Exemplar-Based Recursive Instance Segmentation With Application to Plant Image Analysis

Jin-Gang Yu, Yansheng Li, Changxin Gao, Hongxia Gao, Gui-Song Xia, Zhu Liang Yu, and Yuanqing Li

***Abstract*— Instance segmentation is a challenging computer vision problem which lies at the intersection of object detection and semantic segmentation. Motivated by plant image analysis in the context of plant phenotyping, a recently emerging application field of computer vision, this paper presents the exemplar-based recursive instance segmentation (ERIS) framework. A three-layer probabilistic model is first introduced to jointly represent hypotheses, voting elements, instance labels, and their connections. Afterward, a recursive optimization algorithm is developed to infer the maximum a posteriori (MAP) solution, which handles one instance at a time by alternating among the three steps of detection, segmentation, and update. The proposed ERIS framework departs from previous works mainly in two respects. First, it is exemplar-based and model-free, which can achieve instance-level segmentation of a specific object class given only a handful of (typically less than 10) annotated exemplars. Such a merit enables its use in case that no massive manually-labeled data is available for training strong classification models, as required by most existing methods. Second, instead of attempting to infer the solution in a single shot, which suffers from extremely high computational complexity, our recursive optimization strategy allows for reasonably efficient MAP-inference in full hypothesis space. The ERIS framework is substantialized for the specific application of plant leaf segmentation in this work. Experiments are conducted on public benchmarks to demonstrate the superiority of our method in both effectiveness and efficiency in comparison with the state-of-the-art.**

***Index Terms*— Instance segmentation, exemplar-based, Hough voting, plant leaf segmentation, plant phenotyping.**

Manuscript received March 21, 2018; revised January 25, 2019 and May 21, 2019; accepted June 2, 2019. Date of publication July 11, 2019; date of current version September 23, 2019. This work was supported in part by the National Key R&D Program of China under Grant 2017YFB1002505, in part by the Natural Science Foundation of China under Grant 61703166, Grant 41601352, and Grant 61573150, in part by the Guangzhou Science and Technology Program under Grant 201904010299, and in part by the Fundamental Research Funds for the Central Universities, SCUT, under Grant 2018MS72. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sen-Ching Samson Cheung. *(Corresponding author: Yuanqing Li.)*

J.-G. Yu, H. Gao, Z. L. Yu, and Y. Li are with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: jingangyu@scut.edu.cn; hxgao@scut.edu.cn; zlyu@scut.edu.cn; auyqli@scut.edu.cn).

Y. Li is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (email: yansheng.li@whu.edu.cn).

C. Gao is with the Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: cgao@hust.edu.cn).

G.-S. Xia is with the State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: guisong.xia@whu.edu.cn).

Digital Object Identifier 10.1109/TIP.2019.2923571

I. Introduction

INSTANCE segmentation [1]–[5] is a fundamental vision task which lies at the intersection of object detection and semantic segmentation. While object detection is to localize different object instances roughly with bounding-boxes, and semantic segmentation assigning each pixel with a semantic class label without differentiating object instances, instance segmentation aims at associating each pixel with an instance-level label [1], [2]. Apparently, this task makes possible more in-depth analysis and understanding of images, which can widely find its application in scene understanding [6], robotics [7], autonomous driving [8], etc.

The task of instance segmentation generally considers multiple visual categories, with one or a few instances in each category, where the instances are mostly sparsely distributed [6], [9], and the core challenge lies in the possible partial occlusion among neighboring object instances [5], [10], [11]. The majority of existing approaches rely on massive manually-annotated data to train strong classification models [3]–[5]. In contrast, we are concerned with a different setting of the instance segmentation problem in this paper, *i.e.*, instance segmentation of one specific visual category given only a handful of annotated exemplars. Such a problem setting is quite common in practice. In a variety of applications, the images constitute a crowd of homogenous instances belonging to the same visual category, such as leaf segmentation in plant phenotyping [12], robotic vision in precise agriculture [13], microscopic cell segmentation in bioinformatics [14], pedestrian segmentation in video surveillance [15], just to name a few. In these scenarios, occlusion is usually more severe than in the general problem setting, which makes the instance segmentation very challenging. Furthermore, as the instances are densely distributed and thereby more likely to be partially occluded, pixel-level instance annotation is extremely labour-intensive compared to in the general situation. Hence, the huge number of annotations required to build strong models are usually unavailable, making instance segmentation based on a few exemplars particularly meaningful.

One real-world application motivating this work is image-based plant phenotyping [12], [16], [17], a newly emerging application field of computer vision which necessitates in-depth analysis of plant images so as to extract morphological or physiological traits reflecting the plant's growth status [17]–[20]. A critical step towards high-quality plant phenotyping is to segment plant leaves accurately at the instance level. In this context, the images usually contain

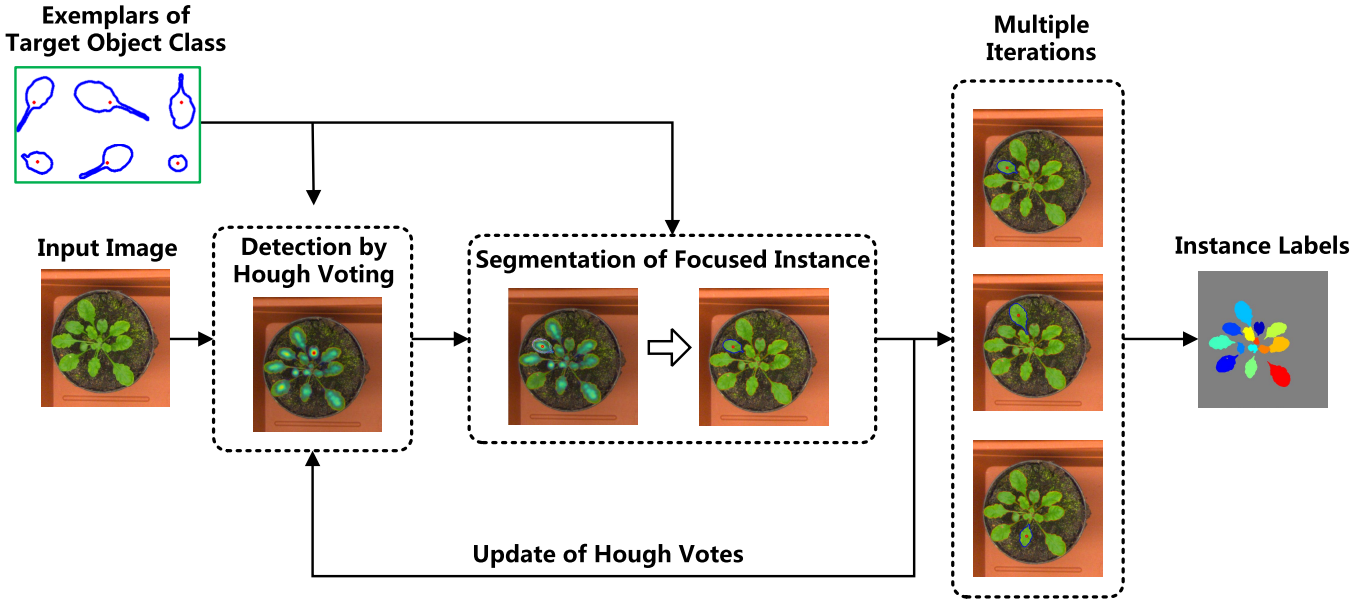


Fig. 1. A schematic illustration of the full pipeline of the proposed ERIS framework. ERIS takes as input an image and a couple of exemplars of the target object class (in the form of complete shape contours or equivalently binary masks), and outputs an instance-level label image. Basically it works in a recursive fashion, which iterates among three steps: 1) Detection by Hough voting: using edge fragments to vote for possible object centers based on the generalized Hough transform; 2) Segmentation of focused instance: localizing the peak in the vote map (marked in red circle), and segmenting the instance around the focused location by using the seeded watershed algorithm, where the seeds are extracted by back-projecting the exemplar masks; 3) Update of Hough votes: updating the assignments as well as the weights of the voting elements according to the current instance segmentation result, and then going to the next iteration. In this way, the method iteratively segments one instance at a time until a certain stop criteria is satisfied.

densely distributed leaves belonging to a known plant type and partially occluding each other, and there is usually lack of massive human annotations. Additionally, the appearance and shape of leaves may vary largely at different growth stages. This defines a representative instance segmentation problem under the setting stated above.

Exploiting a few exemplars as alternative of strong models to address vision problems has been previously demonstrated to be feasible in various contexts, including action recognition [21], shape matching [22], visual tracking [23], occlusion modeling [11], *etc.*. We assume this is also a possible way of solving the instance segmentation problem in lack of abundant annotated data. Most closely related to this work, He and Gould [10] proposed an exemplar-based conditional random Field model for instance segmentation that can jointly model object appearance, shape deformation and occlusion. In [18], Yin *et al.* suggested an optimization method for instance-level plant leaf segmentation by selecting from a candidate set of deformed templates (exemplars) the optimal subset which can best explains the given image. Despite the effectiveness, one common drawback with these approaches is their extremely high computational complexity. The complexity is because that the search space for the instance-level labeling problem is much larger compared to many traditional labeling problems, and these approaches attempt to search for the optimal solution *in a single shot*. To make the reasoning computationally tractable, these approaches commonly take various heuristics to prune the search space to a very limited number of hypotheses in the preprocessing stage (*e.g.*, the method in [10] still costs tens of minutes for a commonly-sized image even reducing to as few as 50 hypotheses), which will inevitably hamper the performance. In another different context, Barinova *et al.* [24] developed a probabilistic framework for multiple object detection which is related to Hough transform. Interestingly, in order to fulfill MAP-inference in the full Hough space (without heuristically pruning the search space), the authors proposed a recursive optimization algorithm which deals with *one instance at a time*. The approach has been demonstrated to be comparable to or more effective than those one-shot inference algorithms while being much more computationally efficient [24].

Motivated by the considerations above, this paper presents a novel exemplar-based recursive method for instance segmentation (termed as ERIS). First, a three-layer probabilistic model is introduced to formulate the instance segmentation problem, which jointly represents instance hypotheses, Hough voting elements, instance labels and their connections in a unified model (detailed in Section III). Second, a recursive optimization algorithm is developed to infer the MAP solution, which handles one instance at a time by alternating among the three steps of detection, segmentation and update. Afterwards, a specific implementation of the ERIS framework is given in the context of plant image analysis for plant phenotyping, and experimental evaluation is conducted on public benchmarks to demonstrate the effectiveness. A schematic illustration of the pipeline of the ERIS framework is presented in Fig. 1. Compared to previous works, our method shares the merit of being exemplar-based and model-free, which is able to fulfill the the detection and segmentation steps by the use of only a couple of annotated exemplars, without the necessity

of massive annotations to build strong models. Meanwhile, it allows for efficient MAP-inference in full hypothesis space due to the recursive optimization strategy. Note that our method is substantialized for the application of plant leaf segmentation in this work, however it can also be adapted to other related application cases.

