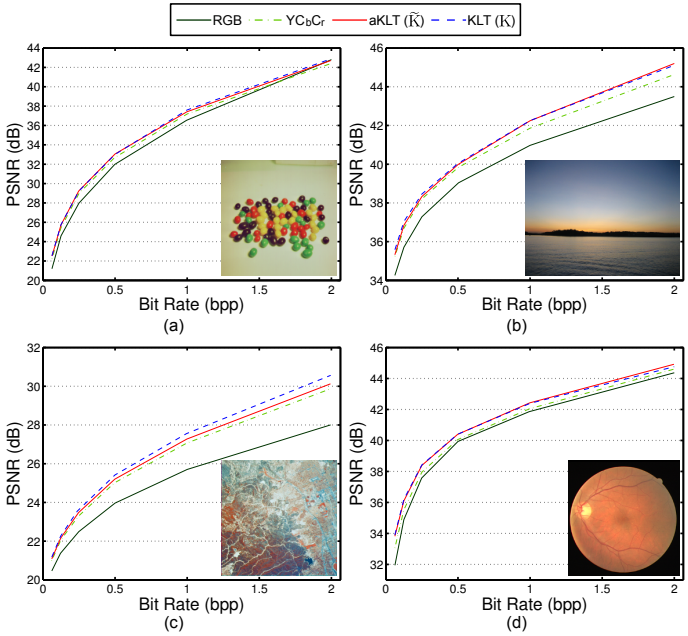


Figure 8.4: Reconstruction accuracy of standard test images, using fixed and data-dependent color transforms (proposed transform is shown with solid red curve). (a) Jelly beans (width × height: 256×256 pixels). (b) Landscape of Bretagne (width × height: 2592×1944 pixels). (c) Aerial photograph of Woodland Hills, Ca. (width × height: 512×512 pixels). (d) Human retina [293] (width × height: 565×584 pixels). For the aKLT, average results are shown, obtained using 100 different initializations (see Section 8.2.1).

Reconstruction accuracy. On the benchmark images of Figure 8.4, all of the decorrelating transforms provide considerable PSNR improvement with respect to the plain RGB color space, with the data-dependent transforms (KLT, aKLT) outperforming the fixed $\mathrm{YC_bC_r}$. Notably, our proposed low-complexity aKLT, $\widetilde{\mathrm{K}}$, exhibits performance very close to the regular KLT, or in some cases slightly superior (cf. red line in Figure 8.4b and 8.4d, higher bit rates).

Table 8.2 reports image fidelity results for the Arabidopsis plant data-

Table 8.2: Reconstruction accuracy comparison for the plant dataset [274] (cf. Figure 8.3a).

	Average PSNR (dB)					
Bit Rate (bpp)	RGB	YC_bC_r	KLT	aKLT	FST	RCA
0.0625	26.75	27.07	27.28	27.23	26.81	25.68
0.125	27.86	28.31	28.44	28.39	27.92	26.26
0.25	29.09	29.53	29.58	29.55	29.13	27.21
0.5	30.53	30.78	30.81	30.90	30.49	28.05
1.0	32.39	32.39	32.28	32.46	32.07	29.23
2.0	34.86	34.68	34.48	34.81	34.46	30.42

set.[5] At low bit rates (< 1 bpp), decorrelating transforms (YC_bC_r, KLT, aKLT) achieve better performance than RGB (0.25 to 0.6 dB improvement in PSNR). Performance of the aKLT is always superior to the YC_bC_r, and for bit rates greater than 0.5 bpp it surpasses the KLT. As also found by others in some cases [196], at higher bit rates the RGB representation may result in higher performance, due to noise amplification effects of the other transformations and reduced quantization (see solid green line in Figure 8.4a, in the range of bit rates close to 2 bpp). The supervised FST, D, shows PSNR performance comparable to RGB, with slight improvement only at low bit rates. On the other hand, the supervised RCA, L, performs worse than baseline RGB, probably due to the lack of orthogonality (Gershikov et al. [106] observe a dependence of PSNR performance on the condition number of the color transformation matrix).

Figure 8.5 offers a visual comparison between the components of the color spaces. The RGB channels appear highly redundant (particularly the first two, i.e. red and green), total signal energy is spread across all channels, and the distributions of intensity values span the entire 0 to 255 range. In the YC_bC_r, the distributions of second and third channel cover a smaller range of values, however signal energy is again dispersed over all three channels. On the other hand, KLT and aKLT present highly similar output, with most of the signal energy (66-70%) compacted in the first channel, and narrow and peaked distributions in second and third

[5] Observe that, in general, major bit rate savings are attained by compression schemes with the combined use of several coding tools. Thus, seemingly small differences in PSNR observed here (i.e. in the order of a fraction of dB) are accounted for by the fact that only the effect of color transformation is tested.

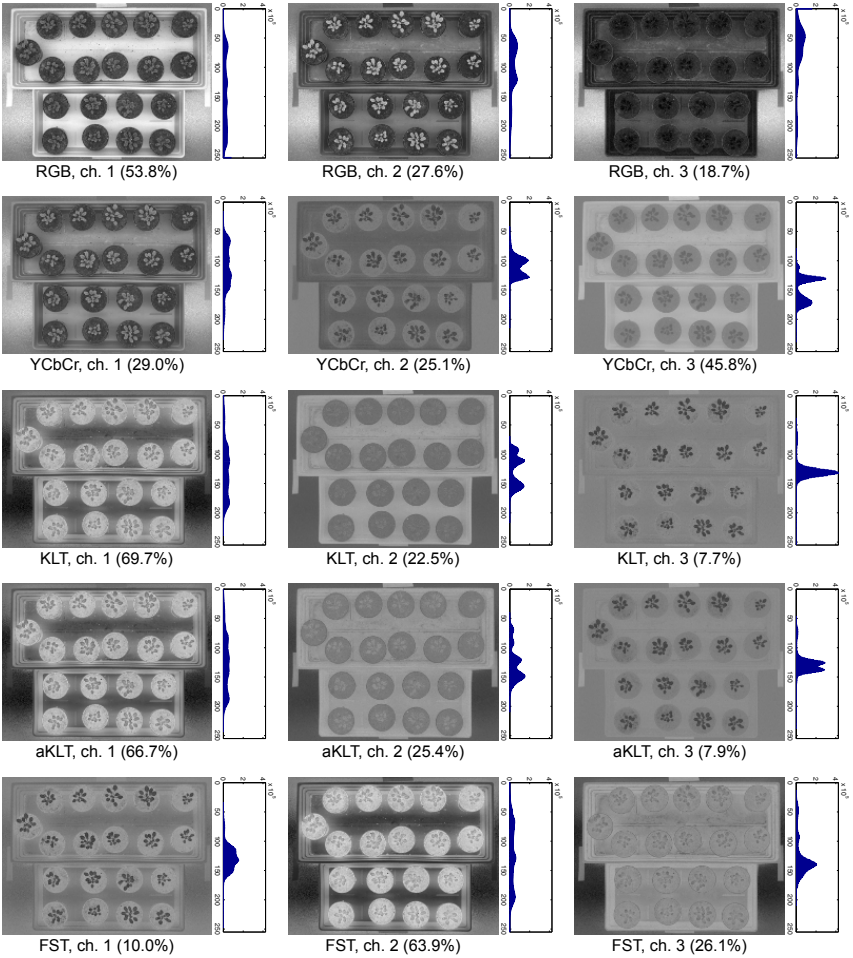


Figure 8.5: Projection of the image of Figure 8.3a in a variety of color spaces. Next to each channel is shown the corresponding histogram of intensity values, and in parentheses the percentage of signal energy contained in that component.

channel, containing a relatively low amount of information. On the other hand, the supervised FST concentrates more energy (64%) in the second channel, as the first one (i.e. projection on Fisher's discriminant vector)

Table 8.3: Mean and standard deviation of reconstruction accuracy performance for the images of Figure 8.4, using the aKLT and 100 different initializations.

	Average PSNR (dB)			
Bit Rate (bpp)	Jelly Beans	Bretagne	Aerial	Retina
0.0625	22.52 ± 0.08	35.32 ± 0.06	21.09 ± 0.03	33.82 ± 0.08
0.125	25.71 ± 0.06	36.85 ± 0.05	22.16 ± 0.07	36.10 ± 0.07
0.25	29.22 ± 0.11	38.33 ± 0.04	23.48 ± 0.09	38.39 ± 0.06
0.5	33.05 ± 0.06	39.98 ± 0.04	25.20 ± 0.10	40.40 ± 0.05
1.0	37.41 ± 0.04	42.25 ± 0.02	27.27 ± 0.12	42.43 ± 0.05
2.0	42.71 ± 0.06	45.20 ± 0.02	30.12 ± 0.17	44.92 ± 0.06

Table 8.4: Average inter-channel linear correlation of the test images of Figure 8.4. For the aKLT, average results are shown, obtained using 100 different initializations (see Section 8.2.1).

	Correlation		
Transform	ch. 1–2	ch. 1–3	ch. 2–3
RGB	0.84	0.71	0.91
YC_bC_r	-0.39	0.11	-0.71
aKLT	0.04	0.09	0.11
KLT	0.00	0.00	0.00

is purposely designed to exhibit good discrimination capabilities of the plant objects. Such features render the KLT, aKLT, and FST ideal for the coding of color images, because the channels accounting for less energy can be effectively subsampled.

Unsupervised transform. In order to assess the sensitivity of the aKLT to the random initialization of the vectors $\boldsymbol{a}_2$ and $\boldsymbol{a}_3$ in the matrix A (cf. Section 8.2.1), we compute 100 different realizations of $\widetilde{\mathrm{K}}$ for each of the test images in Figure 8.4. As shown in Table 8.3, the aKLT behaves consistently, and variations in PSNR performance due to different initial values are on average approximately only 0.2%.

Furthermore, the aKLT exhibits good decorrelating capabilities. As

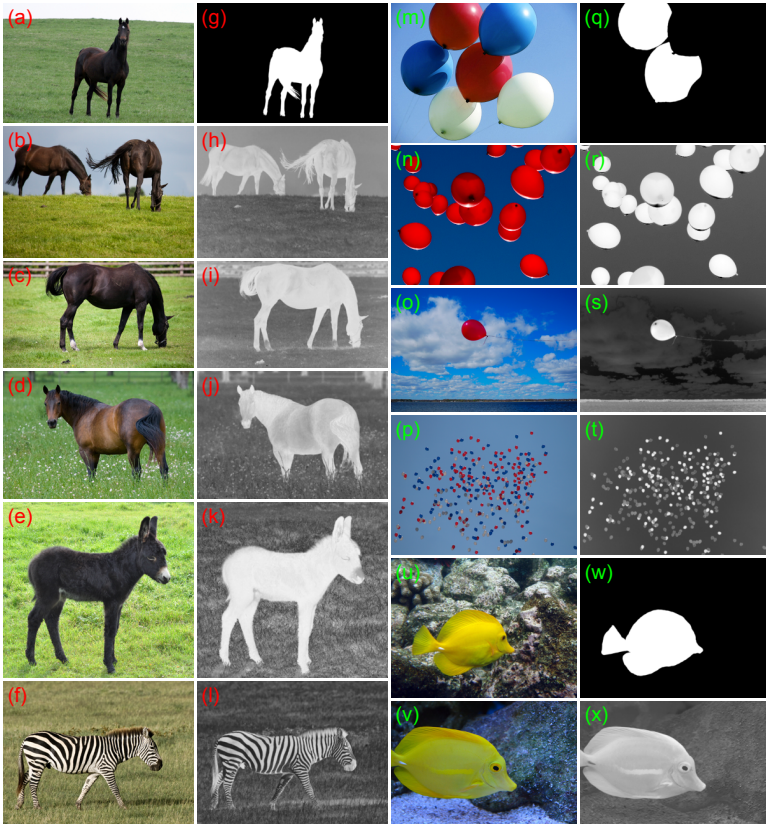


Figure 8.6: Demonstration of the supervised transform using images of: (a)-(f) horses, (m)-(p) balloons, and (u)-(v) fish. For each category a single FST was obtained, using for training images in (a), (m), and (u), and corresponding ground truth segmentations (i.e. (g), (q), and (w), respectively). Images in (h)-(l), (r)-(t), and (x) visualize the projections of the test images on the first component of the FST.

shown in Table 8.4, in the RGB domain, the channels of the test images of Figure 8.4 present on average strong linear correlation. Inter-channel linear correlation is only moderately reduced by the YC_bC_r, whereas the aKLT is able to achieve the almost complete decorrelation obtained by the optimal KLT.

Supervised transforms and application-aware compression. Figure 8.6 provides several visual examples of the supervised transform on a variety of different images, showing its ability to identify the objects of interest in the test images, even when major changes occur in the scene with respect to the training data (e.g., compare background appearance of the images in Figures 8.6u and 8.6v). This approach is chiefly based on color information, therefore after learning the transform D on the image of a black horse (Figures 8.6a and 8.6g), only the black stripes of the zebra in Figure 8.6f result in a high response, while the white stripes are regarded as background (cf. Figure 8.6l). On the other hand, the transform D estimated from training data in Figures 8.6m and 8.6q, is able to selectively identify only the red balloons in the image of Figure 8.6p.

Figure 8.7 compares the approaches from an application standpoint. Color transformation alone provides up to 1.26 dB improvement in PSNR of the foreground (plant) regions relative to RGB, with the FST now obtaining competitive performance. The supervised transforms do not show remarkable improvements with respect to the other approaches, probably due to lacking decorrelation capabilities for these images, causing losses in bit rate performance.

Supervised transform for ROI detection. The separation property of the supervised color transform can be further exploited in applications in which the objects of interest can be discriminated by color features (e.g., plant objects in our dataset can be separated from the background based on color information). Therefore, we envision the use of the supervised color transform to obtain from the transformed image a region of interest (ROI) estimate, that can be used in an encoder with ROI coding capability (e.g. the JPEG 2000 standard [287]). With respect to other approaches obtaining the ROI information from a detection module external to the encoder [216], we propose for the first time to combine color transformation and ROI estimation in a single framework, identifying potential ROI masks solely on the basis of the class separation capabilities of the supervised transform, thus reducing computational overhead at the encoder.

When using the FST approach, the first channel of the FST domain, $y_i^{(1)} = \boldsymbol{d}_1^{\mathsf{T}} \boldsymbol{x}_i$, corresponds to the projection on Fisher's discriminant vector (cf. Figure 8.5, bottom row). In an unseen image, to obtain an ROI estimate, $\Gamma(\mathrm{D}, \theta^*) \in \{0,1\}^n$, we decide the class of a pixel (foreground or background) based on a single threshold θ^* on the values of $y_i^{(1)}$. We estimate

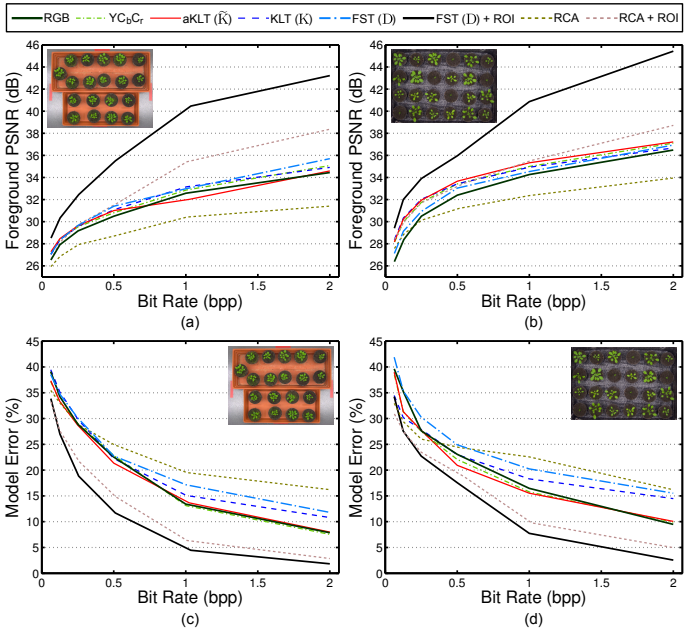


Figure 8.7: R-D performance using the proposed transforms (in solid curves) in comparison to others (dashed curves) on the plant image data of Figure 8.3 using application-aware metrics: (a)-(b) reconstruction accuracy of the objects of interest, and (c)-(d) model error $E_{\mathcal{M}}$ of Eq. (8.11). Results in (a) and (c) are averaged over 19 test images.

θ^* from our training set, maximizing the Dice Similarity Coefficient (DSC):

$$\theta^* = \arg\max_{\theta} \frac{2 \cdot |\Gamma_{\mathrm{GT}} \cap \Gamma(\mathrm{D}, \theta)|}{|\Gamma_{\mathrm{GT}}| + |\Gamma(\mathrm{D}, \theta)|}, \tag{8.12}$$

between the ground truth of pixels, Γ_{GT}, and the classification, $\Gamma(\mathrm{D}, \theta)$, obtained using D and threshold θ on the training data. Supervised transform D and threshold θ^* are generally assumed to be obtained offline, therefore we estimate θ^* using a parameter sweeping strategy. On the other hand, if an application requires that θ^* be obtained at the sensor, statistical assumptions on the distribution of the data (e.g., Gaussian) would lead to closed-form solutions for finding the optimal θ^* efficiently [300]. When using the RCA approach, ROI estimation proceeds analogously.

When used in a spatial decorrelation context to estimate an ROI, combined with the ROI coding feature of JPEG 2000, the FST + ROI approach obtains a major improvement at all bit rates: 2 to 8.8 dB increase in foreground PSNR, and 13 to 77% reduction in application error (cf. black solid line in Figures 8.7a and 8.7c). When using the same FST on a test image of Arabidopsis plants acquired under significantly different scene conditions (Figure 8.3c), the FST + ROI approach proves robust, obtaining again best performance (cf. Figures 8.7b and 8.7d). On the contrary, although the RCA approach is capable of detecting the regions of interest in an image in both testing scenarios, when projecting the images in the so obtained color space, the new intensity values are altered in a way that the benefits of the application-aware transform are diminished (or surpassed) by numerical errors introduced by the combination of forward and reverse color transformation and compression (cf. yellow dashed line in Figure 8.7).

