A Learning Approach for Adaptive Image Segmentation

Vincent Martin, Monique Thonnat

23

A Learning Approach for Adaptive Image Segmentation

Vincent Martin and Monique Thonnat
INRIA Sophia Antipolis, ORION group
France

1. Introduction

Image segmentation remains an issue in most computer vision systems. In general, image segmentation is a key step towards high level tasks such as image understanding, and serves in a variety of applications including object recognition, scene analysis or image/video indexing. This task consists in grouping pixels sharing some common characteristics. But segmentation is an ill-posed problem: defining a criterion for grouping pixels clearly depends on the goal of the segmentation. Consequently, a unique general method cannot perform adequately for all applications. When designing a vision system, segmentation algorithms are often heuristically selected and narrowly tuned by an image processing expert with respect to the application needs. Generally, such a methodology leads to *ad hoc* algorithms working under fixed hypotheses or contexts. Three major issues arise from this approach. First, for a given task, the selection of an appropriate segmentation algorithm is not obvious. As shown in Figure 1, state-of-the-art segmentation algorithms have different behaviours. Second, the tuning of the selected algorithm is also an awkward task. Although default values are provided by authors of the algorithm, these parameters need to be tuned to get meaningful results. But complex interactions between the parameters make the behaviour of the algorithm fairly impossible to predict (see Figure 2). Third, when the context changes, so does the global appearance of images. This can drastically affect the segmentation results. This is particularly true for video applications where lighting conditions are continuously varying. It can be due to local changes (e.g. shadows) and global illumination changes (due to meteorological conditions), as illustrated in Figure 3. The third issue emphasizes the need of automatic adaptation capabilities. As in (Martin et al., 2006), we propose to use learning techniques for adaptive image segmentation. No new algorithms are proposed, but rather a methodology that allows to easily set up a segmentation system in a vision application. More precisely, we propose a learning approach for context adaptation, algorithm selection and parameter tuning according to the image content and the application need.

In order to show the potential of our approach, we focus on two different segmentation tasks. The first one concerns figure-ground segmentation in a video surveillance application. The second segmentation task we focus on is static image adaptive segmentation.

Figure 1. Illustration of the problem of segmentation algorithm selection. Five region-based segmentation algorithms (see Table 1 for details and references) are tuned with default parameters. For better visualization of very small regions, only region boundaries have been represented. Results show differences in terms of number of segmented regions and sensibility to small structures

Figure 2. Illustration of the problem of segmentation algorithm parameter setting. The Hysteresis thresholding algorithm is tuned with two different sets of its two control parameters (T_{low}, T_{high}). A good parameter set might be between these two sets

(a)　　　　　　　　　　　　　　　　(b)

(c)　　　　　　　　　　　　　　　　(d)

(e)　　　　　　　　　　　　　　　　(f)

Figure 3. Illustration of the problem of context variation for a video application. Six frames (from a to f) from an outdoor fixed video surveillance camera have been captured along a day. As lighting conditions change, the perception of the scene evolves. This is visible at a local level as in the zone of pedestrian entrance of the car park (see frames c and d) and at a global level (see frames b and f)

(a)   (b)

(c)   (d)

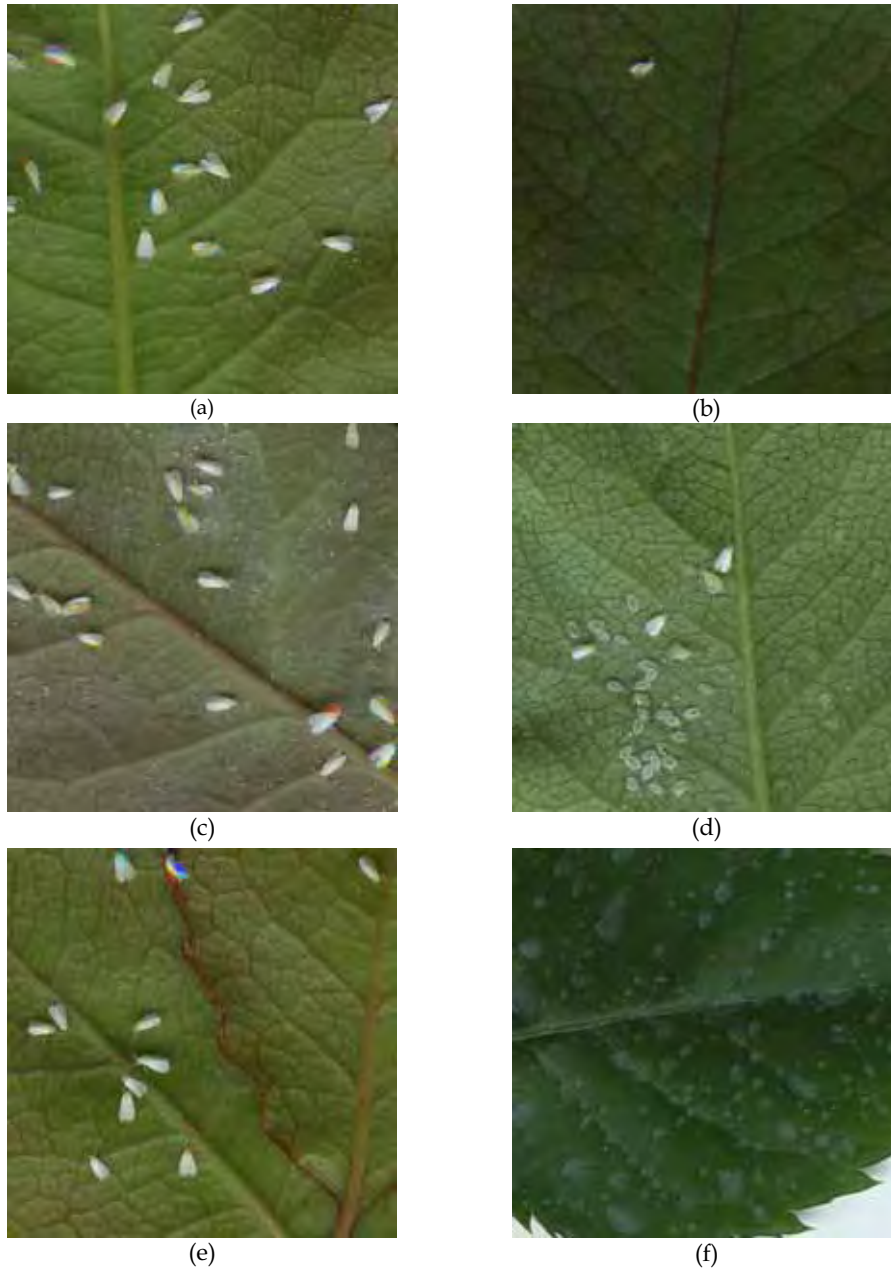(e)   (f)

Figure 4. Illustration of the problem of context variations for a static image application. Objects of interest are small, seen from different point of view and background is highly textured, with complex structures. This makes the segmentation task very difficult

In the first task, the goal is to detect moving objects (e.g. a person) in the field of view of a fixed video camera. Detection is usually carried by using background subtraction methods. A large number of techniques has been proposed in recent years mainly based on pixel intensity variation modeling techniques, e.g. using mixture of gaussians (Grimson & Stauffer, 1999), kernel density (Elgammal et al., 2000) or codebook model (Kim et al., 2005). Strong efforts have been done to cope with quick-illumination changes or long term changes, but coping with both problems altogether remains an open issue (see Figure 3 for example). In these situations, we believe that it should be more reliable to split the background modeling problem into more tractable sub-problems, each of them being associated with a specific context. For this segmentation task, the main contribution of our approach takes place at the context modeling level. By achieving dynamic background model selection based on context analysis, we allow to enlarge the scope of surveillance applications to high variable environments.