In summary, the contributions of our work mainly include the following three aspects:

- A three-layer probabilistic model is proposed to formulate the problem of instance segmentation based on exemplars.
- A recursive optimization algorithm is developed to fulfill the MAP-inference for the model, leading to the Exemplar-Based Recursive Instance Segmentation (ERIS) framework.
- A specific implementation of the proposed ERIS framework is given in the context of plant image analysis, which is experimentally demonstrated to outperform previous approaches.

The rest of this paper is organized as follows: Section II is a brief review of related literature. Section III details the proposed three-layer probabilistic model for the problem formulation. Section IV focuses on the MAP-inference of the model. Section V describes the implementation of the ERIS framework in the application of plant leaf segmentation. Section VI is devoted to experiments and the related analysis. In Section VII, the paper ends up with a conclusion.

II. Related Work

In this section, we briefly review previous works that are closely related to ours, including those on instance segmentation and plant leaf segmentation.

A. Instance Segmentation

Instance segmentation has recently become an active research topic in computer vision with the prosperity of deep learning related methodologies [2]–[5] and the spurt of several large-scale benchmarks [6], [9]. As we focus on instance segmentation in the unavailability of abundant annotated data, we do not put weight on those general approaches based on deep learning in our literature review here.

Arteta *et al.* [25] proposed a structured SVM based method to model partially overlapping instances, which is however more oriented for detection and counting tasks rather than segmentation. Similarly, the authors in [26] and [27] proposed a probabilistic extension of the generalized Hough transform (called Implicit Shape Model) for object detection and segmentation, which is however quite limited in segmenting occluding instances. Winn and Shotton [28] presented a layout consistent random field for recognizing and segmenting partially occluded instances. He *et al.* adopted a random field model similar to [28], but used exemplars instead of part-based object models in order to account for deformation more flexibly, which was also followed by [11]. These methods suffer from very high computational load since they attempt to infer the optimal configuration in a huge search space in a single-shot manner.

Barinova *et al.* [24] inspected the traditional Hough transform from a novel probabilistic perspective, and introduced a recursive inference algorithm for multiple object detection which has been shown to be effective and efficient. However, it targets at the detection problem and cannot generate instance segmentation result. Riemenschneider [29] attempted to adapt the recursive inference mechanism to the instance segmentation problem. This method iteratively raises a seeded binary segmentation procedure by directly analyzing the Hough voting map, which provides no way of incorporating prior information (like shape or appearance) about the target objects. In contrast, our proposed method is exemplar-based, which can effectively inject top-down prior information into the detection and segmentation procedures by the use of exemplars.

B. Plant Leaf Segmentation

Instance-level segmentation of plant leaves is at the core of image-based plant phenotyping, an emerging application field at the interdiscipline of computer vision and plant sciences.

Minervini *et al.* [30] proposed a method for leaf segmentation by combing incremental learning and active contour based method, which unfortunately can only distinguish plant instances from an image containing multiple plants rather than leaf instances within a single plant. The leaf segmentation benchmark and competitions launched by Scharr *et al.* [19], [20] has largely promoted the advances on this topic. Pape and Klukas [31] adopted distance transform and graph representation for leaf instance segmentation by explicitly estimating the split lines among overlapping leaves. Simek and Barnard [32] proposed to model a leaf using a blade and a petiole, each modeled by a Gaussian processes with a smoothness constraint at the boundary. Leaf instance segmentation is performed by optimizing a posterior distribution. Yin *et al.* [18] suggested an optimization framework for joint leaf segmentation and tracking from plant videos. Similar to ours, their method is exemplar-based, that is, using a few templates (exemplars) of target plant to identify a set of leaf candidates from which the true leaf instances are selected by optimizing an objective function. However, as stated previously, the method seeks for the optimal solution in a single shot, which has to rely on heuristics to largely reduce the search space in order to make the inference computationally feasible.

III. Problem Formulation

In this paper, we introduce a unified three-layer probabilistic model to formulate the problem of exemplar-based instance segmentation generally, which will be described in this section. For clarity, we first summarize the notations in Table I.

A. Problem Statement

Suppose we are given an image $\mathbf{I}$, where the pixels are indexed by $i \in \{1, \ldots, N\}$. Let $\mathcal{H} = \{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{N_h}\}$ be a set of object hypotheses, where each $\mathbf{h}_k$ ($k \geq 1$) is a parameter configuration in a certain form that corresponds to one hypothesis about the presence of a target object in the image. For example, $\mathbf{h}_k$ may come in the form of a two-dimensional variable representing the spatial location of

TABLE I
SUMMARY OF NOTATIONS

Notations	Description
$\mathcal{H}$, $\mathbf{h}_k$, N_h	The set of object hypotheses $\mathcal{H}$ of the size N_h, with the k-th hypothesis referenced by $\mathbf{h}_k$.
$\mathbf{z} = \{z_k\}_{k=1}^{N_h}$	The binary variable z_k indicating the activation of $\mathbf{h}_k$.
$\mathcal{F}$, $\mathbf{f}_j$, N_f	The set of voting elements $\mathcal{F}$ of the size N_f, with the j-th voting element referenced by $\mathbf{f}_j$.
$\mathbf{y} = \{y_j\}_{j=1}^{N_f}$	The hypothesis assignment y_j of the voting element $\mathbf{f}_j$ where $y_j = k$ indicates $\mathbf{f}_j$ is assigned to $\mathbf{h}_k$.
$\mathbf{I}$, N	The input image $\mathbf{I}$ with N pixels.
$\mathbf{x} = \{x_i\}_{i=1}^{N}$	The instance label x_i of the pixel i.
$\mathcal{T}$, $\mathbf{t}_m$, N_t	The set of exemplars $\mathcal{T}$ of the size N_t, with the m-th exemplar referenced by $\mathbf{t}_m$.
α, β, λ	Parameters of the algorithm.
$h(y_j, \mathbf{f}_j; \mathcal{T})$	The Hough vote cast by the voting element $\mathbf{f}_j$ for the hypothesis $\mathbf{h}_{y_j}$ (defined by using the exemplars $\mathcal{T}$).
σ_w, σ_l, η	The parameters for defining the Hough vote $h(y_j, \mathbf{f}_j; \mathcal{T})$.
$\mathbf{\Omega}(y_j; \mathcal{T})$	The support region of the set of voting elements assigned with y_j (defined by using the exemplars $\mathcal{T}$).
J	The objective function of ERIS
ψ	The function defining the hypothesis activation consistency.
ϕ	The function defining the instance label consistency.
$\mathcal{C}$, N_c	The shape codebook $\mathcal{C}$ of the size N_c.
$\mathbf{c}_n = (\mathbf{s}_n, \mathbf{d}_n)$	The n-th entry in the codebook $\mathcal{C}$ with $\mathbf{s}_n$ being the shape fragment and $\mathbf{d}_n$ its offset relative to the centroid of the entire shape.
$\mathbf{P} = \{\mathbf{p}_u\}_{u=1}^{N_p}$, $\mathbf{Q} = \{\mathbf{q}_v\}_{v=1}^{N_q}$	Two shape fragments $\mathbf{P}$ and $\mathbf{Q}$ consisting of N_p and N_q edge pixels respectively.
d_{C}, d_{WC}	The chamfer distance and the warping chamfer distance between two shapes respectively.
$W(\mathbf{p}; \mathbf{a})$	The affine warping of the point $\mathbf{p}$ parameterized by $\mathbf{a} = (s, \theta, t_x, t_y)$ with s being the scale, θ the rotation angle, t_x and t_y the translation along the x- and y- axis.
$\mathcal{A}$	The whole parameter space for an affine warping.
$\mathbf{A}$, $\mathbf{B}$	The binary masks obtained by the algorithm and the ground truth.

object center, or a three-dimensional variable parameterizing a circle, *etc*. In particular, $\mathbf{h}_0$ is treated as the hypothesis corresponding to image background. Further, we associate each hypothesis $\mathbf{h}_k$ with a binary variable z_k that $z_k = 1$ indicates $\mathbf{h}_k$ is actually activated in the image and $z_k = 0$ otherwise. Note that we always consider the image background to be active, *i.e.*, we always have $z_0 = 1$.

Let $\mathcal{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_{N_f}\}$ be a set of voting elements extracted from $\mathbf{I}$ which vote for the hypotheses in $\mathcal{H}$, with each $\mathbf{f}_j$ being a feature vector characterizing an image element such as a pixel, patch, edge, and so forth. Likewise, for each $\mathbf{f}_j$ we introduce a variable y_j to indicate its hypothesis assignment, for which $y_j = k \in \{0, 1, \ldots, N_h\}$ implies that $\mathbf{f}_j$ votes for the hypothesis $\mathbf{h}_k$ (or equivalently we say $\mathbf{f}_j$ is generated from the hypothesis $\mathbf{h}_k$), and specially, $y_j = 0$ implies that $\mathbf{f}_j$ belongs to the image background.

Generally, the task of instance segmentation from the given image $\mathbf{I}$ can be stated as to assign each image pixel $i \in \{1, \ldots, N\}$ with an instance label x_i, which can take a value from $\{0, 1, \ldots, N_h\}$, with $x_i = k$ implying the pixel i belongs to the object hypothesis $\mathbf{h}_k$ ($x_i = 0$ the background). One can notice that, this task involves detection and segmentation simultaneously, as the presence of the hypotheses (*i.e.* the configuration of z_k's) remains unknown beforehand. As we resort to exemplars to achieve detection and segmentation, we also assume a set of exemplars of the target object to be given, denoted by $\mathcal{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_{N_t}\}$, where each $\mathbf{t}_m$ is a complete shape contour, or equivalently a binary mask image of the same size as $\mathbf{I}$.

B. The Three-Layer Probabilistic Model

Following the notations above, given the image $\mathbf{I}$ represented as a set of voting elements $\mathcal{F}$, the joint probabilistic distribution over the random variables $\mathbf{x} = \{x_1, \ldots, x_N\}$, $\mathbf{y} = \{y_1, \ldots, y_{N_f}\}$ and $\mathbf{z} = \{z_1, \ldots, z_{N_h}\}$, according to Bayes theorem, is given by

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}|\mathcal{F}) \propto p(\mathcal{F}|\mathbf{x}, \mathbf{y}, \mathbf{z})\, p(\mathbf{z})\, p(\mathbf{y}|\mathbf{z})\, p(\mathbf{x}|\mathbf{y}, \mathbf{z}). \tag{1}$$

The interactions among the variables we take into consideration in our model can be represented by a three-layer graph as illustrated in Fig. 2, where the voting element layer connects directly to the hypothesis layer and the label layer. Explicit expression of the model is based on the given exemplars of target object $\mathcal{T}$. Our goal of instance segmentation can be accomplished by applying maximum a posteriori (MAP) estimation to Eq. (1) to jointly infer the optimal configuration for the variables. In the following, we will generally describe the terms in Eq. (1), while retaining the MAP-inference to the next section.

Likelihood Term. Basically, the likelihood term $p(\mathcal{F}|\mathbf{x}, \mathbf{y}, \mathbf{z})$ is meant to encode the Hough voting procedure. Let us firstly make two reasonable conditional independency assumptions among the variables to simplify the formulation and thereby the inference.

