



Understanding leaves in natural images – A model-based approach for tree species identification ☆,☆☆



Guillaume Cerutti^{a,b,*}, Laure Tougne^{a,b}, Julien Mille^{a,c}, Antoine Vacavant^d, Didier Coquin^e

^a Université de Lyon, CNRS, France

^b Université Lyon 2, LIRIS, UMR5205, F-69676 Bron Cedex, France

^c Université Lyon 1, LIRIS, UMR5205, F-69622 Villeurbanne Cedex, France

^d Clermont Université, Université d'Auvergne, ISIT, F-63001 Clermont-Ferrand, France

^e LISTIC, Domaine Universitaire, F-74944 Annecy le Vieux, France

ARTICLE INFO

Article history:

Received 16 May 2012

Accepted 8 July 2013

Available online 20 July 2013

Keywords:

Plant recognition

Tree leaf

Image segmentation

Natural background

Active contours

Geometric features

Classification

ABSTRACT

With the aim of elaborating a mobile application, accessible to anyone and with educational purposes, we present a method for tree species identification that relies on dedicated algorithms and explicit botany-inspired descriptors. Focusing on the analysis of leaves, we developed a working process to help recognize species, starting from a picture of a leaf in a complex natural background. A two-step active contour segmentation algorithm based on a polygonal leaf model processes the image to retrieve the contour of the leaf. Features we use afterwards are high-level geometrical descriptors that make a semantic interpretation possible, and prove to achieve better performance than more generic and statistical shape descriptors alone. We present the results, both in terms of segmentation and classification, considering a database of 50 European broad-leaved tree species, and an implementation of the system is available in the iPhone application Folia.¹

© 2013 Elsevier Inc. All rights reserved.

1. Foreword

Advances in technology and prevalence of urban settlement undoubtedly contributed to widen a gap between nature and us that was inconceivable a couple of generations ago. Along the triumphant march of alleged progress, we carelessly dropped the knowledge of many plants, trees and herbs that used to be found in our most immediate environment. Knowing how to recognize a plant, and what uses it may provide, is now the mere privilege of a handful of specialists. Today however, with the growing awareness that nature's diversity ought to be treasured, the need to regain some touch with this knowledge never felt so present.

Books are available to the neophyte, but when it comes to precise species identification, it is easy to get lost in the bushy floras, without the minimal theoretical background. However, with the rise of mobile applications that have the potential, yet scarcely

exploited, to place powerful resources in everyone's pocket, the access to such knowledge can be made much more intuitive and flexible for anyone who feels the need. Recently, attempts in this direction have come to light with great success, being with user-based (TreeId,² Fleurs en Poche³) or automatic recognition (Leaf-Snap⁴) on white background images.

Leaves are the most obvious and widespread choice for tree species recognition, even though the botanical classification was not built upon their properties. They can be found almost all year long, are easy to photograph, and their shapes present well studied specificities that make the identification, if not trivial, possible. Our goal with the Folia application is then to build a system for leaf shape analysis that processes, unlike what has been done to date, pictures in a natural environment. With the aim of being an educational tool, it relies on high-level geometric criteria inspired by those used by botanists, that make a semantic interpretation possible, to classify a leaf into a list of species.

In this paper, we propose an attempt for such a system. Other works tackling similar challenges are presented in Section 2, and a global view of our system is presented in Section 3. Section 4

* This paper has been recommended for acceptance by Edwin R. Hancock.

** This work has been supported by the French National Agency for Research with the reference ANR-10-CORD-005 (ReVeS project).

* Corresponding author. Address: LIRIS, Université Lumière Lyon 2, 5 Avenue Pierre Mendès France, 69676 Bron Cedex, France. Fax: +33 4 78 77 23 38.

E-mail addresses: guillaume.cerutti@liris.cnrs.fr (G. Cerutti), laure.tougne@liris.cnrs.fr (L. Tougne), julien.mille@liris.cnrs.fr (J. Mille), antoine.vacavant@iut.u-clermont1.fr (A. Vacavant), didier.coquin@univ-savoie.fr (D. Coquin).

¹ <https://itunes.apple.com/app/fofia/id547650203>.

² <http://isoperla.co.uk/TreeldiPhone.html>.

³ <http://www.antiopa.info/fleursenpoche.htm>.

⁴ <http://leafsnap.com>: developed by researchers from Columbia University, the University of Maryland, and the Smithsonian Institution [1].

introduces the color model we use for segmentation, and Section 5 expounds the leaf model estimation, and the subsequent active contour segmentation algorithm. Section 6 details the leaf-specific contour descriptors we introduce. The classification method we use and its results are shown in Section 7 and perspectives of our work are discussed in Section 8.

2. Related works

2.1. Leaf recognition

Plant recognition and leaf image retrieval have recently been topics of interest for many works in image processing, though most of them limit the problem to leaves on a plain background. Challenges for the community have even been organized such as the ImageCLEF Plant Identification Task [2] since 2011. Some authors [1] also pursued the goal of conceiving a mobile guide with the LeafSnap application, combining established shape descriptors (Inner-Distance Shape Context) and classification methods to perform remarkably well on white-background images. Our goal however is to deal with the kind of complex background images one can take out in the nature.

Many works similarly rely on statistical features such as moments [3], histogram of gradients or local interest points [4,5] or generic shape descriptors such as the Curvature-Scale Space [6]. Such descriptors were not designed to take into account the nature of the object, even if they fit quite well with its specificities.

A commonly used geometrical descriptor for leaf image retrieval is the Centroid-Contour Distance (CCD) curve [7,8] though it can be applied to any type of object. On the other hand, some morphological features explicitly computed on the shape of the object to model its natural properties have also been used [9,10,5]. Even more dedicated methods have been designed, basing their recognition on an explicit representation of the leaf contour and of its teeth [11]. Most of these descriptors are used on white background images where robustness to noise is not an issue since retrieving the exact contour is something trivial.

2.2. Plant segmentation

There are still some authors addressing the hard task of isolating a natural object from a natural environment. Luminosity conditions, complexity of the environment, and presence of very similar objects around the one of interest make it a very difficult problem where generic unconstrained segmentation methods are very likely to fail, as shows Fig. 1.

Considering plants, flowers are obviously an easier target to segment, as their color generally clearly stands out. Introducing a geometric shape model [4] or putting constraints on the image acquisition [12] seems however necessary to perform well.

The case of leaves appears even more challenging, and the use of complex methods, such as 3D reconstruction to take depth into account [8], or multiple step segmentation processes [3], is inevitable to overcome the difficulties. The use of deformable templates to include as much prior knowledge as possible has been used in the restricted case of weed leaves [13] but is a very convincing way to deal with occlusions, overlaps and other problems posed by natural images. Our work follows the same idea, making use of active models to isolate a shape on which we have some notion beforehand.

2.3. Active contour models

The first formulation of such models goes back to active contours, or snakes [14], introduced as a way to solve edge detection problems. They rely on the minimization of an energy functional to adjust splines to the edges in the image. This energy classically presents a two term formulation, summing an internal energy term that constrains the properties (regularity, smoothness) of the contour, and an external or image energy term whose goal is to make sure that the contour corresponds to the content of the image, based on the gradient intensity.

The evolution of the contour can be based on other image characteristics, color consistency is a possible choice when objects are not well defined by gradient [15] using a level-set formulation for the contour [16]. Texture information has also been used [17], the contour being represented in this case by a polygon.

But it can be very beneficial, when some prior information about the shape of the target object is available, to include this knowledge in the evolution of the contour. A deformable template modeling complex objects [18] is a solution that additionally presents the advantage of compressing the representation of the contour by the use of parameters. When objects have a well-determined shape, a strongly constrained template, such as active shape models [19], learned compositions of deformable triangulated polygons [20] or level-set active contours with shape priors [21], is a powerful solution even when the object is occluded. With such methods including prior knowledge, shapes need to be known precisely and not to vary, and they consequently lack the necessary flexibility and expressiveness to capture the diversity of leaf shapes.

In the following section, we give an overview of the proposed method for leaf segmentation and recognition which is based on active contour models.

3. Leaf recognition overview

3.1. Framework

To cope with the hardly solvable challenge of segmenting and identifying leaves in complex images, we decided to limit our focus



Fig. 1. Segmentation issues for leaves on natural background without prior knowledge; results obtained with unconstrained active contours.

to **simple** and **palmately lobed** leaves only, reducing the scope to some 80% of the nearly 200 European native broad-leaved tree species. Our method is designed to process photographs of leaves **in a natural background**, taken by **mobile cameras**, but it is still able to perform on white or plain background images. In any case, we assume that the leaf is **roughly centered** and **vertically-oriented** in the image, so that we start from an area where we are certain to find the leaf. This assumption is anyway legitimate when we consider our final objective of offering a mobile application in which the user is guided for the picture taking as illustrates Fig. 2. The algorithms we propose have to **run on the mobile processor** and consequently fit its requirements in terms of space and execution time.

3.2. Botany as a pattern for recognition

Tree species recognition using leaf features is a well studied problem, when we look at it from a botanist's view. If the expert will identify the tree at a glance, a well established identification process exists to guide the untrained amateur, through the use of dichotomous keys. The diligent user will be asked to look for several significant traits over the leaf (size, global shape, number of lobes, venation, basal and apical shapes, type of margin, ...) following the path towards the species.

These characteristics are described using a dedicated terminology referring generally to canonical, ideal shapes. They are not the exact shapes that are found in nature but rather a human interpretation focusing only on the relevant details. For instance the presence or absence of teeth around the apex does not change the denomination of the apical shape though it certainly creates differ-

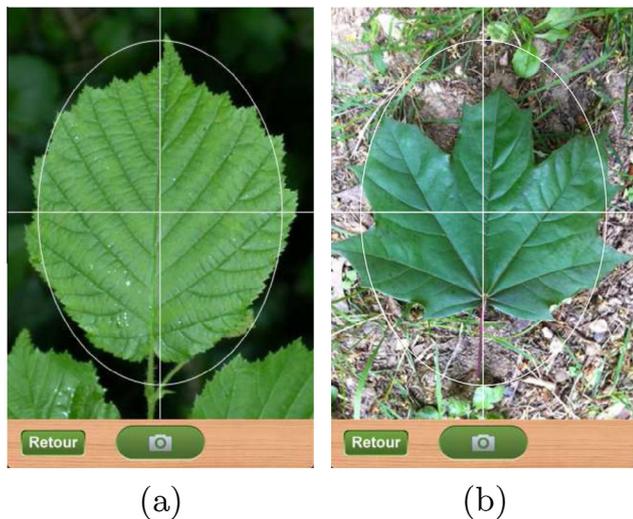


Fig. 2. Example of the picture taking interface for the mobile application, on simple (a) and on a palmately lobed (b) leaves.

ent shapes. The different features are considered independently from one another, an easy abstraction for the human eye. There is consequently a part of subjectivity and vagueness in the use of those botanical terms, but they constitute a determining clue as to where to look at and what to look for.

Knowing in advance the whole set of morphological features one has to search in a leaf opens the possibility of modelling shapes beforehand accordingly. The introduction of specific shape models from the very beginning of the process has the advantage of including prior knowledge, acting like a constraint to ensure that the resulting shapes will correspond to what can be found in the nature. It is also a way of putting semantics on what is extracted in the image, since the models are constructed at the same level of abstraction as the botanist's interpretation.