In the second task, the goal is to segment complex images where both background and objects of interest are highly variables in terms of color, shape and texture. This is well-illustrated in Figure 4. In other words, the segmentation setting of an image to an other one can be completely different. In this situation, the contribution of our approach arises from the need of adaptability of treatments (algorithm selection and parameter tuning) in order to segment the object of interest in an optimal manner for each image. Knowledge-based techniques have been widely used to control image processing (Thonnat et al., 1999; Clouard et al. 1999). One drawback is that a lot of knowledge has to be provided to achieve good parametrization. In our approach, we alleviate the task of knowledge acquisition for the segmentation algorithm parametrization by using an optimization procedure to automatically extract optimal parameters. In the following sections we describe a learning approach that achieves these objectives.

The organization of the chapter is as follows: Section 2 first presents an overview of the proposed approach then a detailed description is given for two segmentation tasks: figure-ground segmentation in video sequence and static image segmentation. In section 3, we present how we apply these techniques for a figure-ground segmentation task in a video surveillance application and a static image segmentation task for insect detection over rose leaves. Section 4 summarizes our conclusions and discusses the possibilities of further research in this area.

2. Proposed Approach

2.1 Overview

Our approach is based on a preliminary supervised learning stage in which the knowledge of the segmentation task is acquired in two steps.
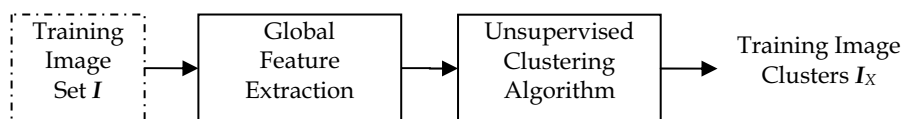

Figure 5. Context analysis schema. The input is a training image set selected by the user. The output is a set of clustered training image I_X

The first step of our approach is dedicated to handle context variations. It aims at modeling context variations based on global image characteristics (see Figure 5). The role of the user is to establish a training image set composed of samples that point out context variations. Low-level information is extracted from the training image set to capture image changes. Then, an unsupervised clustering algorithm is used to cluster this training data feature set. This makes further tasks such as high variable object-class modelization possible by restricting object-class model parameter space.

The second step consists in learning the mapping between the knowledge of the segmentation task and the image characteristics (see Figure 6). The user first defines a set of classes according to the segmentation goal (e.g. background, foreground, object of interest #1, object of interest #2, etc.). This set is used to annotate regions from initial training image segmentation (i.e. grid segmentation, manual segmentation). The goal is to train region classifiers. A region classifier allows to evaluate the membership of a region to a class. Then, a segmentation evaluation metric based on these trained classifiers is defined to assess the quality of segmentation results independently of the segmentation algorithm. This assessment will be further used both for parameter optimization and algorithm ranking.
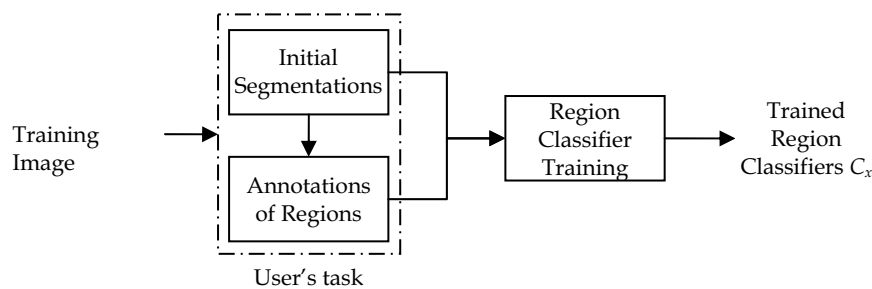

Figure 6. Region classifier training schema. For a cluster of training images I_x belonging to the same context x, the user is invited to annotate template regions from initial segmentations. The output is a set of trained region classifiers C_x, i.e. one classifier per class

After this learning stage our approach proposes an automatic stage for the adaptive segmentation of new images. This stage is devoted to segmentation algorithm parameter control using previously learned knowledge. For an input image, after the context analysis, a global optimization algorithm efficiently explores the parameter space driven by the segmentation quality assessment. The goal is to minimize the assessment value. The main advantage of this procedure is that the search process is independent of both the segmentation algorithm and the application domain. Therefore, it can be systematically applied to automatically extract optimal segmentation algorithm parameters. This scheme is applied to a set of algorithms. By ranking their assessment values, we can select the one which performs the best segmentation for the considered image.

The next sections describe in details each step of our approach for the two investigated segmentation tasks. Figure-ground segmentation task for video surveillance application requires real-time capabilities. In this case, the algorithm selection and parametrization steps are inappropriate because of the necessary computing-time. In static image segmentation task, the computing time is less important.

2.2 Figure-ground Segmentation in Video Sequences

We consider a figure-ground segmentation problem in outdoor with a single fixed video camera. The context variations are mainly due to scene illumination changes such as the nature of the light source (natural and/or artificial), the diffusion effects or the projected shadows. The goal is to segment efficiently foreground regions (i.e. mobile objects) from background regions.

2.2.1 Training Dataset Building by Context Analysis

Segmentation is sensitive to context changes. We study the variability of context in a quantitative manner by using an unsupervised clustering algorithm. The goal is to be able to identify context classes according to a predefined criterion. As context changes alter image both locally and globally, the criterion must be defined to take into account these characteristics. A straightforward approach is to use a global histogram based on pixel intensity distribution as in (Georis, 2006). However, such histograms lack spatial information, and images with different appearances can have similar histograms. To overcome this limitation, we use an histogram-based method that incorporates spatial information (Pass et al., 1997). This approach consists in building a coherent color histogram based on pixel membership to large similarly-colored regions. For instance, an image presenting red pixels forming a single coherent region will have a color coherence histogram with a peak at the level of red color. An image with the same quantity of red pixels but widely scattered, will not have this peak. This is particularly significant for outdoor scene with changing lighting conditions due to the sun rotation, as in Figure 3(a,b).

An unsupervised clustering algorithm is trained using the coherence color feature vectors extracted from the training image set $\boldsymbol{I}$. Let I be an image of the training dataset $\boldsymbol{I}$, for each $I \in \boldsymbol{I}$, the extracted global feature vector is noted g_I. The unsupervised clustering is applied on g_I. Its output is a set of clustered training images $\boldsymbol{I}_X$ composed of n clusters $\boldsymbol{I}_x$:

$$\boldsymbol{I}_X = \bigcup_{i=1}^{n} \boldsymbol{I}_{x_i} \tag{1}$$

The set of cluster identifiers (ID) is noted $X=[x_1,\ldots,x_n]$. In our experiments, we have used a density-based spatial clustering algorithm called DBScan proposed by Ester et al. (Ester et al., 1996). This is well-adapted for clustering noisy data as histograms. Starting from one point, the algorithm searches for similar points in its neighborhood based on a density criteria to manage noisy data.

The next section describes how each cluster of training images is used to train context-specific background classifiers.