First, we assume that the voting elements $\mathcal{F}$, conditioned on $\mathbf{y}$, are independent of the instance labels $\mathbf{x}$ and the hypothesis activations $\mathbf{z}$, implying

$$p(\mathcal{F}|\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathcal{F}|\mathbf{y}). \tag{2}$$

The independency between $\mathcal{F}$ and $\mathbf{x}$ conditioned on $\mathbf{y}$ means the Hough voting does not take into account instance labeling information, which is reasonable because our implementation of Hough voting is essentially shape-based, without using regional appearance features. And the conditional independency between $\mathcal{F}$ and $\mathbf{z}$ conditioned on $\mathbf{y}$ is also reasonable since Hough voting itself is typically a shape matching procedure unaware of the presence (activation) of visual objects.

Second, we assume that each voting element is conditionally independent of the other voting elements as well as their hypothesis assignments, that is,

$$p(\mathcal{F}|\mathbf{y}) = \prod_{j=1}^{N_f} p(\mathbf{f}_j|y_j). \tag{3}$$

Such an assumption implies each voting element votes independently. While some authors have demonstrated the possible

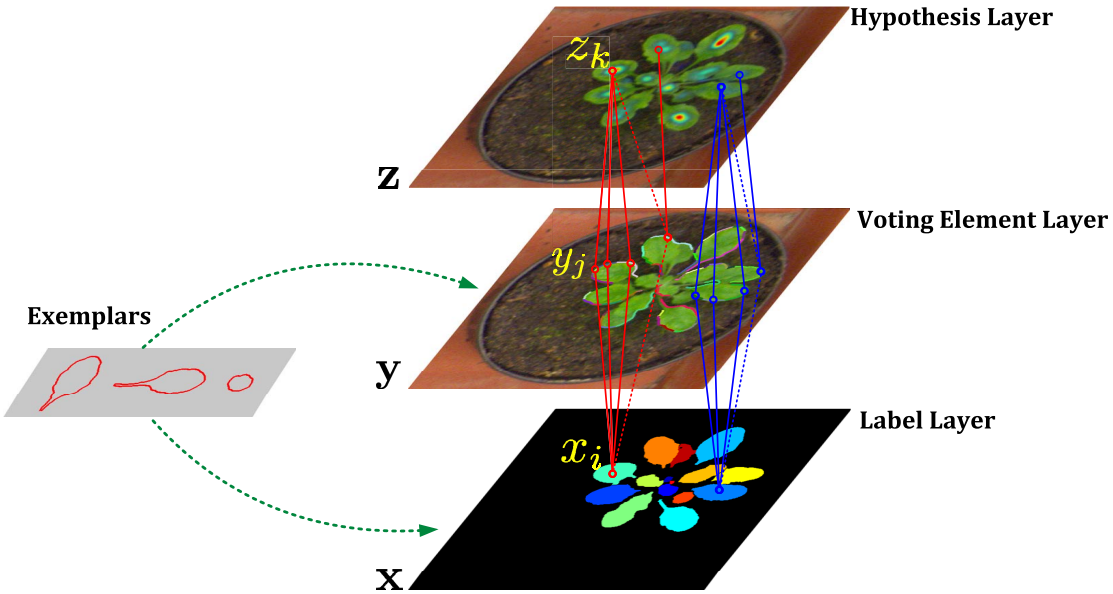


Fig. 2. Illustration of the three-layer probabilistic model for exemplar-based instance segmentation. We jointly model the activation of hypotheses $\mathbf{z} = \{z_1, \ldots, z_{N_h}\}$, the assignment of voting elements $\mathbf{y} = \{y_1, \ldots, y_{N_f}\}$, and the instance labeling of pixels $\mathbf{x} = \{x_1, \ldots, x_N\}$ in a probabilistic way. These random variables and the interactions among them form a three-layer graph, where the voting element layer connects directly to the hypothesis layer and the label layer to encourage the hypothesis activation consistency $p(\mathbf{y}|\mathbf{z})$ and the instance label consistency $p(\mathbf{x}|\mathbf{y})$ respectively. The joint probabilistic distribution over these variables can be expressed explicitly based on exemplars, and maximum a posteriori (MAP) inference is used to find the optimal configuration for the model in order to achieve instance segmentation.

benefits of treating voting elements as groups of dependent entries to vote jointly [33], we still prefer independent voting because those dependent voting strategies may be harmful to the ability of our model in discriminating partially occluded instances.

Since $p(\mathbf{f}_j|y_j) \propto p(y_j|\mathbf{f}_j)/p(y_j)$, for $p(y_j)$ we assume a uniform prior distribution over the hypothesis assignments, which is a reasonable choice if no prior information can be exploited. Then, we further have

$$p(\mathcal{F}|\mathbf{x}, \mathbf{y}, \mathbf{z}) \propto \prod_{j=1}^{N_f} p(y_j|\mathbf{f}_j) = \prod_{j=1}^{N_f} \exp\Big[\alpha h(y_j, \mathbf{f}_j; \mathcal{T})\Big], \quad (4)$$

where α is a parameter, and $h(y_j, \mathbf{f}_j; \mathcal{T})$ is the Hough vote cast by the voting element $\mathbf{f}_j$ for the hypothesis $\mathbf{h}_{y_j}$, which is quantified by matching $\mathbf{f}_j$ to the exemplars $\mathcal{T}$ under the generalized Hough voting framework (see Section V-A for more details).

In the above, we make two conditional independency assumptions to adapt the likelihood term to a shape-based Hough voting procedure, which seems crude to some extent. However, the likelihood term is still able to encode rich spatial constraints between the voting elements and the hypothesized objects (as detailed in Section V-A). Also notice that, we will jointly infer the variables under a unified probabilistic model, and therefore the interactions neglected by our assumptions may be captured by other terms. For instance, the interactions between $\mathbf{y}$ and $\mathbf{z}$ are totally lost in Eq. (2) due to our first assumption, which however can be captured by $p(\mathbf{y}|\mathbf{z})$, the term of hypothesis activation consistency.

Sparsity Prior. The sparsity prior term $p(\mathbf{z})$ is used to express preference to fewer hypothesis activations, that is, explaining the image observations by the use of as few active hypotheses as possible, which can be quantified by

$$p(\mathbf{z}) \propto \exp\left(-\beta \sum_{k=1}^{N_h} z_k\right) \propto \prod_{k=1}^{N_h} \exp\left(-\beta z_k\right), \quad (5)$$

where β is a parameter.

Hypothesis Activation Consistency. The prior term $p(\mathbf{y}|\mathbf{z})$ imposes the constraints of consistency between the configurations of the hypothesis activations $\mathbf{z}$ and the voting element assignments $\mathbf{y}$. More precisely, for any voting element $\mathbf{f}_j$, if it is assigned to the hypothesis $\mathbf{h}_k$, *i.e.*, $y_j = k$, then $\mathbf{h}_k$ must be activated, *i.e.*, $z_k = 1$. Such consistency constraints can be mathematically expressed by

$$p(\mathbf{y}|\mathbf{z}) = \prod_{j=1}^{N_f} \prod_{k=1}^{N_h} \psi(y_j, z_k), \quad (6)$$

where

$$\psi(y_j, z_k) = \begin{cases} 0, & y_j = k \text{ and } z_k = 0, \\ 1, & \text{otherwise.} \end{cases} \quad (7)$$

Instance Label Consistency. As shown in Fig. 2, we perceive the instance labeling $\mathbf{x}$ to depend only on the voting element assignments $\mathbf{y}$, rather than the hypothesis activations $\mathbf{z}$, that is, $p(\mathbf{x}|\mathbf{y}, \mathbf{z}) = p(\mathbf{x}|\mathbf{y})$, which encodes the consistency between $\mathbf{x}$ and $\mathbf{y}$. For each voting element assignment y_j, a support region $\Omega(y_j; \mathcal{T})$ is defined by matching and fitting the exemplars $\mathcal{T}$ to the set of voting elements assigned with y_j (detailed in Section V-B). The consistency then can be stated as, if a pixel i is enclosed by the support region associated to

the voting element assignment y_j, then it should be assigned to the same hypothesis y_j. Formally, let us define

$$\varphi(x_i, y_j) = \begin{cases} 1, & i \in \Omega(y_j; \mathcal{T}) \text{ and } x_i = y_j, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Then,

$$p(\mathbf{x}|\mathbf{y}) = \prod_{i=1}^{N} \prod_{j=1}^{N_f} \varphi(x_i, y_j). \tag{9}$$

It is stressed that, the interaction between the voting element assignment and the pixel labeling, $\varphi(x_i, y_j)$, can be interpreted as using the exemplars $\mathcal{T}$ to guide instance segmentation, which enables the incorporation of top-down information into instance segmentation. Also, we do not consider second-order smoothness in the definition of $p(\mathbf{x}|\mathbf{y})$, a kind of widely-used prior in image labeling problems, which is again because this may hamper the ability of our model in distinguishing partially occluded instances.

IV. MAP-Inference

Upon the establishment of the joint probabilistic distribution $p(\mathbf{x}, \mathbf{y}, \mathbf{z}|\mathcal{F})$, we use maximum a posteriori (MAP) estimation to infer the optimal configuration for the model. By taking the logarithm of Eq. (1), the MAP-inference aims to maximize the following objective function

$$\begin{aligned} J(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \alpha \sum_{j=1}^{N_f} h(y_j, \mathbf{f}_j; \mathcal{T}) + \beta \sum_{k=1}^{N_h} (-z_k) \\ + \sum_{j=1}^{N_f} \sum_{k=1}^{N_h} \log \psi(y_j, z_k) + \sum_{i=1}^{N} \sum_{j=1}^{N_f} \log \varphi(x_i, y_j). \end{aligned} \tag{10}$$

Here we define $\log 0 = -\infty$, which implies that any $\psi(y_j, z_k)$ and $\varphi(x_i, y_j)$ must take the value 1 (hard constraints), otherwise the objective function will get $-\infty$ and therefore cannot be maximized.

A. The ERIS Algorithm

Several standard algorithms exist for solving the optimization problem above, such as loopy belief propagation, simulated annealing, *etc*. However, as the variables in our problem may take a large number of possible values ($x_i, y_j \in \{0, 1, \ldots, N_h\}$), the full search space is extremely large compared to traditional binary labeling problems. Therefore, those algorithms which attempt to optimize over all the variables simultaneously in a single shot are computationally infeasible. Alternatively, Barinova *et al.* [24] suggested a greedy method for solving a similar MAP-inference problem in the context of multiple object detection, which is able to accomplish inference in the full search space (rather than a heuristically reduced one) with satisfactory performance. Inspired by the success of [24], we propose a recursive algorithm for the MAP-inference in Eq. (10).

