Application-Aware Image Compression for Low Cost and Distributed Plant Phenotyping

Massimo Minervini and Sotirios A. Tsaftaris
IMT Institute for Advanced Studies, Lucca, Italy
{massimo.minervini, s.tsaftaris}@imtlucca.it

***Abstract*—Plant phenotyping investigates how a plant's genome, interacting with the environment, affects the observable traits of a plant (phenome). It is becoming increasingly important in our quest towards efficient and sustainable agriculture. While sequencing the genome is becoming increasingly efficient, acquiring phenotype information has remained largely of low throughput, since high throughput solutions are costly and not widespread. A distributed approach could provide a low cost solution, offering high accuracy and throughput. A sensor of low computational power acquires time-lapse images of plants and sends them to an analysis system with higher computational and storage capacity (e.g., a service running on a cloud infrastructure). However, such system requires the transmission of imaging data from sensor to receiver, which necessitates their lossy compression to reduce bandwidth requirements. In this paper, we propose an application aware image compression approach where the sensor is aware of its context (i.e., imaging plants) and takes advantage of the feedback from the receiver to focus bitrate on regions of interest (ROI). We use JPEG 2000 with ROI coding, and thus remain standard compliant, and offer a solution that is low cost and has low computational requirements. We evaluate our solution in several images of Arabidopsis thaliana phenotyping experiments, and we show that both for traditional metrics (such as PSNR) and application aware metrics, the performance of the proposed solution provides a 70% reduction of bitrate for equivalent performance.**

***Index Terms*—image compression; JPEG 2000; ROI coding; plant segmentation; agriculture; plant phenotyping;**

I. Introduction

Plant phenotyping is a branch of biology that studies how a plant's genome, exposed to the interactions with the surrounding environment, maps into phenome, that is the observable traits of a plant. Uncovering a gene's exact function ("functional genomics" [1]) is of great practical interest, because important functions can be matched with agronomically important traits, of interest to breeders. Plant breeding issues are of utmost importance on a worldwide scale, such as plant-based biofuels, resistance of crops to climate changes, increase in global food demand.

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), 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, that is, they involve removing parts of the plant or even harvesting early in the life-cycle.

This "phenotyping bottleneck" can be addressed by combining novel technologies such as noninvasive imaging, spectroscopy, image analysis, robotics, and high-performance computing [2]. Application of these tools in dedicated high-throughput, controlled-environment facilities would improve precision in the results and reduce the need for replication in the field.

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., it is hard to move a cluster of PCs), and (c) the efficiency of using the available computational resources (e.g., idle time is not utilised).

The combination of low-cost smart sensors with Internet connectivity [3] and a cloud infrastructure (e.g., the iPlant Collaborative project (http://www.iplantcollaborative.org/)) can mitigate the above limitations. To keep the cost of the sensor low, minimal robotics and minimal computational power is assumed, with the bulk of analysis occurring at remote infrastructures. 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, both transmission and archival of full resolution uncompressed images can become soon prohibitive. While repeatability of phenotyping experiments is highly desirable (and would be hindered by discarding images), indiscriminate compression for archival purposes may degrade the quality of the data and compromise its utility.

Motivated by findings of Soyak et al. [4], where an application-aware approach to video compression yielded an 80% reduction in bandwith requirements, in this paper we propose an application-aware approach for the compression