2.2.2 Figure-ground Segmentation Knowledge Acquisition by Automatic Annotations of Buckets

Because the point of view of the video camera is fixed, we can easily capture spatial information on image. This is done by using an image bucket partioning where a bucket is a small region at a specific image location. For instance, a bucket can be a square of pixels (see Figure 7) or reduced to only one pixel. The size and the shape of a bucket must be fixed and are equals for all samples of the training image set $\boldsymbol{I}$.

b_1	b_2	b_3
b_4	b_5	b_6
b_7	b_8	b_9

Figure 7. Example of a bucket partioning using a grid segmentation. The image is segmented into nine regions of same size and shape. Each region is a bucket

Let us define the set of bucket partioning B as:

$$B = \bigcup_{i=1}^{m} b_i \tag{2}$$

Where b_i is a bucket among m. Since training image sets are composed of background images, the task of bucket annotations is automatic for a figure-ground segmentation problem. In our approach, this is done by assigning the same background label l to each $b_i \in B$. The role of the user is limited to the selection of video sequences where no mobile objects are present. Then, for each bucket, a feature vector v_b is extracted and makes, with the label a pair sample noted (v_b, l_b). A pair sample represents the association between low-level information (v_b) and high-level knowledge (l_b). If the bucket is a pixel, v_b can be the (R,G,B) value of the pixel. If the bucket is a small region, v_b can be an histogram of the bucket pixel (R,G,B) values. Since all buckets have the same label, the set of all collected pair samples from I_x can be considered as the set of all feature vectors. This constitutes the training dataset T_x as:

$$T_x = \bigcup_{\substack{b \in B \\ I_x \in I_x}} v_b \tag{3}$$

and then,

$$T = \bigcup_{x \in X} T_x \tag{4}$$

T represents the knowledge of the segmentation task. At the end of this automatic annotation process, we obtain $m*n$ training data sets (i.e. one training data set per bucket and per context cluster). The following task is to modelize this knowledge in order to train background classifiers.

2.2.3 Segmentation Knowledge Modelization
For each training image set I_x, we have to train a set of specific background classifiers noted C_x with one background classifier c_x per bucket $b \in B$ as seen in Figure 8.
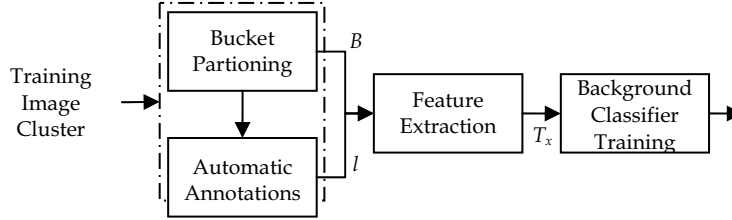
Figure 8. Background classifier training schema for figure-ground segmentation. Since training image sets are composed of only background images, the annotation task is fully automatic

In our approach, we use the background codebook model proposed by Kim et al. (Kim et al., 2005) as background classifier technique. This codebook algorithm adopts a quantization/clustering technique to construct a background model from long observation sequences. For each pixel, it builds a codebook consisting of one or more codewords. For each pixel the set of codewords is built based on a color distortion metric together with brightness bounds applied to the pixel values of the training images I_x. The codewords do not necessarily correspond to single Gaussian or other parametric distributions.

According to this algorithm, a bucket is a pixel and the feature vector v_b is composed of four features: the three (R,G,B) values of the considered pixel and its intensity. At the end of the training, we obtain one background classifier (i.e. a codebook) for each bucket (i.e. a pixel) and for each background cluster I_x.

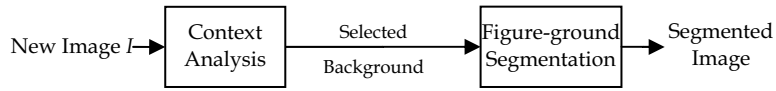### 2.2.3 Real-Time Adaptive Figure-ground Segmentation

Fig 9. Adaptative segmentation schema for figure-ground segmentation

As illustrated in Figure 9, the first step is the dynamic selection of background classifiers. We also use a temporal filtering step to reduce unstability of the clustering algorithm. Indeed, in cluttered scenes, foreground objects can strongly interact with the environnement (e.g. light reflections, projection of shadows) and then add a bias to the context analysis. So, it is important to smooth the analysis by ponderating the current result with respect to previous ones. Our temporal filtering criterion is defined as follows. For an image I, let us define the probability vector of the context analysis ouput for an image I as:

$$p(X \mid g_I) = \left[p(x_1 \mid g_I), \dots, p(x_n \mid g_I) \right] \qquad (5)$$

The most probable cluster x_I with associated probability $p_{max}(x_I)$ for the image I are then:

$$p_{max}(x_I) = \max p(X \mid g_I)$$

$$x_I = \arg \max p(X \mid g_I) \qquad (6)$$

Let us define x the context cluster identifier, x_I the cluster identifier for the incoming image I, μ_x the square mean of cluster probability computed On a temporal window. α is a ponderating coeffcient related to the width w of the temporal filtering window. To decide if x_I is the adequate cluster for an incoming image I, we compare it with the square meanshift of cluster probability μ_x as in the algorithm described in Figure 13. In this algorithm, two cases are investigated. If x_I is the same as the previous one, μ_x is updated based on the context maximum probability $p_{max}(x_I)$ and α. Else if the current x_I is different from the previous one, the current $p_{max}(x_I)$ is tested against μ_x. The square value of $p_{max}(x_I)$ is used to raise the sensibility of temporal filtering to large variations of $p_{max}(x_I)$.

When the cluster identifier x is found, the corresponding background classifiers C_x are selected for the figure-ground segmentation of I as seen in Figure 9.
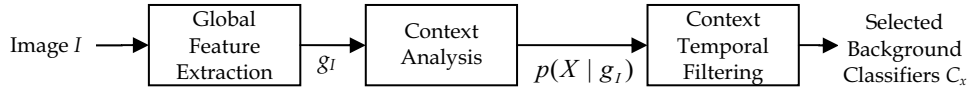

Figure 9. Context analysis in real-time segmentation. From an input image, a global feature vector g_I is first extracted. Then, context analysis computes the vector $p(X \mid g_I)$. Context temporal filtering uses this vector to compute the most probable cluster identifier x_I for the current image depending on previous probabilities

The figure-ground segmentation consists in a vote for each pixel. This vote is based on the results of the background classifiers for each pixel. If a pixel value satisfies both color and brightness distance conditions, it is classified as background ($l = bg$). Otherwise, it is classified as foreground ($l = fg$).

The major problem of this segmentation method is that no spatial coherency is taken into account. To overcome this limitation, we compute in parallel a region-based image segmentation. Our objective is to refine the segmentation obtained with background classifiers (see Figure 12) .

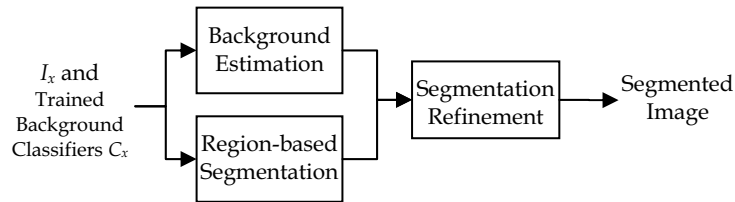Fig 12. Figure-ground segmentation with region spatial refinement

For each region r of the region-based segmentation we compute its label l by testing the percentage of pixels of this region labelled as foreground by the background classifiers. The refinement criterion is defined as follows:

$$\textbf{if } \frac{1}{|r|} \sum_r l_{pix}^{fg} \geq \theta \quad \textbf{then } l = fg \quad \textbf{else } l = bg \tag{7}$$

where θ is a threshold and l_{pix}^{fg} is a pixel classified as being a foreground pixel by its corresponding background classifier.

So, if the foreground pixels inside the region r represent more or equal than θ percent of the region area $|r|$, the region r is considered as a foreground region. In our experiments, we have fixed the threshold θ to 90 percent.