The method we propose to identify leaves is a transposition of this idea. It relies on several models to represent the different botanical criteria defining the shape of a leaf, starting with the global shape, estimated first to guide the segmentation. Each of them focuses on one single element of interest and is designed to avoid taking the other ones into account. This decomposition of shape features into independent characteristics, later to be combined together, fits with the challenges of leaf recognition, where the intra-class variability for a given species may be high for some botanical features whereas other ones will be very discriminating. This specificity is displayed in Fig. 3. This is the reason why it is important to decorrelate features as they are extracted in order to reduce the amount of noise that would prejudice the classification.

3.3. Process overview

As depicted in Fig. 4, the identification process can be divided into four main phases. The first one consists in building a representation of what will make the leaf identifiable in the image. We chose to base this representation on color only and estimate a **model of the color** of the leaf providing information on what may or may not be part of the leaf. The segmentation method directly relies on this representation. Its first step is based on a highly variable **polygonal model** representing the global shape of the leaf. The estimation of this model is performed using the color representation, and later used to retrieve the actual leaf contour. This final segmentation step can be seen as a deformation of the rigid polygonal shape into a contour adjusting to teeth on the leaf margin. The choice of an active contour starting from the polygonal contour and evolving towards the margin is a natural one. The model capturing the global shape acts as a strong shape constraint, which avoids common shortcomings of unsupervised segmentation methods shown in Fig. 1.

To describe the shape obtained after the segmentation, and capture the **discriminating features** of a species, we model successively the characteristics of the margin, base and apex, after interpreting the final contour using the Curvature-Scale Space. To describe those features we introduce dedicated representations, that rely on existing appropriate shape descriptions to capture



Fig. 3. Variability of leaf shapes within a single species: *Populus alba* L..

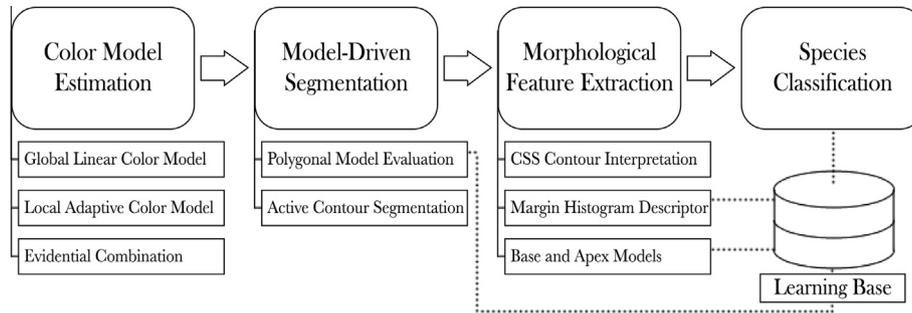


Fig. 4. Overview of the tree leaf identification system.

the interesting morphological properties. The different and independent leaf features are used to build **species models**, and combined in a simple manner to measure their very contribution, without the help of a sophisticated classification algorithm. In the following, we successively detail each of these steps.

4. Leaf color representation

To guide the first phase of the segmentation, we made the decision to rely on the color information. The gradient information is a useful tool, but its efficiency in our case is reduced by the venation that often creates strong edges inside the leaf, and by the fact that, since we start by a polygonal approximation of the leaf's shape, it will hardly be possible to fit exactly the actual edges of the leaf. Texture is another alternative, but most leaves present large-scale texture elements which are consequently detected with less precision, being moreover different over the surface of the leaf.

Building an *a priori* color model that would be always valid, independently of the species, lighting and season is not an achievable goal. It is therefore essential to build a custom color model for each and every leaf we want to process. In our case, we have a good idea of the leaf's location in the first place, ensuring that the central area of the image will contain only leaf pixels. Starting from this initial area, it is possible to build a representation of the leaf's color. The method we present here differs from our previous approach [22] where a 2-Gaussian model was used. That view proved to lack accuracy, and our color representation now tries to estimate both a global color model, that separates leaf pixels from background pixels, and a local mean color model, that focuses on the leaf of interest, and to combine them in a relevant way using belief functions.

4.1. Global linear regression model

The $L*a*b^*$ colorspace is a perceptually intuitive alternative to the regular RGB color representation, and separates well the chrominance information from the luminance. We observed the repartition of leaf pixels in various colorspace ($L*a*b^*$, RGB, HSL) and it appeared that the color distribution generally forms a straight continuous cylinder in $L*a*b^*$, whereas it appears appreciably more curved in the other spaces.

One plausible way to model a leaf's color is then to estimate the main direction of this seeming cylinder. It is important to consider that leaves often contain reflections or shadows which present colors that might not be found in an initial leaf model. Our observations however show that these colors are generally consistent with the colors in a smaller initial area, and that the $L*a*b^*$ cylinder presents roughly the same axis for the whole leaf and for a limited part of the leaf.

It is in any case essential to have a model with the ability to extrapolate colors outside the initial area, and estimate the whole cylinder and not only an ellipsoid. Computing a linear regression

over the $L*a*b^*$ pixels and using it to predict colors is then a good way to achieve this generalization. Fig. 5 shows how the $L*a*b^*$ colorspace is best suited for linear regression, as the prediction line stays almost the same if we consider a small part of the leaf (containing a wide enough luminosity range) or the whole leaf, whereas this line tends to move in the other spaces.

Concretely, given an initial region Ω_0 , we apply a standard linear regression to the pixels $(L_x, a_x, b_x)_{x \in \Omega_0}$ where L is considered as the explanatory variable for both variables a and b . We obtain a color prediction given the luminosity $\hat{a}(L) = a_0 + a_1L$ and $\hat{b}(L) = b_0 + b_1L$ based on which it is possible to compute a dissimilarity map (as shown in Fig. 7) by calculating the distance of every pixel in the image to the color model: $d_{LinReg}(x) = \|(L_x, a_x, b_x) - (L_x, \hat{a}(L_x), \hat{b}(L_x))\|_2$.

This color representation has the double-edged properties of generalization, making it possible for leaf pixels that were not present in the initial area to be considered as close, but risking in the same time to include non-leaf pixels that would fit the model. It remains a global view of the leaf's color that does not consider local distribution changes, which makes it in particular unable to discriminate the leaf of interest from other leaves of similar color in the background.

4.2. Local adaptive mean color

Color inside a leaf may vary. Reflections, venation, shadows or spots may create brutal changes, but the main variation along the surface of the leaf is caused by orientation and illumination differences and is a progressive change. A global color model may be able to acknowledge all these varying colors as leaf colors, but it fails to capture the fact that the leaf may present a specific color only in a given area of the image. And if this color is found in another part of the image, it should not necessarily be considered as a leaf color there. A rigid global model is then not enough, and it is also necessary to have a model that adapts to progressive local color variations.

We build such a model by exploring the image, trying to answer the question "what would be the most probable color in this part of the image if the considered pixel was a leaf pixel?" We start from the same initial region as previously, in which we consider this local expected mean $(\bar{L}_x, \bar{a}_x, \bar{b}_x)$ to be known, taking the mean color of a neighborhood around each pixel. Then, the idea is to propagate, and update if necessary, this mean through the image. As long as the color variation between the pixel and the local expected mean is limited, we consider that the leaf's color is locally changing, and that this change has to be taken into account in the color we expect to find further on. If the change is too brutal, it is certainly not a natural variation of the leaf's color, and we should not change the expected color for the following pixels.

This method relies then on an adaptivity parameter α that defines how fast the mean will adapt to changes, and a tolerance

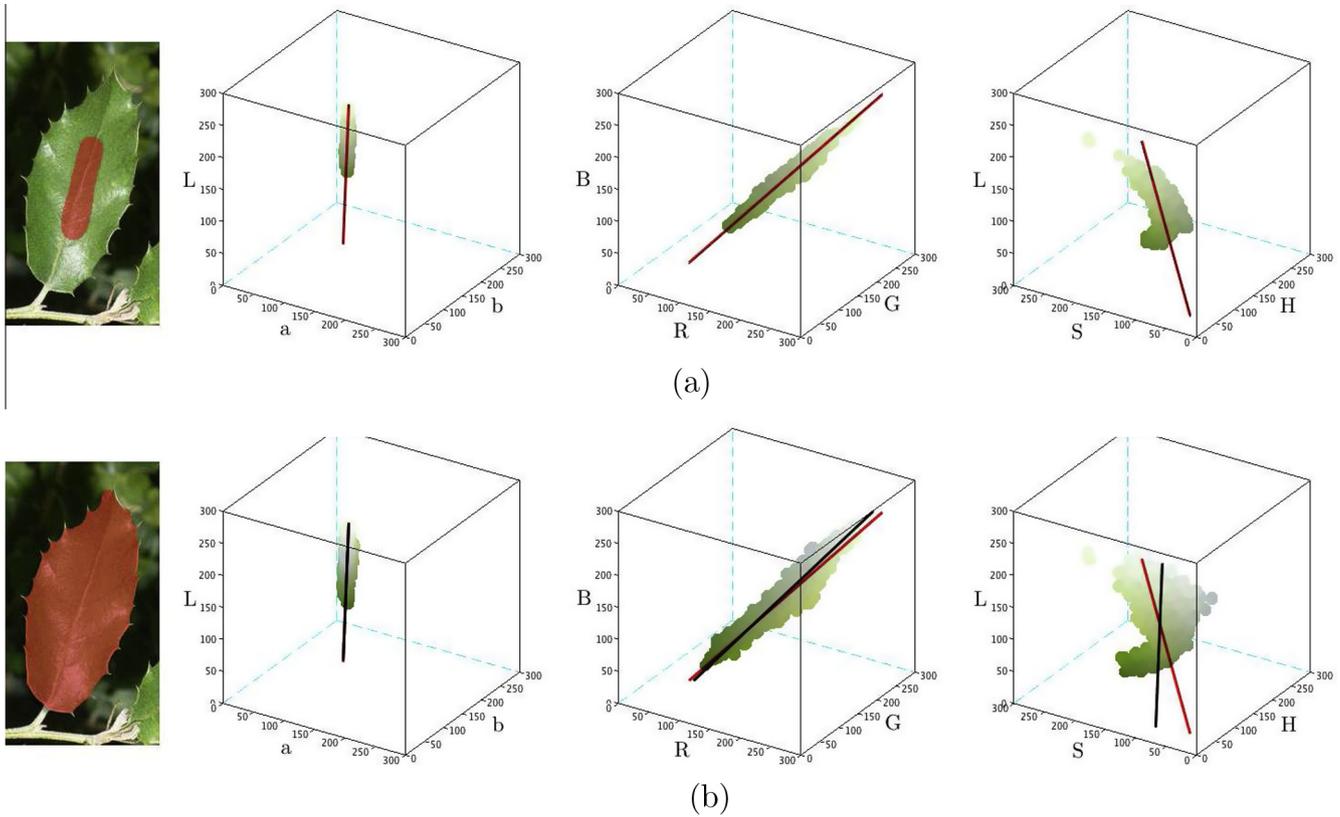


Fig. 5. Comparison between linear regressions of color distributions in $L^*a^*b^*$, RGB and HSL for a leaf image with reflections, on an initial region (a) and on the whole leaf (b).

threshold θ that represents the maximal color distance (between the pixel color and the propagated mean) up to which a pixel will be taken into account to update the mean during its propagation. The pixels are examined once, starting by the center, considering first the current closest pixel in terms of color. This propagation and adaptation process is illustrated by Fig. 6.