978-1-4673-5807-1/13/$31.00 ©2013 IEEE

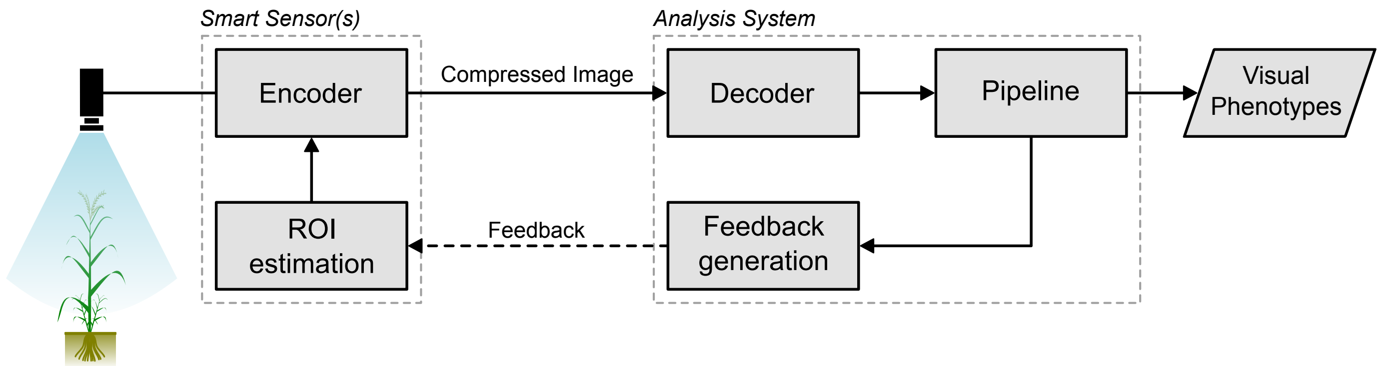


Fig. 1: Graphical representation of the proposed distributed sensing and analysis architecture.

of images in phenotyping environments. We assume that the sensor is aware of its content (i.e., it is imaging plants, as shown in Fig. 2a) and is aware of the final task of extracting features for analysis, which occurs automatically at the cloud and is based on automated computer vision algorithms, as Fig. 1 illustrates. We evaluate our proposed approach in an application-oriented fashion, aiming at maximising both compression efficiency and analysis accuracy at the central location, while maintaining low complexity.

The rest of the paper is organised as follows. In Sec. II we describe our proposed framework for distributed sensing and analysis of plant images from phenotyping experiments. Experimental results are discussed in Sec. III, while Sec. IV offers concluding remarks.

II. Proposed Framework

We propose a distributed architecture for plant phenotypes collection (see Fig. 1), which is characterised by a low computational power sensor (or a grid of sensors) that acquires time-lapse images of a scene containing plants and sends them to a receiver, which we assume to be equipped with higher computational and storage capacity (e.g., a cloud computing service).

An image is acquired by the camera, and a region of interest (ROI) estimation module running on the sensor estimates on the original uncompressed image an ROI indicating where (most likely) plants are located. The raw image is then encoded with the JPEG 2000 compression standard taking advantage of its ROI coding capabilities. Then, the bit stream is transmitted over a link (e.g., Wi-Fi or cabled connection) to a receiver (e.g., a remote workstation, or cloud system), where a decoder reconstructs the image, and automated analysis follows.

On 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 localised, then, a more sophisticated segmentation algorithm is employed to accurately delineate the plant boundaries against the background. The output of this process is a collection of masks that identify pixels belonging to each of the plants in the scene. Such masks enable the extraction of image features (e.g., projected rosette area, average colour intensity) correlated with phenotyping traits.

The output of the analysis system is also utilised to generate feedback information useful for the sensor, to improve its ability in detecting plant objects.

A. Lossy Image Compression with Region-of-Interest Coding

In application-oriented image compression a particularly useful feature is Region-of-Interest (ROI) coding. An ROI is a region in an image that is relevant to the user and, thus, should be preserved in the lossy compression process, by encoding it with better quality than the background. An ROI (possibly composed by multiple objects) in an image I, can be represented as a binary mask M, where $M(i,j) = 1$ means that the pixel at that location is considered part of the foreground, whereas $M(i,j) = 0$ means that the corresponding pixel is part of the background.

In order to compress the acquired images, our system utilises the JPEG 2000 standard [5], based on a Discrete Wavelet Transform (DWT). Compliance to well established standards is preferred in this context to a customised compression scheme, to allow portability of the acquired images virtually on any platform and system, without the need for a specialised decoder. The JPEG 2000 standard also offers ROI coding as a functionality, which is implemented with the Maxshift method [6], described in Part I of the standard. 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 preserving the regions of interest to the highest quality possible for that bit rate.

In plant phenotyping applications, the regions of interest (ROI) in an image should contain plants, and several different approaches for estimating such ROI 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.