The proposed optimization algorithm handles one instance at a time by alternating among three steps: *1) detection by Hough voting; 2) segmentation of focused instance; 3) update of Hough votes.* To be specific, in *step 1)*, it probabilistically accumulates Hough voting scores in order to determine which hypothesis to activate by looking for the peak in the vote map; In *step 2)*, around the currently activated hypothesis (location), the supportive voting elements as well as the given exemplars of target object, are taken together to segment the focused instance; In *step 3)*, the support region corresponding to the currently segmented instance is used to update the Hough votes, where the voting elements enclosed by the support region are allowed to switch their assignments to other hypotheses if such switches can increase the voting scores. These three steps are iterated until a certain stop criteria is satisfied (for example, the peak value of the vote map falls below a predefined threshold λ). The pipeline of the proposed recursive algorithm for solving the MAP-inference in Eq. (10) (termed as ERIS) is summarized in Algorithm 1.

B. Interpretation

The ERIS algorithm presented in Algorithm 1 can be interpreted as a greedy approximation to the optimization in Eq. (10). The basic idea is that, while inferring over the whole variable space jointly in a single shot is computationally unaffordable, one can perform inference greedily over a part of the variables and exactly over the rest. More specifically, in our case we greedily augment the variables $\mathbf{z} = \{z_1, \ldots, z_{N_h}\}$ by activating one at a step, and infer the other variables $(\mathbf{x}, \mathbf{y})$ given the current configuration of $\mathbf{z}$, which iterates until stop.

To realize the greedy inference stated above, the first problem is to maximize over $(\mathbf{x}, \mathbf{y})$ given $\mathbf{z}$ at each iteration. Suppose the activation $\mathbf{z}$ is given, for example, $\mathbf{z} = \hat{\mathbf{z}}$, the maximization of the objective in Eq. (10) can be represented as a function of $\mathbf{z}$ by maximizing out $\mathbf{x}$ and $\mathbf{y}$.

$$\begin{aligned} J_z(\hat{\mathbf{z}}) &= \max_{\mathbf{x},\mathbf{y}} J(\mathbf{x}, \mathbf{y}, \hat{\mathbf{z}}) \\ &= \beta \sum_{k=1}^{N_h} (-\hat{z}_k) + \max_{\mathbf{x},\mathbf{y}} J_{xy}(\mathbf{x}, \mathbf{y}; \hat{\mathbf{z}}), \end{aligned} \tag{11}$$

where

$$\begin{aligned} J_{xy}(\mathbf{x}, \mathbf{y}; \hat{\mathbf{z}}) = \underbrace{\alpha \sum_{j=1}^{N_f} \gamma(y_j) + \sum_{j=1}^{N_f} \sum_{k=1}^{N_h} \log \psi(y_j, \hat{z}_k)}_{G_y(\mathbf{y}; \hat{\mathbf{z}})} \\ + \underbrace{\sum_{i=1}^{N} \sum_{j=1}^{N_f} \log \varphi(x_i, y_j)}_{G_{xy}(\mathbf{x}, \mathbf{y})}, \end{aligned} \tag{12}$$

with $\gamma(y_j) = h(y_j, \mathbf{f}_j; \mathcal{T})$. As aforementioned, the third term $G_{xy}(\mathbf{x}, \mathbf{y})$ in Eq. (12) imposes hard constraints which must be satisfied. Once the constraints are satisfied, this term takes the value 0 and will not influence the objective any more, regardless of the configurations of $\mathbf{x}$ and $\mathbf{y}$. Hence, maximizing $J_{xy}(\mathbf{x}, \mathbf{y}; \hat{\mathbf{z}})$ with respect to $(\mathbf{x}, \mathbf{y})$ is equivalent to first maximizing $G_y(\mathbf{y}; \hat{\mathbf{z}})$ with respect to $\mathbf{y}$ (denoting the

Algorithm 1 Exemplar-Based Recursive Instance Segmentation (ERIS)

Input: The input image $\mathbf{I}$. The exemplars of target object class $\mathcal{T}$.

Output: The instance label of every pixel $\mathbf{x} = \{x_1, ..., x_N\}$.

1: Establish the hypothesis set $\mathcal{H}$; Extract the voting elements $\mathcal{F}$;
2: Set $x_i = 0$ for every pixel i;
3: **for** every voting element $\mathbf{f}_j \in \mathcal{F}$ and every hypothesis
4: $\mathbf{h}_k \in \mathcal{H}$ **do**
5: Compute the Hough vote $h(y_j = k, \mathbf{f}_j; \mathcal{T})$;
6: **end**
7: Set $y_j^{\text{cur}} = 0$ for every voting element $\mathbf{f}_j \in \mathcal{F}$;
8: **repeat**
9: *// Step 1) detection by Hough voting*
10: Set $V(k) = 0$ for every hypothesis $\mathbf{h}_k \in \mathcal{H}$;
11: **for** every voting element $\mathbf{f}_j \in \mathcal{F}$ and every hypothesis
12: $\mathbf{h}_k \in \mathcal{H}$ **do**
13: $\delta_k^j = h(y_j = k, \mathbf{f}_j; \mathcal{T}) - h(y_j = y_j^{\text{cur}}, \mathbf{f}_j; \mathcal{T})$;
14: $V(k) += \max\{\delta_k^j, 0\}$;
15: **end**
16: Find $k^* = \arg\max_k \{V(k)\}$;
17: **if** $V(k^*) < \lambda$
18: Return $\mathbf{x} = \{x_1, ..., x_N\}$;
19: **end**
20: *// Step 2) segmentation of focused instance*
21: Compute the support region $\mathbf{\Omega}(k^*, \mathcal{T})$;
22: **for** every pixel $i \in \mathbf{\Omega}(k^*, \mathcal{T})$ **do**
23: Assign $x_i = k^*$;
24: **end**
25: *// Step 3) update of Hough votes*
26: **for** every voting element $\mathbf{f}_j \in \mathcal{F}$ located within
27: $\mathbf{\Omega}(k^*, \mathcal{T})$
28: **if** $h(y_j = k^*, \mathbf{f}_j; \mathcal{T}) > h(y_j = y_j^{\text{cur}}, \mathbf{f}_j; \mathcal{T})$
29: Update the assignment $y_j^{\text{cur}} = k^*$;
30: **end**
31: **end**
32: **until stop**

result by $\mathbf{y}^*$), and then seeking for the configurations of $\mathbf{x}$ which satisfy the hard constraints in $G_{xy}(\mathbf{x}, \mathbf{y}^*)$. Further, let us consider the maximization of $G_y(\mathbf{y}; \hat{\mathbf{z}})$ with respect to $\mathbf{y}$. The term $\sum_{j=1}^{N_f} \sum_{k=1}^{N_h} \log \psi(y_j, \hat{z}_k)$ also consists of a set of hard constraints, which eventually restrict the values that every y_j can take, *i. e.*, $\forall y_j \in \mathcal{I}_y = \{k' | \hat{z}_{k'} = 1\}$, according to the definition of $\psi(y_j, \hat{z}_k)$ in Eq. (8). Then we have

$$\max_{\mathbf{y}} G_y(\mathbf{y}; \hat{\mathbf{z}}) = \max_{\forall y_j \in \mathcal{I}_y} \sum_{j=1}^{N_f} \gamma(y_j) = \sum_{j=1}^{N_f} \max_{y_j \in \mathcal{I}_y} \gamma(y_j), \qquad (13)$$

which means the joint maximization over $\mathbf{y} = \{y_1, \ldots, y_{N_f}\}$ can be achieved by maximizing over each y_j seperately.

The second problem with the greedy inference is to determine which hypothesis to activate. To this end, the ERIS algorithm considers all the possibilities and picks the one which brings the highest increase to the objective function. This is implemented by maintaining a vote map $V(k)$ to accumulate the increments of Hough votes caused by activating the hypothesis k (Line 10-16 in Algorithm 1). A threshold λ is used for the selection (Line 17-19 in Algorithm 1), which has the effect of imposing sparsity constraint on the activation.

Given the current configuration $\hat{\mathbf{z}}$, maximizing $G_{xy}(\mathbf{x}, \mathbf{y})$ is achieved by segmentation of focused instance (Line 21-24 in Algorithm 1). It is worth noticing that, the Hough voting in step 1) only accumulates the increments of the votes relative to the previous step (see Line 13-14 in Algorithm 1). This provides a mechanism of inhibition of return which is similar to the Non-Maximal Suppression (NMS) scheme, and meanwhile ensures the increase of the objective function. However, different from the traditional NMS, our algorithm allows the voting elements to dynamically update their hypothesis assignments during the iterations (see Line 26-31 in Algorithm 1). We stress that such a dynamic update strategy plays a critical role to the effectiveness of our algorithm, which will be experimentally justified in Section VI-B and VI-C.

As for parameter settings, one can see from Eqs. (11)-(13) that, α and β do not actually take effect in the proposed inference strategy. The only active parameter in Algorithm 1 is λ, which should be smaller if the input image contains less instances, and vice versa. We use a fixed setting $\lambda = 0.2$ throughout all the experiments in this paper.

V. ERIS for Plant Leaf Segmentation

In this section, we substantialize the proposed ERIS framework for the specific application of plant leaf segmentation, which is among the most challenging and representative instance segmentation problems. What remains unclear for the implementation are mainly two problems: 1) the term of exemplar-based Hough voting $h(y_j, \mathbf{f}_j; \mathcal{T})$, and 2) the support region $\Omega(k^*, \mathcal{T})$, that is, segmentation of focused instance. We will detail the implementations of these two problems in what follows.

A. Hough Voting

In order to extract voting elements, Canny edge detector is performed over the given image, followed by the edge link operation [34], to generate a set of edge chains (sequences of edge points). A number of anchor points are then secured on each edge chain through line segment fitting [34]. On an edge chain, by taking a pair of anchor points and the edge points between them, one can obtain an edge fragment. To deal with partial occlusion among leaf instances, each edge chain is fragmented via multiple anchor point pairs and intervals. The edge fragments generated in this way constitutes the voting element set $\mathcal{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_{N_f}\}$, with each $\mathbf{f}_j$ being an edge fragment (see Fig. 3 for an intuitive illustration of edge fragment generation).