Context Temporal Filtering Algortihm

Initialization step :

$x \leftarrow 0, \mu_x = 0, \alpha \leftarrow 0$

For each new image I

    I. $[x_I, p_{max}(x_I)] = ContextAnalysis(g_I)$ // extract $p(X|g_I)$ index of max and its corresp. value

    II. If $(x = x_I \,||\, x = 0)$ // test the context identifier of I (0 = noise)

        (i) $x \leftarrow x_I$

        (ii) $\mu_x \leftarrow \dfrac{\alpha * \mu_c + p^2{}_{max}(x_I)}{\alpha + 1}$ // update the value of μ_x

        (iii) If $(\alpha < w)$ // test α within the width of the temporal window

           $\alpha \leftarrow \alpha + 1$

        Else If $\left(p^2{}_{max}(x_I) \geq \mu_x\right)$

        (i) $x \leftarrow x_I$

        (ii) $\mu_x \leftarrow p^2{}_{max}(x_I)$ // update the value of μ_x with square max prob of $p(X|g_I)$

        (iii) $\alpha \leftarrow 1$ // reinitialize weight α

        End If

    III. return x

End For

Figure 13. Description of the context temporal filtering algorithm. In our experiment, we have fixed w to 40 to consider the last five seconds of the image sequence in the calculation of μ_x (i.e. 40 frames at eight frames per second correspond to five seconds)

Section 3.1 presents experiments of this proposed approach.

2.3 Static Image Adaptive Segmentation

We consider the segmentation task for a static image segmentation. The goal is to segment objects of interest from the background. The objects of interest are small, variable within the background and background is highly textured, with complex structures.

2.3.1 Training Image Set Building by Context Analysis

This step is conducted in the same way as in section 2.2.1. For this segmentation task, the user must provide training images containing both objects of interest and background.

2.3.2 Static Image Segmentation Knowledge Acquisition by Visual Annotations of Regions

In this section, we focus on the knowledge acquisition for static image segmentation. We use the example-based modeling approach as an implicit representation of the knowledge. This approach has been applied successfully in many applications such as detection and segmentation of objects from specific classes (e.g. Schnitman et al., 2006; Borenstein & Ullman, 2004). Starting from representative patch-based examples of objects (e.g. fragments), modeling techniques (e.g. mixture of gaussians, neural networks, naive bayes classifier) are implemented to obtain codebooks or class-specific detectors for the segmentation of images. Our strategy follows this implicit knowledge representation and associates it with machine learning techniques to train region classifiers. In our case, region annotations represents the high-level information. This approach assumes that the user is able to gather a representative set of manually segmented training images, i.e. a set that illustrates the variability of object characteristics which may be found. The result of a manual segmentation for a training image $I \in \mathbf{I}$ image is noted R_I where R is a set of regions.

First, let the user define a domain class dictionary composed of k classes as $L = \{l_1, \ldots, l_k\}$. This dictionary must be designed according to the problem objectives. Once L is defined, the user is invited, in a supervised stage, to label the regions of the segmented training image with respect to V. From a practical point of view, an annotation is done by clicking into a region r and by selecting the desired class label l. At the end of the annotation task, we obtain a list of labelled regions which belong to classes defined by the user. For each region, a feature vector v_r is also extracted and it makes, with the label a pair sample noted (v_r, l_r). The set of all collected pair samples from $\mathbf{I}$ constitutes the training dataset. This training dataset represents the knowledge of the segmentation task and is composed, at this time, of raw information.

In the following section, we address the problem of knowledge modeling by statistical analysis.

2.3.3 Segmentation Knowledge Modelization

The first step towards learning statistical models from an image partition is extracting a feature vector from each region. But which low-level features are the most representative for a specific partition ? This fundamental question, refering to the feature selection problem, is a key issue of most of the segmentation approaches. As said by Draper in (Draper, 2003), we need to avoid relying on heuristically selected domain features. A popular approach is to combine generic features, such as color, texture and geometric features. The final feature vector representing a region is a concatenation of the feature vectors extracted from each cue.
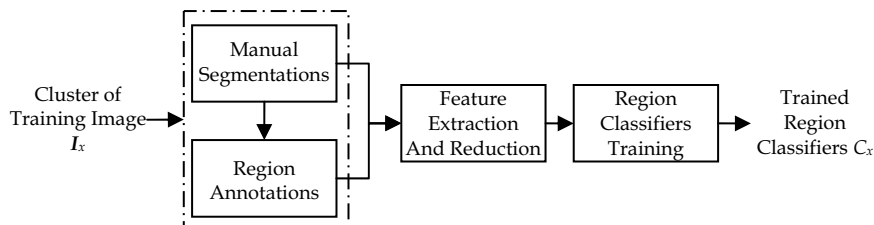

Figure 14. Region Classifier training schema for static image segmentation

Then, applying a feature reduction algorithm, discriminante information is extracted by using a linear component analysis method. In our approach, a generalization of linear principal component analysis, kernel PCA, is exploited to simplify the low-level feature representation of the training dataset T. Kernel PCA was introduced by Scholkopf (Mika et al., 1999) and has proven to be a powerful method to extract nonlinear structures from a data set (Dambreville et al., 2006). Comparing to linear PCA, which may allow linear combinations of features that are unfaithful to the true representation of object classes, kernel PCA combines the precision of kernel methods with the reduction of dimension in the training set. We denote v_r' as the vector of reduced features for the region r.

After reducing feature vector for each region of each training image, the next step is to modelize the knowledge in order to produce region classifiers (one classifier per class) as seen in Figure 14. For a feature vector v_r and a class c,

$$c_l(r) = p(l_r \mid v_r')$$

(9)

with $c_l(r) \in [0,1]$, is the probability estimate associated with the hypothesis: feature vector v_r' extracted from region r is a representative sample of l. The set of these trained region classifiers is noted $C = \{ c_1,\ldots,c_k \}$.

A variety of techniques have been successfully employed to tackle the problem of knowledge modeling. Here we have tested Support Vector Machine (SVM) (Burges, 1998) as a template-based approach. SVM are known to be an efficient discriminative strategy for large-scale classification problems such as in image categorization (Chen & Wang, 2004) or object categorization (Huan & LeCun, 2006). SVM training consists of finding an hyper-surface in the space of possible inputs (i.e. feature vectors labeled by +1 or -1). This hyper-surface will attempt to split the positive examples from the negative examples. This split will be chosen to have the largest distance from the hyper-surface to the nearest of the positive and negative examples. We adopt a one-vs-rest multiclass scheme with probability information (Wu et al., 2004) to train one region evaluator c per class l.

The goal of training region classifiers is not to directly treat the problem of the segmentation as a clustering problem but as an optimization one. Region classifiers express the problem knowledge. Used as performance assessment tools, they define a segmentation evaluation metric. Such functional can then be used in an optimization procedure to extract optimal algorithm parameters. Consequently, we can say that the segmentation optimization is guided by the segmentation task. Next section describes this approach.

2.3.4 Segmentation Knowledge Extraction via Parameter Optimization

While a lot of techniques (Sezgin et al., 2004) have been proposed for adaptive selection of key parameters (e.g. thresholds), these techniques do not accomplish any learning from experience nor adaptation independently of detailed knowledge pertinent to segmentation algorithm. The proposed optimization procedure overcomes such limitations by decomposing the problem into three fundamental and independent components: a segmentation algorithm with its free-parameters to tune, a segmentation evaluation metric and a global optimization algorithm (see Figure 15). To our knowledge, this scheme has already been applied for adaptive segmentation problems by Banu et al. (Bahnu et al., 1995) and by Abdul-Karim et al. (Abdul-Karim et al., 2005). Bahnu et al. used a genetic algorithm to minimize a multiobjective evaluation metric based on a weighted mix of global, local and

symbolic information. Experiments are not very convincing since it has only been tested for one segmentation algorithm and one application (outdoor tv imagery). Abdul-Karim et al. used a recursive random search algorithm to optimize the parameter setting of a vessel-neurite segmentation algorithm. Their system uses the minimum description length principle to trade-off a probabilistic measure of image-content coverage against its conciseness. This trade-off is controlled by an external parameter. The principal limitation of the method is that the segmentation evaluation metric has been defined for the specific task of vessel-neurite segmentation and makes the system unsuitable for other applications. Our approach differs from these ones in the optimization method and above all, in the definition of the evaluation metric.