B. Proposed ROI Estimation with Feedback

As we will see in the results section, there are several challenging aspects in these images, such as the presence

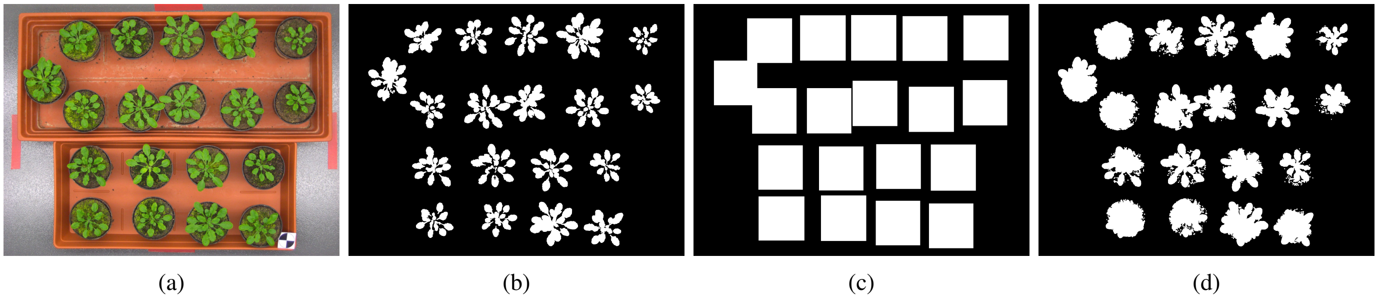


Fig. 2: Examples of ROI masks for the original image (a): proposed method (b), fixed pots (c), and Otsu's thresholding (d).

of moss. 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, in order 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 initialisation 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

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

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 characterised by the density function

$$p(x|\Theta) = \sum_{j=1}^{M} \pi_j p(x|\Theta_j), \quad (2)$$

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 $0 \le \pi_j \le 1$, 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 maximising the log-likelihood function

$$L(\Theta) = \sum_{i=1}^{N} \log\left(\sum_{j=1}^{M} \pi_j p(x_i|\Theta_j)\right), \quad (3)$$

using the Expectation-Maximisation algorithm [7].

We represent pixel intensities in the 1976 CIE $L^*a^*b^*$ colour space [8], 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 colour 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 colour look-up table (LUT) [9].) Then, the intensity vector of each pixel is evaluated in Eq. (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 penalising compression efficiency), the obtained binary mask undergoes a post-processing: 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 utilize the receiver and its feedback to suggest optimal thresholds. 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 maximise 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.

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, and assume it to send $\varphi = \{\Theta, T, A_{max}\}$ 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 added for their transmission amounts approximately to only a hundred bytes.

C. Application-Specific 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. 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 bitrate savings [4].

As part of our methodology, along with traditional PSNR, we propose to evaluate compression with specialised 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 scenario for a given algorithm, as suggested in [4]. 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)* [10], 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 [11], 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 colour 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 localisation tasks, and the gradients are a low-level feature utilised 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).

III. Results and Discussion

In this section we first describe our experimental set-up, 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.

A. Experimental Setup

We evaluated our proposed system on time-lapse images of *Arabidopsis thaliana* rosettes, acquired with a 7 megapixel commercial camera (Canon PowerShot SD1000), in a small laboratory setting [3]. Figure 2a shows an example image from the dataset. We implemented our system using Matlab R2011b by Mathworks, on a machine equipped with Intel Core 2 Duo CPU E8200 2.66 GHz and 4 GB memory, running 64-bit Linux. We adopted the JJ2000 software implementation (from http://code.google.com/p/jj2000/) of the JPEG 2000 standard, to compress the original images at various bit rates with and without the ROI. For evaluation purposes, we also included in the comparison the traditional JPEG standard, using the codec implementation available in Matlab. The α_i weights in *Accuracy* were set to 0.125 for OCE and gradients, and for all other metrics were set to 0.25. For the proposed approach, the receiver estimated the GMM and feedback based on another image (and its segmentation) which was not used in our experiments to eliminate bias.