The exemplars $\mathcal{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_{N_t}\}$ are given in the form of complete shape contours (and also equivalent binary masks),

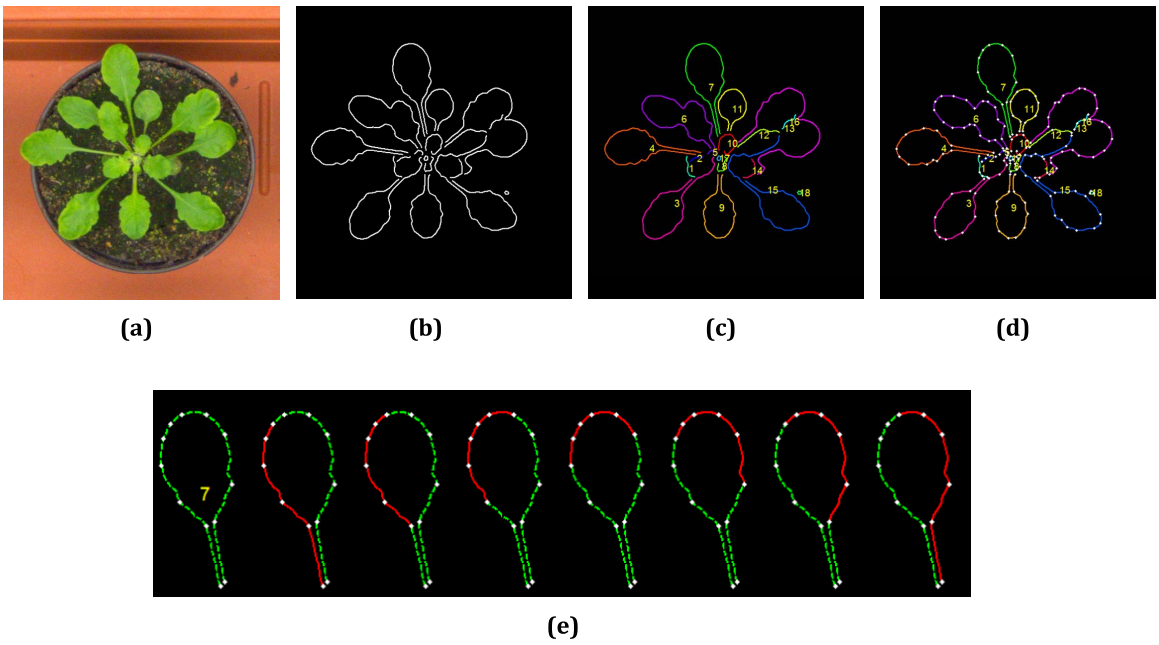


Fig. 3. Illustration of the procedure for edge fragment generation. Canny edge detector is first performed over (a) the given image to obtain (b) the binary edge map. Then, connected components are labeled and chained to generate (c) a set of edge chains using the edge link operation [34], and (d) anchor points (shown as white dots) are secured on each edge chain through line segment fitting. To account for partial occlusion among leaves, each edge chain is fragmented from different starting anchor points and using different intervals. An example of edge fragment generation (over the 7-th edge chain using an interval of 5) is shown in (e), where the generated edge fragments are drawn in red.

where each $\mathbf{t}_m$ is further converted into a number of shape fragments in exactly the same way as that used to extract the voting elements described above. A codebook of shape fragments $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{N_c}\}$ can then be obtained, where the m-th entry is represented as $\mathbf{c}_m = (\mathbf{s}_m, \mathbf{d}_m)$ with $\mathbf{s}_m$ being the shape fragment (a sequence of points) and $\mathbf{d}_m$ the offset of the shape fragment relative to the centroid of the entire shape. Acquisition of the exemplars will be discussed in Section VI-A.

In this work, the hypotheses $\mathcal{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_{N_h}\}$ are substantialized to be (two-dimensional) discrete image coordinates representing object centers. In another word, we treat every pixel location in the image as a hypothesis standing for a possible object center (and hence there are totaly $N_h = N$ hypotheses). Chamfer matching is adopted to match the voting elements $\mathcal{F}$ against the exemplars $\mathcal{T}$ (actually the set of shape fragments $\mathcal{C}$) so as to cast votes for the hypotheses $\mathcal{H}$, *i.e.*, to calculate the Hough vote $h(y_j, \mathbf{f}_j; \mathcal{T})$.

Chamfer Matching. Let $\mathbf{P} = \{\mathbf{p}_u\}_{u=1}^{N_p}$ and $\mathbf{Q} = \{\mathbf{q}_v\}_{v=1}^{N_q}$ be two shapes (sequences of edge pixels). The chamfer distance from $\mathbf{P}$ to $\mathbf{Q}$ is defined by the average of the distances between each point in $\mathbf{P}$ and its nearest neighbor in $\mathbf{Q}$, or formally

$$d_{\mathrm{C}}(\mathbf{P}, \mathbf{Q}) = \frac{1}{N_p} \sum_{\mathbf{p} \in \mathbf{P}} \min_{\mathbf{q} \in \mathbf{Q}} \|\mathbf{p} - \mathbf{q}\|_2. \tag{14}$$

It is well-known that chamfer distance can be computed efficiently via a pre-computed distance transform image $\mathrm{DT}(\mathbf{p}) = \min_{\mathbf{q} \in \mathbf{Q}} \|\mathbf{p} - \mathbf{q}\|_2$, and accordingly, $d_{\mathrm{C}}(\mathbf{P}, \mathbf{Q}) = \frac{1}{N_p} \sum_{\mathbf{p} \in \mathbf{P}} \mathrm{DT}(\mathbf{p})$.

To capture shape deformation, shapes are allowed to undergo affine transformation during matching. For this purpose, we define an affine warping over a point $\mathbf{p}$ by

$$W(\mathbf{p}; \mathbf{a}) = s \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{p} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \tag{15}$$

where $\mathbf{a} = (s, \theta, t_x, t_y)$ is the set of parameters corresponding respectively to scale, rotation angle, translation along the x- and y- axis, and we use $W(\mathbf{P}; \mathbf{a})$ to stand for the warping over all the points in a shape $\mathbf{P}$. The warping chamfer distance is then given by

$$d_{\mathrm{WC}}(\mathbf{P}, \mathbf{Q}) = \min_{\mathbf{a} \in \mathcal{A}} d_{\mathrm{C}}(W(\mathbf{P}; \mathbf{a}), \mathbf{Q}), \tag{16}$$

which implies searching over the whole parameter space $\mathcal{A}$ to find the configuration giving the minimum distance between the affinely warped $\mathbf{P}$ and $\mathbf{Q}$.

For every voting element $\mathbf{f}_j \in \mathcal{F}$, we match it to every entry in the shape codebook $\mathbf{c}_m = (\mathbf{s}_m, \mathbf{d}_m) \in \mathcal{C}$ according to the warping chamfer distance in Eq. (16). Note that, during the matching we first shift $\mathbf{s}_m$ to ensure its centroid coincides with that of $\mathbf{f}_j$, and only vary scales and rotation angles to minimize the chamfer distance. Let us denote by $\mathbf{a}^*_{mj}$ the optimal warping parameters corresponding to $d_{\mathrm{WC}}(\mathbf{s}_m, \mathbf{f}_j)$. Once $\mathbf{a}^*_{mj}$ is known, the spatial location $\boldsymbol{\ell}_{mj}$ where this match should cast a Hough vote can be computed by $\boldsymbol{\ell}_{mj} = W(\mathbf{d}_m; \mathbf{a}^*_{mj})$.

Given all the above, the Hough vote $h(y_j, \mathbf{f}_j; \mathcal{T})$ can be quantified as

$$h(y_j, \mathbf{f}_j; \mathcal{T}) = \sum_{m=1}^{N_t} \xi_{mj} \exp\left\{-\frac{d^2_{\mathrm{WC}}(\mathbf{s}_m, \mathbf{f}_j)}{\sigma_w^2}\right\} \cdot \exp\left\{-\frac{\|\boldsymbol{\ell}_{mj} - \mathbf{h}_{y_j}\|_2^2}{\sigma_l^2}\right\}, \tag{17}$$

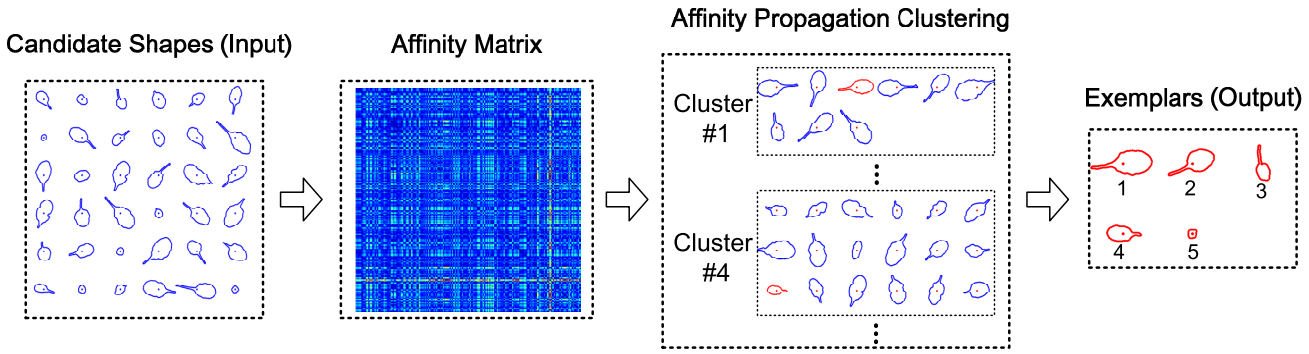


Fig. 4. Illustration of exemplar acquisition. A number of candidate shape contours are taken as input. A pairwise affinity matrix is constructed using the warping chamfer distance in Eq. (16) and the Affinity Propagation algorithm [35] is used to cluster the shapes, where each cluster (shown partially) has a representative data point (shown in red). These representatives are taken as the output exemplars.

where σ_w and σ_l are parameters, and ξ_{mj} is an indicator given by

$$\xi_{mj} = \begin{cases} 1, & d_{\mathrm{WC}}(\mathbf{s}_m, \mathbf{f}_j) < 0.6, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

We consider 120 rotation angles $\theta \in \{0, 3, \ldots, 357\}$ (degrees) and 9 scales $s \in \{2^{-1}, 2^{-\frac{3}{4}}, \ldots, 2^{1}\}$, and use a fixed setting for the Hough voting related parameters $\sigma_l = 5$ (pixels) and $\sigma_w = 0.2$ throughout the experiments in this paper.

B. Segmentation of Focused Instance

The calculation of support region $\Omega(y_j, \mathcal{T})$ is actually to segment the instance around the currently activated hypothesized location $\mathbf{h}_{y_j}$ (focused instance). One possible solution is to back-project the voting elements (edge fragments) supportive of the hypothesis $\mathbf{h}_{y_j}$ onto the exemplars $\mathcal{T}$ (shape masks) to derive a foreground probability map and further a binary segmentation, as in [26] and [27]. However, we observed this method is sometimes unreliable, especially in the presence of occlusion, or if the supportive edge fragments for a focused instance are insufficient.

In our implementation, we adopt an improved strategy as follows: First, we select all the edge fragments supportive of the hypothesis y_j, denoted by $\mathcal{F}_j = \{\mathbf{f}_l | y_l = y_j\}$; Second, we turn to the given exemplars to find $\mathbf{t}_r \in \mathcal{T}$ which best matches $\mathcal{F}_j$ according to the warping chamfer distance in Eq. (16), denoting by $\mathbf{a}_r^*$ the affine parameters corresponding to the best match found. We then back-project $\mathbf{t}_r$ onto the target image by $\mathbf{t}_r^* = W(\mathbf{t}_r; \mathbf{a}_r^*)$; Third, by taking the region $\mathbf{t}_r^*$ as foreground seed, we raise a seeded watershed segmentation procedure over the distance map of the edge image, yielding the support region $\Omega(y_j, \mathcal{T})$. One can see that, this procedure is exemplar-based, which thereby provides a way of integrating top-down information about the target object class into the instance segmentation.

C. Exemplar Acquisition

The proposed ERIS method is distinct for being exemplar-based, so how to acquire exemplars is a critical issue. Suppose we need to acquire N_t exemplars. One straightforward way is to manually select and annotate the exemplars, which may be feasible in certain circumstances since the number N_t is small (typically $N_t < 10$). However, in other cases where the dataset is very large or there exist severe variations among the instances, it may be practically difficult to pick up the small number of good exemplars appropriately. Hence, in our implementation, we suggest a more general approach to exemplar acquisition so as to avoid possible biases in manual selection.