When considering a new pixel x , if the pixel color is close enough to the expected mean propagated from a previous neighbor, the mean we propagate to its unexplored neighbors is a weighted sum of the mean and the color; otherwise it is the same mean. The neighbors that are not yet to be considered are then placed inside a queue, according to their distance to the propagated mean. In other terms, if we call $N_8(x)$ the 8 neighbors of the pixel x :

$$\forall y \in N_8(x), \quad (\bar{L}_y, \bar{a}_y, \bar{b}_y) = \begin{cases} \alpha(L_x, a_x, b_x) + (1 - \alpha)(\bar{L}_x, \bar{a}_x, \bar{b}_x), & \text{if } \|(L_x, a_x, b_x) - (\bar{L}_x, \bar{a}_x, \bar{b}_x)\|_2 < \theta \\ (\bar{L}_x, \bar{a}_x, \bar{b}_x), & \text{otherwise} \end{cases} \quad (1)$$

Estimating the adaptive expected mean in every pixel of the image makes it possible to compute a complementary dissimilarity map, by calculating the distance of each pixel to the local mean: $d_{Adaptive}(x) = \|(L_x, a_x, b_x) - (\bar{L}_x, \bar{a}_x, \bar{b}_x)\|_2$.

The exploration method however presents the risk of finding a flaw, a path of progressive color transition between the leaf and the background, that may ultimately lead to consider a large part of the background as close. A good tolerance/adaptivity compromise allows to minimize this risk, but it is an inherent property of the method to consider as leaf color any color reachable by a gradual color path starting from the inside of the leaf.

4.3. Evidence-based combination

Both previous approaches constitute accurate ways to model the color of the leaf, but have their drawbacks. To keep only the best of each, it is essential to bear in mind that they do not measure the same thing in the image. It is important to take this into account to combine them effectively, as a simple, flat combination

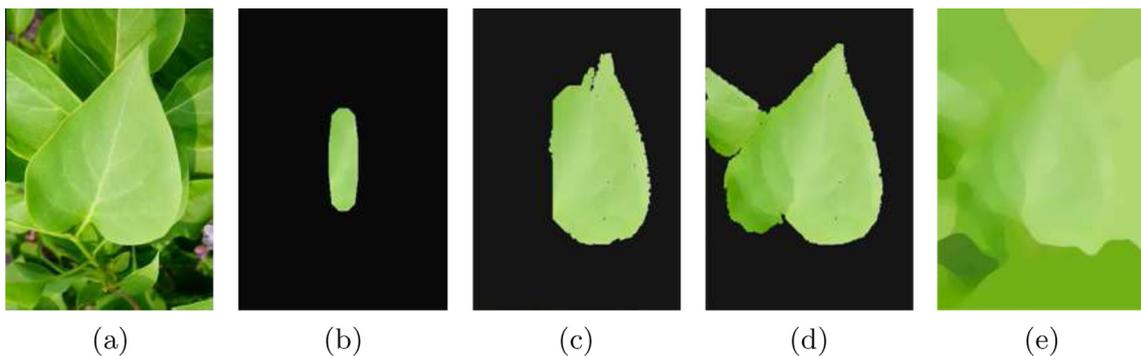


Fig. 6. Building a local adaptive mean color: image (a), initial region (b), propagation (c), (d) and final estimated local mean (e).

as the mean or the minimum would include both their advantages and disadvantages, and therefore not be sufficient.

The combination of these two representations could be performed using a probabilistic method such as Bayesian inference (as it is done in [23]) but we chose to rely on Dempster–Shafer theory of evidence [24,25] that is a framework generalizing probability theory. It has notably the advantage to reason with uncertainty, and to explicitly model the absence of knowledge when the sources do not provide tangible information.

As a matter of fact, a pixel appearing close in the global linear regression approach is generally a leaf pixel (not necessarily from the one of interest) whereas a distant pixel might also belong to the leaf but result being far from the prediction line if it is outside the learning luminosity range. This is what typically happens with reflections when not enough bright pixels belong to the initial region. On the other hand, a pixel being close in the local approach only means that a progressive path exists between the leaf and it, which reciprocally implies that a distant pixel will most certainly not belong to the leaf since no path could be found. Intuitively, we see that the global approach puts confidence on what

belongs to the leaf whereas the local approach tends more to distrust what lies outside the leaf.

Viewing it as a binary problem for each pixel, we define two opposite basic belief assignments that will be combined using Dempster’s rule. In the following, we consider the problem for one given pixel x . The linear regression distance puts mass on the “leaf” hypothesis $\{L\}$, and the adaptive distance will conversely be used only to place belief mass on the “non-leaf” hypothesis $\{\bar{L}\}$, the missing mass being in any case assigned to the whole set $\{L, \bar{L}\}$, representing uncertainty.

The combination of these two mass distributions using Dempster’s rule gives a resulting couple of belief and plausibility of the pixel x being a leaf pixel, that are used to compute a distance-like term $d_{\text{belief}}(x)$ and produce a dissimilarity map. More details about the exact computation of the distance can be found in Section A. Fig. 7 shows an example of such a map, and illustrates the behavior of our combination compared to more basic possibilities. A more thorough justification of this evidence-based approach in terms of segmentation performance is provided in Section B.

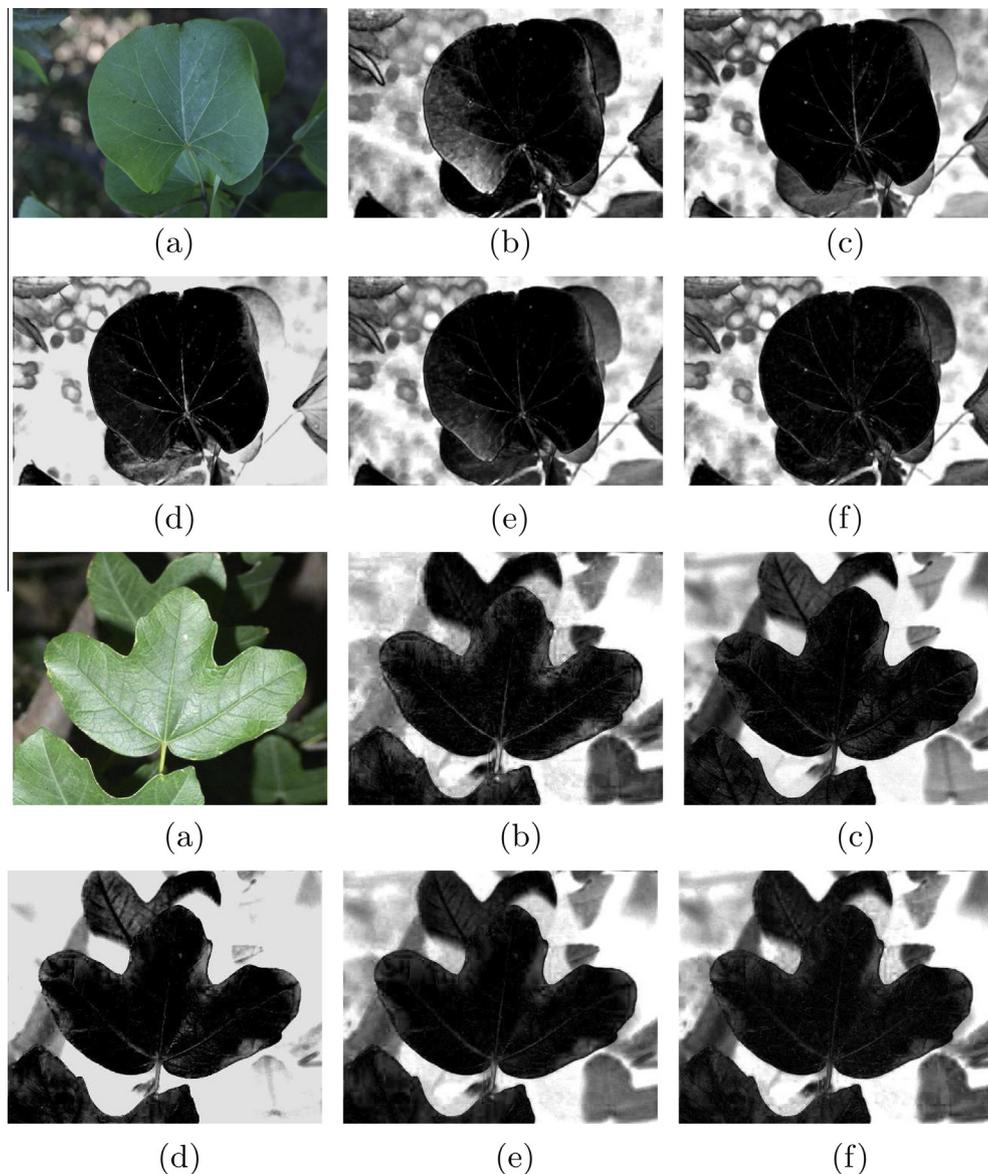


Fig. 7. Combination of distance maps of a leaf image (a) obtained from linear regression (b) and local mean (c) color models: using Dempster–Shafer theory (d) compared to the mean distance (e) and the minimum (f).

5. A model for leaf segmentation

The definition of a leaf, despite seeming intuitively easy to comprehend, covers actually a wide diversity of objects, which are surprisingly different in terms of shape, ranging from pine needles to spread-out bipinnate leaves. Such variety in shape makes it almost impossible to come up with a single well-defined model that could be applied to any possible leaf, even though they all develop with the same goal and result from a common growing process.

However, to facilitate the segmentation of leaves in complex images, it is necessary to integrate as much general shape information as possible, and exploit prior knowledge to constraint the resulting shapes. We propose to use a general leaf model to guide the segmentation, in an accurate and flexible way, without leaving too much adaptivity to non-leaf shapes. This is a way to ensure the coherence of the obtained shapes with both the content of the image and the natural possibilities.

5.1. Modeling the leaf object

A polygonal model based on the notion of global leaf shape in botany and covering the whole set of possible shapes for simple leaves was first introduced [22] and extended to the trickier case of palmately lobed leaves [26]. This model relies on a compact set of numerical parameters, accounting for geometric characteristics, to recreate the approximated shape of a leaf, as shown in Fig. 8. It needs only two points B and T , representing the base and top, and the values for the various parameters we recall here:

- w , the relative maximal width,
- p , the relative position where this width is reached,
- α_T , the opening angle at the apex,
- α_B , the opening angle at the base,
- n_L , the number of pairs of lobes (1 means simple),
- $(L(i))_{i=1\dots n_L}$, the relative length of each pair of lobes,
- $(\alpha(i))_{i=1\dots n_L}$, the angle of each pair of lobes.

These values can directly be used as descriptors of the global shape of the leaf, which is a discriminant feature used by botanists to identify leaves [27]. The model's parameters are additionally significant enough to constitute a first step towards a semantic interpretation, while keeping the precision and flexibility of a numerical description. The idea is then to use this model with a deformable template approach, where the elementary deformations are simply the variations of the parameters.

5.2. Parametric active polygon

This color dissimilarity d_{Belief} is then the main force that will guide the evolution of the polygonal model. The deformable model approach relies on an energy functional that is minimized by the model during its evolution. At each step, it will select a deformation that leads to the greatest decrease of the energy until it remains stable.

5.2.1. Energy definition

The energy is classically split into an image, or external, term and an internal term. Here, the external energy is directly based on the color distance. Since the goal is to gather pixels that are close to the leaf's color model, the value that needs to be minimized is the overall color dissimilarity of the pixels inside the polygonal region Π defined by a model built on the two points B and T and the set of parameters P . It just has to be balanced by a balloon force d_{max} to prevent the optimal model from being an empty one.

$$E_{\text{ext}}(B, T, P) = \sum_{x \in \Pi(B, T, P)} (d_{\text{Belief}}(x) - d_{\text{max}}) \quad (2)$$

The value d_{max} pushes the region to grow, but can also be seen as a threshold for which a pixel will be costly to add into the polygon.