1) Plant Segmentation: Although our framework is generic, in our experiments we adopt a state-of-the-art approach to plant phenotyping that incorporates incremental learning via appearance models and a level set segmentation [12]. Briefly described, when processing a new incoming image, the analysis system employs several steps including a localisation step to separate plants (implemented using a K-means clustering algorithm), a level set segmentation algorithm to accurately delineate plant objects from the background, a plant labelling algorithm to assign disconnected objects to same plant, and also learns an appearance model, which assists the localization and level set initialisation. In the following, due to the pertinence to the performance of the system, we outline the level set algorithm. It is based on an active contour model for vector-valued images [13], which is capable of robustly detecting objects characterised by a complex and fragmented shape (e.g., a plant rosette). For an image I with N channels, the model is initialised with an initial contour in the spatial

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

Fig. 3: (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.

domain, and then evolves this contour with a level set method, minimising the following energy functional:

$$\begin{aligned} F\left(c^{+}, c^{-}, \phi\right) &= \mu \cdot \text{Length}(C) \\ &+ \int_{in(C)} \frac{1}{N} \sum_{i=1}^{N} \lambda_i^{+} e_i^{+}(z)\, dz \\ &+ \int_{out(C)} \frac{1}{N} \sum_{i=1}^{N} \lambda_i^{-} e_i^{-}(z)\, dz, \end{aligned} \tag{4}$$

where ϕ is the level set function, and $e_i^{+}(z) = \left|I_i - c_i^{+}\right|^2$, and $e_i^{-}(z) = \left|I_i - c_i^{-}\right|^2$, for $i = 1, \ldots, N$. The contour C determines the boundary between foreground and background regions; c_i^{+} and c_i^{-} are, respectively, foreground and background average intensity for the i-th component; λ_i^{+}, λ_i^{-} are positive parameters weighing each channel; and μ is the parameter of the contour length term. By operating on multiple channels (in our context $a*$ and $b*$), this model can detect objects present in at least one of the channels.

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 was fixed for each image and kept constant across the experiment. This level set process is executed for each image (compressed and uncompressed) providing the S, $\hat{S}$ segmentation masks needed for performance evaluation.

2) Baseline ROI approaches: With the goal of demonstrating the accuracy of our method, and the complexity of finding a good ROI without computationally intense processes, we implemented 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 2c shows an ROI mask where pots are assumed to be in fixed positions, and is provided to the encoder. This approach puts 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 out of the ROI, which will affect the validity of the phenotyping analysis.

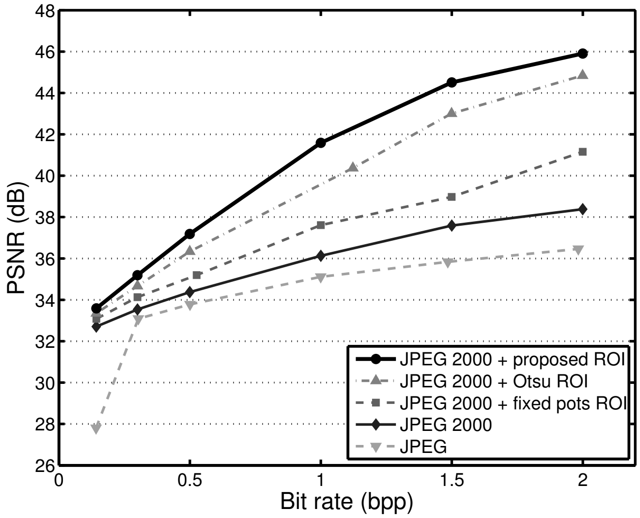
To implement the second baseline approach, we transform the original RGB image to the Excess Green (*ExG*) domain, with $ExG = 2G - R - B$, where R, G and B are red, green, and blue channels of the RGB colour space, respectively [14]. Then, we use Otsu's method [15] to identify an optimum threshold. 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.