A visual comparison of reconstructed images after compression with JPEG 2000 and all color transforms adopted in this work is shown in Figure 8.8. The RGB image appears oversmoothed, whereas the decorrelating transforms (YC_bC_r, aKLT, and KLT) exhibit higher image fidelity and appear increasingly richer in details (cf. Figures 8.8b, 8.8c, 8.8d, and 8.8e). The supervised FST alone already provides good reconstruction accuracy, however, the FST + ROI outperforms all other methods (cf. Figures 8.8g and 8.8h). The artifacts introduced by the RCA are evident in Figure 8.8i, and even when coupled with ROI coding the approach produces a noisy image (Figure 8.8j).

The results envision different use cases for the proposed approaches. The aKLT is general purpose and can be efficiently calculated on a per image basis to target reconstruction accuracy. On the other hand, the supervised approach is best suited for application-aware compression or enhancement scenarios, and since it does require supervision (which can be costly to obtain at the sensor) is assumed to be computed offline. The regularized versions of Eq. (8.10) are highly dependent on the free parameter λ and their performance is found to lie within the bounds of the other two. When varying the value of λ, the new transform D' exhibits behavior very close to either the supervised or the unsupervised transform, respectively. Therefore, it is best to exploit the classification abilities of the supervised FST to focus bits in appropriate places in the image, which is considerably less computationally demanding.

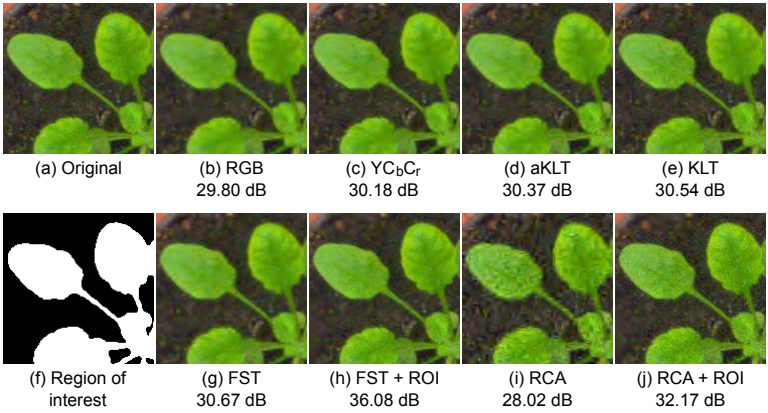


Figure 8.8: (a) Detail of the image in Figure 8.3a. (b)-(e), (g)-(j) Reconstructed images after compression at 0.5 bpp using the JPEG 2000 standard and several color space transformations. Foreground PSNR between (a) and each of the reconstructed versions, calculated for the plant region indicated in (f), is also reported.

8.4 Summary

In this chapter we addressed the problem of designing image-adaptive color space transformations for image and video compression applications. In recognition of the superior data-dependent KLT with respect to fixed transforms such as the YC_bC_r (as also confirmed by our experimental results), we derived a low-complexity approximation, the aKLT, capable of comparable performance. Our proposed aKLT achieves lower computational complexity than other KLT approaches in the literature, which is expected to result in proportionally reduced computation time, when devising optimized implementations. This will ease adoption on resource-constrained devices such as our affordable sensor, or in time-critical applications.

We also considered an application-aware compression setting, in which prior knowledge is available at the encoder on the objects of interest in the scene. Here this prior knowledge takes on the form of a color transform, which can be obtained at the receiver and communicated to the sensor. We formulated a novel approach to design color transforms with class separation capabilities, using supervised learning methods. Inspired by

the linear discriminant analysis, we measured class separability using the Fisher's discrimination criterion, and adopted the Foley-Sammon transform to obtain an orthogonal application-aware color transform. We also adopted metric learning approaches, however they focus only on class separation (renouncing also orthogonality) and were found to result in lower performance in a compression context. The proposed unsupervised and supervised approaches, for which closed-form solutions were presented, address different requirements, therefore we also considered optimization strategies to combine the two approaches. In the experiments, we also showcased the use of the separation property of the supervised transforms to detect regions of interest in an image, and inform the encoder where to focus bit rate spatially.

When coupled with quantizer design even greater bit rate savings are possible, but that would in general violate standard compliance. Increased image resolution or video applications are expected to emphasize the benefits of the proposed approaches. While we adopted the JPEG 2000 standard, our methodology is general and can be adapted to other coding schemes. Reversible integer approximations of the proposed transforms can also be derived [122, 123], for lossless or progressive lossy-to-lossless compression of color images.

In the next chapter we will complete our investigation of application-aware compression approaches, by letting the encoder make rate-distortion decisions, based on an estimate of the accuracy the analysis system will achieve on the reconstructed image.

9 Application-aware rate-distortion optimization

9.1 Introduction

In the previous chapters we described approaches to application-aware image compression that inform the encoder of the image regions to compress at better quality. However, these 'explicit' approaches can only be used when regions of interest can be clearly defined (e.g., plant objects vs. background). On the contrary, when the information to preserve appears all over the image (e.g., edges), an ROI-based approach would not be beneficial. To overcome this limitation and provide a more general solution to application-aware image compression, in this chapter we inject prior knowledge in the rate-distortion optimization process of the encoder.

Modern image and video coding schemes typically split the input signal into independent coding units (CUs), corresponding for example to non-overlapping image blocks or sub-bands. Hence, for each such

This chapter is partly based on:

- M. Minervini, S. A. Tsaftaris, "Classification-aware distortion metric for HEVC intra coding", submitted to the *International Conference on Visual Communications and Image Processing*, 2015.

CU, the encoder is faced with the problem of selecting a configuration of coding options (e.g., quantizer, CU size, bit stream ordering, intra- or inter-prediction mode) from a finite set of admissible values, that will result in different bit rate requirements and levels of distortion, for both the single CU and the entire image or video signal.

In an operational rate-distortion (R-D) optimization framework [232, 296], the resource allocation problem is addressed by minimizing a given distortion metric between original and reconstructed signal, subject to a bit budget constraint (or, alternatively, by minimizing bit rate for a target level of distortion). For computational efficiency, simple distortion metrics based on the mean squared error (MSE), such as the sum of absolute differences (SAD) or the sum of squared differences (SSD), are routinely adopted by most image and video encoders. Recognizing that optimizing for MSE-like metrics does not necessarily entail the best perceptual quality (i.e. for a human observer) at a given bit rate [327], other metrics have been also adopted (or adapted) for rate-distortion optimized encoding, that are more correlated with the way the human visual system perceives distortion (e.g., visual attention schemes [183,297], just noticeable distortion [55,186], structural similarity [42,137,265,323,338]).

This approach to application-aware compression is more flexible than those based on spatial or color information only, because in modern compression standards (e.g., HEVC) increasingly more decision among different coding options are made by selecting the one yielding minimum rate-distortion cost. Therefore, a distortion metric designed specifically for the application at hand will have profound effects on compression performance.

In this chapter, we propose to direct the rate control scheme of the encoder to focus bits in critical for the application regions, i.e. those that are more likely to jeopardize the performance of automated analysis tasks, thus aiming at maximizing post-compression classification accuracy. To that end, in Section 9.2 we define a new distortion metric for pixel-level classification accuracy based on a geometric heuristic. We validate our approach by implementing our metric in the rate-distortion optimization framework of the recent High Efficiency Video Coding (HEVC) standard [295], and in Section 9.3 we derive optimal model parameters for the λ-domain rate control algorithm, based on curve fitting procedures and training data. Our approach involves only encoder-side optimizations, and the resulting bit stream is standard compliant.[1] In Section 9.4, we

[1]Notice that while in Chapters 6 and 8 we tested our approaches using JPEG 2000, here

validate our application-aware R-D optimization approach on a time-lapse sequence of plant images (cf. Chapter 3), observing a reduction in segmentation error at all test bit rates (between 0.3 and 2 bpp) with respect to the baseline image fidelity R-D optimization approach.

9.2 Classification-aware distortion metric for rate-distortion optimization

In this section we address the problem of rate-distortion optimization in an application-aware image compression setting. First, we state the general rate-distortion optimization problem and the notation adopted throughout the chapter. We proceed by discussing the relation between compression and classification, on the basis of which we define a new distortion metric to evaluate pixel-level classification accuracy.

9.2.1 Rate-distortion optimization: Notation and problem statement

In current transform coding schemes (e.g., JPEG, JPEG 2000, H.264, HEVC), an original input image (or a frame) $\mathbf{X}$ is transformed with a reversible operation (e.g., discrete cosine transform, discrete wavelet transform) and split into coding units (CUs), or blocks. Then, the CUs undergo a lossy operation (e.g., quantization, bit stream truncation), resulting in approximations that are further entropy coded.

Optimal trade-off between compression ratio and average distortion with respect to the original image can be achieved by solving a bit allocation problem among the CUs. This task is accomplished by the encoder, that seeks to find the set of parameters Θ to encode the CUs, such that a given distortion metric D between original, $\mathbf{X}$, and reconstructed, $\hat{\mathbf{X}}$, image is minimized, while satisfying a bit rate constraint [283]:

$$\begin{aligned} \underset{\Theta}{\text{minimize}} \quad & D(\mathbf{X}, \hat{\mathbf{X}}; \Theta) \\ \text{subject to} \quad & R(\mathbf{X}; \Theta) \leq R_{\text{tot}}, \end{aligned} \tag{9.1}$$

we rely on the more recent HEVC standard (first version finalized in 2013), which is expected to result in a broader install base of software and hardware-accelerated decoders, following the lead of its predecessor H.264 [332].

where $R(\mathbf{X};\Theta)$ is the output rate, and R_{tot} is the total available bit budget. The optimization problem of Eq. (9.1) is usually solved using the generalized Lagrange multiplier method [89, 283], resulting in a set of encoding parameters that make the best use of the available resources according to D, i.e. while minimizing total (average) distortion. According to the generalized Lagrange multiplier method [89], the solution Θ^* to the unconstrained problem:

$$\underset{\Theta}{\text{minimize}}\ D(\mathbf{X},\hat{\mathbf{X}};\Theta)+\lambda R(\mathbf{X};\Theta), \tag{9.2}$$

for any $\lambda \geq 0$, is also the solution to the constrained problem of Eq. (9.1), whose constraint is $R(\mathbf{X};\Theta) \leq R(\mathbf{X};\Theta^*)$. The Lagrange multiplier λ sets the trade-off between rate and distortion, and is selected according to the budget constraint R_{tot}. While it would be desirable to select the value of λ in a way to achieve complete resource usage, i.e. $R(\mathbf{X};\Theta^*) = R_{\text{tot}}$, the parameter space of coding options Θ is in fact discrete and finite, thus, in general almost full resource usage is attained by selecting the smallest λ such that $R(\mathbf{X};\Theta^*) \leq R_{\text{tot}}$. In practice, the value of λ^* can be found by sweeping over a range of values, using a bisection algorithm if the R-D curve is monotone, or by estimating it based on a model.

At the pixel level, the sum of squared errors (SSE) is typically used as the criterion to measure distortion:

$$D_{\text{SSE}}(\boldsymbol{x},\hat{\boldsymbol{x}};\Theta) = \|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2, \tag{9.3}$$

where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ denote pixel values respectively of original and reconstructed CUs encoded with parameters Θ, and $\|\cdot\|$ denotes the ℓ_2 norm. At the CU level, the sum of pixel-level distortion values provides an aggregated estimate of the distortion experienced by a CU encoded with parameters Θ. The SSE or akin error metrics (e.g., mean squared error) are simple and efficient to compute [327], and they satisfy the additivity condition assumed by the generalized Lagrange multiplier method [89], i.e. if D is such that overall image distortion corresponds to the sum of individual distortions of the CUs, then given the optimal λ^*, the minimization of Eq. (9.2) can be conducted for each CU independently.

However, MSE-like metrics remain agnostic to the application, here defined as a classifier that assigns a class label to each image pixel. In the following we discuss the relationship between compression error and classification. Next, we define a new distortion metric based on geometric heuristics that accounts for post-compression classification accuracy.

9.2.2 On the relation between compression and classification

Several pattern recognition tasks (e.g., object detection, segmentation, scene interpretation, image retrieval) often rely on the classification of image pixels into labels drawn from a finite set $\mathcal{Y}$ (see [190] for a comprehensive overview of image classification approaches). Let $\mathcal{X}$ be the domain of image pixel values (e.g., gray scale or RGB color intensities), or in general a (possibly) multi-dimensional feature space of feature vectors associated to each pixel (e.g., a combination of intensity, texture, and other descriptors accounting local regional properties). A discriminant function $\delta\colon \mathcal{X} \to \mathcal{Y}$ can be defined to decide the class of a pixel value.

When an image $\mathbf{X}$ undergoes lossy compression, artifacts and loss of details in the reconstructed image $\hat{\mathbf{X}}$ (and consequently in the corresponding features that are possibly extracted from it), will affect the performance of the classifier.[2] For a given target rate R_{tot}, our goal is to select coding parameters Θ in a way to achieve maximum possible resource usage, while minimizing classification distortion, i.e. discrepancy between the outcome of δ on original and reconstructed image, respectively.

For a simple two-class problem, several metrics exist (e.g., Hamming distance, Dice dissimilarity coefficient, Hausdorff distance) to compare the binary output obtained from the evaluation of δ on $\mathbf{X}$ and $\hat{\mathbf{X}}$. However, this would render the optimization problem of Eq. (9.1) difficult and possibly inefficient to solve on devices. Besides, such an approach could not be easily extended to a multi-class classification scenario.

A pixel value $\boldsymbol{x}$ in the original image $\mathbf{X}$, after compression of $\mathbf{X}$ with parameters Θ, can be modeled as $\hat{\boldsymbol{x}}(\Theta) = \boldsymbol{x} + \boldsymbol{\varepsilon}(\Theta)$, where $\boldsymbol{\varepsilon}(\Theta)$ denotes additive quantization error. For all possible choices of Θ available at the encoder, the corresponding points $\hat{\boldsymbol{x}}(\Theta)$ will populate a hypercube in the image intensity space $\mathcal{X}$, centered in $\boldsymbol{x}$ and with 'radius' (i.e. maximum extent) determined by the Euclidean distance between $\boldsymbol{x}$ and the farthest approximation $\hat{\boldsymbol{x}}(\Theta)$, i.e. the one with maximum distortion. In general, a R-D optimization approach that aims at minimizing the MSE will select coding parameters Θ in a way that reconstructed pixels on average are close (in a Euclidean sense) to the original values.

However, the relation between quantization error in image (or feature) domain $\mathcal{X}$ and classification error cannot be modeled in general solely on

[2] The case studies presented in Chapter 5 illustrate and discuss the effects of compression on plant segmentation obtained via pixel classification.

the basis of MSE. From a classification standpoint, reconstructed pixels $\hat{\boldsymbol{x}}$ with equivalent MSE distortion (i.e. same distance from $\boldsymbol{x}$), may result in very diverse outcomes of the classifier. On the other hand, a classifier can be interpreted geometrically in terms of the boundaries that separate the decision regions in the feature space. Among the factors that contribute to the performance of the classifier on the reconstructed image pixels, we identify for example: (a) distance of $\boldsymbol{x}$ from the decision boundary H, (b) extent of the error hypercube, and (c) distribution of the approximations within the error hypercube.

Chances of misclassification on the reconstructed image pixels can be reduced by allocating less resources to image regions that are 'easy' from a classification perspective, in favor of the more 'difficult' and error-prone ones. This motivates the introduction of a distortion metric that takes into account how a classifier would be affected by the different decisions made at the encoder.

9.2.3 Proposed classification-aware distortion metric

For simplicity and without loss of generality, we initially present our distortion metric for a two-class problem (e.g., single object segmentation), i.e. $\mathcal{Y} \equiv \{0, 1\}$. Let δ_H be the discriminant function of a linear classification model, where $H = \{\boldsymbol{x} \in \mathcal{X} \,|\, \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + w_0 = 0\}$ denotes the decision hyperplane. (Later in this section we will discuss the generalization of the proposed approach to multi-class classification and nonlinear discrimination functions.)

Figure 9.1 shows a graphical representation of a two-dimensional feature space, and different approximations of a point $\boldsymbol{x}$ resulting from compression errors. Shown is also a decision hyperplane H, representing the prior knowledge of the application available at the encoder. Notice that H is a surrogate of the actual application at the receiver, which may involve more sophisticated vision algorithms but operates only on compressed data. Points $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$ are in the same region as the original $\boldsymbol{x}$ with respect to H, thus, classification outcome of these reconstructed data points will likely be identical to the uncompressed one, i.e. $\delta_H(\boldsymbol{x}) = \delta_H(\hat{\boldsymbol{x}}_1) = \delta_H(\hat{\boldsymbol{x}}_2)$. Notice that $\hat{\boldsymbol{x}}_2$ is closer to the decision boundary, thus ambiguity of its classification at the receiver may increase. On the other hand, $\hat{\boldsymbol{x}}_1$ moves farther from H than $\boldsymbol{x}$, thus a denoising effect of compression is observed here. Finally, $\hat{\boldsymbol{x}}_3$ is located in a different region of the problem space, hence the compression error that maps $\boldsymbol{x}$ into $\hat{\boldsymbol{x}}_3$ will (possibly) lead to

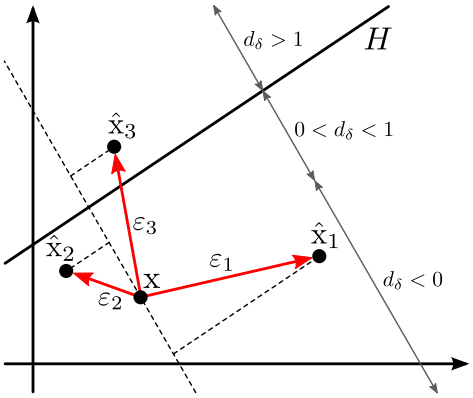


Figure 9.1: Graphical representation of the proposed distortion metric d_δ of Eq. (9.6) in a two-dimensional feature space. Shown are the decision hyperplane H of a binary classifier, and an original image pixel (or features) value $\boldsymbol{x}$. Data points $\hat{\boldsymbol{x}}_1$, $\hat{\boldsymbol{x}}_2$, and $\hat{\boldsymbol{x}}_3$ are example approximations after lossy compression (e.g., quantization) of $\boldsymbol{x}$, with $\boldsymbol{\varepsilon}_1$, $\boldsymbol{\varepsilon}_2$, and $\boldsymbol{\varepsilon}_3$ denoting corresponding error vectors.

a misclassification of that point at the receiver. The example shown in Figure 9.1 leads to the following two observations.