The internal energy can be considered to be largely implicit, contained in the rules of the polygon's construction. But given only its definition, there is still a risk that the resulting shape does not look like a leaf at all. To ensure that consistency, we add a constraint of the values of the parameters. Using K-means on the parameters from hand-tuned models on white background leaf images, the centroids $(\mu_{k,i})_{k=1\dots K} = (\{w_k, p_k, \alpha_{Tk}, \alpha_{Bk}\})_{k=1\dots K}$ of 10 classes of archetypal shapes appear, and the corresponding standard deviations $(\sigma_{k,i})_{k=1\dots K}$ can be computed to account for the width of each class. To be of a leaf-consistent shape, the model has to be close to at least one of these shape classes in the parameter space. We subsequently define an internal energy term expressing this constraint, based on the normalized Euclidean distance to the classes of a model built with parameters $(P_i) = \{w, p, \alpha_T, \alpha_B\}$:

$$E_{\text{int}}(P) = \min_{k=1\dots K} \left(\sum_i \frac{(P_i - \mu_{k,i})^2}{\sigma_{k,i}^2} \right) \quad (3)$$

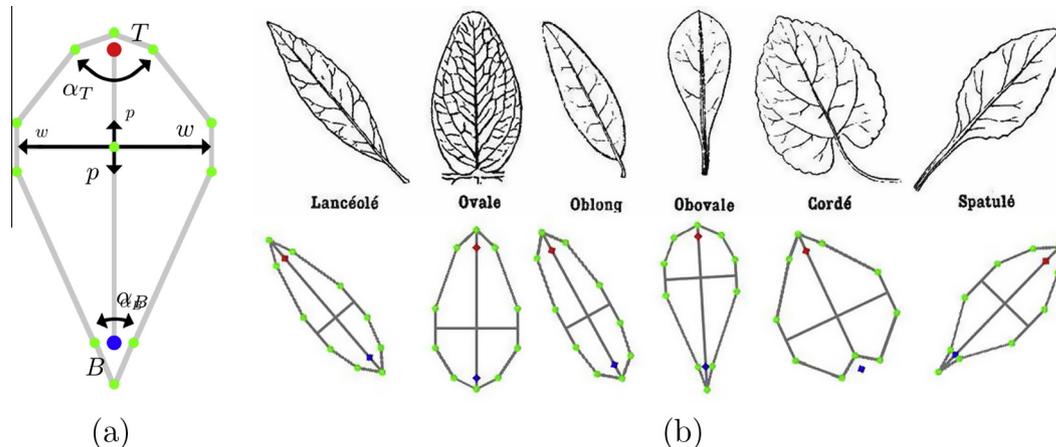


Fig. 8. Construction of the simple leaf model (a) and examples of recreated leaf shapes: Lanceolate, Ovate, Oblong, Obovate, Cordate and Spatulate (b) Images taken from [27].

The final expression of the energy to be minimized is a combination of these two energy, with a coefficient ω_{int} weighting the internal term, which sums up the problem to finding the optimal region Π^* defined by points B^* , T^* and parameters P^* , such as:

$$(B^*, T^*, P^*) = \left(\operatorname{argmin}_{(B,T,P)} (E_{ext}(B, T, P) + \omega_{int} E_{int}(P)) \right) \quad (4)$$

5.2.2. Model evolution

Starting from an initial position at the center of the image, the model will undergo a sequence of elementary deformations until it converges. In our case, the optimization is performed over the model's parameters and points, making the number of deformations to be considered at each step very limited. We look at the energy resulting from the two points B and T moving in one of the four directions, and from each parameter varying positively or negatively. The deformation giving the lowest energy is selected, accordingly modifying the model.

To avoid local minima, a heuristic close to simulated annealing is employed, with a temperature variable making possible to choose positive energy variations, but also conditioning the amplitude of the deformations. This temperature is slowly decreasing and regularly raised to a high value, until the model remains stable between two temperature rises.

All parameters are optimized simultaneously following this process, except for the number of lobes, that has to be treated differently given the drastic changes in shape it involves. To retrieve the accurate number of lobes, the model is initialized with a clearly excessive number of lobes (greater than the maximal number of lobes that can be found in nature) and redundant lobes are eliminated at the end of each temperature cycle (Fig. 9). Lobes in excess tend to group inside the actual lobes of the leaf, and by computing an overlap factor it is possible to detect which lobes are unnecessary and suppress them [26].

5.3. Guided active contour

The model $\Pi^* = \Pi(B^*, T^*, P^*)$ obtained after this evolution constitutes both a rough segmentation and a generally pertinent representation of the leaf's global shape. But to have a more detailed look at the particular morphological characteristics of the leaf, it is necessary to refine this segmentation and to obtain a contour

corresponding precisely to the actual leaf margin, where most botanical differential features appear.

The interesting point is that we do not start from nothing, but we can use the polygonal model to initialize an active contour and to guide its evolution. The active contour implementation we use differs from the original model [14] since it uses an implicit representation based on the binary level-set framework [28], an approximation for the original level-set evolution [29]. The constraints on the shape takes the form of an additional term in the energy functional minimized by the contour, that is then defined, for a contour Γ , by:

$$E(\Gamma) = \omega_L E_{Leaf}(\Gamma) + \omega_{\Pi} E_{Shape}(\Gamma, \Pi^*) + \omega_{\nabla} E_{Gradient}(\Gamma) + \omega_S E_{Smooth}(\Gamma) - \omega_{Balloon} \quad (5)$$

Instead of having an external energy term based on color consistency, or distance to a mean, we decided to reuse the dissimilarity map from the previous step, considering we have already an efficient measure of how well a pixel should fit in the leaf, in terms of color. The E_{Leaf} term is then simply the sum of the dissimilarity inside the region.

To use the polygonal model as a shape constraint, we decided to force the contour to remain close to the polygon Π^* . The E_{Shape} energy term ensures this guidance by summing a distance function, based on the distance of a pixel to an extended version $\widehat{\Pi}^*$ of the polygon, over the contour [26]. Fig. 10 shows an example of the distance to such an extended polygon, as well as the evolution of the contour constrained by this shape. This constraint controls the overflow that would occur when a boundary between the leaf of interest and a neighboring leaf is neither clear in terms of color nor gradient. The extension performed on the polygon, at the base and at the top, gives more liberty to the contour in these areas where it is crucial to find the actual margin to analyze the local shape.

However, all these energies are hard to harmonize into a coherent whole, independently of the image. To avoid conflicting interactions, we chose to have the coefficient of the leaf dissimilarity energy α initially set to a very low value, so that the behavior of the contour preeminently relies on the gradient in the first steps of its evolution. This coefficient is then gradually raised to its final value, so that the final contour minimizes always the same energy. This choice makes it possible for the contour to reach high

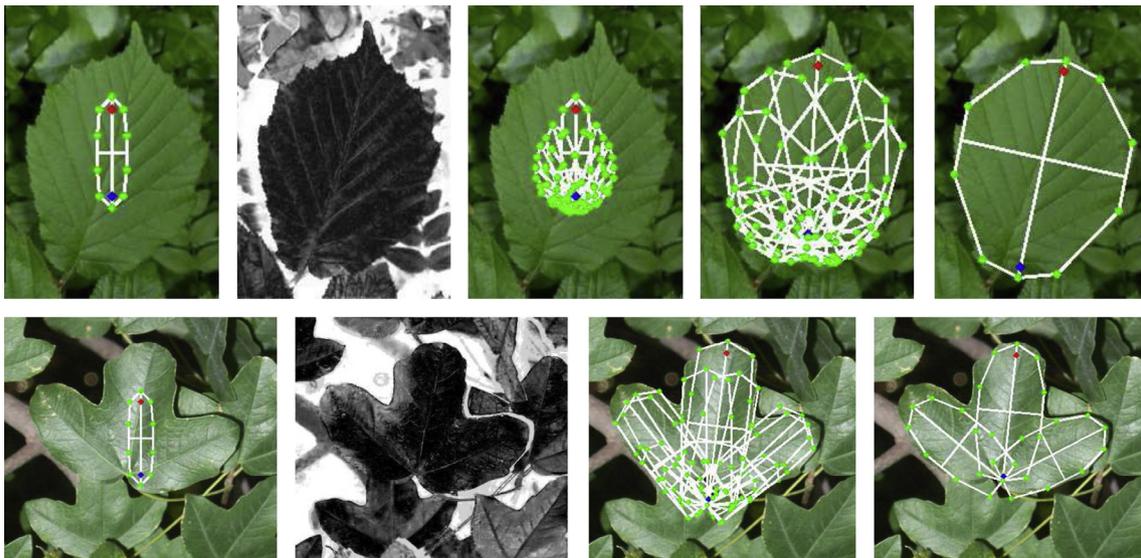


Fig. 9. Evolution of the parametric active polygon using the distance map, and redundant lobe suppression.

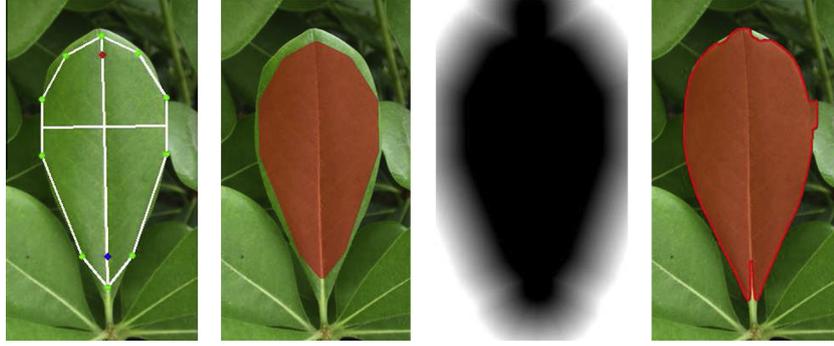


Fig. 10. Guided active contour results: polygonal model (left) initial contour (center left) guiding shape function (center right) and final contour (right).

gradients corresponding to the leaf margin even in color distant areas, and to keep sticking to these boundaries even when the leaf term reaches its normal value.

Examples of the whole segmentation process over various natural background leaf images can be found in Section B.

6. Capturing the local morphology

To describe the segmented leaves and make their identification possible, we want to represent their different botanical properties through independent sets of descriptors. Considering that the global shape is characterized by the parameters of the polygonal model, we have to look at a smaller scale and to focus on the interest areas that are the base and the apex of the leaf, and the potential teeth one may find on the margin. We propose separate descriptions for these three local leaf shapes, using curvature to locate and describe teeth, and introducing deformable models to approximate basal and apical shapes.

6.1. Teeth and pits on the margin

The places we want to locate to examine local shapes are generally easy to spot in terms of geometry, as they stand out clearly when giving a look at the contour's curvature. The main problem of this intuitive definition is that the level of detail where the apex of the leaf or smaller teeth appear is never the same, and looking at the contour with a fixed scale is consequently not sufficient.

6.1.1. Curvature scale space transform

A richer representation of the contour that avoids this obstacle is the Curvature-Scale Space (CSS), that has already been used in the context of shape-based leaf recognition [6]. It piles up curvature measures at each point of the contour over successive scales, summing up the information into a map where concavities and convexities clearly appear, as well as the relative scale until which they persist. For shape matching applications, the CSS contours, namely the curves formed by zero crossings of the curvature, or more exactly their maximal scale points, are generally used as a shape representation, losing some information on the actual curvature value.