Instead of annotating exactly N_t exemplars, we first annotate $N_t' > N_t$ candidate exemplars, and then use a certain clustering algorithm to automatically learn the N_t exemplars from these candidates. More precisely, as illustrated in Fig. 4, we take N_t' instance masks (or equivalently shape contours) as input (the candidate exemplars) and adopt the well-known Affinity Propagation (AP) [35] (due to its demonstrated effectiveness) to cluster the shapes, where the required similarity matrix is constructed by the use of the warping chamfer distance in Eq. (16). Note that the AP algorithm originally does not allow for specifying the number of clusters. Whenever needed, we first run the algorithm to automatically generate clusters, and then greedily merge the two most closest clusters at a time until reaching the desired number N_t of clusters. The representatives of the N_t clusters are taken as the learned exemplars. Also note that we should use unoccluded shapes for exemplar learning.

Although this approach will slightly add the burden of manual annotation, it may largely alleviate the practical difficulty in picking up a small number of good exemplars from a large or complicated dataset. The impact of exemplars on performance will be experimentally investigated in Section VI-D.

D. Remarks

Although substantialized by leaf segmentation in plant phenotyping, the proposed ERIS framework can be possibly adapted to other similar applications. One example is accurate segmentation of cells from microscopy images for bioinformatics uses [14]. Another example is robotic vision in precise agriculture [13], where fruits (e.g., apples) on the tree should

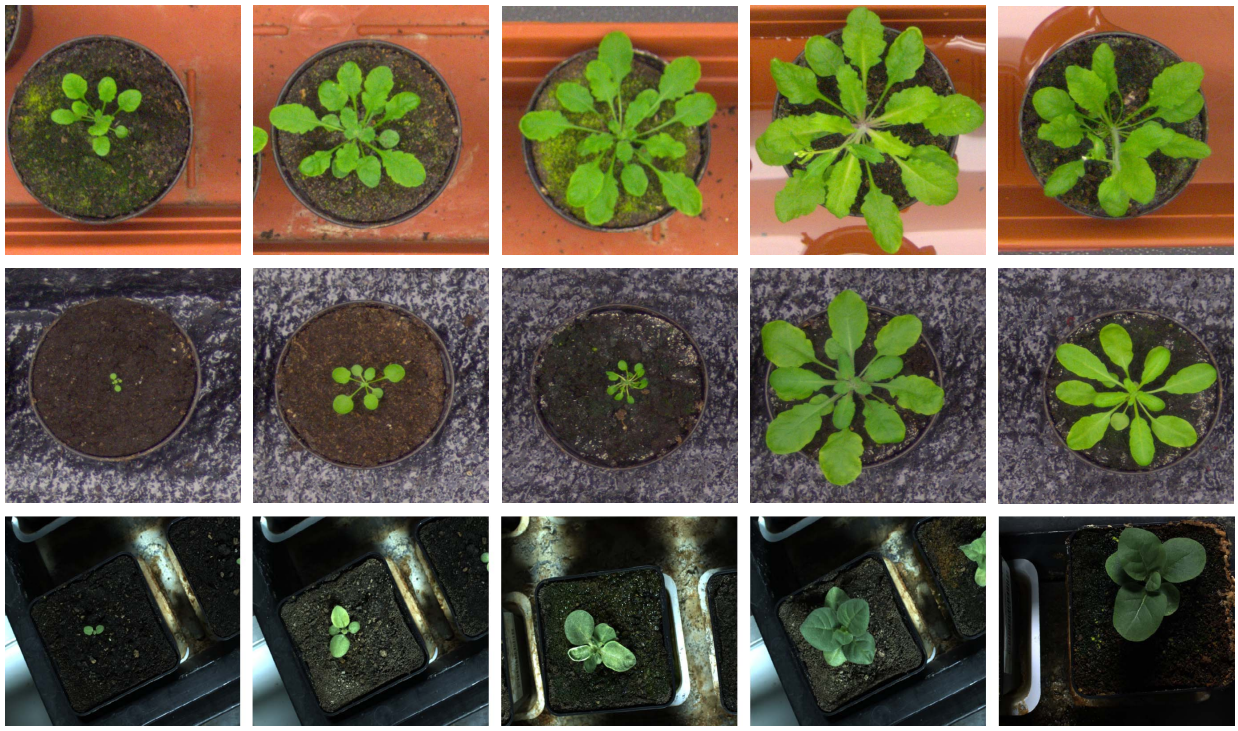


Fig. 5. Exemplary images from the LSC dataset. From top to bottom rows are images from LSC-A1 (Arabidopsis), LSC-A2 (Arabidopsis) and LSC-A3 (Tobacco) respectively.

be first segmented so that robots can localize and grasp them accurately.

As we have justified in Section I, the ERIS framework is basically category-specific, which means it can only accommodate one visual category. Hence, when deployed to a new category, it usually requires "retraining" by using the new data, including acquiring the exemplars and possibly setting the key parameters.

VI. Experiments and Results

Experiments have been carried out on public benchmarks to demonstrate the effectiveness of the proposed ERIS framework. Both qualitative and quantitative analysis will be presented in this section with four major purposes: (1) To quantitatively compare our method against the state-of-the-art, and qualitatively demonstrate the advantages of our method by some examples; (2) To further verify the effectiveness of our method by comparison with some baselines; (3) To test the computational efficiency; (4) To analyze the limitations of our method. All the experiments were conducted on a desktop equipped with dual-core 3.4GHz CPU and 16GB RAM, and our method was implemented in Matlab.

A. Experimental Settings

Dataset. We adopt the LSC dataset [19] for our experiments, which may be the most representative dataset publicly available for this research topic. The dataset was originally released for the Leaf Segmentation Challenge in conjunction with the ECCV Workshop on Computer Vision Problems in Plant Phenotyping in 2014, constituting three subsets termed as **LSC-A1**, **LSC-A2** and **LSC-A3** respectively. LSC-A1 consists of 128 images of Arabidopsis, the most well-known model plant in plant science research. LSC-A2 is also a set of Arabidopsis images, which has a number of 31 images, and LSC-A3 includes 27 images of the Tobacco plant. Pixelwise manual annotations of leaf instances are provided along with each image.

Some exemplary images from the LSC dataset are shown in Fig. 5. Images in plant phenotyping are usually acquired by special imaging equipments (e.g., the LemnaTec system) under controlled environment, so there will not be very complex background in the images [19], [20]. However, there exist other particular aspects of challenging factors: (1) An image may contain a crowd of leaf instances with occlusion among each other; (2) There are significant variations in leaf shapes and sizes, due to the intrinsic difference among individuals, or the growth of an individual.

Performance Measure. Following [18]–[20], we adopt Symmetric Best Dice (SBD) as the performance measure for quantitative evaluation. Let $\mathbf{A}$ be the instance label image obtained by algorithm and $\mathbf{A}_i$ $(1 \leq i \leq |A|)$ the binary mask corresponding to the i-th instance. Let $\mathbf{B}$ be the ground truth and $\mathbf{B}_j$ $(1 \leq j \leq |B|)$ the binary mask corresponding to the j-th instance. We define Best Dice (BD) by

$$\mathrm{BD}(\mathbf{A}, \mathbf{B}) = \frac{1}{|A|} \sum_{i=1}^{|A|} \max_{1 \leq j \leq |B|} \frac{2|\mathbf{A}_i \cap \mathbf{B}_j|}{|\mathbf{A}_i| + |\mathbf{B}_j|}, \tag{19}$$

where $|\cdot|$ stands for the number of non-zeros pixels. Further, Symmetric Best Dice (SBD) is defined by

$$\mathrm{SBD}(\mathbf{A}, \mathbf{B}) = \min\left\{\mathrm{BD}(\mathbf{A}, \mathbf{B}), \mathrm{BD}(\mathbf{B}, \mathbf{A})\right\}. \tag{20}$$

Note that SBD $\in [0, 1]$, and larger values indicate higher consistency between the algorithmic result and the ground truth and thereby better performance.

TABLE II
SBD(%) OBTAINED BY THE COMPARED METHODS ON THE LSC DATASETS

	LSC-A1	LSC-A2	LSC-A3	ALL
ERIS	70.3	63.5	67.4	68.9
MSU [18]	68.1	64.4	63.6	67.1
IPK [31]	64.6	64.3	46.2	62.4

TABLE III
SBD(%) OBTAINED BY ERIS AND THE BASELINES ON THE LSC-A1 DATASET

	ERIS	Baseline-1	Baseline-2	Baseline-3
SBD(%)	70.3	63.5	67.1	66.9

Implementation Details. We use the method described in Section V-C to obtain exemplars. As the datasets provide full instance-level annotations for all the images (note this is not required by our approach), we randomly select a number of images as the training set to learn exemplars and use the remaining images for testing (the annotations are only used for performance evaluation). Since each training image contains multiple partially occluded leaf instances, we only pick up the unoccluded ones for exemplar learning, where a shape in an annotation image is considered to be unoccluded if none (or few) of its pixels are not connected to those belonging to other shapes. More specifically, we randomly select 20 images (including $\sim$150 unoccluded leaves typically) exemplars to extract $N_t = 9$ exemplars for LSC-A1, and 10 images (including $\sim$ 60 unoccluded leaves typically) to extract $N_t = 5$ exemplars for LSC-A2 and LSC-A3 (the impact of N_t will be discussed later). Whenever such random selection is needed, we repeat for 10 times and report the average performance. The numbers of images used for testing are 108, 21 and 17 respectively for LSC-A1, LSC-A2 and LSC-A3.

B. Comparison With State-of-the-Art

Two state-of-the-art methods, referred to as **MSU** [18] and **IPK** [31] (our proposed method is termed as **ERIS**), are considered for comparison, which are both among the top methods as comprehensively evaluated in [19]. For MSU [18], we use the Matlab source code kindly provided by the authors. Since this method also requires a set of shape templates (exemplars) to be given, for fair comparison, we use the same methodology for exemplar generation and evaluation as that used by our method described above. Also note that, this method was developed for leaf segmentation and tracking in plant videos, where the tracking procedure however can also be applied to a single image for refining the segmentation results. IPK [31] relies on distance transform to identify leaf centers, and uses a graph representation to localize split lines between overlapping leaves in order to segment leaf instances. A Java-based implementation of the core components of this method was released by the authors[1], which however cannot be used off-the-shelf for our evaluation, so we implemented it in Matlab by ourselves based on the Java version.

Quantitative Comparison. The performance of the various methods in terms of SBD are comparatively reported in Table II. As can be observed, our method achieves higher overall SBD score than the other two compared methods, which indicates its effectiveness. One may also notice that, the SBD score achieved by our method is slightly lower on LSC-A2. The reason may be that, nearly half of the images in LSC-A2 contain very tiny plants, making the extraction of edge fragments (voting elements) inaccurate, which consequently degrades the performance since voting elements are essential to our method.