Observation 1. Distortion estimation should be inversely proportional to the (signed) distance $\Delta(\boldsymbol{x}, H)$ between an original data point $\boldsymbol{x}$ and the decision hyperplane H:

$$\Delta(\boldsymbol{x}, H) = \frac{\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + w_0}{\|\boldsymbol{w}\|}\,, \tag{9.4}$$

where $\Delta(\boldsymbol{x}, H) > 0$ if $\boldsymbol{x}$ lies on the same side of the plane H as the normal vector $\boldsymbol{w}$ and negative otherwise.

Observation 2. Distortion should be proportional to the component of the error vector $\boldsymbol{\varepsilon}_{\|\boldsymbol{w}}$ in the direction normal to the decision hyperplane:

$$\boldsymbol{\varepsilon}_{\|\boldsymbol{w}} = \frac{(\hat{\boldsymbol{x}} - \boldsymbol{x})^\mathsf{T}\boldsymbol{w}}{\|\boldsymbol{w}\|}\,, \tag{9.5}$$

where the error vector is oriented from $\boldsymbol{x}$ to $\hat{\boldsymbol{x}}$.

Based on these heuristics, we define our proposed classification-aware distortion metric d_δ as:

$$\begin{aligned} d_\delta(\boldsymbol{x}, \hat{\boldsymbol{x}}; H) &= -\operatorname{sgn}(\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + w_0) \cdot \frac{\|\boldsymbol{w}\|}{\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + w_0} \cdot \frac{(\hat{\boldsymbol{x}} - \boldsymbol{x})^\mathsf{T}\boldsymbol{w}}{\|\boldsymbol{w}\|} \\ &= -\frac{(\hat{\boldsymbol{x}} - \boldsymbol{x})^\mathsf{T}\boldsymbol{w}}{\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + w_0}, \end{aligned} \tag{9.6}$$

where $\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + w_0 \neq 0$. The first term adjusts for the sign, such that: $d_\delta < 0$ if $\hat{\boldsymbol{x}}$ is farther than $\boldsymbol{x}$ with respect to H; $0 < d_\delta < 1$ if $\hat{\boldsymbol{x}}$ is between $\boldsymbol{x}$ and H; and $d_\delta > 1$ if $\hat{\boldsymbol{x}}$ crosses the decision boundary. Notice that the denominator of Eq. (9.6) is zero if the original point lies exactly on H. In this case, prior knowledge available at the encoder is not sufficient to make an informed decision: any displacement of such a point may result in a misclassification at the receiver, thus distortion is 'infinite'.[3]

To satisfy the conditions of the generalized Lagrange multiplier method [89], distortion values must be nonnegative. Thus, we exponentiate the value of d_δ in Eq. (9.6):

$$D_\delta(\boldsymbol{x}, \hat{\boldsymbol{x}}; H) = \exp(d_\delta(\boldsymbol{x}, \hat{\boldsymbol{x}}; H)), \tag{9.7}$$

If $\hat{\boldsymbol{x}}$ is farther than $\boldsymbol{x}$ with respect to H, contribution of application error is between 0 and 1; while it is greater than 1 for $\hat{\boldsymbol{x}}$ closer to H and grows rapidly after the decision boundary is crossed.

Distortion at the CU level is obtained as the sum over all pixel-level distortion values:

$$D_\delta^{CU} = \sum_{i=1}^{N} D_\delta(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i; H), \tag{9.8}$$

where $\boldsymbol{x}_1 \ldots \boldsymbol{x}_N$ are the pixel values in a CU.

Depending on the application, a trade-off may be desirable between application accuracy and reconstruction accuracy, e.g., if images are also viewed by humans for visual assessment or inspection. Therefore, the distortion measure of Eq. (9.6) can be combined with traditional squared error distortion:

$$D_{\text{app}}(\boldsymbol{x}, \hat{\boldsymbol{x}}; \varphi, H) = (1 - \alpha)\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 + \alpha D_\delta \tag{9.9}$$

[3] To avoid singularity a small quantity could be added to the denominator. Alternatively, the largest finite value of the data type used in the implementation to define distortion values could be used.

where $0 \leq \alpha \leq 1$ sets the trade-off between reconstruction accuracy (α close to 0) and application accuracy (α close to 1). Besides, further error terms can be included in Eq. (9.9) to account for other applications (e.g., images are processed by several classifier trained for different purposes).

Extensions to nonlinear and multi-class classification problems

In applications where the classes are not separable by linear discrimination functions, nonlinear boundaries can be found using kernel methods. Therefore, a nonlinear classifier would require that Eq. (9.6) be adapted to calculate the distance of a data point from the decision boundary. Using the notation of support vector machines (SVM) [31], the separating hyperplane is defined in the input space as a linear combination $\boldsymbol{w} = \sum_{i=k}^{p} \alpha_k y_k \boldsymbol{x}_k$ of the support vectors $\boldsymbol{x}_k$, where p is the number of support vectors, y_k are their target values, and $\alpha_k \geq 0 \; \forall k$ are their weights (i.e. Lagrange multipliers arising from the dual formulation associated with SVM training). Using the so-called 'kernel trick', we modify Eq. (9.6) as follows:

$$d_\delta(\boldsymbol{x}, \hat{\boldsymbol{x}}; H) = -\frac{\sum_{k=1}^{p} \alpha_k y_k K(\boldsymbol{x}_k, \hat{\boldsymbol{x}} - \boldsymbol{x})}{\sum_{i=k}^{p} \alpha_k y_k K(\boldsymbol{x}_k, \boldsymbol{x}) + w_0}, \tag{9.10}$$

where K is a kernel function that computes the inner product between a data point and the parameter vector $\boldsymbol{w}$ in the kernel-induced dual space.

In many computer vision applications, multiple categories can be defined in the scene (e.g., an optical character recognition application for mail sorting, in which characters are to be preserved and digits are more important than other characters, or a traffic surveillance application, in which objects are classified as vehicles or pedestrians). A variety of approaches exist to decompose a multi-class classification task in terms of binary classification problems [267], which provide a natural way of adapting our proposed distortion function to handle multiple classes. For example, in a one-vs-all (or, equivalently, in an all-vs-all) strategy, an importance-weighted relation accounting for all of the N binary classifiers could be adopted, resulting in the following distortion metric:

$$d_\delta^{MC}(\boldsymbol{x}, \hat{\boldsymbol{x}}; H_1, \ldots, H_N) = -\sum_{i=1}^{N} \gamma_i \frac{(\hat{\boldsymbol{x}} - \boldsymbol{x})^\mathsf{T} \boldsymbol{w}_i}{\boldsymbol{w}_i^\mathsf{T} \boldsymbol{x} + w_i}, \tag{9.11}$$

where $N = |\mathcal{Y}| > 2$ is the number of classes, and γ_i are weights indicating the importance to the application of each class.

9.3 Application-aware rate-distortion optimization in HEVC

To validate our approach we implement the proposed distortion metric in the rate-distortion optimization framework of the HEVC compression standard [295]. First, we provide an overview of the λ-domain rate control algorithm [179] currently adopted in HEVC. Next, we discuss how we obtain suitable model parameters based on curve fitting procedures and training data.

9.3.1 Overview of the λ-domain rate control algorithm

In the HEVC coding standard [295], R-D optimization is employed at the encoder to decide the majority of coding parameters. The goal is to minimize the following cost function:

$$J = D_{\mathrm{SSE}} + \lambda R\,, \tag{9.12}$$

where $D_{\mathrm{SSE}} = \sum_{i=1}^{N}(Rec_i - Org_i)^2$ measures distortion between original and reconstructed image pixels as the sum of squared errors, R is the total amount of bits required to encode the picture, and λ is the Lagrange multiplier that trades off between rate and distortion. For computational efficiency, the λ parameter is estimated based on a model.

The JCT-VC has recently adopted the λ-domain rate control algorithm proposed by Li et al. [178, 179], which models the R-D function based on the relationship between λ and R. The relationship between R and D is modeled in HEVC using a hyperbolic function, which was found to outperform other models (e.g., the exponential function):

$$D(R) = CR^{-K}, \tag{9.13}$$

where C and K are model parameters. Accordingly, the slope λ of $D(R)$ in Eq. (9.12) can be obtained by differentiating Eq. (9.13):

$$\lambda = -\frac{\partial D}{\partial R} = CKR^{-K-1} = \alpha R^{\beta}\,, \tag{9.14}$$

where α and β are model parameters that depend on the characteristics of the source. Based on experimental observation [178], α = 3.2003 and β = -1.367 are used in HEVC as initial values and are updated with the encoding process, as described in [179].

The rate control scheme adopted in the test model (HM) [201] operates at different levels, administering the available bit budget among groups of pictures (GOP), pictures within a GOP, and blocks of a single picture. At the lowest level of the rate control hierarchy, the largest coding unit (LCU) is represented by a coding tree unit (CTU). Target bit rate T_{LCU} for a LCU is estimated on the basis of the available to that point bit budget and an estimate of the bits that will be required to encode the remaining LCUs of the frame.

With the target rate T_{LCU} known, the value of λ for a LCU can now be calculated using the $R-\lambda$ model of Eq. (9.14). Based on this λ value, all coding parameters for the LCU are determined by exhaustive search, evaluating for each option Eq. (9.12) and eventually selecting the configuration with minimum R-D cost.

On the other hand, to reduce encoding complexity, the QP value used to quantize the transformed coefficients of the LCU is obtained as a function of λ, instead of performing multiple-QP optimization. In HEVC, the linear-log QP-λ model proposed in [180] is used:

$$QP = a \log \lambda + b\,, \tag{9.15}$$

where $a = 4.2005$ and $b = 13.7122$ are model parameters, experimentally determined by multiple-QP optimization on several sequences [180]. To ensure spatial (between neighboring LUCs) and temporal (between consecutive frames) quality consistency, both λ and QP values are restricted to a narrow range, as detailed in [179].

Due to differences in the R-D characteristics of I frames and B frames, Li et al. [179] focus their discussion on B frames and estimate the model parameters α and β under this assumption. To accommodate such differences between I and B frames, the JCT-VC adopted for intra frames the R-λ model proposed by Karczewicz and Wang [157,158]:

$$\lambda = \frac{\alpha}{256}\left(\frac{C^{\beta_1}}{R}\right)^{\beta_2}, \tag{9.16}$$

where $\alpha = 6.7542$ and $\beta_2 = 1.7860$ are model parameters, C is a complexity measure of the LCU (or frame), and $\beta_1 = 1.2517$. Eq. (9.16) is used to estimate λ at the frame level and also for each LCU in an I-frame. In order to achieve higher accuracy in the rate allocation, the authors of [157] propose to calculate λ based on R, and also a complexity measure C based

on the Sum of Absolute Transformed Differences (SATD):

$$\text{SATD} = \sum_{i=1}^{N} \sum_{j=1}^{N} |h_{ij}| \,, \tag{9.17}$$

where $N = 8$, and h_{ij} are the coefficients obtained after applying the Hadamard transform to an $N \times N$ block of original pixel values. The complexity factor of Eq. (9.16) is then obtained at the LCU (resp. frame) level by summing the SATD calculated on all $N \times N$ blocks, normalized by the number of pixels in the LCU (resp. frame).

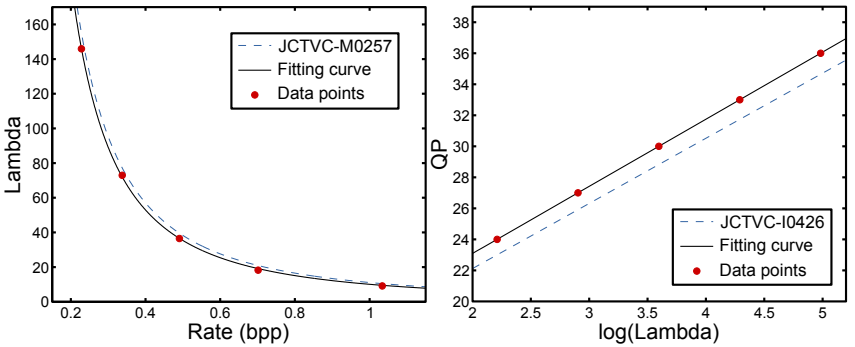
9.3.2 Rate-distortion modeling using the proposed metric

We describe the relation between rate and λ according to our proposed distortion metric D_δ defined in Eq. (9.7). Rate-distortion optimized rate control is achieved in HEVC by jointly using the R-λ and λ-QP models. Following an approach similar to [179], we obtain suitable model parameters based on training data and curve fitting procedures.

To collect data samples, we encode training images using different values of the quantization parameter (QP). In fixed-QP mode, the value of λ is calculated by the encoder at a frame level and then is propagated to all LCUs in that frame, thus making all of the LCUs operate at a constant slope point on their R-D curve. We record for each frame the values of λ, QP, SATD, and the resulting rate R.

To model the relationship between R and λ, we use the Levenberg-Marquardt algorithm [175,197] to fit Eq. (9.14) to the R-λ statistics collected from the encoded frames, thus obtaining an estimate of the parameters α and β. We proceed analogously to estimate α and β_2 for the R-λ model of Eq. (9.16) used for intra frames. Finally, to model the relationship between λ and QP we use a least squares fitting, obtaining an estimate of the parameters a and b of Eq. (9.15).

We measure goodness of fit using the coefficient of determination $R^2 = 1 - \sum_i (y_i - f_i)^2 / \sum_i (y_i - \bar{y})^2$, where y_i and f_i denote respectively observed and estimated values for the i^{th} data point, and i ranges over all available data points.



(a) $\alpha = 5.6344$, $\beta_2 = 1.8110$, $R^2 = 0.99$ (b) $a = 4.3281$, $b = 14.4329$, $R^2 = 1.0$

Figure 9.2: R-λ curve fitting and frame statistics obtained from a training image encoded with a flat QP approach, using QP $\in \{24, 27, 30, 33, 36\}$. For comparison, the default models adopted in HM are shown (dashed lines).

9.4 Results and discussion

9.4.1 Experimental settings

Image data. We evaluate our proposed approach on a time-lapse image sequence of growing plants, extracted from the Ara2012 dataset described in Chapter 3. The sequence used here is composed of 21 frames (width×height: 1080×432), showing a top view on 11 *Arabidopsis thaliana* subjects. We convert the original RGB images to gray-scale, by keeping only the green (G) channel. Figure 9.4a shows an example input image.

Codec settings. We implement our proposed metric (cf. Eq. 9.9) in the HM reference encoder, version 16.3. In the experiments we use default settings for all parameters except the following ones. To encode images with a single component we specify `InputChromaFormat` = 400 and `Profile` = 'monochrome'. The sequences are encoded with an 'intra-only coding approach', setting `FrameRate` = 1 frame/s and I-frame period to `IntraPeriod` = 1. We enable rate control (`RateControl` = 1) and encode the test sequence at a variety of bit rates up to 2 bpp, selected with the option (`TargetBitrate`).

To estimate the parameters of the λ-domain rate control algorithm, we

proceed as described in Section 9.3.2. We encode the first image of the time-lapse sequence with a flat QP approach, using QP $\in \{24, 27, 30, 33, 36\}$. In the curve fitting procedures, each model parameter is initialized using the corresponding default value in HM. Figure 9.2 shows original image statistics and corresponding fitting curves. We obtain on our data best-fit parameters ($r^2 \approx 1$, cf. Figure **??**) for R-λ model ($\alpha = 5.6344$, $\beta_2 = 1.8110$) and λ-QP model ($a = 4.3281$, $b = 14.4329$).

Since video quality consistency among neighboring LCUs may be unnecessary in an application-aware context, we disable the clipping operation of λ and QP values estimated by the models to a narrow range normally occurring at the encoder. We only ensure that $0 \leq \text{QP} \leq 51$, while we pose no restrictions on λ.

For comparison, we include in the experiments as baseline approach the HM reference encoder with default (SSE-based) rate control settings, referred to as 'HM16.3'. To decode all compressed bit streams we use the HM reference decoder, version 16.3.

Pixel classifier. To validate our approach we adopt a pixel-level classifier based on a logistic regression model operating on pixel intensity values. Classes are defined as 'foreground' (i.e. plant object) and 'background'. Based on a training (uncompressed) image and corresponding ground-truth pixel labels, model parameters $\boldsymbol{\beta}$ (i.e. the regression coefficients) are found using maximum likelihood estimation. We use for training the first image in the time-lapse sequence, which is subsequently excluded from the sequence being encoded for testing. Let $\boldsymbol{x} \in \mathcal{X}$ be a pixel value in a test image and y the corresponding (unknown) label. We predict the probability of y being 'foreground' as:

$$\mathcal{P}(y = \text{foreground}|\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{x}}}\,, \tag{9.18}$$

and consequently the probability of y being 'background' as the complement. We decide the class of a pixel ('foreground' or 'background') based on a single threshold θ^* on the probability values of Eq. (9.18). We estimate θ^* from the same training image data by sweeping over a range of values in the $[0, 1]$ interval and selecting the value:

$$\theta^* = \arg\max_{\theta} \frac{2|\mathrm{Y}^{gt} \cap \mathrm{Y}^{ar}|}{|\mathrm{Y}^{gt}| + |\mathrm{Y}^{ar}|}\,, \tag{9.19}$$

that maximizes the Dice similarity coefficient between the ground truth of pixels Y^{gt} and the classification Y^{ar} obtained using Eq. (9.18) and threshold θ with given model parameters $\boldsymbol{\beta}$.