We used a curvature measure based on the estimated derivatives of the contour points, but whereas the original method estimates the same curvature on Gaussian smoothings of the contour with growing kernels, we chose to keep the same contour but to estimate the derivatives in each point on a growing neighborhood. Our scale variable represents then the size of the considered neighborhood. The CSS consists of the kind of matrix displayed in Fig. 11, containing curvature values K for a given scale S and at a given point $(x; y)$ defined by its normalized arc length u :

$$K(u, S) = \frac{x'_S(u) \cdot y''_S(u) - x''_S(u) \cdot y'_S(u)}{(x'_S(u)^2 + y'_S(u)^2)^{3/2}}$$

$$x'_S(u) = \frac{1}{S} \left(x \left(u + \frac{S}{2} \right) - x \left(u - \frac{S}{2} \right) \right)$$

$$y'_S(u) = \frac{1}{S} \left(y \left(u + \frac{S}{2} \right) - y \left(u - \frac{S}{2} \right) \right) \quad (6)$$

$$x''_S(u) = \frac{1}{S^2} \left(x \left(u + \frac{S}{2} \right) - 2x(u) + x \left(u - \frac{S}{2} \right) \right)$$

$$y''_S(u) = \frac{1}{S^2} \left(y \left(u + \frac{S}{2} \right) - 2y(u) + y \left(u - \frac{S}{2} \right) \right)$$

The CSS transform is a very powerful description of a contour but is too informative to be used as a descriptor and to build class models by averaging several of them. As we wish to locate and characterize precisely curvature-defined elements on the contour, it was a more judicious choice to look, not at zero-curvature points, but on the contrary at the maxima and minima of curvature. The fact that these points visually stand out in the CSS image, and can be matched obviously with convex or concave structures on the contour, argues at least in favor of this choice.

6.1.2. Detection of dominant points

To locate these points, we actually want to detect dominant points on the leaf contour and to find the scale of the concavity or convexity they correspond to. This is a well studied problem [30,31] and the method we use is largely inspired of existing works. The same steps are repeated at each scale, starting with $S = 4$, and the set of dominant points at each scale $D(S)$ is initialized with all the contour points. The process is then the following:

1. Extract points for which the curvature value is a local optimum:

$$D(S) \leftarrow D(S) \cap \left\{ u, \forall u' / |u' - u| < \frac{S}{2}, |K(u, S)| \geq |K(u', S)| \right\}$$

2. Suppress points the curvature intensity of which is below a threshold k_{min} :

$$D(S) \leftarrow D(S) \cap \{u, |K(u, S)| > k_{min}\}$$

3. Suppress points that cannot be traced to a dominant point at the previous scale (if $S > 4$):

$$D(S) \leftarrow D(S) \cap \left\{ u, \exists u' / |u' - u| < \frac{S}{2}, u' \in D(S-1) \right\}$$

4. Keep only the median point of potential remaining groups of size $\leq S$: if $u \in D(S)$: $g_u = \{u', |u' - u| < \frac{S}{2}, u' \in D(S)\}$

$$D(S) \leftarrow (D(S) \setminus g_u) \cup \{\text{median}(g_u)\}$$

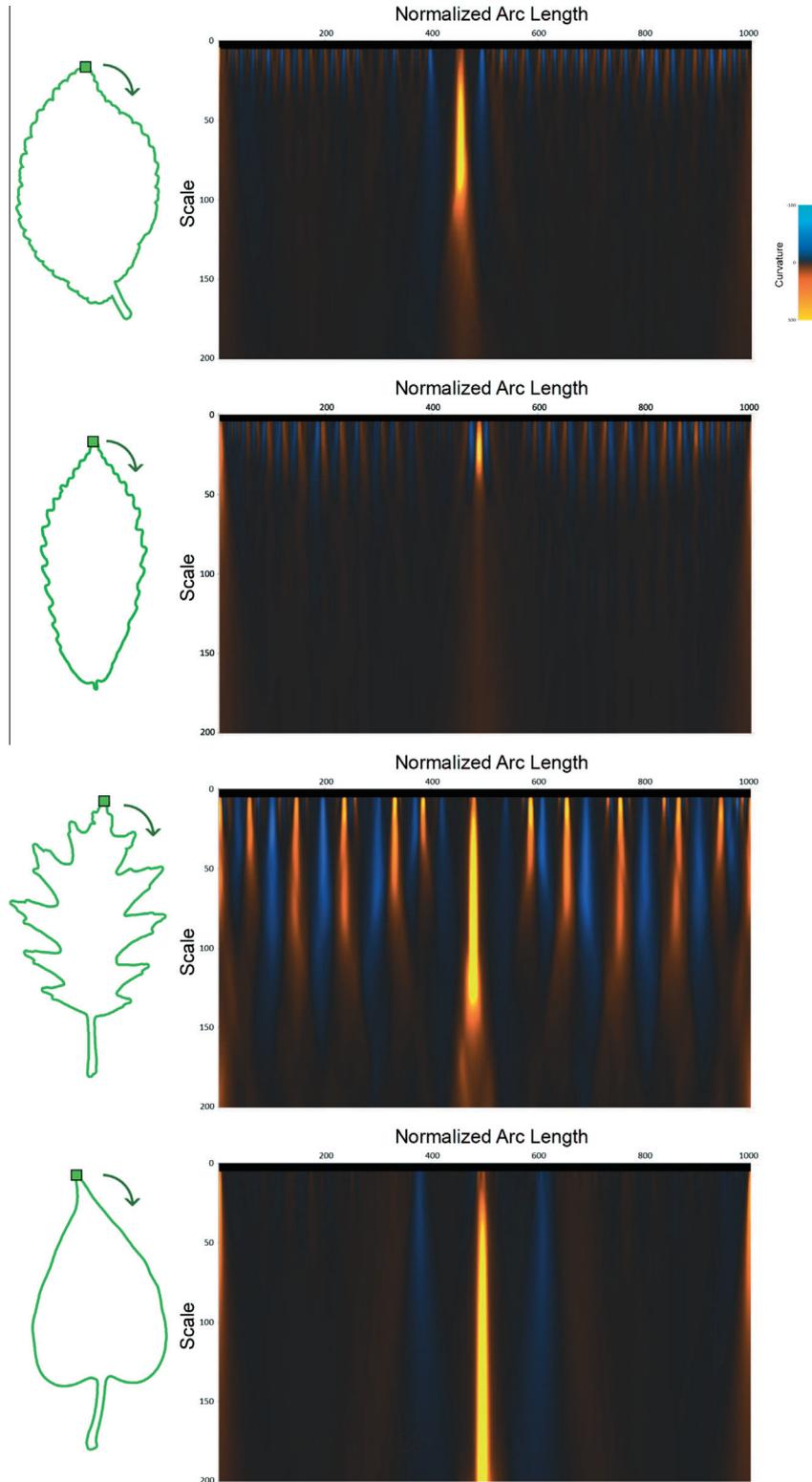


Fig. 11. Example of Curvature-Scale Space transforms of three different contours (green square marks the first point): brightness represents curvature intensity, negative curvatures in blue, positive curvatures in orange. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The result we obtain is a map of the located dominant points at each scale, as presented in Fig. 12. This representation reveals the teeth and pits of the contour in the form of chains of dominant points. Given the particularities of our method, dominant points may shift between scales, leading to an imprecision on the exact location of the point on the contour, but giving some potentially

useful information on the orientation of the corresponding concave or convex structure.

6.1.3. Relevant scale selection

For the sake of interpretation, the information we want to retrieve is the visible curvature and size of every located structure

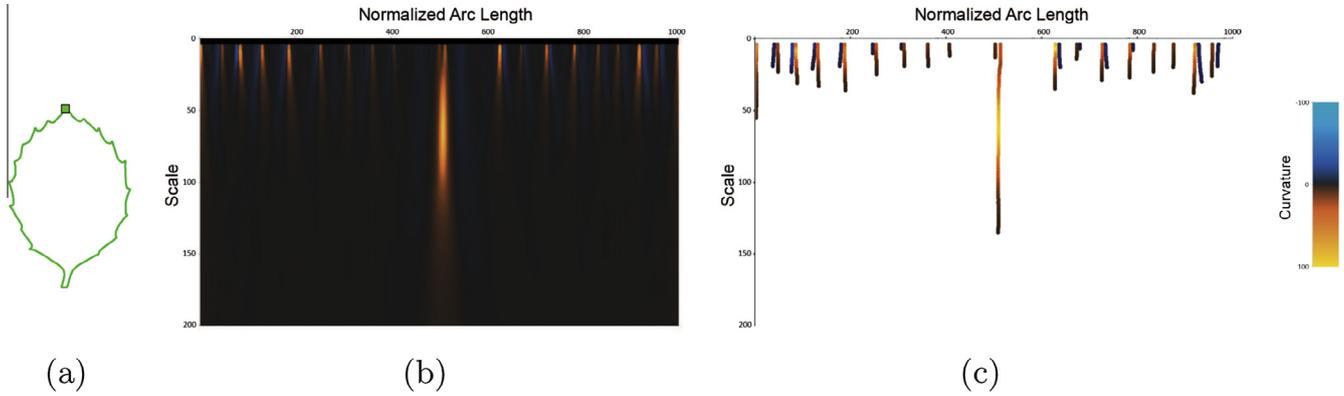


Fig. 12. Traced dominant points over scale (c) and their curvature, obtained from the Curvature-Scale Space transform (b) of a leaf contour (a).

on the contour. Each chain that appears on the dominant points map corresponds indeed to a convex or a concave part, and its actual size, as a human eye would interpret it, is the scale until which it persists, that is to say the scale of the chain's end point.

To keep only these terminal points, we scan the dominant points starting from the highest scale. When a dominant point of arc-length u is found at scale S , all the dominant points of the same curvature sign are suppressed at lower scales, in a neighborhood of size S .

$$\forall S' < S, D(S')$$

$$\leftarrow D(S') \setminus \left\{ u' \mid |u' - u| < \frac{S}{2}, \text{sign}(K(u, S)) = \text{sign}(K(u', S')) \right\}$$

This way, we ensure that a single structure is not counted twice, and small structures which are included in larger ones are merged into one. In the case of large-scale elements, we consider then that only the largest appearing structure is relevant, and that smaller concave or convex parts it includes are part of it and can be discarded. The only exception concern the leaf's base and apex, which are generally strongly marked convex parts, and would encompass all the smaller teeth that should be kept and analyzed.

6.1.4. Understanding the leaf contour

Prior to this processing, the contour needs to be divided into areas corresponding to the tip, to the base, to the potential lobe tips, and finally to the rest of the margin, so that the most salient features do not absorb the rest of the information. Here we can advantageously rely on the polygonal shape model to generate this partition, that would otherwise be a much more complicated task. Supposing that the axes of the polygonal model are accurate, we can use them to mark off contour points that are in one of these areas of interest.

To achieve this labelling, as shown in Fig. 13, we only keep the intersection of the contour with a fixed angular sector, built around the corresponding axis, issuing from the opposite point (base for the tips, and main tip for the base) and with a minimal distance to this point.