Qualitative Analysis. Some representative label images generated by the various methods are demonstrated in Fig. 6 for visual comparison. One can observe that, our method generally yields better segmentation results in terms that 1) the shapes of the leaf instances are preserved better, especially for the partially overlapping leaves (the fourth and sixth rows); 2) it performs better for varying sizes of plants, including tiny ones (the fifth row); 3) it can perform more consistently on different types of plants (Arabidopsis in the first to fifth rows and Tobacco in the last two rows). There are mainly three aspects of factors accounting for the superior performance of our method: 1) It is exemplar-based, which can introduce top-down shape information into the detection and segmentation; 2) The adaptive update mechanism makes it possible to amend the incorrect results in previous iterations. 3) The voting elements used in our method are edge fragments, rather than holistic shapes, which is essentially better for handling occluded leaves. In contrast, IPK splits overlapping leaves via straight lines, which cannot well adapt to the true shapes. In addition, this method relies heavily on the identification of local peaks of distance transform, which tends to split leaves undesirably if the mode seeking is inaccurate. MSU relies much on the choice of templates since it is based on holistic shapes, which is however a difficult problem in practice. So, this method may wrongly split a leaf or identify two neighboring leaves as a single one in existence of occlusion.

We further show in Fig. 7 the intermediate results generated during the iteration procedure for two representative plants. For the first plant (see the top two rows), the two overlapping leaves are wrongly segmented as a single one at iteration #7, which is however corrected desirably at iteration #16 with the dynamic update of the Hough vote map. Similarly, for the second plant (the bottom two rows), the incorrectly split leaf at iteration #6 is corrected at iteration #17. These two examples can demonstrate the advantages of the dynamic update strategy used in our algorithm.

C. Comparison With Baselines

To further verify the effectiveness of our method, we construct three baselines as follows for comparison.

- **Baseline-1:** The recursive pipeline of ERIS is somewhat like the Non-Maximal Suppression (NMS) widely used

[1]https://github.com/OpenImageAnalysisGroup/IAP

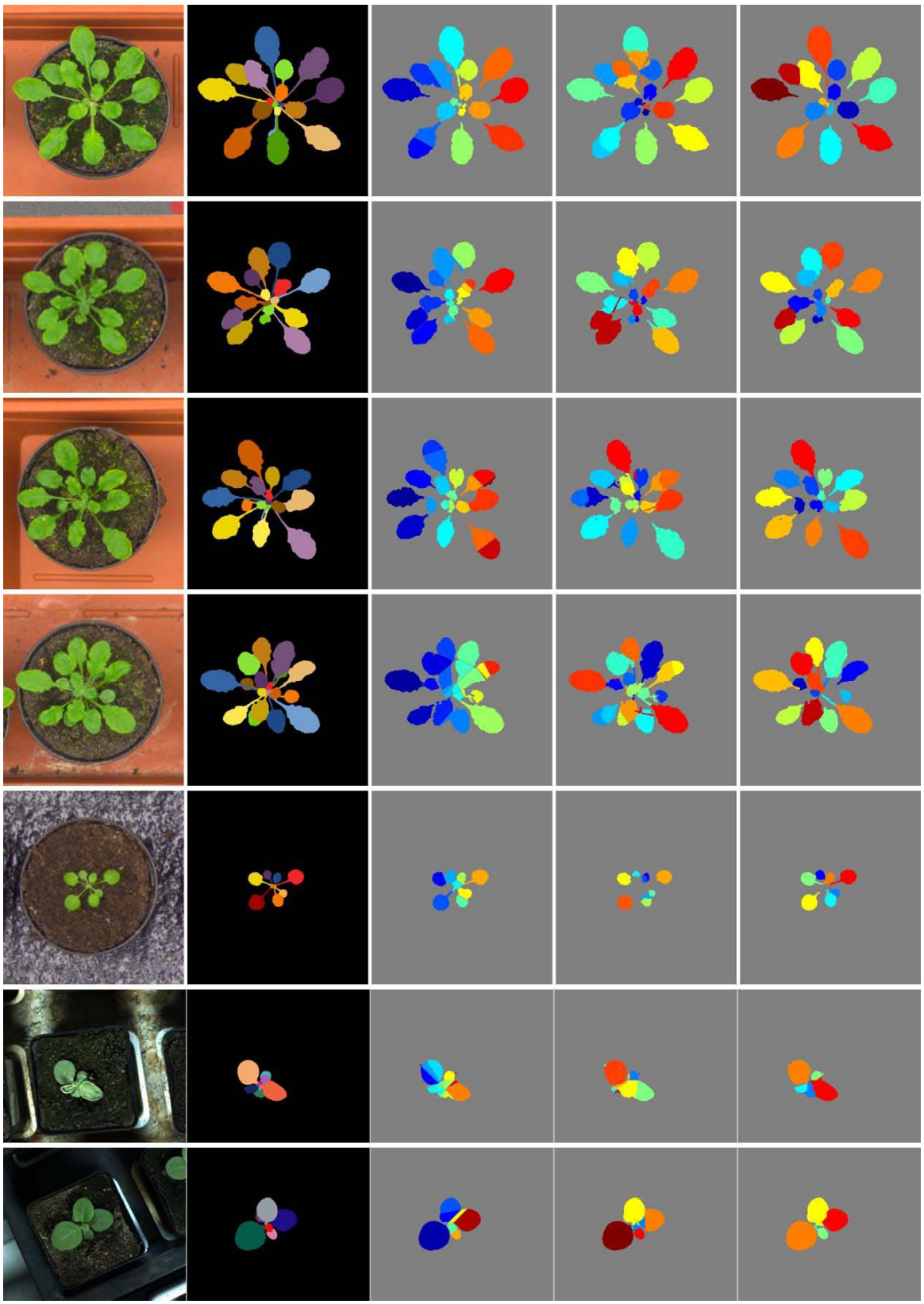


Fig. 6. Representative instance segmentation results obtained by of the compared methods. In each row, from left to right are the input image, the ground truth annotation, the instance label images obtained by IPK [31], MSU [18] and our ERIS, respectively. From top to bottom, the first to fourth rows are from LSC-A1 (Arabidopsis), the fifth row is from LSC-A2 (Arabidopsis), and the last two rows are from LSC-A3 (Tobacco).

in object detection. However, the intrinsic difference is that, ERIS can adaptively update the Hough votes and adjust the assignment of voting elements during iterations. In contrast, NMS just simply nullifies the related part in the confidence map once a hypothesis is activated. In this baseline, we replace the adaptive update

Fig. 7. Intermediate segmentation results generated by ERIS during the iteration procedure for two representative plants. The top two rows correspond to one plant and the bottom two rows to another. The first and third rows are the dynamically updated Hough vote maps at several iteration steps, where the blue circles indicate the identified global peaks, and the last column is the ground truth. The second and fourth rows are the corresponding instance segmentation results at each iteration, with the last column being the final instance labels.

TABLE IV
SBD(%) OBTAINED BY VARYING THE NUMBER OF EXEMPLARS ON THE LSC-A1 DATASET

N_t	3	4	5	6	7	8	9
SBD(%)	67.98 ± 3.07	68.98 ± 1.34	68.96 ± 1.26	69.37 ± 0.89	69.35 ± 0.86	70.02 ± 1.08	70.25 ± 0.63

scheme in ERIS by NMS, that is, when the hypothesis $\mathbf{h}_{k^*}$ is activated, the Hough vote $V(k)$ is set to 0 if $\mathbf{h}_k$ spatially lies within the radius R from $\mathbf{h}_{k^*}$.

- **Baseline-2:** We modify the update scheme in ERIS, that is, instead of reassigning all the voting elements within the segmented region $\Omega(k^*, \mathcal{T})$, we reassign the voting elements located within the fixed radius R from $\mathbf{h}_{k^*}$.
- **Baseline-3:** In this baseline, we consider another variant of the update scheme in ERIS, that is, each voting element is only allowed to be assigned once. In another word, reassignment of voting elements is performed for $\mathbf{f}_j$ only if $y_j^{\text{cur}} = 0$ (unassigned previously).

We run these baselines on LSC-A1 (using the same set of exemplars as those used by ERIS), and the performance in terms of SBD is reported in Table III. It can be noticed that, ERIS outperforms all the three baselines, which indicates the effectiveness of the proposed recursive algorithm as well as the update scheme.

D. Impact of Exemplars

As the proposed method is exemplar-based, the selection of exemplars is of critical importance. While we have previously detailed our strategy for exemplar acquisition, one may be concerned with how much the choice of exemplars influences the performance, *i.e.*, the sensitivity of our method to exemplar selection. Towards this end, we conduct another experiment on LSC-A1 by varying the numbers of exemplars to be $N_t = \{3, 4, 5, 6, 7, 8, 9\}$ respectively. For each configuration, we randomly select 20 images for training and use the rest for testing, repeat the experiment for 10 times as aforementioned. The mean and the standard variance of the SBD scores obtained by the various configurations are listed in Table IV and plotted in Fig 8. Generally, with the increase of N_t, the mean SBD grows higher with lower variance, which can be explained by that more exemplars can better cover the

TABLE V
COMPARISON OF THE CPU TIME OF VARIOUS METHODS ON THE LSC-A1 DATASET

	ERIS	MSU [18]	IPK [31]	MSU-opt	ERIS-opt
SBD(%)	70.3	68.1	64.6	-	-
CPU Time (s)	27.1	560.3	5.27	505.9	7.9

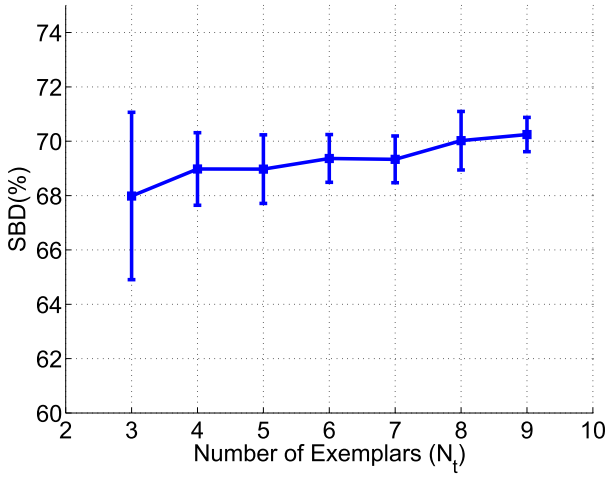


Fig. 8. SBD(%) obtained by varying the number of exemplars on the LSC-A1 dataset.

Fig. 9. Typical failure cases for our method to demonstrate its limitations.

variations in the input data. Meanwhile, as N_t enlarges from 4 to 9, the mean SBD only varies by 1.3%, which suggests the proposed ERIS method can indeed work well with only a couple of exemplars, and is insensitive to the choice of these exemplars.