Let I be an image of the training dataset $\mathbf{I}$, A be a segmentation algorithm and $\mathbf{p}^A$ a vector of parameters for the algorithm A. The result R_I^A of the segmentation of I with algorithm A is defined as:

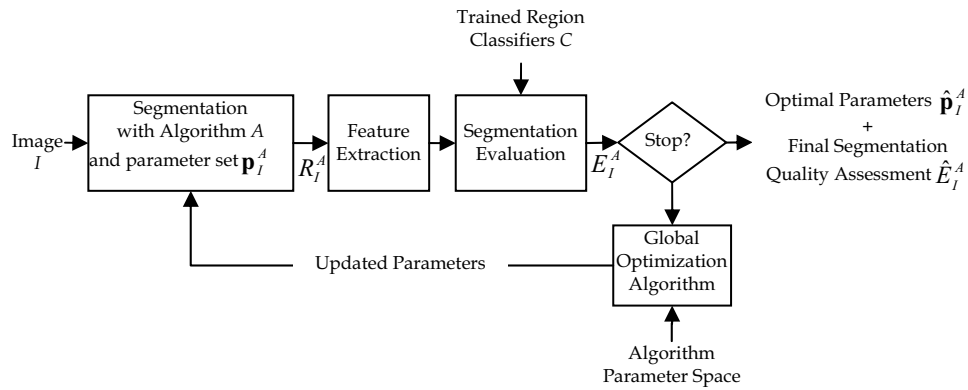$$R_I^A = A(I, \mathbf{p}^A) \tag{10}$$

where R is a set of regions.

Fig 15. Algorithm parameter optimization schema. Given an input image and trained region classifiers, the ouput of the module is the set of optimal parameter for the segmentation algorithm associated with the final segmentation quality assessment value

Several considerations motivate the selection of a direct search method (the simplex algorithm in our implementation) as a preferred strategy compared to other available alternatives. First, exhaustive search is time prohibitive because of the size of the search space. Second, in our approach, the performance metric ρ has no explicit mathematical form and is non-differentiable with respect to $\mathbf{p}_I^A$, mainly because the mapping itself is not differentiable. Thus, standard powerful optimization techniques like Newton-based methods cannot be applied effectively. Simplex algorithm reachs these two conditions: it is able to work on non-smooth functions and the number of segmentation runs to obtain the optimal parameter settings is low (from experiments, under 50 runs in mean).

Let us define the performance evaluation of the segmentation as:

$$E_I^A = \rho(R_I^A, \mathrm{C})$$ (11)

where E_I^A is a scalar, R_I^A the result of the segmentation of I with algorithm A and C the set trained region classifiers. The purpose of the optimization procedure is to determine a set of parameter values $\hat{\mathbf{p}}_I^A$ which minimizes E_I^A:

$$\begin{aligned} \hat{\mathbf{p}}_I^A &= \arg\min_{\mathbf{p}_I^A} \rho(R_I^A, \mathrm{C}) \\ &= \arg\min_{\mathbf{p}_I^A} \rho(A(I, \mathbf{p}^A), \mathrm{C}) \end{aligned}$$ (12)

In order to be goal-oriented, ρ must take into account the knowledge of the problem. In our approach, this knowledge is represented by the set of previously trained region classifiers C. Each region classifiers returns the class membership probability $c(r)$ depending on the feature vector v_r extracted from r. The analysis of the classifier output values allows to judge the quality of the segmentation of each segmented region. The performance metric ρ is then considered as a discrepency measure based of the responses of region classifiers as:

$$\rho(R_I^A, C) = \frac{1}{|I|} \sum_{r_i \in R} |r_i| \cdot \left(1.0 - \max_j c_j(r_i)\right)$$ (13)

where $|I|$ and $|r_i|$ are respectively the image area and the area of the ith region. ρ is borned between zero (i.e. optimal segmentation according to C) and one (i.e. all classifier responses to zero).Our metric takes also into account the region sizes by lowering the weight of small regions.

2.2.3 Adaptive Static Image Segmentation

From a new image and a set of algorithms, the clustering algorithm determines to which context cluster the image belongs to. Then, corresponding region classifiers are used for algorithm parameter optimizations. A set of segmentation assessment values is obtained (one per algorithm). This is used to rank algorithms. Finally, the algorithm with the best assessment value is selected and parametrized with the corresponding optimal parameter set for the segmentation of the image (see Figure 16).
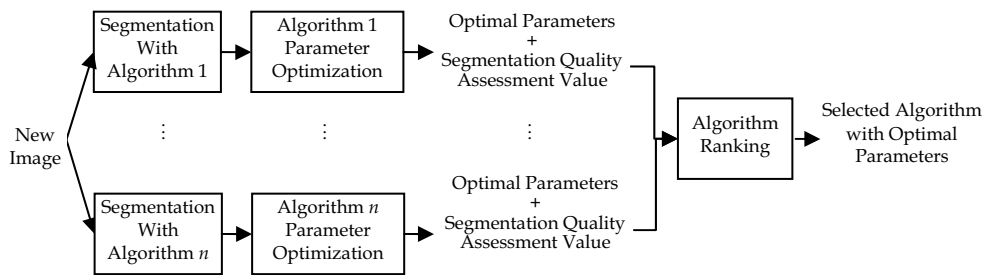

Figure 16. Adaptive static image segmentation schema

Section 3.2 presents experiments of this proposed approach.

3. Experiments

In this section, we present two experiments. The first experiment is a figure-ground segmentation task for video surveillance. It shows the interest of our approach for context adaptation issues. The second experiment is a segmentation task for object detection on static images. The application is in the scope of biological organism detection in greenhouse crops. It shows the interest of our approach for the three issues, i.e. context adaptation, algorithm selection and parameter tuning.

3.1 Figure-ground Segmentation in Video Sequences

The experimental conditions are the followings: the video data are taken during a period of 24 hours at eight frames per second, the field of view is fixed and the video camera parameters are set in automatic mode. In this application our goal is to be able to select the best appropriate background model according to the current context analysis. The size of the images is 352x288 pixels. Our approach is implemented in C++ and a 2,33 GHz Dual Core Xeon system with 4 Go of RAM is used for the experiments.
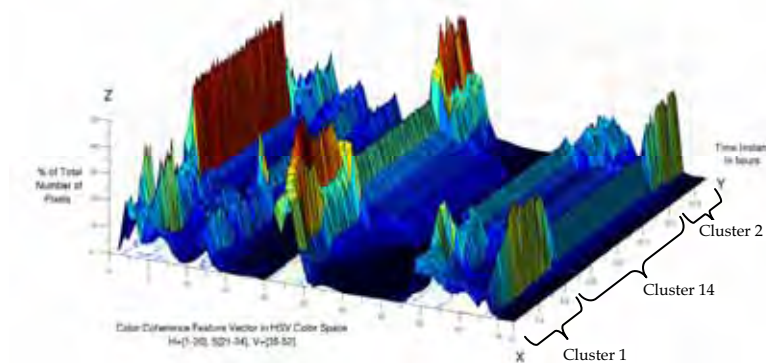

Figure 17. 3-D histogram of the image sequence used during the experiment (see Figure 3 for samples). Each X-Z slice is an histogram which represents the percentage of the number of pixels (Z axis) belonging to a given color coherent feature (X axis). The coherent color feature scale has been divided into 3 intervals for the three HSV channels. Histograms are ordered along the Y axis which represents the time in the course of a day. Several clusters of histograms can be easily visually discriminated as notified for cluster number 1, 14 and 2. Others clusters not represented here are intermediate ones and mainly correspond to transitions states between the three main clusters

In the learning stage, we have manually defined a training image set I composed of 5962 background frames (i.e. without foreground objects) along the sequence. This corresponds to pick one frame every 15 seconds in mean. First, the context clustering algorithm is trained using coherence color feature vectors g_I as inputs. Figure 17 gives a quick overview of the feature distribution along the sequence. Sixteen clusters I_x are found (see Figure 18 for context class distribution). For each cluster, the corresponding frames are put together and automatically annotated by assigning the same (background) label to each pixel. The resulting training data set T is used to train background classifiers C_x (i.e. codebooks).