The interest of this partition is twofold. First, it is now possible to locate accurately the base point \hat{B} and the tip point \hat{T} of the leaf as the most salient dominant points in the corresponding area. And secondly, outside of these areas, we have, for each side of the leaf, located the teeth and the pits of the margin $\cup_s D(S)$, at their maximal scale S , allegedly their actual size. To complete this size information, we estimate the actual curvature $\bar{K}(u, S)$ of each dominant point, as the average curvature of the dominant points belonging to the same chain. This computation is done considering scales from the highest to the lowest, while suppressing the points at lower scales:

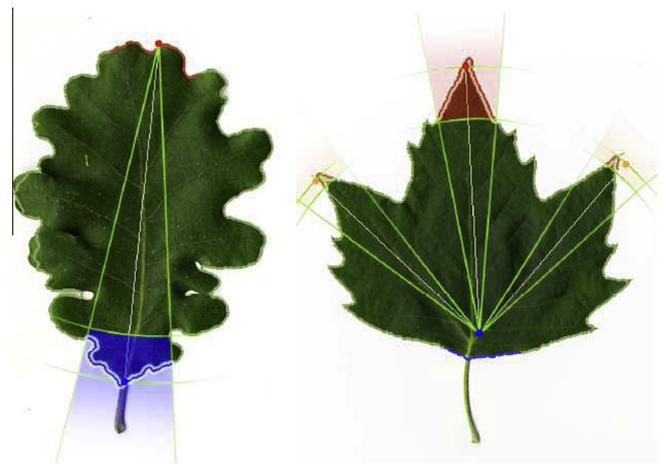


Fig. 13. Labelling contour parts corresponding to leaf areas.

$$\forall S, \forall u \in D(S), \bar{K}(u, S) = \frac{1}{S} \sum_{s=1}^S \max_{u' \in D(s)} K(u', s) \quad (7)$$

$$|u' - u| < \frac{s}{2}$$

Finally, what we produce is a semantically rich representation of the leaf contour, where the base and tip points are precisely located, and where teeth are detected and characterized independently in terms of size and curvature. This representation is displayed in Fig. 14.

6.2. A descriptor for the leaf margin

Even if this representation of the contour is full of information and could be used to compare two leaves, it still does not come in very handy if we want to average several of them for classification purposes. It is necessary to sum up this information into a condensed descriptor that will make it possible to capture the properties of a species, and allow a semantic interpretation of its specificities.

6.2.1. Margin shapes in botany

To describe the shape of the leaf margin, botanists use a terminology that refers both to the properties of teeth taken separately and to their repartition over the margin. Some of these terms are shown in Fig. 15. For instance a “doubly dentate” leaf implies two levels of teeth with different sizes and frequencies, with bigger teeth divided into smaller sub-teeth. The words themselves are vague enough so that leaves with teeth that look rather different may fall inside the same term.

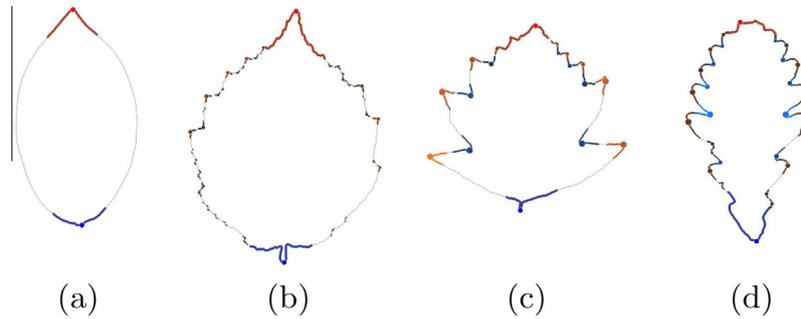


Fig. 14. Various leaf contours with detected base, tip, teeth and pits; tip area in red, base area in dark blue; convexities in orange, concavities in blue, brightness representing curvature intensity, extent representing scale. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

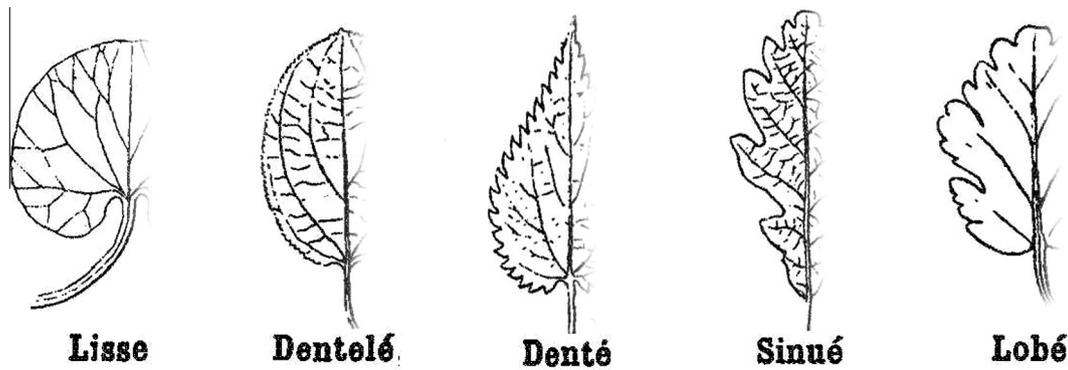


Fig. 15. Examples of leaf margin shapes: Entire, Denticulate, Dentate, Sinuate, Lobate. Images taken from [27].

The margin shape constitutes however a very discriminant criterion for species identification that generally presents less variability than the global shape, with the exception of some pathological species. Our descriptor will have to capture the differences regarding size, sharpness, frequency and variability of the teeth implied in the botanical terms, but also to benefit from the use of numerical values to be more precise and discriminate leaves of different aspect that would be called the same.

6.2.2. Curvature-scale histogram

The descriptor we propose to represent leaf margin shapes is simply a square two dimensional histogram summing up the properties of the detected teeth and pits over scale and curvature dimensions. Since every detected dominant point $u \in \cap_s D(S)$ is characterized by its maximal scale S and average curvature $\bar{K}(u, S)$, it will contribute at the location $(\bar{K}(u, s), S)$ of the histogram. To simplify the notations, we now write $((K_i; S_i))_{i=1..N} = \cup_s \{(\bar{K}(u, S); S), u \in D(S)\}$ to enumerate the dominant points.

We want to count all the vertices of the contour, and count them once, to have a valid statistics of what proportion of the contour is part of a concavity or convexity of given properties. Consequently, the weight that is put in the histogram for one structure has to be proportional to its scale S_i , since, given their definition, detected structures account for S_i vertices. For smoothing purpose, we chose to place in the histogram a Gaussian G centered on (K_i, S_i) and whose total weight is equal to S_i . If we set $\sigma_s(S_i) = \frac{S_i}{4}$ and $\sigma_k = 4$ the values of the histogram can then be expressed in function of $((K_i; S_i))_{i=1..N}$:

$$H_{CS}(k, s) = \sum_{i=1}^n G_{\sigma_s(S_i), \sigma_k}((k; s) - (K_i; S_i)) \quad (8)$$

$$\text{where } G_{\sigma_s, \sigma_k}(K; S) = \frac{1}{2\pi} \exp\left(-\left(\frac{S^2}{2\sigma_s^2} + \frac{K^2}{2\sigma_k^2}\right)\right)$$

In this way, we cumulate the structures found along the margin. We build two distinct histograms for the left margin and the right margin, as they are detected when locating the different areas of the leaf. Then we compute their geometric mean to keep only teeth that are present on both sides, and reduce the influence of a possibly bad segmentation. This resulting histogram is normalized back to the sum of the weights of the two previous histograms. The total weight of the histogram at this step is then equal to the number of vertices detected as part of a convexity or concavity along both sides. The missing weight, *i.e.* the number of vertices of either side that have not yet been counted, because they do not belong to any concavity or convexity, is added as such in the bin $H(0, 0)$ that represents then the proportion of the margin that is smooth. The histogram is then normalized again so that the weight is the same for every leaf.

The final histograms are presented in Fig. 16 where we can see how an entire margin in (a) will produce an empty histogram (where all the weight is actually in one single bin) and how the different levels of teeth in (b) and (c) appear in the histogram.

This descriptor represents well the characteristics of the teeth of a leaf, and is generally able to reflect the levels of dentition even when there are several. However, it has the disadvantage of compressing the information and losing trace of the spatial repartition of teeth along the margin. It is still a very dedicated tool that captures with accuracy what needs to be observed on a leaf margin.

6.3. Local deformable curves for tip and base

Concerning the tip and the base of the leaf, other regions of great interest for identification, a single curvature-scale measure would not be enough (especially at the base where the hypothetical presence of the petiole makes this measure irrelevant) and a finer way to characterize the local shape of the contour around these points is a necessity.

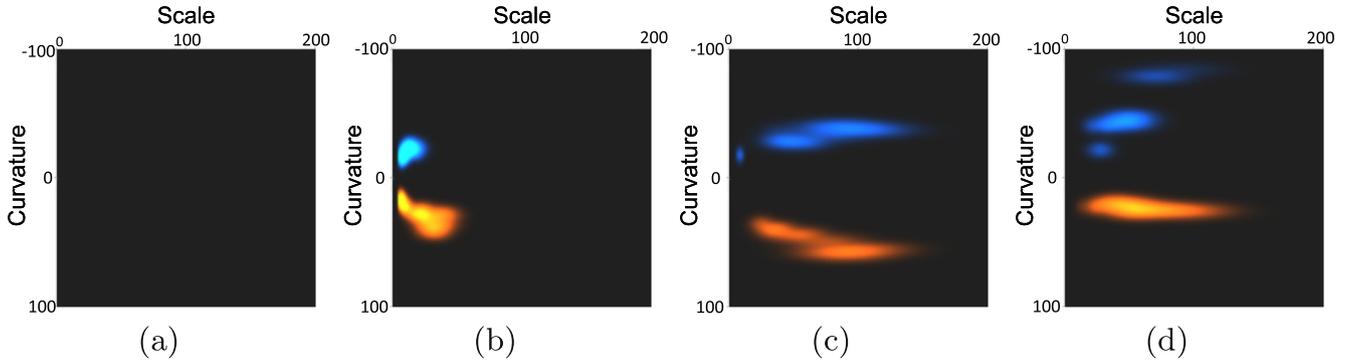


Fig. 16. Curvature scale histograms for the 4 leaves of Fig. 14: curvature on the vertical axis, scale on the horizontal axis; convexities in orange, concavities in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Similarly to our approach for the global shape, we decided to use a deformable model approach, defining a simple model able to re-create the various shapes that can be found in the botanical literature. And again, we expect the construction parameters, which will be optimized during the model's evolution, to be significant and representative measures for the resulting shape.

6.3.1. Parametric Bézier curves

The models for the base and for the tip are basically defined in the same way. If we take for instance the tip, the model is built on the tip point \hat{T} that has been located based on curvature (Section 6.1.4). We consider then the contour around this point as two curves that join at the tip, making a pointy, blunt or even squared shape, with either convex or concave edges. The contour can then be modelled by two simple Bézier curves, that are expressive enough to form these shapes, the two sides being not necessarily symmetrical, but of equal length.

The set of parameters we use to build these two curves, or more precisely the control points they are built upon, are the following:

- α , the global angle of the model,
- o , the orientation angle relatively to the leaf axis
- $(\alpha_{t1})_{l,r}$, for each side, the tangent angle at the tip point,
- $(\alpha_{t2})_{l,r}$, for each side, the tangent angle at the end point,
- $(\delta_{t1})_{l,r}$, for each side, the distance of the 1st control point from the tip,
- $(\delta_{t2})_{l,r}$, for each side, the distance of the 2nd control point from the end.