E. Computational Efficiency

One merit of the proposed ERIS method is the capability of performing MAP-inference efficiently in an recursive way. To demonstrate its efficiency, we have tested the computational time of the three compared methods on the LSC-A1 dataset where all the images are of the size 530×500, as shown in Table V. As suggested by the results, our method without special optimization is over 20 times faster than MSU while achieving better performance. IPK is the fastest among the three since it is a simple and heuristic method, but its performance is limited. We also decouple the MSU and our method to test the running time of the optimization algorithms alone (termed as 'MSU-opt' and 'ERIS-opt' respectively), and ERIS-opt is 60 times faster than MSU-opt. These results demonstrates the efficiency of the proposed recursive MAP-inference algorithm compared to the one-shot optimization approaches like MSU (Note that the CPU time of MSU in Table V is significantly different from that reported in [18], which is because the resolution of images in our experiment is much higher and the computational platform may be different).

F. Limitations

In spite of its effectiveness and efficiency, the proposed method still has several limitations as illustrated in Fig. 9. First, our method may miss the stem part of a leaf (see the blue markers in the top row), which is because it uses the seeded watershed algorithm as the component of segmentation, and it sometimes fails to extract the required seed from the very thin leaf stems. Second, when the contour edge of a leaf is broken into too short fragments due to very complicated occlusion, or when a leaf itself is very much small (see the red markers in the top row), our method is likely to fail to identify them accurately. This is because our method relies on edge fragments for detection and segmentation, but in these cases, it often fails to acquire the edge fragments accurately. Third, while our method has good tolerance to occlusion, it may fail to distinguish two leaves (see the yellow mark in the bottom row) if the occlusion is so heavy that the visible part of a shape is insufficient to support the detection, or especially if the two occluded shapes appear to be a single leaf.

VII. CONCLUSION

In this paper, we have proposed the Exemplar-Based Recursive Instance Segmentation (ERIS) framework, motivated by the specific application of plant leaf segmentation in plant phenotyping. The instance segmentation is formulated by a unified three-layer probabilistic model, and a recursive optimization algorithm is developed to infer the MAP solution. The proposed ERIS framework requires only a couple of annotated exemplars while being able to achieve efficient MAP-inference in full hypothesis space. Note that our method is substantialized for the specific application of plant image analysis in this work, however it can be straightforwardly adapted to other related application cases. Extensive experiments on open benchmarks have been performed to verify its effectiveness and efficiency through both quantitative comparison and qualitative analysis. In addition, the robustness and the

limitations of the proposed method have also been analyzed. In our future work, we plan to apply the ERIS algorithm to real-world plant phenotyping studies by collaborating with plant scientists. We will also attempt to adapt the ERIS framework to the instance segmentation problems in other contexts.

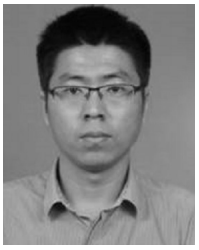
ACKNOWLEDGMENT

The authors would like to thank Dr. Xi Yin for kindly sharing the source code and the helpful discussion on experiments.

REFERENCES

[1] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 297–312.
[2] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3150–3158.
[3] A. Arnab and P. H. S. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 441–450.
[4] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 75–91.
[5] M. Ren and R. S. Zemel, "End-to-end instance segmentation with recurrent attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 293–301.
[6] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3213–3223.
[7] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 345–360.
[8] Z. Zhang, S. Fidler, and R. Urtasun, "Instance-level segmentation for autonomous driving with deep densely connected MRFs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 669–677.
[9] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[10] X. He and S. Gould, "An exemplar-based CRF for multi-instance object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 296–303.
[11] Y.-T. Chen, X. Liu, and M.-H. Yang, "Multi-instance object segmentation with occlusion handling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3470–3478.
[12] F. Fiorani and U. Schurr, "Future scenarios for plant phenotyping," *Annu. Rev. Plant Biol.*, vol. 64, pp. 267–291, Apr. 2013.
[13] K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. Ben-Shahar, "Computer vision for fruit harvesting robots—State of the art and challenges ahead," *Int. J. Comput. Vis. Robot.*, vol. 3, no. 1, pp. 4–34, 2012.
[14] S. Dimopoulos, C. E. Mayer, F. Rudolf, and J. Stelling, "Accurate cell segmentation in microscopy images using membrane patterns," *Bioinformatics*, vol. 30, no. 18, pp. 2644–2651, Sep. 2014.
[15] X. Liang *et al.*, "Learning to segment human by watching YouTube," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1462–1468, Jul. 2017.
[16] D. Houle, D. R. Govindaraju, and S. Omholt, "Phenomics: The next challenge," *Nature Rev. Genetic*, vol. 11, no. 12, pp. 855–866, 2010.
[17] M. Minervini, H. Scharr, and S. A. Tsaftaris, "Image analysis: The new bottleneck in plant phenotyping," *IEEE Signal Process. Mag.*, vol. 32, no. 4, pp. 126–131, Jul. 2015.
[18] X. Yin, X. Liu, J. Chen, and D. M. Kramer, "Joint multi-leaf segmentation, alignment, and tracking for fluorescence plant videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1411–1423, Jun. 2018.
[19] H. Scharr *et al.*, "Leaf segmentation in plant phenotyping: A collation study," *Mach. Vis. Appl.*, vol. 27, no. 4, pp. 585–606, 2016.
[20] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," *Pattern Recognit. Lett.*, vol. 81, pp. 80–89, Oct. 2016.
[21] J.-F. Hu, W.-S. Zheng, J. Lai, S. Gong, and T. Xiang, "Exemplar-based recognition of human–object interactions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 4, pp. 647–660, Apr. 2016.
[22] D. M. Gavrila, "A Bayesian, exemplar-based approach to hierarchical shape matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1408–1421, Aug. 2007.
[23] C. Gao, F. Chen, J.-G. Yu, R. Huang, and N. Sang, "Robust visual tracking using exemplar-based detectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 300–312, Feb. 2017.
[24] O. Barinova, V. S. Lempitsky, and P. Kohli, "On detection of multiple object instances using Hough transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1773–1784, Sep. 2012.
[25] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect partially overlapping instances," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3230–3237.
[26] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, no. 1, pp. 259–289, May 2008.
[27] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Proc. Workshop Stat. Learn. Comput. Vis. (ECCV)*, 2004, pp. 1–16.
[28] J. Winn and J. Shotton, "The layout consistent random field for recognizing and segmenting partially occluded objects," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 37–44.
[29] H. Riemenschneider, S. Sternig, M. Donoser, P. M. Roth, and H. Bischof, "Hough regions for joining instance localization and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 258–271.
[30] M. Minervini, M. M. Abdelsamea, and S. A. Tsaftaris, "Image-based plant phenotyping with incremental learning and active contours," *Ecol. Inform.*, vol. 23, pp. 35–48, Sep. 2014.
[31] J.-M. Pape and C. Klukas, "3-D histogram-based segmentation and leaf detection for rosette plants," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 61–74.
[32] K. Simek and K. Barnard, "Gaussian process shape models for Bayesian segmentation of plant leaves," in *Proc. Comput. Vis. Problems Plant Phenotyping (CVPPP)*, 2015, pp. 4–11.
[33] P. Yarlagadda and B. Ommer, "Beyond the sum of parts: Voting with groups of dependent entities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1134–1147, Jun. 2015.
[34] P. D. Kovesi, *MATLAB and Octave Functions for Computer Vision and Image Processing*. Accessed: 2000. [Online]. Available: http://www.peterkovesi.com/matlabfns/
[35] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

Jin-Gang Yu received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2005, and the M.S. and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2007 and 2014, respectively. He was a Postdoctoral Research Associate with the Department of Computer Science and Technology, University of Nebraska at Lincoln, Lincoln, NE, USA, from 2014 to 2016. He spent three years as a Research and Development Engineer with ZTE Corporation, Shenzhen, China, and with Nortel Networks Corporation, Guangzhou, China, before starting the Ph.D. Program at HUST. He joined the South China University of Technology, Guangzhou, China, in 2016, where he is currently an Associate Professor. His research interests include computer vision, pattern recognition, and machine learning.



Yansheng Li received the B.S. degree in information and computing science from Shandong University, Weihai, China, in 2010, and the Ph.D. degree in pattern recognition and intelligent system from the Huazhong University of Science and Technology, Wuhan, China, in 2015.

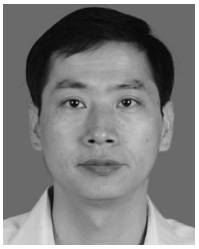
He was an Assistant Professor with Wuhan University (WHU), Wuhan, in 2015, where he became an Associate Research Fellow in 2017. From 2017 to 2018, he was a Visiting Assistant Professor with the Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA. He is currently an Associate Professor with the School of Remote Sensing and Information Engineering, WHU. He has authored over 30 peer-reviewed articles in international journals from multiple domains such as remote sensing and computer vision. His research interests mainly lay in the field of computer vision, machine learning, and their applications in remote sensing big data analysis.

Changxin Gao received the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology in 2010. He is currently an Associate Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests are image analysis and surveillance video analysis.

Hongxia Gao received Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Science, Beijing, China, in 2003. Since 2003, she has been with the South China University of Technology, Guangzhou, China. She is currently a Professor focused on computer vision. Her main scientific interests are in various fields of pattern recognition, computer vision, and bioinformatics. She has given contribution in image reconstruction, nonlinear image restoration, and compressed sensing. She has published over 40 papers in journals and international conferences of this field.

Gui-Song Xia received the B.S. degree in electronic engineering and the M.S. degree in signal processing from Wuhan University, Wuhan, China, in 2005 and 2007, respectively, and the Ph.D. degree in image processing and computer vision from CNRS LTCI, Télécom ParisTech, Paris, France, in 2011. Since 2011, he has been a Postdoctoral Researcher with the Centre de Recherche en Mathématiques de la Decision, CNRS, Paris-Dauphine University, Paris, for one and a half years. He is currently a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. His current research interests include mathematical image modeling, texture synthesis, image indexing and content-based retrieval, structure from motion, perceptual grouping, and remote sensing imaging.

Zhu Liang Yu received the B.S.E.E. and M.S.E.E. degrees in electronic engineering from the Nanjing University of Aeronautics and Astronautics, China, in 1995 and 1998, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2006. In 2000, he joined the Center for Signal Processing, Nanyang Technological University, as a Research Engineer, where he became a Group Leader in 2001. In 2008, he joined the College of Automation Science and Engineering, South China University of Technology, where he was promoted to be a Full Professor in 2010. His research interests include signal processing, machine learning, and their applications in biomedical engineering and intelligent robotics.



Yuanqing Li received the B.S. degree in applied mathematics from Wuhan University, Wuhan, China, in 1988, the M.S. degree in applied mathematics from South China Normal University, Guangzhou, China, in 1994, and the Ph.D. degree in control theory and applications from the South China University of Technology, Guangzhou, in 1997. From 2002 to 2004, he was a Researcher with the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Saitama, Japan. From 2004 to 2008, he was a Research Scientist with the Laboratory for Neural Signal Processing, Institute for Infocomm Research, Singapore. Since 1997, he has been with the South China University of Technology, where he became a Full Professor in 2004. He has authored or coauthored over 60 scientific papers in journals and conference proceedings. His research interests include blind signal processing, sparse representation, machine learning, brain–computer interface, electroencephalograph, and fMRI data analysis.