In the automatic stage the figure-ground segmentation is performed in real-time. For each new frame *I*, context analysis with temporal filtering is used to select a background classifier C_x. Then, background segmentation is computed using the selected C_x. We compute in parallel a static region-based segmentation using the EGBIS algorithm with parameter σ set to 0.2 and parameter *k* set to 100. We use this segmentation to refine the one resulting from the background segmentation. Exemple of segmentation refinement is presented in Figure 19. The testing set is composed of 937 frames different from the training set *I*. We present in Figure 20 four representative results of figure-ground segmentation illustrating different context situations. To show the potential of our approach, we have compared the results obtained with our approach with the results obtained without context adaptation, i.e. using background classifiers trained on the whole sequence. We can see that the detection of moving objects is improved with our approach.
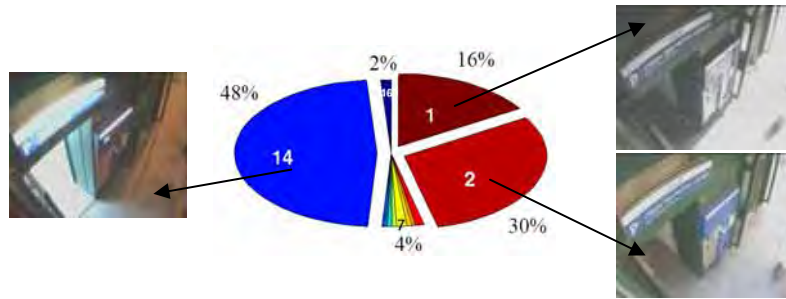

Figure 18. Pie chart of the context class distribution for the image sequence used for the experiments. Three major clusters can be identified (number 1, 2 and 14). The order of class representation does not necessary correspond to consecutive time instants. Cluster 1 corresponds to noon (sunny context), cluster 2 correspond to the morning (lower contrast) and cluster 14 to the night
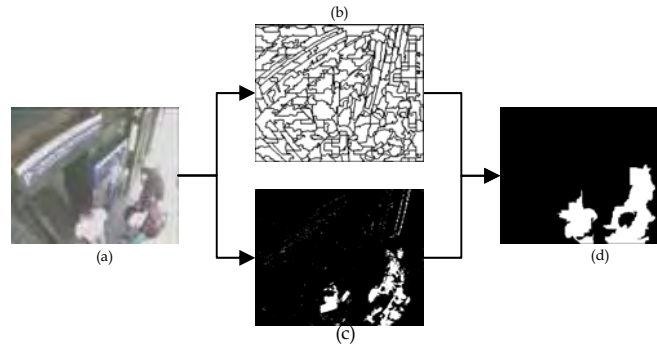

Figure 19. Illustration of the segmentation refinement. An input image (a) is segmented using a region-based segmentation algorithm. The result is presented in (b). In parallel, a figure-ground segmentation (c) is computed using the background classifiers. The final result (d) is a combination of the two segmentations with respect to the criteria defined in Equation 7

Concerning the computational-time, without any optimization of the implementation, the background segmentation takes less than 0,02 second and the region-based segmentation takes 0,4 second. The total processing time allows to segment two frames per second in mean. This validates our approach for real-time applications.

(a)   Context 2

(b)   Context 6

(c)   Context 8

(d)   Context 3

Figure 20. Segmentation results illustrating different context situations. Boundaries of the detected foreground regions (mobile objects) are shown in red. Images of the left column are those obtained without context adaptation. Images of the right column are segmentation results with context adaptation. The third column corresponds to the identified context cluster. We can see that the persons are better detected using our method (rows a, c and d). Moreover, false detection are reduced (rows b, c and d)

3.2 Static Image Adaptive Segmentation

This experiment is related to a major challenge in agronomy: the early pest detection in rose crops (Boissard et al., 2003). The experimental conditions are the followings: images are obtained from scanned rose leaves. The objects of interest are white flies (Trialeurode vaporariorum) at mature stage. The white fly wings are half-transparent and the insect has many appendices as antennas and paws. They are shown from different points of view. The image background (i.e. the rose leaf) is highly structured and textured and also varies in color in function of the specy and the age of the plant. Concerning the set of segmentation algorithms used for this experiment, we have selected from the literature (Freixenet et al, 2002), four algorithms which illustrate different state-of-the-art approaches of image segmentation: Efficient Graph-Based Image Segmentation (Felzenszwalb & Huttenlocher, 2004), Color Structure Code (Priese et al., 2002), Statistical Region Merging (Nock & Nielsen, 2004) and Color Watershed Adjency Graph Merge (Alvarado, 2001). They are summarized in Table 1, along with their free parameters and default values used in our experiment.

Algorithm	Free Parameter	Range	Default Value
EGBIS	σ: smooth control on input image	0.0-1.0	0.50
	k: color space threshold	0.0-2000.0	500.0
CSC	t: region merging threshold	5.0-255.0	20.0
SRM	Q: coarse-to-fine scale control	1.0-255.0	32.0
CWAGM	M: Haris region merge threshold	0.0-2000.0	100.0
	k: Haris minimal region number	1.0-100.0	10.0
	t: Min prob for wathershed threshold	0.0-1.0	0.45

Table 1. Components of the segmentation algorithm bank, their names, parameters to tune with range and default values

In the learning stage, we have defined a training image set I composed of 100 sample images of white flies over rose leaves. The size of an image is 350x350 pixels. First, the context clustering algorithm is trained using coherence color feature vectors g_I as inputs. We have obtained four context clusters. Each training image cluster is manually segmented into regions by marking white fly boundaries out. This represent a total of 557 regions. Then, each region is annotated with a *white fly* or a *leaf* label and a feature vector v_r is extracted. We compute the (H,S,V) histogram of the region pixel values quantified into 48 bins (i.e. 16 bins per channel). Each cluster of feature vectors is reduced by using kernel PCA. The size of a reduced feature vector v_r' varies from 22 features to 28 depending on the context cluster. Then, the region classifiers C_x are trained using the linearly scaled feature vectors v_r'. We use SVM with radial basis function (RBF) as region classifiers. To fit the C and γ parameters of the RBF kernel to the problem data, we perform a five fold cross-validation on training data to prevent overfitting problems.

In the automatic stage, a new image I is initially segmented with an algorithm A tuned with default parameters $\mathbf{p}^A$ (i.e. with values given by the author of the algorithm). Then, parameter optimization is achieved and returns an optimal parameter set $\hat{\mathbf{p}}_I^A$ and a segmentation quality assessment quality value E_I^A as output. Once all segmentation algorithm parameter optimizations are processed, we can rank the segmentation algorithms in accordance to their E_I^A. This algorithm selection technique is illustrated in Figure 21. We

can see that for the four presented algorithms, the assessment values are very closed. This is in accordance with the visual observation of the results. The small differences between the algorithms can be explained by the detection (or not) of small appendices of white flies (e.g. antenna, paw). We also see that the SRM algorithm gets the best result (i.e. the smallest assessment value) without performing the finest segmentation (appendices are not detected). This is mainly due to the fact that white fly classifiers have been trained with manually segmented regions for which, most of the time, small details like the appendices are missed. Consequently, segmentation is better evaluated when the appendices are not parts of white fly regions.

original test image	EGBIS	CSC

SRM	CWAGM	Algorithm segmentation quality assessment values: (0=perfect, 1=null)

Algorithm segmentation quality assessment values: (0=perfect, 1=null)