B. Results

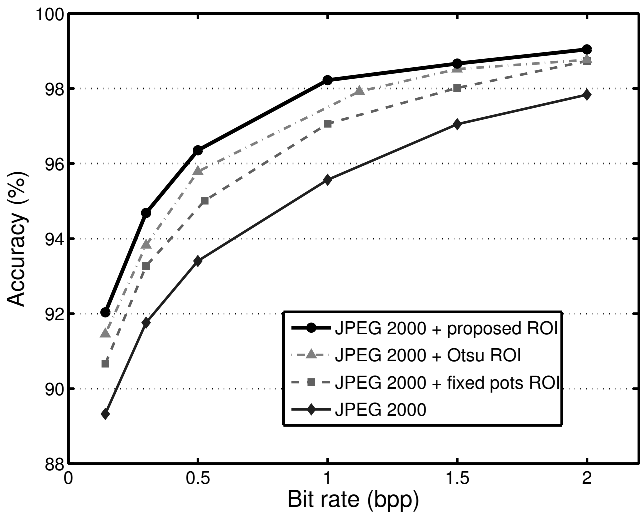
Figure 2 shows examples of the ROI masks used by the encoder based on the proposed and baseline approaches. It is evident that the proposed method provides more accurate ROI masks without actually increasing the complexity in the sensor. The other methods lead to over-segmentation either by design (fixed squares) or due to complexity in the scene (presence of moss is a challenging problem). This results in bits spent encoding information not related to our true objects of interest. This fact can be appreciated also visually in Fig. 3. Traditional JPEG exhibits colour distortion and blocking artifacts (due to its block-based discrete cosine transform). On the other hand, JPEG 2000 introduces a blurring artifact that over-smooths textured regions (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.

In order to quantify the effects of using ROI coding and feedback to the sensor across bit rates, we evaluate application-specific accuracy (as defined in Sec. II) 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 (Fig. 4a) confirm the superiority of our method in visual findings: JPEG 2000 shows an increase of up to 2 dB with respect to JPEG, while up to 8 dB can be



(a)



(b)

Fig. 4: PSNR (a), and application-specific accuracy (b), at various bit rates.

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 favour of the foreground.

Application-specific accuracy (Fig. 4b) reflects an analogous ranking of the systems (curve for JPEG was 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.

IV. Conclusions

We propose a distributed sensing and analysis framework for plant images from phenotyping experiments. While such approach keeps the cost of the sensor low, and allows to carry out sophisticated image analysis tasks (by exploiting the potential of cloud computing infrastructures), it also introduces the need for compressing the transmitted data. We investigate the effects of lossy image compression on plant segmentation, and in order to keep the performance consistent with a scenario where no compression occurs, we propose a smart sensor, which compresses the acquired images using the JPEG 2000 standard and ROI coding, and adopts a Gaussian Mixture Model to find an accurate estimate of the ROI. In order to relieve the sensor from learning the GMM, we propose 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 confirm the efficacy of the proposed approach, both in the fidelity of the reconstructed images, and in the accuracy achieved in the application (i.e., the plant segmentation).

Acknowledgment

This work was supported in part by a Marie Curie Action: "Reintegration Grant" (ref# 256534) of the EU's Seventh Framework Programme (FP7).

References

[1] H. Holtorf, M.-C. Guitton, and R. Reski, "Plant functional genomics," *Die Naturwissenschaften*, vol. 89, no. 6, pp. 235–249, Jun. 2002.

[2] R. T. Furbank and M. Tester, "Phenomics - technologies to relieve the phenotyping bottleneck," *Trends in Plant Science*, vol. 16, no. 12, pp. 635–644, Nov. 2011.

[3] S. A. Tsaftaris and C. Noutsos, *Plant Phenotyping with Low Cost Digital Cameras and Image Analytics*, ser. Environmental Science and Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ch. 18, pp. 238–251.

[4] 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, Oct. 2011.

[5] 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, Sep. 2001.

[6] 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, Jan. 2002.

[7] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.

[8] "Colorimetry," CIE Publication, Vienna: Central Bureau of the CIE, Tech. Rep. 15.2, 1986.

[9] 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, Jul. 1997.

[10] 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, Jul. 2009.

[11] 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, Apr. 2004.

[12] M. Minervini, M. M. Abdelsamea, and S. A. Tsaftaris, "Image-based plant phenotyping with incremental learning and active contours," *Ecological Informatics*, under review.

[13] 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, Jun. 2000.

[14] M. R. Golzarian, M. K. Lee, and J. M. A. Desbiolles, "Evaluation of Color Indices for Improved Segmentation of Plant Images," *Transactions of the ASABE*, vol. 55, no. 1, pp. 261–273, 2012.

[15] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979.