Evaluation metrics. We evaluate the accuracy of our approach based on both reconstruction accuracy and post-compression classification accuracy. We measure image fidelity using Peak Signal-to-Noise Ratio (PSNR, cf. Eq. (5.2) on page 103).

To measure classification accuracy, we adopt the Global Consistency Error (GCE) [199] to compare the segmentation of the image reconstructed after compression with the ground-truth segmentation obtained manually.[4] Let $\mathrm{P}(S, p_i)$ be the set of pixels corresponding to the region in segmentation S that contains pixel p_i. Then, the local refinement error between segmentations S_1 and S_2 at the pixel location p_i is defined as:

$$E(S_1, S_2, p_i) = \frac{|\mathrm{P}(S_1, p_i) \setminus \mathrm{P}(S_2, p_i)|}{|\mathrm{P}(S_1, p_i)|}\,, \tag{9.20}$$

where $\setminus$ denotes set difference, and $|\cdot|$ set cardinality. The unidirectional local refinement errors are combined in the GCE:

$$\mathrm{GCE}(S_1, S_2) = \frac{1}{n} \min \left\{ \sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i) \right\}, \tag{9.21}$$

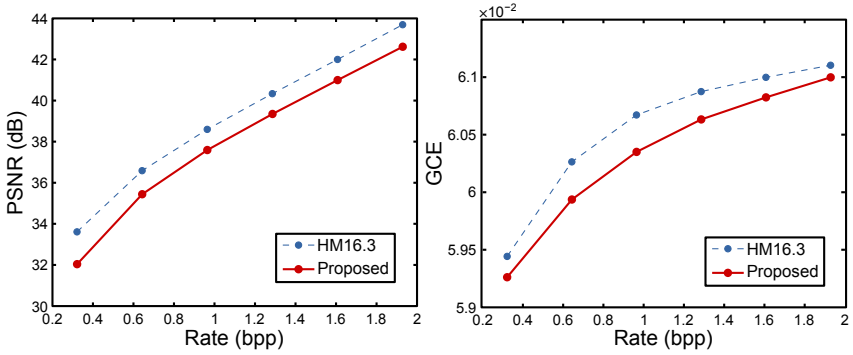
where n is the number of image pixels.

9.4.2 Results

Figure 9.3 shows average results across bit rates. It is readily seen that with the proposed approach GCE is consistently lower than HM16.3 at all test bit rates, demonstrating that using our classification-aware distortion metric the encoder focuses bit rate in a way to preserve classification accuracy. As expected, PSNR is reduced with the proposed approach, since the encoder does not optimize for image fidelity.

To appreciate visually the benefits of our approach, the outcome of post-compression classification is shown in Figure 9.4 for an example frame

[4] Although in Chapter 7 we discussed the modified Hausdorff distance (MHD) [85] for plant phenotyping applications, here we do not rely on the MHD because it is computationally complex, nor we employ the efficient approximation described in Section 7.3 in order to decorrelate from approximation errors.



(a) (b)

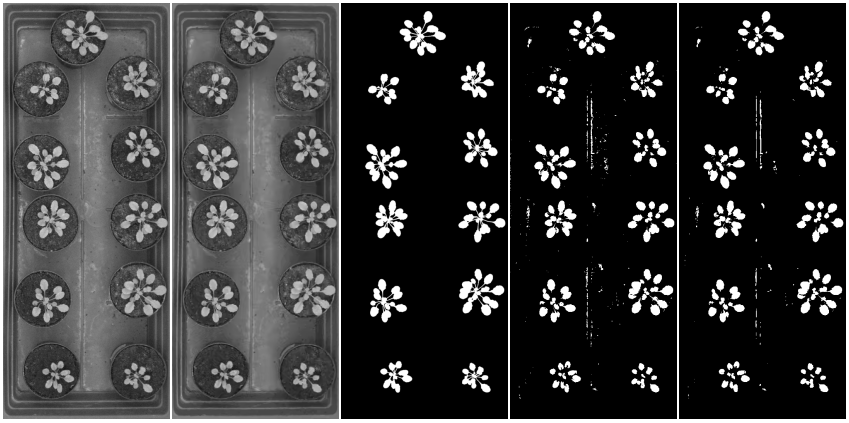
Figure 9.3: R-D curves showing compression performance according to: (a) image fidelity (PSNR, higher is better), and (b) segmentation accuracy (GCE, lower is better).

from the time-lapse test sequence. The classifier on uncompressed data and data compressed at 0.3 bpp using the proposed application-aware rate allocation strategy. Despite a reduction in bit rate of approximately 96% between uncompressed and compressed image, segmentation appears very similar, and in certain regions (e.g., center of the tray and around some plants) the segmentation obtained on compressed data appears even less noisy and closer to the ground truth than the segmentation obtained on the original image.

9.5 Summary

In this chapter we investigated application-aware compression from a rate-distortion optimization point of view. We defined a new distortion metric that permits the evaluation of errors introduced by lossy compression with respect to pixel-level classification.

While our metric is general and could be adapted to different coding standards (and potentially to different types of signal than images), here we implemented our approach in the rate-distortion optimization framework of the state-of-the-art HEVC video coding standard. We derived suitable model parameters for the λ-domain rate control algorithm adopted by HEVC, and compared our approach with R-D optimization based on



(a) (b) (c) (d) (e)

Figure 9.4: Example of post-compression classification: (a) original image, (b) reconstructed image after compression with the proposed approach at 0.3 bpp, (c) ground-truth segmentation obtained manually, (d) algorithmic segmentation of (a) obtained with the classifier, (e) algorithmic segmentation of (b) obtained with the same classifier.

image fidelity. A reduction in segmentation error (measured by Global Consistency Error) was observed in a range of bit rates.

To demonstrate our methodology we used gray-scale images and an intra-only coding strategy. However, our framework is general and in the future we will devise an implementation to test it on multi-component sources (e.g., RGB or YC_bC_r color images) and using generic group of pictures (GOP) structures, thus allowing also inter-coded frames.

Directions for future work also include an extension of the proposed metric to the transform (e.g., DCT) domain to permit a more efficient calculation that does not require that pixels be transformed back to the original domain.

With this chapter we complete our investigation of application-aware image and video compression approaches. While in Chapters 7 and 8 we explicitly informed the encoder on image regions to preserve (either by defining regions of interest, or via color transforms), in the approach based on R-D optimization presented in this chapter this behavior emerges implicitly from all decision made by the encoder.

10 Conclusions

10.1 Concluding remarks

Plant phenotyping is central to the understanding of plant function and is a tool that can enable us to meet agricultural demands of the future. Image-based approaches to plant phenotyping are gaining attention among plant researchers as a technology with the promise (and the capability) to increase throughput by orders of magnitude with respect to traditional approaches based on manual measurements. In the last decade, the computer vision community also started considering plant image analysis as fertile ground for interesting applications. In Chapter 2 we have discussed the key role of image processing and computer vision in plant phenotyping, outlining how the challenges that arise when examining the dimensions of the problem can motivate fundamental research and the investigation of novel methodologies.

Current versatile solutions for phenotype collection and analysis are costly and their adoption remains limited to few large organizations, thus high throughput in plant phenotyping is available mostly on a local basis. To overcome this limitation and enable 'global high throughput', we have proposed in this thesis an affordable solution for image-based plant phenotyping based on distributed sensing and analysis, which is expected to benefit labs (particularly in developing countries) and small companies such as breeders.

In Chapter 3 we have presented our affordable sensing solution based

on the Raspberry Pi single-board computer to acquire time-lapse image sequences of plants and transmit them to a remote location for storage and analysis. To test our approach, we have set up two plant experiments with different genotypes of *Arabidopsis thaliana*, and therein we have deployed our sensor to collect image datasets. We have annotated these images manually (e.g., plant and leaf object delineation, leaf centers) and used them throughout the thesis to validate the proposed methodologies. As a benefit for the scientific community, we also have released a curated collection of benchmark datasets (accompanied by manual annotations) aimed at the development and evaluation of computer vision algorithms in the context of plant phenotyping.

Our affordable sensor is complemented by a plant image analysis system. We have deployed our software solution on the scientific cloud platform provided by the iPlant Collaborative project to plant researchers. To remain agnostic to the specific laboratory setting and adapt to varying scene compositions, in Chapter 4 we have proposed a method integrating an active contour model and prior knowledge on plant appearance for the segmentation and the automated analysis of time-lapse plant images from phenotyping experiments.

In Part III of the thesis we have focused on the transmission to the analysis system of the images acquired at the sensor. To assess the effects of compression on plant phenotyping, we first have presented in Chapter 5 several proof-of-concept experiments demonstrating that image compression does introduce error in vision-based measurements and more importantly in the phenotypic analyses.

When conducting image-based plant phenotyping experiments, a single lab can routinely acquire hundreds of high-resolution images per day, hence compression is necessary to cope with the vast amounts of data at hand (especially in a distributed sensing scenario). More importantly, a pipeline of computer vision algorithms will process the images to automatically extract visual phenotypes, and a human eye most likely will never look at those images (and in most cases the differences between consecutive images can hardly be appreciated by visual inspection). Therefore, compression should be tailored to the application and optimized according to the algorithms running on the analysis system, rather than the human visual system. Chapters 6 to 9 have been devoted to the investigation of application-aware image and video compression strategies to increase compression efficiency while retaining application accuracy. The proposed solutions are characterized by low complexity, since they are

deployed on an affordable sensing device.

The proposed distributed sensing and analysis framework has been presented in Chapter 6. In Chapter 7 we have described an approach to compress the images in an application-aware fashion based on region-of-interest (ROI) coding and low-complexity ROI estimation, that exploits feedback from the receiver to the sensor. We have also investigated evaluation metrics for plant image segmentation, finding that the modified Hausdorff distance is ideally suited for this task. To permit the adoption of such a computationally complex metric on an affordable, resource-constrained sensing device, we have found efficient approximations using sparse linear regression models and simpler to compute metrics as features.

In Chapter 8 we have investigated color space transformation for image coding. We have described two frameworks to obtain linear maps of the RGB color data that minimize the loss of information due to compression. The first adapts to the image data and aims at reconstruction accuracy, representing an efficient approximation of the Karhunen-Loève transform. The second adapts to the application for which the images are used (in particular, the image classification task).

Finally, in Chapter 9 we have injected the concept of application-aware compression in the rate control algorithm of a video codec, by letting the encoder make rate-distortion decisions (which have an impact on several coding aspects), based on an estimate of the accuracy the analysis system will achieve on the reconstructed image data. To this end, we have defined a new distortion metric for pixel-level classification accuracy, and devised an implementation based on the recent HEVC compression standard.

10.2 Future directions

Several research avenues remain open for exploration regarding certain aspects that have been discussed in the thesis. Throughout this work we have described the details of our affordable phenotyping solution, but more importantly we have presented a general framework.

While we have devised an implementation based on an optical camera, other imaging modalities (e.g., thermal or infrared imaging) could be adopted by plugging different sensors. Additional depth information can be computed from a pair of images obtained by varying acquisition parameters (e.g., using depth from defocus techniques), or by deploying

the affordable sensors in pairs (e.g., adopting stereo vision techniques).

In our small growth chamber setup we have deployed a single static sensor. To cover a larger area a grid of sensors could be employed, or affordable robotics such as a cable-suspended camera systems could be adopted to obtain high flexibility. In our distributed sensing and analysis framework the sensor is free to move, thus when going from the greenhouse to the field the sensor could be hosted on board an autonomous robotic carrier, a tractor, or even an unmanned aerial vehicle (UAV). For field applications, energy-aware algorithms should be investigated to reduce energy usage on autonomous vehicles with limited resources. Additionally, the receiver may experience loss of information if the acquired images are transmitted over an unreliable communication channel, thus application-aware error detection and correction schemes could be investigated to ensure the delivery of important for the application information.

Our analysis system focuses on growth estimation via accurate plant segmentation, however more fine-grained phenotypic analyses will be based on individual leaf information. Robust leaf counting and segmentation algorithms should be therefore investigated and software solutions could be devised as modules of cloud-based analysis system.

In this thesis we have investigated different compression approaches in isolation to assess their value, but their combination would result in a complete solution for application-aware image and video compression. Also, we have presented our compression solutions in a plant phenotyping context, but our methodologies are general and could be beneficial to several other applications involving automated analysis of compressed image data (e.g., video surveillance, telemedicine, remote sensing). While we have focused on distributed sensing, the presented application-aware compression methodologies are beneficial also when transmission is not involved (e.g., for archival purposes), and in a phenotyping context their adoption may enable even higher throughput experiments in the future.

Our investigations of application-aware rate-distortion optimization have been restricted to two-class, linearly separable problems. While we outline possible directions to address nonlinear and multi-class classification problems, further investigations could lead to a generalization of the framework. Finally, from an implementation perspective, adapting the approach to multi-component (e.g., color) images and inter-frame coding will have numerous practical implications.

A Appendix

This appendix provides further details and instructions to setup and operate the affordable sensing solutions for plant phenotyping described in Chapter 3, which are used to collect the data adopted throughout this thesis. In Section A.1 we describe our proposed solution based on the Raspberry Pi, *Rpi*. Next, in Section A.2 we describe the implementation of the solution based on a commercial camera, *Canon*, that was first proposed in [306].

A.1 *Rpi* sensing solution

In this section we provide further information on the *Rpi* solution and detailed instructions to setup and operate the system. Figure A.1 illustrates the system at first configuration time. In the following paragraphs, first we describe the RaspiCam image sensor and the web-based interface that we developed to operate the sensor; next, we describe how to setup the Raspberry Pi device.

Imaging sensor. The "RaspiCam" camera module is a fixed-focus 5 megapixel CMOS image sensor produced by OmniVision Technologies.[1]

[1] `http://www.ovt.com`

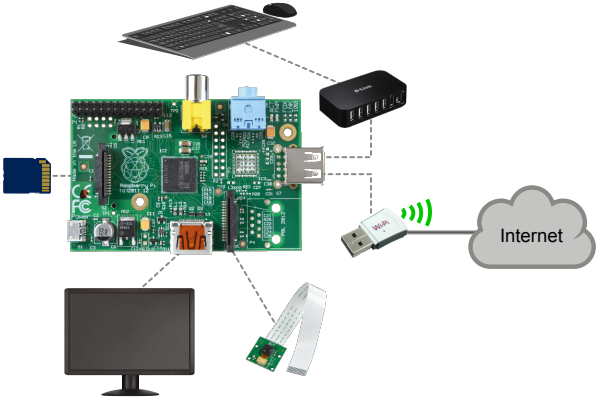


Figure A.1: Schematic of our sensing solution based on the Raspberry Pi, showing system setup and configuration. Note that after the initial setup, the system does not require input/output peripherals, and can be operated remotely (cf. Figure 3.2 on page 37).

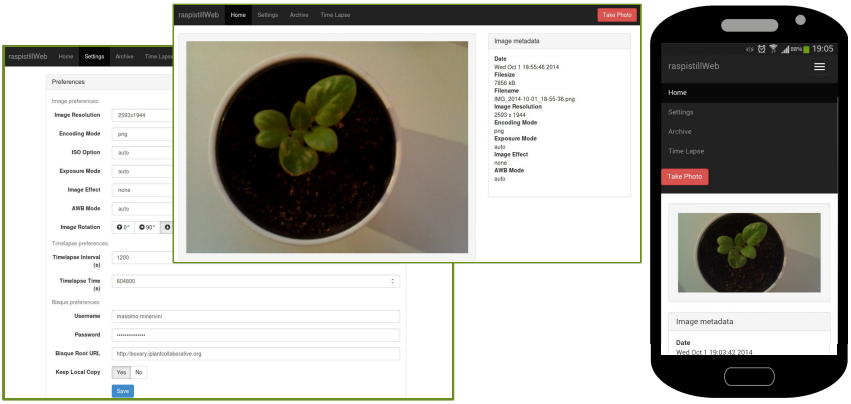
Two command line utilities are available on the Raspberry Pi to operate the camera: `raspistill`, to capture still photos, and `raspivid`, to record HD video. For example, to acquire a picture and save it in the PNG format [315] we use the following command line options:

`raspistill -n -e png -awb fluorescent -rot 180 -o` *filename*

The `raspistill` utility is easy to use and offers several options[2] to configure image acquisition. Using a job scheduler (e.g., the Cron software utility), it is possible to run the `raspistill` periodically, to acquire a time-lapse sequence of images. However, any changes to the mode of operation require that the Raspberry Pi be connected to all input/output peripherals to provide the user with physical access. This procedure involves attaching and detaching cables, which may cause undesired displacements of the sensing device or the plants. Furthermore, in some scenarios, physical access to the device may not be possible or desirable after the initial setup, thus introducing the need for a solution that enables remote control.

The Raspian runs by default an SSH (secure shell) server, allowing remote access to the command line of the Raspberry Pi using the following

[2] `http://www.raspberrypi.org/documentation/raspbian/applications/camera.md`



(a) Desktop interface (b) Mobile interface

Figure A.2: Screen captures of our web-based interface to operate the Raspberry Pi camera, based on the raspistillWeb project.

command:

```
ssh -p 22 pi@<IP>
```

where `<IP>` is the local (or remote) IP address of the Raspberry Pi (executing the command `hostname -I` will display the IP address assigned to the Raspberry Pi, e.g., `192.168.1.4`). However, operating the camera module from the command line may reveal problematic for some users. Therefore, we devise a web-based interface to configure and operate the camera module of the Raspberry Pi easily from another computer (e.g., a laptop or even a smartphone, cf. Figure A.2).