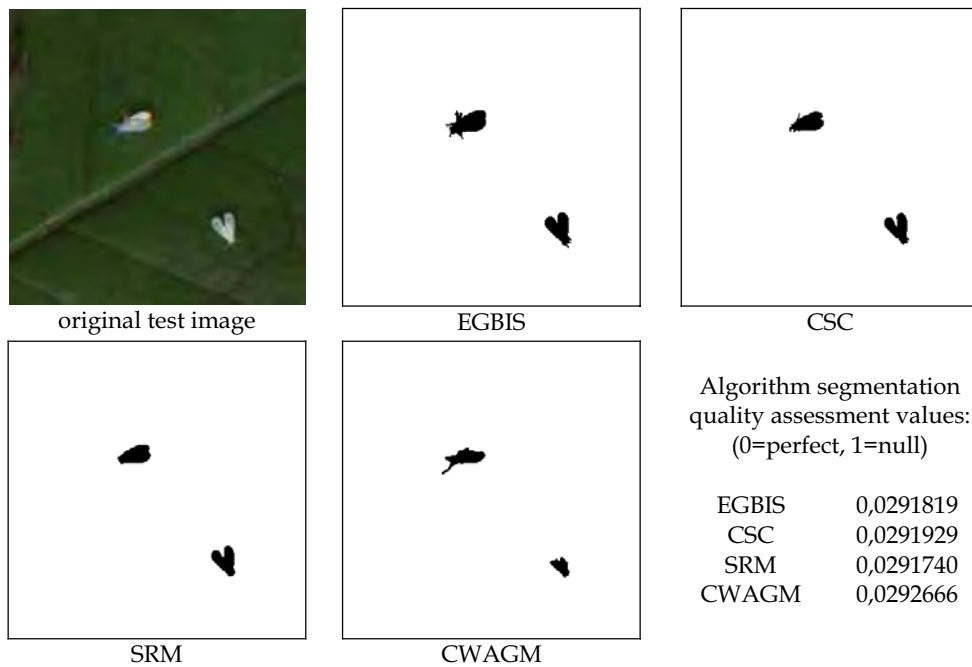|	
EGBIS	0,0291819
CSC	0,0291929
SRM	0,0291740
CWAGM	0,0292666

Figure 21. Segmentation results from test samples illustrating the algorithm selection issue. After parameter optimization, final algorithm segmentation quality assessment values can be compared to rank the algorithms

Figure 22 is shown to illustrate the parameter tuning issue. We clearly see that optimization of parameters is useful and tractable for different segmentation algorithms. However, we can see for the first image of Figure 22 that two white flies are miss-detected. This discrepancy has two explanations: first, it reveals that classifiers have not been trained enough and second, that our dictionary does not discriminate enough differences between classes. The first issue can be achieved by training classifiers on more training images and the second issue can be achieved by using more specific classes as one classe for each white fly body parts (e.g. head, wings and abdomen). Obviously, this also demands more efforts to the user.

Regarding the conputation-time, we have used the same hardware system than in section 3.2. Both context analysis and algorithm ranking are inconsiderable (less than 0,01 second).

An optimization closed-loop takes between 5 and 35 seconds for an image. The duration depends on the algorithm segmentation computation-time (between 0.08 second and 0.8 second), the number of iterations (between 8 and 50) and the segmentation evaluation time depending on the number of regions to process (between 1 and 300). So the total processing time of the automatic adaptive segmentation is between 5 and 35 seconds, using the same system as in section 3.1.
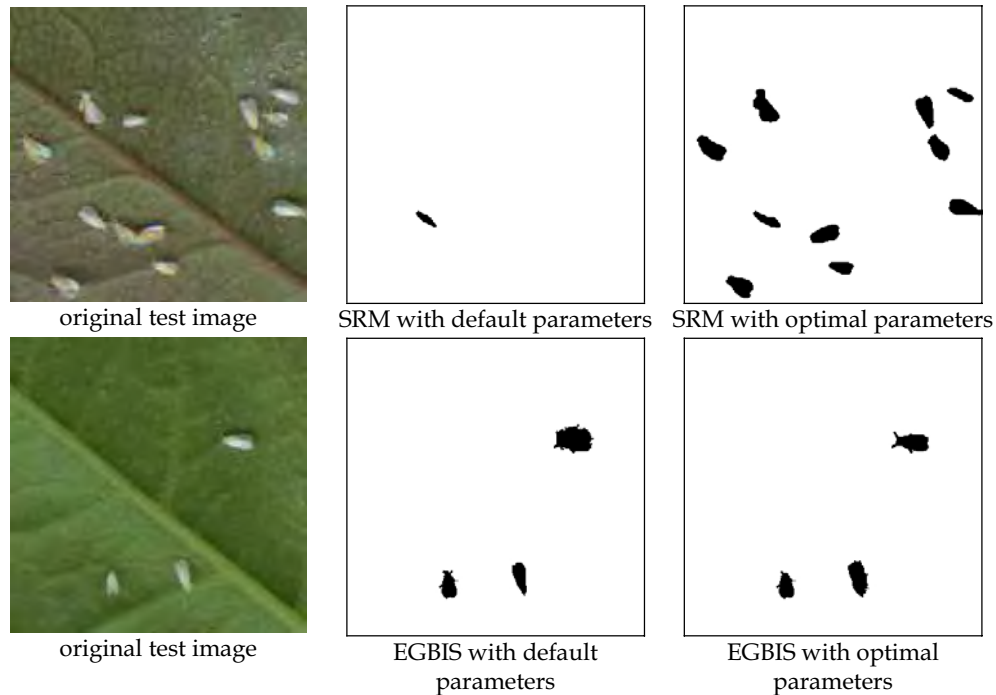

| original test image | SRM with default parameters | SRM with optimal parameters |

| original test image | EGBIS with default parameters | EGBIS with optimal parameters |

Figure 22. Segmentation results from test samples illustrating the algorithm parameter tuning issue. For two different images , two algorithms are first run with their default parameters (central column). Results after the parameter optimization step are presented in the last column. We can see that the detection of the object of interest is better with optimal parameters than with the default parameters

4. Conclusion and Discussion

In this chapter, we have proposed a learning approach for three major issues of image segmentation: context adaptation, algorithm selection and parameter tuning according to the image content and the application need. This supervised learning approach relies on hand-labelled samples. The learning process is guided by the goal of the segmentation and therefore makes the approach reliable for a broad range of applications. The user effort is restrained compared to other supervised methods since it does not require image processing skills: the user has just to click into regions to assign labels; he/she never interacts with algorithm parameters. For the figure-ground segmentation task in video application, this annotation task is even automatic. When all images of the training set are labelled, a context

analysis using an unsupervised clustering algorithm is performed to divide the problem into context clusters. This allows the segmentation to be more tractable when context is highly variable. Then, for each context cluster, region classifiers are trained with discriminative information composed of a set of image features. These classifiers are then used to set up a performance evaluation metric reflecting the segmentation goal. The approach is independent of the segmentation algorithm. Then, a closed-loop optimization procedure is used to find algorithm parameters which yield optimal results.

The contribution of our approach is twofold: for the image segmentation community, it can be seen as an objective and goal-oriented performance evaluation method for algorithm ranking and parameter tuning. For computer vision applications with strong context variations (e.g. multimedia applications, video surveillance), it offers extended adaptability capabilities to existing image-sequence segmentation techniques.

The ultimate goal of this approach is to propose the best available segmentation for a given task. So, the reliability of the approach entirely depends on the inner performance of the segmentation algorithms used. One other limitation of the approach is that the adaptability ability is depending on the sampling of the training data. More the training dataset is representative of different contexts, more the system will be precise to select and tune the algorithms.