The model's construction is illustrated in Fig. 17. The angle α defines two end points T_l and T_r as the opposite vertices of an isosceles triangle. The apex of the triangle is the fixed tip point \hat{T} , the length of the two equal sides is set constant and the main axis differs from the leaf axis (found with the polygonal model) of an angle o . Then for each side apart, the tangent lines at the tip and at the

corresponding end point are defined by angles α_{t1} and α_{t2} , a positive angle placing the tangent towards the outside in every location. The distance parameters δ_{t1} and δ_{t2} define the position of the two control points $(t_{1right})_{l,r}$ and $(t_{2})_{l,r}$ on their respective tangent lines. And the 4 points \hat{T} , $t_{1,l}$, $t_{2,l}$ and T_l (resp. \hat{T} , $t_{1,r}$, $t_{2,r}$ and T_r) make it possible to build the Bézier curve.

6.3.2. Fitting the curves

The main goal is to obtain curves that are as close as possible to the leaf contour. Consequently the model's evolution has to be driven by an energy based on the distance to this contour. The distance transform of the final contour Γ gives a distance map indicating for each pixel x in the image its distance $d(x, \Gamma)$ to the leaf contour. The distance of the two end points to the contour need to be considered with a greater weight because these points need to be actual contour points for the tangent parameters to be relevant. If we consider the model Ψ formed by the two curves built with parameters $P = \{\alpha, o, (\alpha_{ti,k})_{i=1,2,k=l,r}, (\delta_{ti,k})_{i=1,2,k=l,r}\}$, the external energy term is then defined as:

$$E_{ext}(P) = \sum_{x \in \Psi(\hat{T}, P)} (d(x, \Gamma)) + \eta \text{left}(d(T_l, \Gamma) + d(T_r, \Gamma)) \quad (9)$$

The curves are free to evolve, and may take asymmetrical shapes, but it is important to limit their expressiveness to plausible shapes and avoid aberrant behavior. This is why we added soft symmetry constraints, since leaves are mostly symmetrical objects, in the form of an internal energy, as a weighted sum of the differences between parameters for the left curve and the right curve:

$$E_{int}(P) = \sum_{i=1}^2 w_\alpha |\alpha_{ti,l} - \alpha_{ti,r}| + w_\delta |\delta_{ti,l} - \delta_{ti,r}| \quad (10)$$

And combining those two energies, the evolution process is strictly the same as described for the polygonal model Section (5.2.2), selecting at each step the variation of one parameter that creates

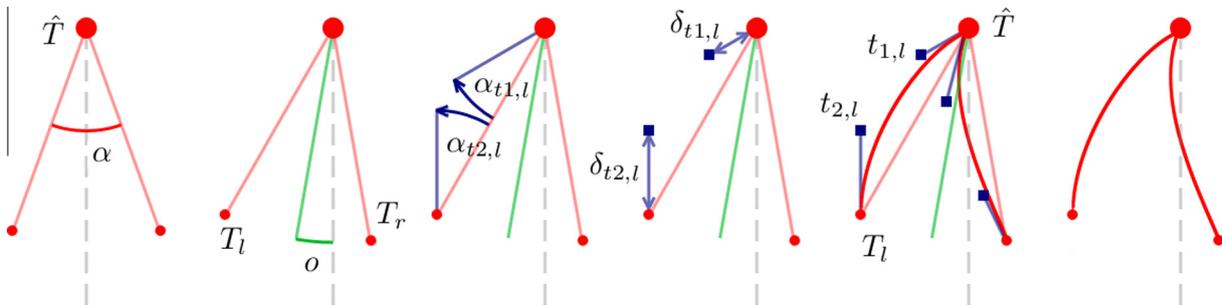


Fig. 17. Construction of the two Bézier curves defining the tip model.



Fig. 18. Evolution of the tip model, based on the distance transform of the leaf contour.

the greatest decrease in the energy. The model is initialized with an angle α set to the corresponding angle parameter α_T^* or α_B^* in the polygonal model, and all the other angle parameters set to 0 so that the curves are initially straight. An example of such evolution is displayed in Fig. 18.

6.3.3. Dealing with the petiole

The presence of a petiole, the stalk linking the leaf to the stem, is a source of trouble when considering the shape of the base. One of the main problems is the uncertainty of its presence, some species having very short to absent petiole (sessile leaves), or the petiole being too different in terms of color to eventually be part of the contour. However, when it is there, it will appear as a major convexity in the base area, which generally leads to placing the base point \hat{B} (around which the base model is built) at the end of the petiole. The estimated “base shape” in this case would be irrelevant, since it would represent more the shape of the petiole.

To overcome this problem, the base point is made moveable in the base model, along one direction only: the model axis. To include it in the deformation process, we added a shift parameter s that defines the position of the actual leaf base \hat{B}_0 on the axis defined by the detected base \hat{B} and the orientation o . The control points are then built with \hat{B}_0 as base point (the end points being still built from \hat{B}) and the shift, though being meaningless for shape description, is optimized along with the other parameters. To make sure that the base will move, the curves are pushed to remain convex (which is generally the case if we consider botanical base shapes) by adding a component in the internal energy that pushes the angles to remain positive:

$$\hat{E}_{int}(P) = E_{int}(P) - \gamma \sum_{i=1,2} \min(\alpha_{t,k}, 0) \quad (11)$$

$$k = l, r$$

As shown in Fig. 19, even when starting with an erroneous base point, the model will be able to fit the actual leaf margin, with the base moving up along the petiole. This way, the angles obtained at the end of the evolution have a real significance, which would not be the case if the base remained in its first location.

The descriptors introduced in this section use very explicit representations of the contour, and can be expected to capture the specificities of leaves at a high level. It could be interesting to connect them with a semantic representation, such as the terms used in botany. This perspective holds furthermore a great potential for educational and interactive purposes.



Fig. 19. Evolution of the base model, with moving base point.

7. Identification results

7.1. Datasets

The database used to test our algorithms is a subset of the Pl@ntLeaves [2] database, keeping only 3603 leaf images (out of 5436) on white, plain or natural backgrounds of the 50 species (out of 71) with non-compound leaves. This database is divided into two sets, Train and Test, containing leaves from different individuals.

7.2. Classification method

To assess the relevance of our descriptors, and compare their performance with more usual generic shape descriptors, we decided to use a basic classification method, assuming that results with a simple approach measure accurately the significance of features. The descriptors are separated into independent sets, since we consider they represent uncorrelated features of the leaf's morphology. Samples are first classified using only one set of descriptors, and results of such classifications are to be fused together to obtain a final class.

The descriptors we extracted on all the leaves in the database can be distinguished between those we introduced:

- Parameters of the polygonal model ($P_{T,i}$) [MP].
- Parameters of the base and tip models ($P_{B,i}$) and ($P_{T,i}$) [B/T].
- Curvature-scale histogram ($H_{CS}(k,s)$) [HCS].

And some usual shape descriptors to perform comparisons:

- Central moments (m_{ij}) [CM].
- Hu moments (h_i) [HM].
- Centroid contour distance curve ($\Delta(u)$) [CCD].

The moments have interesting invariance properties that make them popular general shape descriptors [3,6]. We extract the central moments of degrees 2 and 3, along with the 7 Hu moments over the segmentation results. The CCD curve is also commonly used [7,8] and the one we compute is a function of the normalized arc-length which starts at the detected tip point of the leaf, making comparisons easier, and allowing to compute means without shifting and resampling.

The database formed by all the parameters from all the images is first centered and normalized, so that every parameter has its global mean set to 0 and its standard deviation set to 1. Assuming that each parameter simply follows a Gaussian distribution, we compute class centroids and standard deviations with examples from the Train dataset. We chose to use the number of lobes n_L from the polygonal model as a first filter, computing actually, for each species, one class model per plausible number of lobes. For each value of n_L up to 3, and for each species S we build a class $\Phi_{S,n_L} = (\mu_{S,n_L,i}, \sigma_{S,n_L,i})$ if at least 10% of all the species examples have this value.

It is important to notice that we extracted two different types of descriptors here: those for which each parameter taken independently conveys significance ($(P_{\Pi,i}), (P_{T,i}), (P_{B,i}), (M_{i,j}), (H_i)$) and those which need to be considered as a whole ($(H_{CS}(k,s)), (D(u))$). Considering that one single bin of a histogram provides interpretable information does not make sense, and the same goes for the distance to the center of one given contour point.

For the first type, a class can be seen as an hyper-ellipsoid in the parameter space, whose axes are given by a covariance matrix. Again for simplicity purposes, we assume here that this matrix is always diagonal, making the abusive hypothesis that the descriptors are not correlated. On the other hand, for the other descriptors, speaking of a parameter space seems inappropriate, and it is easier to see a class as an envelope inside which the curves, or the histogram peaks, have to fit.

To classify a new example, we will simply have to compare its descriptors with the centroids of the classes sharing the same number of lobes n_L . For each one of the 6 sets of parameters, we compute a distance term. The distance for the first type of descriptors is an estimated Euclidean distance to the surface of the class ellipsoid in the parameter space. Its expression, for a set of descriptors P , is given by:

$$\mathcal{D}(P, \Phi_{S,n_L}) = \|P - \mu_{S,n_L}\|_2 \max\left(1 - \frac{1}{\|P - \mu_{S,n_L}\|_M}, 0\right) \quad (12)$$

More details on what is represented by this distance are to be found in Section C. In this equation $\|P - \mu_{S,n_L}\|_M$ is the Mahalanobis distance, which is simply in our diagonal case a normalized Euclidean distance:

$$\|P - \mu_{S,n_L}\|_M = \sqrt{\sum_i \frac{(P_i - \mu_{S,n_L,i})^2}{\sigma_{S,n_L,i}^2}} \quad (13)$$

For the other descriptors, the class model can be seen as thick curve, or surface in the case of the histogram, to which we want to compare another regular curve or surface. A measure that is used to compare two CCD curves ($\Delta_1(u)$ and $\Delta_2(u)$) of same length L , but not necessarily aligned, is:

$$\mathcal{D}(\Delta_1, \Delta_2) = \min_{k=0..L-1} \left(\sqrt{\sum_{u=0}^{L-1} (\Delta_1(u) - \Delta_2(u+k \bmod L))^2} \right) \quad (14)$$

In our case, since the curves are aligned, there is no need to compute L different sums, and we compute the distance of a curve $(\Delta(u))_{u=0..L-1}$ to a mean thick curve centered at each point on $\mu_{S,n_L}(u)$ and of width $2\sigma_{S,n_L}(u)$. This curve can be written $(\bar{\Delta}_{S,n_L}(u))_{u=0..L-1} = (\mu_{S,n_L}(u) - \sigma_{S,n_L}(u); \mu_{S,n_L}(u) + \sigma_{S,n_L}(u))_{u=0..L-1}$ and the distance is then given by:

$$\begin{aligned} \mathcal{D}(\Delta, \Phi_{S,n_L}) &= \sqrt{\sum_{u=0}^{L-1} \min_{\delta \in \bar{\Delta}_{S,n_L}(u)} (\Delta(u) - \delta)^2} \\ &= \sqrt{\sum_{u=0}^{L-1} (\max(|\Delta(u) - \mu_{S,n_L}(u)| - \sigma_{S,n_L}(u), 0))^2} \quad (15) \end{aligned}$$

We use the same idea to compute a χ^2 -like distance between a new histogram and the thick histogram surface representing the class:

$$\mathcal{D}(H_{CS}, \Phi_{S,n_L}) = \sum_k \sum_s \frac{(\max(|H(k,s) - \mu_{S,n_L}(k,s)| - \sigma_{S,n_L}(k,s), 0))^2}{H(k,s) + \mu_{S,n_L}(k,s) + \sigma_{S,n_L}(k,s)} \quad (16)$$

Each one of the 6 distance terms corresponding to the 6 sets of descriptors is then weighted differently after having learned the classes, dividing it by the average distance of the correct class over the training base, which weights it accordingly to its significance. The final distance we use for classification is simply the sum of these weighted terms, and we finally produce a list of species ordered by their distance.