We implement our interface[3] as a fork of the *raspistillWeb*[4] project (version 0.1), i.e. a web interface for the `raspistill` tool, implemented using the Python programming language and the Pyramid[5] web framework. We adapt the original software platform to the requirements of our application, adding the following key features:

(a) the user can select among different image file formats, including lossy (JPEG [146], GIF [63]) and lossless (BMP [222], PNG [315]) coding standards;

[3] Available for download at: `http://www.phenotiki.com`
[4] `https://github.com/TimJuni/raspistillWeb`
[5] `http://www.pylonsproject.org`

(b) the user can start and interrupt time-lapse image acquisitions;

(c) more detailed information and metadata are displayed about acquired images; and

(d) acquired images can be transmitted to the iPlant Collaborative [111] cloud infrastructure for storage and analysis (cf. Chapter 4 for a discussion of the Bisque modules developed for the analysis).

As shown in Figure A.2, graphical control elements of the user interface are intuitive and self-explanatory, thus rendering the web application easy-to-use. The 'Settings' page allows to configure parameters regarding image acquisition, time-lapse photography, and transmission to the iPlant. 'Home' and 'Time Lapse' pages allow to capture single still images and initiate a time-lapse acquisition, respectively. Detailed information about acquired images is displayed in 'Home' and 'Archive' pages. In the 'Archive' page the user can browse previously acquired images and download time-lapse sequences as compressed archives. Our interface also adapts to small screens and can be displayed on mobile devices such as smartphones and tablets (cf. Figure A.2).

Acquired images are transmitted to the iPlant using the Bisque Python APIs and the user's credentials (username and password) for iPlant. The user can also decide to delete the local copy of the image files after their transmission, to save storage space on the Raspberry Pi.

Setting up the Raspberry Pi. To implement our imaging sensor based on the Raspberry Pi we use the following hardware equipment:

- Raspberry Pi Model B;
- "RaspiCam" camera module;
- USB Micro power supply;
- 8 GB Secure Digital (SD) memory card;
- HDMI monitor and cable;
- USB keyboard and mouse;
- USB wireless dongle;
- self-powered USB hub.

Setting up the Raspberry Pi requires only few simple steps, described in detail in the official documentation.[6] In order to install the necessary software and perform initial configurations, the device must be attached

[6] `http://www.raspberrypi.org/documentation/setup/`

to input/output peripherals (monitor, keyboard, and mouse), as shown in Figure A.1. To connect the camera module to the Raspberry Pi, we proceed as shown in the instructional video available at `http://www.youtube.com/watch?v=GImeVqHQzsE`. Subsequently, the Raspberry Pi can be started headless and operated remotely from a computer connected to the same local network (cf. Figure 3.2 on page 37).

Next, we download and install the latest version of the Raspbian operating system.[7] We extract the system image from the ZIP archive and install it on the SD card (also used for local storage), following the instructions in the official documentation.[8] For example, on Linux, we run the following command with superuser privileges:

```
dd bs=4M if=2014-06-20-wheezy-raspbian.img of=/dev/sdc
```

to install the Raspbian image `2014-06-20-wheezy-raspbian.img` on the device named `/dev/sdc` associated with the SD card.

On first booting (default user name and password are, respectively, 'pi' and 'raspberry'), the Raspberry Pi configuration tool `raspi-config` is automatically started, allowing system-level configuration tasks. In particular, we edit: 'Internationalization Options', to change locale, timezone, and keyboard layout; 'Enable Camera', to enable the camera module; and 'Enable Boot to Desktop/Scratch', to enable automatic booting into a desktop environment (this option should be disabled after all configurations are done, to reduce overhead on the device when it is started headless).

With the USB wireless adapter plugged into the Raspberry Pi, we configure access to a local Wi-Fi connection using the WiFi Config tool available in the Raspbian (alternatively, wired connection is possible using an Ethernet network cable). Once the connection is established, we run in a terminal:

```
sudo rpi-update
sudo apt-get update
sudo apt-get -y dist-upgrade
```

to upgrade the device firmware and all installed packages to the latest available version (this operation may take some time, depending on network speed). Then, we install the software dependencies necessary for our operations:

```
sudo apt-get install python2.7-dev python-virtualenv \
```

[7] Available at: `http://downloads.raspberrypi.org/raspbian_latest`

[8] `http://www.raspberrypi.org/documentation/installation/`

```
        python-setuptools python-imaging python-picamera \
        python-lxml python-bs4
sudo pip install requests==2.4.1
```

We build and install the Python APIs for Bisque, with the following commands:

```
hg clone http://biodev.ece.ucsb.edu/hg/bisque/
cd bisque/bqapi/
python setup.py build_py
cp -r bqapi/RequestsMonkeyPatch/ build/lib.linux-armv6l-2.7/bqapi/
sudo python setup.py install
```

Our fork of the raspistillWeb software can be installed with the following series of commands:

```
mkdir ~/Development
cd ~/Development
virtualenv env
cd env
wget http://.../raspistillWeb-PHIDIAS.tar.gz
tar -xvf raspistillWeb-PHIDIAS.tar.gz
```

To start the raspistillWeb service (by default listening on port number 6543) we execute on the Raspberry Pi:

```
cd Development/env/raspistillWeb
../bin/pserve development.ini
```

Then, from a web browser on a computer connected to the same local network, we access the raspistillWeb interface at the address `http://192.168.1.4:6543`. Note that by properly configuring the router of the local network, it is possible to enable remote access to the raspistillWeb via the Internet. In order to start the raspistillWeb automatically on system boot, we edit the `/etc/rc.local` file, adding the following lines:

```
cd ~/Development/env/raspistillWeb
../bin/pserve development.ini
```

before the line `exit 0` at the bottom of the file.

A.2 *Canon* sensing solution

In this section we provide overview information to: (a) configure the wireless network of the Eye-Fi memory card, and (b) obtain and install

Figure A.3: Canon camera, model PowerShot SD1000, with labels indicating buttons used to operate the CHDK firmware.

the CHDK on a Canon camera. Figure A.3 shows the layout of the camera model employed here.

Configuring the Eye-Fi. Normal mode of operation of the Eye-Fi involves transmission of image files from the camera to the user's PC via the Eye-Fi servers. Alternatively, with the Eye-Fi Pro X2 card, it is possible to setup a wireless ad-hoc network between the card and the PC, following the instructions provided by the vendor.[9] This mode of operation is termed *Direct Mode* and allows to directly transfer photos from the camera to the PC as they become available. To enable Direct Mode we connect the Eye-Fi card to a computer and using the Eyefi Center software we obtain SSID and WPA2 password necessary to connect to the wireless network established by the Eye-Fi card. We use a computer equipped with GNU/Linux operating system as a server to receive the image files acquired and transmitted by the camera. The Eye-Fi natively only supports Windows operating systems, therefore we adopt the EyeFiServer,[10] an open source Eye-Fi server written in Python. We configure the EyeFiServer to listen on a port for connections from the Eye-Fi card, and to execute a script to develop the incoming raw camera images into RGB.

Installing the CHDK. The correct CHDK build must be first identified, according to camera model and firmware version. Advanced firmware information can be displayed on the camera, such as:

```
Canon PowerShot SD1000
```

[9] `https://x2help.eyefi.com/hc/en-us/articles/200143167-Set-up-Direct-Mode-for-Your-Computer`

[10] `https://github.com/tachang/EyeFiServer`

```
P-ID:314F NT V
Firmware Ver GM1.01B
No Error
Mar 14 2007  09:48:23
```

We download a CHDK build[11] suitable for our PowerShot SD1000 camera with firmware version 1.01B, and install it on the Eye-Fi memory card (notice that no permanent changes are made to the camera). From the CHDK menu we enable the camera to save raw images, while we devise a Lua script (also installed on the Eye-Fi) for the acquisition of time-lapse images with preset parameters. After each acquisition, both raw and JPEG pictures are transmitted by the Eye-Fi to the receiving computer, where the raw images are automatically developed into TIFF format.

[11] http://mighty-hoernsche.de

Bibliography

[1] A. Abadpour and S. Kasaei, "Color PCA eigenimages and their application to compression and watermarking," *Image and Vision Computing*, vol. 26, no. 7, pp. 878–890, 2008.

[2] M. M. Abdelsamea and S. A. Tsaftaris, "Active contour model driven by globally signed region pressure force," in *International Conference on Digital Signal Processing*, 2013, pp. 1–6.

[3] G. J. Agin, "Computer vision systems for industrial inspection and assembly," *Computer*, vol. 13, no. 5, pp. 11–20, 1980.

[4] E. E. Aksoy, A. Abramov, F. Wörgötter, H. Scharr, A. Fischbach, and B. Dellen, "Modeling leaf growth of rosette plants using infrared stereo image sequences," *Computers and Electronics in Agriculture*, vol. 110, pp. 78–90, 2015.

[5] G. Alenya, B. Dellen, S. Foix, and C. Torras, "Robotized plant probing: Leaf segmentation utilizing time-of-flight data," *IEEE Robotics & Automation Magazine*, vol. 20, no. 3, pp. 50–59, 2013.

[6] G. Alenya, B. Dellen, and C. Torras, "3D modelling of leaves from color and ToF data for robotized plant measuring," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3408–3414.

[7] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley, 2003.

[8] P. Andrade-Sanchez, M. A. Gore, J. T. Heun, K. R. Thorp, A. E. Carmo-Silva, A. N. French, M. E. Salvucci, and J. W. White, "Development and evaluation of a field-based high-throughput phenotyping platform," *Functional Plant Biology*, vol. 41, no. 1, pp. 68–79, 2013.

[9] M. Andrecut, "Parallel GPU implementation of iterative PCA algorithms," *Journal of Computational Biology*, vol. 16, no. 11, pp. 1593–1599, 2009.

[10] C. Andrews, "Easy as Pi," *Engineering Technology*, vol. 8, no. 3, pp. 34–37, 2013.

[11] M. Andrle, L. Rebollo-Neira, and E. Sagianos, "Backward-optimized orthogonal matching pursuit approach," *IEEE Signal Processing Letters*, vol. 11, no. 9, pp. 705–708, 2004.

[12] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[13] A. Arora, A. Gupta, N. Bagmar, S. Mishra, and A. Bhattacharya, "A plant identification system using shape and morphological features on segmented leaflets: Team IITK, CLEF 2012," in *CLEF (Online Working Notes/Labs/Workshop)*, Sep. 2012.

[14] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect partially overlapping instances," in *Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3230–3237.

[15] S. Arvidsson, P. Pérez-Rodríguez, and B. Mueller-Roeber, "A growth phenotyping pipeline for Arabidopsis thaliana integrating image analysis and rosette area modeling for robust quantification of genotype effects," *The New Phytologist*, vol. 191, no. 3, pp. 895–907, 2011.

[16] J. Askelöf, M. L. Carlander, and C. Christopoulos, "Region of interest coding in JPEG 2000," *Signal Processing: Image Communication*, vol. 17, no. 1, pp. 105–111, 2002.

[17] S. Atwell, Y. S. Huang, B. J. Vilhjalmsson, G. Willems, M. Horton, Y. Li, D. Meng, A. Platt, A. M. Tarone, T. T. Hu, R. Jiang, N. W. Muliyati, X. Zhang, M. A. Amer, I. Baxter, B. Brachi, J. Chory, C. Dean, M. Debieu, J. de Meaux, J. R. Ecker, N. Faure, J. M. Kniskern, J. D. G. Jones, T. Michael, A. Nemri, F. Roux, D. E. Salt, C. Tang, M. Todesco, M. B. Traw, D. Weigel, P. Marjoram, J. O. Borevitz, J. Bergelson, and M. Nordborg, "Genome-wide association study of 107 phenotypes in Arabidopsis thaliana inbred lines," *Nature*, vol. 465, no. 7298, pp. 627–631, 2010.

[18] M. Augustin, Y. Haxhimusa, W. Busch, and W. G. Kropatsch, "Image-based phenotyping of the mature arabidopsis shoot system," in *Computer Vision – ECCV 2015*, 2015, vol. 8928, pp. 231–246.

[19] K. O. Babalola, B. Patenaude, P. Aljabar, J. Schnabel, D. Kennedy, W. Crum, S. Smith, T. F. Cootes, M. Jenkinson, and D. Rueckert, "Comparison and evaluation of segmentation techniques for subcortical structures in brain MRI," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 5241, 2008, pp. 409–416.

[20] J. Bankoski, P. Wilkins, and Y. Xu, "Technical overview of VP8, an open source video codec for the web," in *International Conference on Multimedia and Expo*, 2011, pp. 1–6.

[21] L. Bar, T. F. Chan, G. Chung, M. Jung, N. Kiryati, R. Mohieddine, N. Sochen, and L. A. Vese, "Mumford and Shah model and its applications to image segmentation and image restoration," in *Handbook of Mathematical Methods in Imaging*, 2011, pp. 1095–1157.

[22] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a Mahalanobis metric from equivalence constraints," *Journal of Machine Learning Research*, vol. 6, pp. 937–965, Dec. 2005.

[23] O. Barinova, V. S. Lempitsky, and P. Kohli, "On detection of multiple object instances using Hough transforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1773–1784, 2012.

[24] D. Basch, G. G. Hartmann, A. Ricci, and J. Siemer, "Harvesting opportunities for a sustainable food supply," RobecoSAM, 2012.

[25] P. N. Benfey and T. Mitchell-Olds, "From genotype to phenotype: Systems biology meets natural variation," *Science*, vol. 320, no. 5875, pp. 495–497, 2008.

[26] B. Berger, B. Parent, and M. Tester, "High-throughput shoot imaging to study drought responses," *Journal of Experimental Botany*, vol. 61, no. 13, pp. 3519–3528, 2010.

[27] S. Bergsträsser, D. Fanourakis, S. Schmittgen, M. Cendrero-Mateo, M. Jansen, H. Scharr, and U. Rascher, "HyperART: non-invasive quantification of leaf traits using hyper-

spectral absorption-reflectance-transmittance imaging," *Plant Methods*, vol. 11, no. 1, 2015.

[28] K. Billiau, H. Sprenger, C. Schudoma, D. Walther, and K. I. Köhl, "Data management pipeline for plant phenotyping in a multisite project," *Functional Plant Biology*, vol. 39, no. 11, pp. 948–957, 2012.

[29] B. Biskup, H. Scharr, U. Schurr, and U. Rascher, "A stereo imaging system for measuring structural parameters of plant canopies," *Plant, Cell and Environment*, vol. 30, pp. 1299–1308, 2007.

[30] B. R. Bochner, "New technologies to assess genotype-phenotype relationships," *Nature Reviews Genetics*, vol. 4, no. 4, pp. 309–314, 2003.

[31] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *5th Annual Workshop on Computational Learning Theory (COLT)*, 1992, pp. 144–152.

[32] L. E. Boucheron, N. R. Harvey, and B. S. Manjunath, "A quantitative object-level metric for segmentation performance and its application to cell nuclei," in *Advances in Visual Computing*, 2007, vol. 4841, pp. 208–219.

[33] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[34] C. Briese, T. Bodewein, and F. Fiorani, "Plant shoot image segmentation using colour features and support vector machines," in *International Workshop on Image Analysis Methods for the Plant Sciences*, 2013.

[35] T. Brown, C. Zimmermann, W. Panneton, N. Noah, and J. Borevitz, "High-resolution, time-lapse imaging for ecosystem-scale phenotyping in the field," in *High-Throughput Phenotyping in Plants*, 2012, vol. 918, pp. 71–96.

[36] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[37] J. Bruinsma, "The resource outlook to 2050: By how much do land, water and crop yields need to increase by 2050?" in *FAO Expert Meeting on 'How to feed the world in 2050'*, 2009.

[38] A. Bucksch, J. Burridge, L. M. York, A. Das, E. Nord, J. S. Weitz, and J. P. Lynch, "Image-based high-throughput field phenotyping of crop roots," *Plant Physiology*, 2014.

[39] E. Candes, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[40] D. Casanova, J. B. Florindo, W. N. Gonçalves, and O. M. Bruno, "IFSC/USP at ImageCLEF 2012: Plant identification task," in *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[41] M. E. Celebi, H. A. Kingravi, and F. Celiker, "Fast colour space transformations using minimax approximations," *IET Image Processing*, vol. 4, no. 2, pp. 70–80, Apr. 2010.

[42] F. Cen, Q. Lu, and W. Xu, "SSIM based rate-distortion optimization for intra-only coding in HEVC," in *International Conference on Consumer Electronics*, 2014, pp. 17–18.

[43] G. Cerutti, V. Antoine, L. Tougne, J. Mille, L. Valet, D. Coquin, and A. Vacavant, "ReVeS Participation – Tree species classification using random forests and botanical features," in *Conference and Labs of the Evaluation Forum*, 2012.

[44] G. Cerutti, L. Tougne, A. Vacavant, and D. Coquin, "A parametric active polygon for leaf segmentation and shape estimation," in *Advances in Visual Computing*, 2011, vol. 6938, pp. 202–213.