Future works consist in improving these issues. For instance, incremental learning could be used to learn on-the-fly new situations and then enrich the knowledge of the problem. In this chapter, we have proposed a method based on class models of visual objects. This method exploits features in a discriminative manner. For very difficult cases where intra-class information (i.e. object appearance) is very heterogeneous and/or inter-class information is poorly discriminative, selection of representative features is tricky and leads to poor performances. In this case, approaches based on shared visual features across the classes as boosted decision stumps should be more appropriated and effective. Finally, by addressing the problem of adaptive image segmentation, we have also addressed underlying problems such as feature extraction and selection, segmentation evaluation and mapping between low-level and high-level knowledge. Each of these well-known challenging problems are not easily tractable and still demands to be intensively considered. We have designed our approach to be modular and upgradeable so as to take advantage of new progresses in these topics.

5. Acknowledgments

6. References

Abdul-Karim, M.-A. and Roysam, B. and Dowell-Mesfin, N.M. and Jeromin, A. and Yuksel, M. and Kalyanaraman, S. (2005). Automatic Selection of Parameters for Vessel/Neurite Segmentation Algorithms, *IEEE Transactions on Image Processings*, Vol.14, No.9, September 2005, pp. 1338-1350, ISSN: 1057-7149

Alvarado, P. and Doerfler, P. and Wickel, J. (2001). Axon2 - A visual object recognition system for non-rigid objects, *Proceedings of the International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA)*, pp.235-240, ISBN: 0-88986-293-1, July 2001, Rhodes, Greece

Bahnu, B. and Lee, S. and Das, S. (1995). Adaptive Image Segmentation Using Genetic and Hybrid Search Methods, *IEEE Transactions on Aerospace and Electronic Systems*, Vol.31, No.4, October 1995, pp.1268-1291, ISSN: 0018-9251

Boissard, P. and Hudelot, C. and Thonnat, M. and Perez, G. and Pyrrha, P. and Bertaux, F. (2003). An automated approach to monitor the sanitary status and to detect early biological attacks on plants in greenhouse – Examples on flower crops, *Proceedings of the International Symposium on Greenhouse Tomato*, Avignon, France, September, 2003

Borenstein, E. and Ullman, S. (2004). Learning to segment, *Proceedings of European Conference on Computer Vision*, pp. 315-328, ISBN 978-3-540-21982-8, Prague, Czech Republic, May 2004, Springer, Berlin / Heidelberg

Burges, Christopher J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery,* Vol.2 ,No.2 , pp.121-167, June 1998, Kluwer Academic Publishers, Hingham, MA, USA, ISSN:1384-5810

Chen, Y. and Wang, James Z. (2004). Image Categorization by Learning and Reasoning with Regions, *The Journal of Machine Learning Research* , Vol.5, pp.913-939, December, MIT Press, Cambridge, MA, USA, ISSN:1533-7928

Clouard, R. and Elmoataz, A. and Porquet, C. and Revenu, M. (1999). Borg: A knowledge-based system for automatic generation of image processing programs , *IEEE Transaction on Pattern Analysis and Machine Intelligence,* Vol. 21, No. 2, February, 1999, pp. 128-144, ISSN: 01 62-8828

Dambreville, S. and Rathi, Y. and Tannenbaum, A. (2006). Shape-Based Approach to robust Image Segmentation using Kernel PCA, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol.1, pp.977-984, ISBN: 0-7695-2597-0, ISSN: 1063-6919, June 2006, IEEE Computer Society, Washington, DC, USA

Draper, Bruce A. (2003). From Knowledge Bases to Markov Models to PCA, *Proceedings of Workshop on Computer Vision System Control Architectures*, Graz, Austria, March 2003

Elgammal A., Harwood D. and Davis L. (2000). Non-parametric Model for Background Subtraction, *Proceedings of the 6th European Conference on Computer Vision-Part II*, Vol.1843, pp. 751-767, ISBN:3-540-67686-4, Dublin, Ireland, June 2000, Springer-Verlag, London, UK

Ester M., Kriegel H-P., Sander J., Xu X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226-231, Portland, Oregon, USA, 1996, AAAI Press, ISBN 1-57735-004-9

Felzenszwalb, Pedro F. and Huttenlocher, Daniel P. (2004). Efficient Graph-Based Image Segmentation, *International Journal of Computer Vision*, Vol.59, No.2, September 2004, pp. 167-181, Kluwer Academic Publishers, Hingham, MA, USA, ISSN: 0920-5691

Freixenet J., Muñoz X., Raba D., Martí J., Cufí X. (2002). Yet Another Survey on Image Segmentation: Region and Boundary Information Integration, *Proceedings of the 7th European Conference on Computer Vison Part III*, Vol. 2352/2002,pp.408-422, Copenhagen, Denmark, May 2002, Springer Berlin / Heidelberg, ISSN: 0302-9743

Georis, B. (2006). Program supervision techniques for easy configuration of video understanding ststems, Ph. D. Thesis. Louvain Catholic University, Belgium, January 2006

Grimson W.E.L and Stauffer C. (1999). Adaptive background mixture models for real-time tracking, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.2, pp.252, ISBN: 0-7695-0149-4, Fort Collins, CO, USA, June 1999, IEEE Computer Society

Huang Fu Jie and LeCun Y. (2006). Large-scale Learning with SVM and Convolutional Nets for Generic Object Categorization, *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.1, pp. 284 – 291, ISBN:1063-6919, New York, USA, June 2006, IEEE Computer Society, Washington, DC, USA

Kim, K. and Chalidabhongse, T.H. and Harwood, D. and Davis, L.S. (2005). Real-time foreground-background segmentation using codebook model, *Real-time imaging* , Vol.11,No.3,pp.172-185, June 2005, Elsevier, Oxford, United Kingdom, ISSN:1077-2014

Mika, S. and Scholkopf, B. and Smola, A. (1999). Kernel PCA and De-noising in Feature Spaces, *Proceedings of the 1998 conference on Advances in neural information processing systems II*, Vol.11, pp. 536-542, ISBN: 0-262-11245-0, Denver, Colorado, USA, December 1998, MIT Press, Cambridge, MA, USA

Martin, V. Thonnat, M. Maillot, N. (2006). A Learning Approach for Adaptive Image Segmentation, *IEEE International Conference on Computer Vision Systems,* pp.40, ISBN: 0-7695-2506-7, New York, NY, USA, January 2006, IEEE Computer Society

Nock, R. and Nielsen, F. (2004). Statistical Region Merging , *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.26, No.11, November 2004, pp.1452-1458, IEEE Computer Society , Washington, DC, USA, ISSN:0162-8828

Pass, G. and Zabih, R. and Miller, J. (1997). Comparing Images Using Color Coherence Vectors, *Proceedings of the fourth ACM international conference on Multimedia*, pp.65-73, ISBN: 0-89791-871-1, Boston, Massachusetts, United States, November 1996, ACM Press, New York, USA

Priese, Lutz Rehrmann, Volker and Sturm, P. (2002). Color Structure Code, url = http://www.uni-koblenz.de/

Sezgin M., Sankur B. (2004). Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging, Vol.13, pp. 146-165, January 2004

Schnitman, Y. and Caspi, Y. and Cohen-Or, D. and Lischinski, D. (2006). Inducing Semantic Segmentation from an Example, *Proceedings of the 7th Asian Conference on Computer Vision - ACCV*, Vol.3852, No.22,pp. 384-393, Hyderabad, India, January 13-16, 2006, Springer-Verlag

Thonnat, M. and Moisan, S. and Crubezy M. (1999). Experience in integrating image processing programs, *Proceedings of the First International Conference on Computer Vision Systems*, pp.200-215, ISBN: 3-540-65459-3, Henrok Christensen, Las Palmas Gran Canaria, Spain, January 1999, Springer-Verlag

Wu T. and Lin C. and Weng R. (2004). Probability estimates for multi-class classification by pairwise coupling, *The Journal of Machine Learning Research*, Vol.5, December 2004, pp.975-1005, MIT Press, Cambridge, MA, USA, ISSN: 1533-7928