[45] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
[46] T. F. Chan, B. Y. Sandberg, and L. A. Vese, "Active contours without edges for vector-valued images," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 130–141, 2000.
[47] J. Chao, R. Huitl, E. Steinbach, and D. Schroeder, "A novel rate control framework for SIFT/SURF feature preservation in H.264/AVC video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, 2014.
[48] J. Chao, H. Chen, and E. Steinbach, "On the design of a novel JPEG quantization table for improved feature detection performance," in *International Conference on Image Processing*, 2013, pp. 1675–1679.
[49] J. Chao and E. Steinbach, "Preserving SIFT features in JPEG-encoded images," in *International Conference on Image Processing*, 2011, pp. 301–304.
[50] ——, "SIFT feature-preserving bit allocation for H.264/AVC video compression," in *International Conference on Image Processing*, 2012, pp. 709–712.
[51] A. Chavarría-Krauser, W. Jäger, and U. Schurr, "Primary root growth: a biophysical model of auxin-related control," *Functional Plant Biology*, vol. 32, no. 9, pp. 849–862, 2005.
[52] A. Chavarría-Krauser, K. A. Nagel, K. Palme, U. Schurr, A. Walter, and H. Scharr, "Spatio-temporal quantification of differential growth processes in root growth zones based on a novel combination of image sequence processing and refined concepts describing curvature production," *New Phytologist*, vol. 177, no. 3, pp. 811–821, 2008.
[53] S. Chen and R. J. Radke, "Level set segmentation with both shape and intensity priors," in *International Conference on Computer Vision*, 2009, pp. 763–770.
[54] Y. Chen, P. Hao, and A. Dang, "Optimal transform in perceptually uniform color space and its application in image coding," in *International Conference on Image Analysis and Recognition*, 2004, pp. 269–276.
[55] Z. Chen and C. Guillemot, "Perceptually-friendly H.264/AVC video coding based on foveated just-noticeable-distortion model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 806–819, 2010.
[56] Y. Chéné, D. Rousseau, P. Lucidarme, J. Bertheloot, V. Caffier, P. Morel, E. Belin, and F. Chapeau-Blondeau, "On the use of depth camera for 3D phenotyping of entire plants," *Computers and Electronics in Agriculture*, vol. 82, pp. 122–127, 2012.
[57] M. Cho, Y. M. Shin, and K.-M. Lee, "Unsupervised detection and segmentation of identical objects," in *Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1617–1624.
[58] C. Christopoulos, J. Askelöf, and M. Larsson, "Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard," *IEEE Signal Processing Letters*, vol. 7, no. 9, pp. 247–249, 2000.
[59] C. Clausen and H. Wechsler, "Color image compression using PCA and backpropagation learning," *Pattern Recognition*, vol. 33, no. 9, pp. 1555–1560, 2000.
[60] P. Clauw, F. Coppens, K. De Beuf, S. Dhondt, T. Van Daele, K. Maleux, V. Storme, L. Clement, N. Gonzalez, and D. Inze, "Leaf responses to mild drought stress in natural variants of Arabidopsis thaliana," *Plant Physiology*, 2015.
[61] A. Clément and B. Vigouroux, "Unsupervised segmentation of scenes containing vegetation (Forsythia) and soil by hierarchical analysis of bi-dimensional histograms," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1951–1957, 2003.
[62] J. Cobb, G. DeClerck, A. Greenberg, R. Clark, and S. McCouch, "Next-generation phenotyping: requirements and strategies for enhancing our understanding of genotype-

phenotype relationships and its relevance to crop improvement," *Theoretical and Applied Genetics*, vol. 126, no. 4, pp. 867–887, 2013.
[63] CompuServe Inc., "Graphics interchange format (GIF)," Jul. 1990, version 89a. [Online]. Available: http://www.w3.org/Graphics/GIF/spec-gif89a.txt
[64] C. Connolly and T. Fleiss, "A study of efficiency and accuracy in the transformation from RGB to CIELAB color space," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 1046–1048, 1997.
[65] D. Cremers, S. J. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk," in *Pattern Recognition*, vol. 3175, 2004, pp. 36–44.
[66] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr, "Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 295–313, 2002.
[67] F. Cuomo, E. Mibuari, K. Weldemariam, and O. Stewart, "Leveraging Raspberry Pi for interactive education," in *Annual Symposium on Computing for Development*, ser. ACM DEV-4 '13, 2013.
[68] H. Cwiek, C. Pommier, U. Scholz, M. Lange, J. Chen, A. D. van Dijk, J.-P. Nap, J. van Oeveren, P. J. Kersey, and P. Krajewski, "On minimum reporting requirements and standard formatting of plant phenotypic data," in *International Plant & Animal Genome XXII Conference*, 2014.
[69] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *International Conference on Machine Learning*, 2007, pp. 209–216.
[70] J. De Vylder, D. Ochoa, W. Philips, L. Chaerle, and D. Van Der Straeten, "Leaf segmentation and tracking using probabilistic parametric active contours," in *International Conference on Computer Vision/Computer Graphics Collaboration Techniques*, ser. MIRAGE 2011, 2011, pp. 75–85.
[71] J. De Vylder, F. J. Vandenbussche, Y. Hu, W. Philips, and D. Van Der Straeten, "Rosette Tracker: an open source image analysis tool for automatic quantification of genotype effects," *Plant Physiology*, vol. 160, no. 3, pp. 1149–1159, 2012.
[72] A. Declercq and J. H. Piater, "Online learning of Gaussian mixture models: a two-level approach," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2008, pp. 605–611.
[73] B. Dellen, G. Alenyà, S. Foix, and C. Torras, "Segmenting color images into surface patches by exploiting sparse depth data," in *WACV*, 2011, pp. 591–598.
[74] B. Dellen, H. Scharr, and C. Torras, "Growth signatures of rosette plants from time-lapse video," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. PP, no. 99, 2015.
[75] S. Dhondt, N. Gonzalez, J. Blomme, L. De Milde, T. Van Daele, D. Van Akoleyen, V. Storme, F. Coppens, G. T.S. Beemster, and D. Inzé, "High-resolution time-resolved imaging of in vitro arabidopsis rosette growth," *The Plant Journal*, vol. 80, no. 1, pp. 172–184, 2014.
[76] S. Dhondt, D. Van Haerenborgh, C. Van Cauwenbergh, R. M. Merks, W. Philips, G. T. Beemster, and D. Inzé, "Quantitative analysis of venation patterns of arabidopsis leaves by supervised image analysis," *The Plant Journal*, vol. 69, no. 3, pp. 553–563, 2012.
[77] S. Dhondt, N. Wuyts, and D. Inzé, "Cell to whole-plant phenotyping: the best is yet to come," *Trends in Plant Science*, vol. 18, no. 8, pp. 428–439, 2013.
[78] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[79] S. Dikbas, T. Arici, and Y. Altunbasak, "Chrominance edge preserving grayscale transformation with approximate first principal component for color edge detection," in *International Conference on Image Processing*, 2007, pp. 261–264.
[80] B. M. Dogahe and M. N. Murthi, "Quantization for classification accuracy in high-rate quantizers," in *Digital Signal Processing Workshop*, 2011, pp. 277–282.
[81] Y. Dong, S. Chang, and L. Carin, "Rate-distortion bound for joint compression and classification with application to multiaspect scattering," *IEEE Sensors Journal*, vol. 5, no. 3, pp. 481–492, 2005.
[82] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, 1996, pp. 155–161.
[83] Q. Du and J. E. Fowler, "Hyperspectral image compression using JPEG2000 and principal component analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 201–205, 2007.
[84] ——, "Low-complexity principal component analysis for hyperspectral image compression," *International Journal of High Performance Computing Applications*, vol. 22, no. 4, pp. 438–448, 2008.
[85] M. P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," in *12th IAPR International Conference on Pattern Recognition*, 1994, pp. 566–568.
[86] M. Eberius, *Arabidopsis Morphological Phenotype Assessment*, LemnaTec GmbH, Aachen, Germany, available at: http://www.lemnatec.com/fileadmin/user_upload/Literature/PlantGrowth/06-Arabidopssis-phenotype-assessment.pdf.
[87] D. Edwards and J. Batley, "Plant bioinformatics: from genome to phenome," *Trends in Biotechnology*, vol. 22, no. 5, pp. 232–237, 2004.
[88] ——, "Plant genome sequencing: applications for crop improvement," *Plant Biotechnology Journal*, vol. 8, no. 1, pp. 2–9, 2010.
[89] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, no. 3, pp. 399–417, Jun. 1963.
[90] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[91] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.
[92] P. Favaro and S. Soatto, *3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*. Springer-Verlag, 2007.
[93] P. F. Felzenszwalb, "Representation and detection of deformable shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 208–220, 2005.
[94] L. Fiaschi, K. Gregor, B. Afonso, M. Zlatic, and F. A. Hamprecht, "Keeping count: Leveraging temporal context to count heavily overlapping objects," in *International Symposium on Biomedical Imaging*, 2013, pp. 656–659.
[95] L. Fiaschi, U. Köthe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in *International Conference on Pattern Recognition*, 2012, pp. 2685–2688.
[96] E. Finkel, "With 'phenomics,' plant scientists hope to shift breeding into overdrive," *Science*, vol. 325, no. 5939, pp. 380–381, 2009.
[97] F. Fiorani and U. Schurr, "Future scenarios for plant phenotyping," *Annual Review of Plant Biology*, vol. 64, no. 1, pp. 267–291, 2013.

[98] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
[99] ——, "Some remarks on the methods formulated in a recent article on 'The quantitative analysis of plant growth'," *Annals of Applied Biology*, vol. 7, no. 4, pp. 367–372, 1921.
[100] M. Fleury, A. C. Downton, and A. F. Clark, "Karhünen-Loève transform: An exercise in simple image-processing parallel pipelines," in *Euro-Par '97 Parallel Processing*, 1997, vol. 1300, pp. 815–819.
[101] D. H. Foley and J. W. Sammon, "An optimal set of discriminant vectors," *IEEE Transactions on Computers*, vol. C-24, no. 3, pp. 281–289, 1975.
[102] A. French, S. Ubeda-Tomás, T. J. Holman, M. J. Bennett, and T. Pridmore, "High-throughput quantification of root growth using a novel image-analysis tool," *Plant Physiology*, vol. 150, no. 4, pp. 1784–1795, 2009.
[103] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic Press Professional, 1990.
[104] R. T. Furbank and M. Tester, "Phenomics – Technologies to relieve the phenotyping bottleneck," *Trends in Plant Science*, vol. 16, no. 12, pp. 635–644, 2011.
[105] G. Gerig, M. Jomier, and M. Chakos, "Valmet: A new validation tool for assessing and improving 3D object segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2001, vol. 2208, pp. 516–523.
[106] E. Gershikov and M. Porat, "Does decorrelation really improve color image compression?" in *International Conference on Systems Theory and Scientific Computation*, 2005, pp. 306–309.
[107] A. R. Gillespie, A. B. Kahle, and R. E. Walker, "Color enhancement of highly correlated images. I. Decorrelation and HSI contrast stretches," *Remote Sensing of Environment*, vol. 20, no. 3, pp. 209–235, 1986.
[108] ——, "Color enhancement of highly correlated images. II. Channel ratio and "chromaticity" transformation techniques," *Remote Sensing of Environment*, vol. 22, no. 3, pp. 343–365, 1987.
[109] D. Giordano, I. Kavasidis, S. Palazzo, and C. Spampinato, "Nonparametric label propagation using mutual local similarity in nearest neighbors," *Computer Vision and Image Understanding*, vol. 131, pp. 116–127, 2015.
[110] H. Goëau, P. Bonnet, A. Joly, I. Yahiaoui, D. Barthélémy, B. Nozha, and J.-F. Molino, "The ImageCLEF 2012 plant identification task," in *CLEF*, 2012.
[111] S. A. Goff, M. Vaughn, S. McKay, E. Lyons, A. E. Stapleton, D. Gessler, N. Matasci, L. Wang, M. Hanlon, A. Lenards, A. Muir, N. Merchant, S. Lowry, S. Mock, M. Helmke, A. Kubach, M. Narro, N. Hopkins, D. Micklos, U. Hilgert, M. Gonzales, C. Jordan, E. Skidmore, R. Dooley, J. Cazes, R. McLay, Z. Lu, S. Pasternak, L. Koesterke, W. H. Piel, R. Grene, C. Noutsos, K. Gendler, X. Feng, C. Tang, M. Lent, S.-J. Kim, K. Kvilekval, B. S. Manjunath, V. Tannen, A. Stamatakis, M. Sanderson, S. M. Welch, K. A. Cranston, P. Soltis, D. Soltis, B. O'Meara, C. Ane, T. Brutnell, D. J. Kleibenstein, J. W. White, J. Leebens-Mack, M. J. Donoghue, E. P. Spalding, T. J. Vision, C. R. Myers, D. Lowenthal, B. J. Enquist, B. Boyle, A. Akoglu, G. Andrews, S. Ram, D. Ware, L. Stein, and D. Stanzione, "The iPlant collaborative: Cyberinfrastructure for plant biology," *Frontiers in Plant Science*, vol. 2, no. 34, 2011.
[112] M. R. Golzarian, M. K. Lee, and J. M. A. Desbiolles, "Evaluation of color indices for improved segmentation of plant images," *Transactions of the American Society of Agricultural and Biological Engineers*, vol. 55, no. 1, pp. 261–273, 2012.

[113] M. Golzarian, R. Frick, K. Rajendran, B. Berger, S. Roy, M. Tester, and D. Lun, "Accurate inference of shoot biomass from high-throughput images of cereal plants," *Plant Methods*, vol. 7, no. 1, pp. 2+, 2011.

[114] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[115] C. Granier, L. Aguirrezabal, K. Chenu, S. J. Cookson, M. Dauzat, P. Hamard, J.-J. Thioux, G. Rolland, S. Bouchier-Combaud, A. Lebaudy, B. Muller, T. Simonneau, and F. Tardieu, "PHENOPSIS, an automated platform for reproducible phenotyping of plant responses to soil water deficit in Arabidopsis thaliana permitted the identification of an accession with low sensitivity to soil water deficit," *New Phytologist*, vol. 169, no. 3, pp. 623–635, 2006.

[116] C. Granier and D. Vile, "Phenotyping and beyond: modelling the relationships between traits," *Current Opinion in Plant Biology*, vol. 18, pp. 96–102, 2014.

[117] M. C. Grant and S. P. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, 2008, vol. 371, pp. 95–110.

[118] J. Green, H. Appel, E. M. Rehrig, J. Harnsomburana, J.-F. Chang, P. Balint-Kurti, and C.-R. Shyu, "Phenophyte: a flexible affordable method to quantify 2d phenotypes from imagery," *Plant Methods*, vol. 8, no. 1, p. 45, 2012.

[119] S. D. Gupta and Y. Ibaraki, Eds., *Plant Image Analysis: Fundamentals and Applications*. CRC Press, 2014.

[120] S.-E. Han, B. Tao, T. Cooper, and I. Tastl, "Comparison between different color transformations for the JPEG 2000," in *PICS 2000*, 2000, pp. 259–263.

[121] P. Hao and Q. Shi, "Comparative study of color transforms for image coding and derivation of integer reversible color transform," in *International Conference on Pattern Recognition*, vol. 3, 2000, pp. 224–227.

[122] ——, "Matrix factorizations for reversible integer mapping," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2314–2324, 2001.

[123] ——, "Reversible integer KLT for progressive-to-lossless compression of multiple component images," in *International Conference on Image Processing*, 2003, pp. 633–636.

[124] A. Hartmann, T. Czauderna, R. Hoffmann, N. Stein, and F. Schreiber, "HTPheno: An image analysis pipeline for high-throughput plant phenotyping," *BMC Bioinformatics*, vol. 12, no. 1, p. 148, 2011.

[125] S. Haug, A. Michaels, P. Biber, and J. Ostermann, "Plant classification system for crop/weed discrimination without segmentation," in *Winter Conference on Applications of Computer Vision*, 2014, pp. 1142–1149.

[126] X. He and S. Gould, "Multi-instance object segmentation with exemplars," in *International Conference on Computer Vision Workshops (ICCVW)*, 2013, pp. 1–4.

[127] ——, "An exemplar-based CRF for multi-instance object segmentation," in *Conference on Computer Vision and Pattern Recognition*, 2014, pp. 296–303.

[128] M. Heiler and C. Schnörr, "Natural image statistics for natural image segmentation," *International Journal of Computer Vision*, vol. 63, no. 1, pp. 5–19, 2005.

[129] M. Hernandez-Cabronero, V. Sanchez, M. W. Marcellin, and J. Serra-Sagrista, "A distortion metric for the lossy compression of DNA microarray images," in *Data Compression Conference*, 2013, pp. 171–180.

[130] N. J. Higham, *Functions of Matrices: Theory and Computation*. SIAM, 2008.

[131] C. Hoffa, G. Mehta, T. Freeman, E. Deelman, K. Keahey, B. Berriman, and J. Good, "On the use of cloud computing for scientific workflows," in *International Conference on eScience*, 2008, pp. 640–645.

[132] H. Holtorf, M.-C. Guitton, and R. Reski, "Plant functional genomics," *Die Naturwissenschaften*, vol. 89, no. 6, pp. 235–249, 2002.
[133] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America A*, vol. 5, no. 7, pp. 1127–1135, 1988.
[134] M. J. Hossain, Dewan, K. Ahn, and O. Chae, "A linear time algorithm of computing Hausdorff distance for content-based image analysis," *Circuits, Systems, and Signal Processing*, vol. 31, no. 1, pp. 389–399, 2012.
[135] D. Houle, D. R. Govindaraju, and S. Omholt, "Phenomics: the next challenge," *Nature Reviews Genetics*, vol. 11, no. 12, pp. 855–866, 2010.
[136] P. Howarth and S. Rüger, "Evaluation of texture features for content-based image retrieval," in *Image and Video Retrieval*, 2004, vol. 3115, pp. 326–334.
[137] Y.-H. Huang, T.-S. Ou, P.-Y. Su, and H. Chen, "Perceptual rate-distortion optimization using structural similarity index as quality metric," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1614–1624, 2010.
[138] E. R. Hunt, W. D. Hively, S. J. Fujikawa, D. S. Linden, C. S. T. Daughtry, and G. W. McCarty, "Acquisition of NIR-green-blue digital photographs from unmanned aircraft for crop monitoring," *Remote Sensing*, vol. 2, no. 1, pp. 290–305, 2010.
[139] R. Hunt, "Growth analysis, individual plants," in *Encyclopedia of Applied Plant Sciences*, 2003, pp. 579–588.
[140] ICL, "Intelligent computing laboratory plant leaf dataset," 2013.
[141] R. Ierusalimschy, L. H. de Figueiredo, and W. C. Filho, "Lua – an extensible extension language," *Software: Practice and Experience*, vol. 26, no. 6, pp. 635–652, 1996.
[142] A. Ilie and G. Welch, "Ensuring color consistency across multiple cameras," in *International Conference on Computer Vision*, vol. 2, 2005, pp. 1268–1275.
[143] International Commission on Illumination (CIE), "Colorimetry," CIE Publication, Tech. Rep. CIE 015:2004, 2004.
[144] ——, "CIE Colorimetry - Part 4: 1976 L*a*b* Colour Space," CIE Publication, Joint ISO/CIE Standard ISO 11664-4:2008(E)/CIE S 014-4/E:2007, 2007.
[145] *Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios*, International Telecommunication Union ITU-R Recommendation BT.601-7, 2011.
[146] *Information technology – Digital compression and coding of continuous-tone still images*, International Telecommunication Union (ITU) ITU-T Recommendation T.81, 1992.
[147] *Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios*, International Telecommunication Union (ITU) ITU-R Recommendation BT.601-5, 1995.
[148] *High efficiency video coding*, International Telecommunication Union (ITU) ITU-T Recommendation H.265, 2013.
[149] V. Jain, B. Bollmann, M. Richardson, D. R. Berger, M. N. Helmstaedter, K. L. Briggman, W. Denk, J. B. Bowden, J. M. Mendenhall, W. C. Abraham, K. M. Harris, N. Kasthuri, K. J. Hayworth, R. Schalek, J. C. Tapia, J. W. Lichtman, and H. S. Seung, "Boundary learning by optimization with topological constraints," in *Computer Vision and Pattern Recognition*, 2010, pp. 2488–2495.
[150] M. Jansen, F. Gilmer, B. Biskup, K. Nagel, U. Rascher, A. Fischbach, S. Briem, G. Dreissen, S. Tittmann, S. Braun, I. D. Jaeger, M. Metzlaff, U. Schurr, H. Scharr, and A. Walter, "Simultaneous phenotyping of leaf growth and chlorophyll fluorescence via GROWSCREEN FLUORO allows detection of stress tolerance in Arabidopsis thaliana and other rosette plants," *Functional Pant Biology*, vol. 36, no. 10/11, pp. 902–914, 2009.

[151] *Exchangeable image file format for digital still cameras: Exif Version 2.2*, Japan Electronics and Information Technology Industries Association (JEITA) Std. CP-3451, 2002.
[152] J. Jin and L. Tang, "Corn plant sensing using real-time stereo vision," *Journal of Field Robotics*, vol. 26, no. 6–7, pp. 591–608, 2009.
[153] Z. Jin, J.-Y. Yang, Z.-S. Hu, and Z. Lou, "Face recognition based on the uncorrelated discriminant transformation," *Pattern Recognition*, vol. 34, no. 7, pp. 1405–1416, 2001.
[154] I. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, 2002.
[155] G. A. Jones, N. Paragios, and C. S. Regazzoni, Eds., *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*. Kluwer Academic Publishers, 2002.
[156] R. Jošth, J. Antikainen, J. Havel, A. Herout, P. Zemčík, and M. Hauta-Kasari, "Real-time PCA calculation for spectral imaging (using SIMD and GP-GPU)," *Journal of Real-Time Image Processing*, vol. 7, no. 2, pp. 95–103, 2012.
[157] M. Karczewicz and X. Wang, "Intra frame rate control based on SATD," in *13th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC)*, no. JCTVC-M0257, Apr. 2013, pp. 1–5.
[158] ——, "Intra rate control for video encoding based on sum of absolute transformed difference," U.S. Patent Application 14/245,864, Apr., 2014.
[159] A. A. Kassim and W. S. Lee, "Embedded color image coding using SPIHT with partially linked spatial orientation trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 2, pp. 203–206, 2003.
[160] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato, "An innovative web-based collaborative platform for video annotation," *Multimedia Tools and Applications*, vol. 70, no. 1, pp. 413–432, 2014.
[161] D. B. Kell, "Genotype-phenotype mapping: genes as computer programs," *Trends in Genetics*, vol. 18, no. 11, pp. 555–559, 2002.
[162] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade, "Distributed cosegmentation via submodular optimization on anisotropic diffusion," in *International Conference on Computer Vision*, 2011, pp. 169–176.
[163] H. M. Kim, W.-S. Kim, and D.-S. Cho, "A new color transform for RGB coding," in *International Conference on Image Processing*, 2004, pp. 107–110.
[164] J. Knüfer, C. Briese, M. Jansen, S. Kleinen, A. Putz, S. Kumar, D. Stelling, U. Schurr, and A. Wiese-Klinkenberg, "A non-invasive imaging procedure to identify drought-stress tolerant oilseed rape plants expressing genes from high altitude plants," in *InterDrought-IV Conference*, 2013.
[165] J. Kokorian, G. Polder, J. J. B. Keurentjes, D. Vreugdenhil, and M. Olortegui Guzman, "An ImageJ based measurement setup for automated phenotyping of plants," in *Proceedings of the ImageJ User and Developer Conference*, 2010, pp. 178–182.
[166] R. K. Kouassi, J. C. Devaux, P. Gouton, and M. Paindavoine, "Application of the Karhunen-Loeve transform for natural color images analysis," in *Asilomar Conference on Signals, Systems, and Computers*, vol. 2, 1997, pp. 1740–1744.
[167] R. Kountchev and R. Kountcheva, "New method for adaptive Karhunen-Loeve color transform," in *International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services*, 2009, pp. 209–216.
[168] B. Kulis, "Metric learning: A survey," *Foundations and Trends in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2013.
[169] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[170] N. Kumar, P. Belhumeur, A. Biswas, D. Jacobs, W. Kress, I. Lopez, and J. Soares, "Leafsnap: A computer vision system for automatic plant species identification," in *Computer Vision – ECCV 2012*, 2012, pp. 502–516.
[171] C. P. Lee, W. Snyder, and C. Wang, "Supervised multispectral image segmentation using active contours," in *International Conference on Robotics and Automation*, 2005, pp. 4242–4247.
[172] J.-S. Lee and T. Ebrahimi, "Perceptual video compression: A survey," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 684–697, 2012.
[173] D. Leister, C. Varotto, P. Pesaresi, A. Niwergall, and F. Salamini, "Large-scale evaluation of plant growth in Arabidopsis thaliana by non-invasive image analysis," *Plant Physiology and Biochemistry*, vol. 37, no. 9, pp. 671–678, 1999.
[174] V. S. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010, pp. 1324–1332.
[175] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly Journal of Applied Mathmatics*, vol. II, no. 2, pp. 164–168, 1944.
[176] M. Leventon, O. Faugeras, E. Grimson, and W. Wells, "Level set based segmentation with intensity and curvature priors," in *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, 2000, pp. 4–11.
[177] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.
[178] B. Li, H. Li, L. Li, and J. Zhang, "Rate control by R-lambda model for HEVC," in *11th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC)*, no. JCTVC-K0103, Oct. 2012, pp. 1–11.
[179] ——, "λ domain rate control algorithm for High Efficiency Video Coding," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3841–3854, 2014.
[180] B. Li, D. Zhang, H. Li, and J. Xu, "QP determination by lambda value," in *9th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC)*, no. JCTVC-I0426, May 2012, pp. 1–10.
[181] J. Li, R. M. Gray, and R. Olshen, "Joint image compression and classification with vector quantization and a two dimensional hidden Markov model," in *Data Compression Conference*, 1999, pp. 23–32.
[182] L. Li, Q. Zhang, and D. Huang, "A review of imaging techniques for plant phenotyping," *Sensors*, vol. 14, no. 11, pp. 20 078–20 111, 2014.
[183] Z. Li, S. Qin, and L. Itti, "Visual attention guided bit allocation in video compression," *Image and Vision Computing*, vol. 29, no. 1, pp. 1–14, 2011.
[184] J. O. Limb, C. B. Rubinstein, and J. Thompson, "Digital coding of color video signals – a review," *IEEE Transactions on Communications*, vol. 25, no. 11, pp. 1349–1385, 1977.
[185] Y. Liu, Z. G. Li, and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 134–139, 2008.
[186] Z. Liu, L. J. Karam, and A. B. Watson, "JPEG2000 encoding with perceptual distortion control," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1763–1778, 2006.
[187] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, "Annotated high-throughput microscopy image sets for validation," *Nature Methods*, vol. 9, no. 7, pp. 637–637, 2012.
[188] G. Lobet, X. Draye, and C. Périlleux, "An online database for plant image analysis software tools," *Plant Methods*, vol. 9, no. 1, p. 38, 2013.
[189] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[190] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, no. 5, pp. 823–870, 2007.
[191] M. Majchrowicz, P. Kapusta, L. Was, and S. Wiak, "Application of general-purpose computing on graphics processing units for acceleration of basic linear algebra operations and principal components analysis method," in *Man-Machine Interactions 3*, 2014, vol. 242, pp. 519–527.
[192] H. S. Malvar, G. J. Sullivan, and S. Srinivasan, "Lifting-based reversible color transformations for image compression," in *Proceedings of SPIE, Applications of Digital Image Processing XXXI*, vol. 7073, 2008.
[193] J. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion of the encoding of images," *IEEE Transactions on Information Theory*, vol. 20, no. 4, pp. 525–536, 1974.
[194] L. A. C. Mansilla and P. A. V. Miranda, "Image segmentation by oriented image foresting transform: Handling ties and colored images," in *International Conference on Digital Signal Processing*, 2013, pp. 1–6.
[195] E. R. Mardis, "Next-generation DNA sequencing methods," *Annual review of genomics and human genetics*, vol. 9, no. 1, pp. 387–402, 2008.
[196] D. Marpe, H. Kirchhoffer, V. George, P. Kauff, and T. Wiegand, "Macroblock-adaptive residual color space transforms for 4:4:4 video coding," in *International Conference on Image Processing*, 2006, pp. 3157–3160.
[197] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
[198] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.
[199] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *International Conference on Computer Vision*, vol. 2, 2001, pp. 416–423.
[200] D. A. Matos, B. J. Cole, I. P. Whitney, K. J.-M. MacKinnon, S. A. Kay, and S. P. Hazen, "Daily changes in temperature, not the circadian clock, regulate growth rate in brachypodium distachyon," *PLoS One*, vol. 9, no. 6, p. e100072, 2014.
[201] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. Sullivan, "High Efficiency Video Coding (HEVC) test model 16 (HM 16) encoder description," in *18th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC)*, no. JCTVC-R1002, Jul. 2014, pp. 1–58.
[202] M. Meddeb, M. Cagnazzo, and B. Pesquet-Popescu, "Region-of-interest-based rate control scheme for high-efficiency video coding," *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.
[203] ——, "Region-of-interest based rate control scheme for high efficiency video coding," in *International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 7338–7342.
[204] M. Meilă, "Comparing clusterings by the variation of information," in *Computational Learning Theory*, vol. 2777, 2003, pp. 173–187.
[205] D. W. Meinke, J. M. Cherry, C. Dean, S. D. Rounsley, and M. Koornneef, "Arabidopsis thaliana: A model plant for genome analysis," *Science*, vol. 282, no. 5389, pp. 662–682, 1998.
[206] B. Méndez-Vigo, M. T. de Andrés, M. Ramiro, J. M. Martínez-Zapater, and C. Alonso-Blanco, "Temporal analysis of natural variation for the rate of leaf production and its

relationship with flowering initiation in Arabidopsis thaliana," *Journal of Experimental Botany*, vol. 61, no. 6, pp. 1611–1623, 2010.

[207] V. Mezaris, I. Kompatsiaris, and M. Strintzis, "Still image objective segmentation evaluation using ground truth," in *5th COST 276 Workshop*, 2003, pp. 9–14.

[208] J. L. Micol, "Leaf development: time to turn over a new leaf?" *Current Opinion in Plant Biology*, vol. 12, no. 1, pp. 9–16, 2009.

[209] M. Minervini, M. M. Abdelsamea, and S. A. Tsaftaris, "Image-based plant phenotyping with incremental learning and active contours," *Ecological Informatics*, vol. 23, pp. 35–48, Sep. 2014, special Issue on Multimedia in Ecology and Environment.

[210] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," *Pattern Recognition Letters*, 2015, under review.

[211] M. Minervini, C. Rusu, and S. A. Tsaftaris, "Learning computationally efficient approximations of complex image segmentation metrics," in *8th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Trieste, Italy, Sep. 2013, pp. 60–65.

[212] ——, "Unsupervised and supervised approaches to color space transformation for image coding," in *21st International Conference on Image Processing (ICIP)*, Paris, France, Oct. 2014.

[213] ——, "Computationally efficient data and application driven color transforms for the compression and enhancement of images and video," in *Color Image and Video Enhancement*. Springer, 2015, ch. 12.

[214] M. Minervini, H. Scharr, and S. A. Tsaftaris, "Image analysis: The new bottleneck in plant phenotyping," *IEEE Signal Processing Magazine*, vol. 32, no. 4, pp. xx–yy, 2015.

[215] ——, "The significance of image compression in plant phenotyping applications," *Functional Plant Biology*, 2015, under review.

[216] M. Minervini and S. A. Tsaftaris, "Application-aware image compression for low cost and distributed plant phenotyping," in *18th International Conference on Digital Signal Processing (DSP)*, Santorini, Greece, Jul. 2013, pp. 1–6.

[217] ——, "Classification-aware distortion metric for HEVC intra coding," in *International Conference on Visual Communications and Image Processing (VCIP)*, 2015, under review.

[218] S. J. Mooney, T. P. Pridmore, J. Helliwell, and M. J. Bennett, "Developing x-ray computed tomography to non-invasively image 3-D root systems architecture in soil," *Plant and Soil*, vol. 352, no. 1–2, pp. 1–22, 2012.

[219] D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu, and R. Bultje, "The latest open-source video codec VP9 – An overview and preliminary results," in *Picture Coding Symposium*, 2013, pp. 390–393.

[220] M. Müller-Linow, F. Pinto-Espinosa, H. Scharr, and U. Rascher, "The leaf angle distribution of natural plant populations: assessing the canopy with a novel software tool," *Plant Methods*, vol. 11, no. 1, p. 11, 2015.

[221] M. J. Murillo, J. A. Paco, and D. Wright, "Long-distance telecommunication in remote poor areas: From partnerships and implementation to sustainability," *IEEE Technology and Society Magazine*, vol. 34, no. 1, pp. 19–30, 2015.

[222] J. D. Murray and W. VanRyper, *Encyclopedia of graphics file formats*. O'Reilly, 1994.

[223] K. A. Nagel, A. Putz, F. Gilmer, K. Heinz, A. Fischbach, J. Pfeifer, M. Faget, S. Blossfeld, M. Ernst, C. Dimaki, B. Kastenholz, A.-K. Kleinert, A. Galinski, H. Scharr, F. Fiorani, and U. Schurr, "GROWSCREEN-Rhizo is a novel phenotyping robot enabling simultaneous measurements of root and shoot growth for plants grown in soil-filled rhizotrons," *Functional Plant Biology*, vol. 39, no. 11, pp. 891–904, 2012.

[224] K. R. Namuduri and V. N. Ramaswamy, "Feature preserving image compression," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2767–2776, 2003.
[225] T. H. Nawaz and A. Cavallaro, "A protocol for evaluating video trackers under real-world conditions," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1354–1361, 2013.
[226] C. Nieuwenhuis and D. Cremers, "Spatially varying color distributions for interactive multilabel segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1234–1247, 2013.
[227] C. Nieuwenhuis, E. Toeppe, and D. Cremers, "A survey and comparison of discrete and continuous multi-label optimization approaches for the Potts model," *International Journal of Computer Vision*, vol. 104, no. 3, pp. 223–240, 2013.
[228] M.-E. Nilsback and A. Zisserman, "Delving deeper into the whorl of flower segmentation," *Image and Vision Computing*, vol. 28, no. 6, pp. 1049–1062, 2010.
[229] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445–456, 2010.
[230] K. L. Oehler and R. M. Gray, "Combining image compression and classification using vector quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 461–473, 1995.
[231] R. C. O'Malley and J. R. Ecker, "Linking genotype to phenotype using the arabidopsis unimutant collection," *The Plant Journal*, vol. 61, no. 6, pp. 928–940, 2010.
[232] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.
[233] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
[234] M. Otte and H.-H. Nagel, "Optical flow estimation: advances and comparisons," in *European Conference on Computer Vision*, vol. I, 1994, pp. 51–60.
[235] G. Pajares and J. M. de la Cruz, "A wavelet-based image fusion tutorial," *Pattern Recognition*, vol. 37, no. 9, pp. 1855–1872, 2004.
[236] S. K. Pal, M. Liput, M. Piques, H. Ishihara, T. Obata, M. C. M. Martins, R. Sulpice, J. T. van Dongen, A. R. Fernie, U. P. Yadav, J. E. Lunn, B. Usadel, and M. Stitt, "Diurnal changes of polysome loading track sucrose content in the rosette of wildtype arabidopsis and the starchless pgm mutant," *Plant Physiology*, vol. 162, no. 3, pp. 1246–1265, 2013.
[237] H. Palus, "Representations of colour images in different colour spaces," in *The Colour Image Processing Handbook*. Springer, 1998, pp. 67–90.
[238] J. D. Paola and R. A. Schowengerdt, "The effect of lossy image compression on image classification," in *International Geoscience and Remote Sensing Symposium*, vol. 1, 1995, pp. 118–120.
[239] J.-M. Pape and C. Klukas, "3-D histogram-based segmentation and leaf detection for rosette plants," in *Computer Vision – ECCV 2014 Workshops*, 2015, vol. 8928, pp. 61–74.
[240] D.-C. Park and Y.-J. Woo, "Weighted centroid neural network for edge preserving image compression," *IEEE Transactions on Neural Networks*, vol. 12, no. 5, pp. 1134–1146, 2001.
[241] J. D. Peleman and J. R. R. van der Voort, "Breeding by design," *Trends in Plant Science*, vol. 8, no. 7, pp. 330–334, 2003.
[242] B. Penna, T. Tillo, E. Magli, and G. Olmo, "A new low complexity KLT for lossy hyperspectral data compression," in *International Geoscience and Remote Sensing Symposium*, vol. 7, 2006, pp. 3525–3528.

[243] G. A. Pereyra-Irujo, E. D. Gasco, L. S. Peirone, and L. A. N. Aguirrezábal, "GlyPh: a low-cost platform for phenotyping plant growth and water use," *Functional Plant Biology*, vol. 39, no. 11, pp. 905–913, 2012.
[244] A. Perina, M. Cristani, and V. Murino, "2LDA: Segmentation for recognition," in *International Conference on Pattern Recognition*, 2010, pp. 995–998.
[245] W. S. Peters and N. Bernstein, "The determination of relative elemental growth rate profiles from segmental growth rates (a methodological evaluation)," *Plant Physiology*, vol. 113, no. 4, pp. 1395–1404, 1997.
[246] R. Pieruschka, A. Chavarría-Krauser, K. Cloos, H. Scharr, U. Schurr, and S. Jahnke, "Photosynthesis can be enhanced by lateral CO_2 diffusion inside leaves over distances of several millimeters," *New Phytologist*, vol. 178, no. 2, pp. 335–347, 2008.
[247] R. Pieruschka and H. Poorter, "Phenotyping plants: genes, phenes and machines," *Functional Plant Biology*, vol. 39, no. 11, pp. 813–820, 2012.
[248] M. Polak, H. Zhang, and M. Pi, "An evaluation metric for image segmentation of multiple objects," *Image and Vision Computing*, vol. 27, no. 8, pp. 1223–1227, 2009.
[249] F. Porikli and O. Tuzel, "Fast construction of covariance matrices for arbitrary size image windows," in *International Conference on Image Processing*, 2006, pp. 1581–1584.
[250] M. P. Pound, A. P. French, E. H. Murchie, and T. P. Pridmore, "Automated recovery of 3D models of plant shoots from multiple colour images," *Plant Physiology*, 2014.
[251] C. Poynton, "A guided tour of color space," in *New Foundations for Video Technology*, 1995, pp. 167–180.
[252] W. K. Pratt, "Spatial transform coding of color images," *IEEE Transactions on Communication Technology*, vol. 19, no. 6, pp. 980–992, 1971.
[253] C. A. Price, O. Symonova, Y. Mileyko, T. Hilley, and J. S. Weitz, "LEAF GUI: segmenting and analyzing the structure of leaf veins and areoles," *Plant Physiology*, vol. 155, no. 1, pp. 236–245, 2010.
[254] T. P. Pridmore, A. P. French, and M. P. Pound, "What lies beneath: underlying assumptions in bioimage analysis," *Trends in Plant Science*, vol. 17, no. 12, pp. 688–692, 2012.
[255] L. Pu, M. W. Marcellin, A. Bilgin, and A. Ashok, "Image compression based on task-specific information," in *International Conference on Image Processing*, 2014, pp. 4817–4821.
[256] J. Puhl, "Im Silicon Savannah," *Der Spiegel*, vol. 2013, no. 48, pp. 118–122, 2013.
[257] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. Kang, "Image-based plant modeling," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 599–604, 2006, proc. SIGGRAPH '06.
[258] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.
[259] A. Redondi, L. Baroffio, J. Ascenso, M. Cesano, and M. Tagliasacchi, "Rate-accuracy optimization of binary descriptors," in *International Conference on Image Processing*, 2013, pp. 2910–2914.
[260] A. Redondi, L. Baroffio, M. Cesana, and M. Tagliasacchi, "Compress-then-analyze vs. analyze-then-compress: Two paradigms for image analysis in visual sensor networks," in *International Workshop on Multimedia Signal Processing*, 2013, pp. 278–282.
[261] A. Redondi, M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization in visual wireless sensor networks," in *International Conference on Image Processing*, 2012, pp. 1105–1108.
[262] F. Remondino and C. Fraser, "Digital camera calibration methods: considerations and comparisons," in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVI, 2006, pp. 266–272.

[263] C. Reuzeau, J. Pen, V. Frankard, J. de Wolf, R. Peerbolte, W. Broekaert, and W. van Camp, "TraitMill: a discovery engine for identifying yield-enhancement genes in cereals," *Molecular Plant Breeding*, vol. 3, no. 5, pp. 753–759, 2005.
[264] F. J. Richards, "A flexible growth function for empirical use," *Journal of Experimental Botany*, vol. 10, no. 2, pp. 290–301, 1959.
[265] T. Richter and K. J. Kim, "A MS-SSIM optimal JPEG 2000 encoder," in *Data Compression Conference*, 2009, pp. 401–410.
[266] H. Riemenschneider, S. Sternig, M. Donoser, P. M. Roth, and H. Bischof, "Hough regions for joining instance localization and segmentation," in *Computer Vision – ECCV 2012*, 2012, vol. 7574, pp. 258–271.
[267] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.
[268] C. Rosenberger, S. Chabrier, H. Laurent, and B. Emile, "Unsupervised and supervised image segmentation evaluation," in *Advances in Image and Video Segmentation*. IRM Press, 2006, ch. 18, pp. 365–393.
[269] D. Rousseau, H. Dee, and T. Pridmore, "Imaging methods for phenotyping of plant traits," in *Phenomics in Crop Plants: Trends, Options and Limitations*, 2015, pp. 61–74.
[270] M. Rousson and R. Deriche, "A variational framework for active and adaptative segmentation of vector valued images," in *IEEE Workshop on Motion and Video Computing*, 2002, pp. 56–61.
[271] C. B. Rubinstein and J. O. Limb, "Statistical dependence between components of a differentially quantized color signal," *IEEE Transactions on Communications*, vol. 20, no. 5, pp. 890–899, 1972.
[272] L. Sass, P. Majer, and E. Hideg, "Leaf hue measurements: A high-throughput screening of chlorophyll content," in *High-Throughput Phenotyping in Plants*, 2012, vol. 918, pp. 61–69.
[273] K. Sayood, *Introduction to Data Compression*. Morgan Kaufmann, 2012.
[274] H. Scharr, M. Minervini, A. Fischbach, and S. A. Tsaftaris, "Annotated image datasets of rosette plants," Forschungszentrum Jülich GmbH, Jülich, Germany, Tech. Rep. FZJ-2014-03837, Jul. 2014. [Online]. Available: http://hdl.handle.net/2128/5848
[275] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo Muntión, J.-M. Pape, G. Polder, D. Vukadinovic, X. Yin, and S. A. Tsaftaris, "Leaf segmentation in plant phenotyping: A collation study," *Machine Vision and Applications*, 2015, under review.
[276] D. Schilling and P. Cosman, "Preserving step edges in low bit rate progressive image compression," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1473–1484, 2003.
[277] D. Schmundt, M. Stitt, B. Jähne, and U. Schurr, "Quantitative analysis of the local rates of growth of dicot leaves at a high temporal and spatial resolution, using image sequence analysis," *The Plant Journal*, vol. 16, no. 4, pp. 505–514, 1998.
[278] D. Schneider, "Camera hacking," *IEEE Spectrum*, vol. 47, no. 12, pp. 18–19, 2010.
[279] P. H. Schönemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.
[280] C. Severance, "Eben Upton: Raspberry Pi," *Computer*, vol. 46, no. 10, pp. 14–16, 2013.
[281] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*, 2nd ed. CRC Press, 2008.
[282] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: the rising trend of vision based measurement," *IEEE Instrumentation Measurement Magazine*, vol. 17, no. 3, pp. 41–47, 2014.

[283] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, 1988.
[284] C.-R. Shyu, J. Gree, D. P. K. Lun, T. Kazic, M. Schaeffer, and E. Coe, "Image analysis for mapping immeasurable phenotypes in maize [life sciences]," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 115–118, 2007.
[285] L. Silva, M. Koga, C. Cugnasca, and A. Costa, "Comparative assessment of feature selection and classification techniques for visual inspection of pot plant seedlings," *Computers and Electronics in Agriculture*, vol. 97, pp. 47–55, 2013.
[286] P. F. Silva, A. R. Marçal, and R. M. A. da Silva, "Evaluation of features for leaf discrimination," in *Image Analysis and Recognition*, 2013, pp. 197–204.
[287] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.
[288] R. Slovak, C. Göschl, X. Su, K. Shimotani, T. Shiina, and W. Busch, "A scalable open-source pipeline for large-scale root phenotyping of arabidopsis," *Plant Cell*, vol. 26, no. 6, pp. 2390–2403, 2014.
[289] J. V. B. Soares and D. W. Jacobs, "Efficient segmentation of leaves in semi-controlled conditions," *Machine Vision and Applications*, vol. 24, no. 8, pp. 1623–1643, 2013.
[290] Y. Song, R. Wilson, R. Edmondson, and N. Parsons, "Surface modelling of plants from stereo images," in *International Conference on 3-D Digital Imaging and Modeling (3DIM '07)*, 2007, pp. 312–319.
[291] E. Soyak, S. A. Tsaftaris, and A. K. Katsaggelos, "Low-complexity tracking-aware H.264 video compression for transportation surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1378–1389, 2011.
[292] E. P. Spalding and N. D. Miller, "Image analysis is driving a renaissance in growth measurement," *Current Opinion in Plant Biology*, vol. 16, no. 1, pp. 100–104, 2013.
[293] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.
[294] M. Subbarao and G. Surya, "Depth from defocus: A spatial domain approach," *International Journal of Computer Vision*, vol. 13, no. 3, pp. 271–294, 1994.
[295] G. J. Sullivan, J. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
[296] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, 1998.
[297] C.-W. Tang, C.-H. Chen, Y.-H. Yu, and C.-J. Tsai, "Visual sensitivity guided bit allocation for video coding," *IEEE Transactions on Multimedia*, vol. 8, no. 1, pp. 11–18, 2006.
[298] X. Tang, M. Liu, H. Zhao, and W. Tao, "Leaf extraction from complicated background," in *IEEE International Congress on Image and Signal Processing*, 2009, pp. 1–5.
[299] C.-H. Teng, Y.-T. Kuo, , and Y.-S. Chen, "Leaf segmentation, classification, and three-dimensional recovery from a few images with close viewpoints," *Optical Engineering*, vol. 50, no. 3, 2011.
[300] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Academic Press, 2009.
[301] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 58, no. 1, pp. 267–288, 1996.

[302] S. Tisné, Y. Serrand, L. Bach, E. Gilbault, R. Ben Ameur, H. Balasse, R. Voisin, D. Bouchez, M. Durand-Tardif, P. Guerche, G. Chareyron, J. Da Rugna, C. Camilleri, and O. Loudet, "Phenoscope: an automated large-scale phenotyping platform offering high spatial homogeneity," *The Plant Journal*, vol. 74, no. 3, pp. 534–544, 2013.
[303] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.
[304] T. K. To and J.-M. Kim, "Epigenetic regulation of gene responsiveness in arabidopsis," *Frontiers in Plant Science*, vol. 4, no. 548, 2014.
[305] S. A. Tsaftaris, "A scientist's guide to cloud computing," *Computing in Science Engineering*, vol. 16, no. 1, pp. 70–76, 2014.
[306] S. A. Tsaftaris and C. Noutsos, "Plant phenotyping with low cost digital cameras and image analytics," in *Information Technologies in Environmental Engineering*, 2009, pp. 238–251.
[307] M. Tuceryan and A. K. Jain, *Texture Analysis*. World Scientific, 1993, ch. Texture Analysis, pp. 235–276.
[308] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929–944, 2007.
[309] E. Upton and G. Halfacree, *Raspberry Pi User Guide*. Wiley, 2014.
[310] G. van der Heijden, Y. Song, G. Horgan, G. Polder, A. Dieleman, M. Bink, A. Palloix, F. van Eeuwijk, and C. Glasbey, "SPICY: towards automated phenotyping of large pepper plants in the greenhouse," *Functional Plant Biology*, vol. 39, no. 11, pp. 870–877, 2012.
[311] E. J. van Henten, D. Goense, and C. Lokhorst, Eds., *Precision Agriculture '09*. Wageningen Academic, 2009.
[312] A. Vedaldi and B. Fulkerson, "VLFeat - an open and portable library of computer vision algorithms," in *ACM International Conference on Multimedia*, 2010.
[313] C. Verleysen, N. Merlin, and C. D. Vleeschouwer, "Adapting JPEG2000 bit allocation to preserve features of interest," in *International Congress on Image and Signal Processing*, vol. 2, 2011, pp. 602–606.
[314] L. Vese and T. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.
[315] *Portable Network Graphics (PNG) Specification*, W3C Recommendation ISO/IEC 15 948:2003 (E), Rev. 2, 2003. [Online]. Available: http://www.w3.org/TR/PNG/
[316] M. Wallenberg, M. Felsberg, P.-E. Forssén, and B. Dellen, "Channel coding for joint colour and depth segmentation," in *Pattern Recognition, 33rd DAGM Symposium*, 2011, pp. 306–315.
[317] A. Walter and U. Schurr, "The modular character of growth in Nicotiana tabacum plants under steady-state nutrition," *Journal of Experimental Botany*, vol. 50, no. 336, pp. 1169–1177, 1999.
[318] A. Walter, H. Scharr, F. Gilmer, R. Zierer, K. A. Nagel, M. Ernst, A. Wiese, O. Virnich, M. M. Christ, B. Uhlig, S. Jünger, and U. Schurr, "Dynamics of seedling growth acclimation towards altered light conditions can be quantified via GROWSCREEN: a setup and procedure designed for rapid optical phenotyping of different plant species," *New Phytologist*, vol. 174, no. 2, pp. 447–455, 2007.
[319] A. Walter and U. Schurr, "Dynamics of leaf and root growth: Endogenous control versus environmental impact," *Annals of Botany*, vol. 95, no. 6, pp. 891–900, 2005.

[320] A. Walter, B. Studer, and R. Kölliker, "Advanced phenotyping offers opportunities for improved breeding of forage and turf species," *Annals of Botany*, vol. 110, no. 6, pp. 1271–1279, 2012.
[321] J. Wang, J. He, Y. Han, C. Ouyang, and D. Li, "An adaptive thresholding algorithm of field leaf image," *Computers and Electronics in Agriculture*, vol. 96, pp. 23–39, 2013.
[322] L. Wang, I. V. Uilecan, A. H. Assadi, C. A. Kozmik, and E. P. Spalding, "HYPOTrace: Image analysis software for measuring hypocotyl growth and shape demonstrated on arabidopsis seedlings undergoing photomorphogenesis," *Plant Physiology*, vol. 149, no. 4, pp. 1632–1637, 2009.
[323] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao, "SSIM-motivated rate-distortion optimization for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 4, pp. 516–529, 2012.
[324] X.-F. Wang, D.-S. Huang, and H. Xu, "An efficient local Chan-Vese model for image segmentation," *Pattern Recognition*, vol. 43, no. 3, pp. 603–618, 2010.
[325] Y. Wang, L.-Y. Duan, J. Lin, T. Huang, and W. Gao, "Joint optimization of JPEG quantization table and coefficient thresholding for low bitrate mobile visual search," in *International Conference on Image Processing*, 2014, pp. 3978–3982.
[326] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[327] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.
[328] C. Weight, D. Parnham, and R. Waites, "LeafAnalyser: a computational method for rapid and large-scale analyses of leaf shape variation," *The Plant Journal*, vol. 53, no. 3, pp. 578–586, 2008.
[329] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.
[330] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.
[331] J. W. White, P. Andrade-Sanchez, M. A. Gore, K. F. Bronson, T. A. Coffelt, M. M. Conley, K. A. Feldmann, A. N. French, J. T. Heun, D. J. Hunsaker, M. A. Jenks, B. A. Kimball, R. L. Roth, R. J. Strand, K. R. Thorp, G. W. Wall, and G. Wang, "Field-based phenomics for plant genetics research," *Field Crops Research*, vol. 133, pp. 101–112, 2012.
[332] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.
[333] P. Wohlhart, M. Donoser, P. M. Roth, and H. Bischof, "Detecting partially occluded objects with an implicit shape model random field," in *Asian Conference on Computer Vision*, 2013, pp. 302–315.
[334] S. G. Wolf, R. Ginosar, and Y. Y. Zeevi, "Spatio-chromatic model for colour image processing," in *International Conference on Pattern Recognition*, vol. 1, 1994, pp. 599–601.
[335] B. Wu and R. Nevatia, "Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 185–204, 2009.
[336] S. Wu, F. Bao, E. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *IEEE Int. Symp. Signal. Proc. Inf. Tech. (ISSPIT)*, 2007, pp. 11–16.

[337] Y. Yang, P. Yuhua, and L. Zhaoguang, "A fast algorithm for YCbCr to RGB conversion," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1490–1493, 2007.
[338] C. Yeo, H. L. Tan, and Y. H. Tan, "SSIM-based adaptive quantization in HEVC," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 1690–1694.
[339] X. Yin, X. Liu, J. Chen, and D. M. Kramer, "Multi-leaf tracking from fluorescence plant videos," in *International Conference on Image Processing*, 2014, pp. 408–412.
[340] A. L. Yuille, D. S. Cohen, and P. W. Hallinan, "Feature extraction from faces using deformable templates," in *Conference on Computer Vision and Pattern Recognition*, 1989, pp. 104–109.
[341] J. Zabalza, J. Ren, J. Ren, Z. Liu, and S. Marshall, "Structured covariance principal component analysis for real-time onsite feature extraction and dimensionality reduction in hyperspectral imaging," *Applied Optics*, vol. 53, no. 20, pp. 4440–4449, 2014.
[342] K. Zhang, L. Zhang, H. Song, and W. Zhou, "Active contours with selective local or global segmentation: A new formulation and level set method," *Image and Vision Computing*, vol. 28, no. 4, pp. 668–676, 2010.
[343] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, vol. 43, no. 4, pp. 1531–1549, 2010.
[344] L. Zhang, Z. Gu, and H. Li, "SDSP: A novel saliency detection method by combining simple priors," in *International Conference on Image Processing*, 2013, pp. 171–175.
[345] X. Zhang, R. J. Hause, and J. O. Borevitz, "Natural genetic variation for growth and development revealed by high-throughput phenotyping in Arabidopsis thaliana," *G3*, vol. 2, no. 1, pp. 29–34, 2012.
[346] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *Journal of Computational Physics*, vol. 127, no. 1, pp. 179–195, 1996.
[347] W. Zheng, L. Zhao, and C. Zou, "Foley-Sammon optimal discriminant vectors using kernel approach," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 1–9, 2005.