7.3. Performance estimation

We compared the classification results obtained using different sets of descriptors. First, we used only the descriptors introduced in our work [MP + B/T + HCS]. These three sets were designed to represent the properties of a leaf both globally and locally, and it is no surprise that they give better results than generic shape descriptors alone [CM + HM + CCD]. To be fair, we considered that the curvature-scale histogram, based on curvature, captures details that these descriptors cannot. This means that the actually relevant comparisons are to be made between [CM + HM + CCD] and [MP + B/T], and also between [CM + HM + CCD + HCS] and [MP + B/T + HCS]. Finally, we want to measure the information we miss with our three sets of dedicated descriptors by comparing the classification scores obtained with [MP + B/T + HCS] and with [MP + B/T + HCS + CM + HM + CCD].

We computed classification rates for the different types of images in the Pl@ntLeaves Database (Scan, Pseudoscan and Photograph) and measured the presence of the true species among the top k answers (k going from 1 to 10) returned by the classification algorithm. These measures were obtained on the Train database, using a cross validation approach: class models were learned on a random two thirds of the database and scores estimated on the remaining third.

Fig. 20 presents these results for the different sets of parameters. Obviously our descriptors [MP + B/T + HCS] perform better than generic shape descriptors alone [CM + HM + CCD] (dark blue and orange curves) since they take more aspects into account, such as the curvature information. What appears here is first the great improvement in performance induced by the use of [HCS] in the classification, when we look at the gap between the curves for [CM + HM + CCD] and [CM + HM + CCD + HCS] (orange and red curves) and between [MP + B/T] and [MP + B/T + HCS] (light blue and dark blue curves). It also seems that, as far as the general shape is concerned, our dedicated descriptors [MP + B/T] are sufficient to outperform the joint moments and CCD [CM + HM + CCD] (light blue and orange curves). However, when all the descriptors are taken into account, the performance seems better, yet slightly, which tends to prove that some of the shape characteristics are not captured by our descriptors.

A more objective performance measure is given by the official score computation tool for the Pl@ntLeaves Test database, that considers the classification to be correct only if the true species is the first (top-1) answer. The scores we present for our classification in the first line of Table 1 are to be compared with the best fully-automatic participation submitted to ImageCLEF 2011 [5] whose results are recalled on the second line, as well as our former participation and the maximal reachable scores given our limitations (no compound species, pictures of one single leaf).

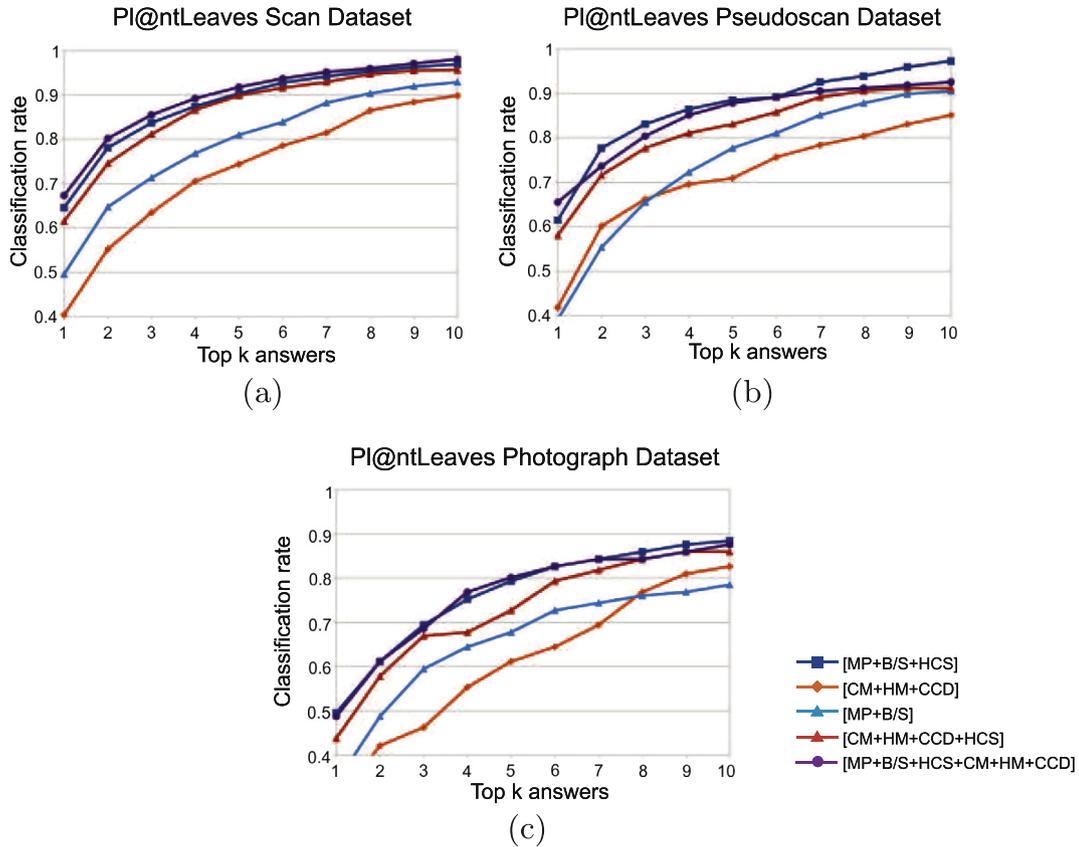


Fig. 20. Classification rates on the Pl@ntLeaves Database on scan (a) pseudoscan (b) and photograph (c) images.

Table 1

Official scores on the Pl@ntLeaves Test database.

	Scan	Pseudoscan	Photograph	Mean
[MP + B/T + HCS]	0.581	0.538	0.374	0.498
INRIA lmedia 2011	0.685	0.464	0.197	0.449
LIRIS 2011	0.546	0.513	0.251	0.437
Maximal Scores	0.824	0.835	0.716	0.792

8. Looking further

The system presented in this article constitutes a solid starting structure for the recognition of tree species from natural leaf images. The use of scarcely employed curvature-based features proves to be an essential asset for identification, as well as intuitive leaf shape characteristics. Everything relies however on the robustness of the segmentation part which is obviously the toughest task in a complex natural environment. The methods we presented deal with a good number of obstacles one can encounter in such images, but are not yet the perfect key to unlock this problem. A survey of other existing image models and segmentation techniques is engaged to assess their effectiveness in our context.

We aim at recognizing most of native European tree species, which sum up to nearly 150, but the models, upon which our recognition is built, need extension and refining to fit more complicated leaf shapes. The case of compound leaves is a challenging problem that will be at the heart of our further developments. With the number of potential classes increasing, it will become also a necessity to reduce the scope of the search, by including geographical information linked to the GPS system that now exists in every smartphone. Knowing in advance which species are likely to

be found in the geographical area where the user stands may be a decisive step towards a truly reliable identification.

The implementation of our current algorithms on mobile devices through the Folia application is a way to put them into concrete use, with the hope of making it accessible for users who are not necessarily specialists. The performance in terms of processing time is also a concern, as the execution that now lasts a few seconds on a computer, is generally 10 times longer on a smartphone. The introduction of high-level botany-inspired descriptors makes it also conceivable to develop an educational interface, introducing a semantic level displayed to the user. It is of course a great plus to explain in intelligible terms what led the system to its decision. And it would also be beneficial in terms of identification, where interaction and correction would be possible if the words presented to the user do not fit what lies in front of him.

We hope now to have laid the foundations for a new tree identification application, covering the diversity of European species, and destined to anyone with interest in plants but without the otherwise compulsory botanical background. The goal is still to open a little door to this field, as a way to transmit recognition skills that are today a rare knowledge, and to satisfy in the same time the everlasting desire to understand nature.

Appendix A. Evidence-based dissimilarity computation

We consider the problem of determining the actual dissimilarity of a pixel x to the leaf color model, using an evidence-based combination of its distance $d_{LR} = d_{LinReg}(x)$ to the global model and its distance $d_{AM} = d_{Adaptive}(x)$ to the local adaptive mean.

Considering the binary problem of knowing whether the pixel x is a leaf pixel (L) or not (\bar{L}), we place ourselves in a universe

$X = \{L, \bar{L}\}$. In the following we will work in the power set $\mathcal{P}(X) = \{\{L\}, \{\bar{L}\}, \{L, \bar{L}\}\}$.

The first step is then to define two basic belief assignments m_{LR} and m_{AM} over $\mathcal{P}(X)$. As explained in Section 4.3, d_{AM} is only used to put belief mass on the \bar{L} hypothesis whereas d_{LR} does on L . The functions we use to assign masses are affine functions of the distances that have to remain between 0 and 1. The coefficients $\lambda_0, \lambda_1, \alpha_0$ and α_1 have been set experimentally, but could also be have been learned.

$$\begin{aligned} m_{LR}\{L\} &= \max(\lambda_1 d_{LR} + \lambda_0, 0) & m_{AM}\{L\} &= 0 \\ m_{LR}\{L, \bar{L}\} &= 0 & m_{AM}\{\bar{L}\} &= \min(\alpha_1 d_{AM} + \alpha_0, 1) \\ m_{LR}\{L, \bar{L}\} &= 1 - m_{LR}\{L\} & m_{AM}\{L, \bar{L}\} &= 1 - m_{AM}\{\bar{L}\} \end{aligned} \quad (\text{A.1})$$

Dempster's rule of combination provides a way to compute the joint mass distribution $m_{LR \oplus AM} = m_{LR} \oplus m_{AM}$, resulting from the combination of evidence from both sources. It relies on a measure K of the conflict between the two distributions:

$$\begin{aligned} K &= \sum_{\substack{A, B \in \mathcal{P}(X) \\ A \cap B = \emptyset}} m_{LR}(A) \cdot m_{AM}(B) \\ &= m_{LR}\{L\} m_{AM}\{\bar{L}\} + m_{LR}\{\bar{L}\} m_{AM}\{L\} = m_{LR}\{L\} m_{AM}\{\bar{L}\} \end{aligned} \quad (\text{A.2})$$

$$\begin{aligned} m_{LR \oplus AM}\{L\} &= \frac{1}{1-K} \sum_{\substack{A, B \in \mathcal{P}(X) \\ A \cap B = \{L\}}} m_{LR}(A) \cdot m_{AM}(B) \\ &= \frac{1}{1-K} (m_{LR}\{L\} m_{AM}\{L\} + m_{LR}\{L\} m_{AM}\{L, \bar{L}\} + m_{LR}\{L, \bar{L}\} m_{AM}\{L\}) \\ &= \frac{1}{1-K} m_{LR}\{L\} m_{AM}\{L, \bar{L}\} \end{aligned} \quad (\text{A.3})$$

$$\begin{aligned} m_{LR \oplus AM}\{\bar{L}\} &= \frac{1}{1-K} \sum_{\substack{A, B \in \mathcal{P}(X) \\ A \cap B = \{\bar{L}\}}} m_{LR}(A) \cdot m_{AM}(B) \\ &= \frac{1}{1-K} (m_{LR}\{\bar{L}\} m_{AM}\{\bar{L}\} + m_{LR}\{\bar{L}\} m_{AM}\{L, \bar{L}\} + m_{LR}\{L, \bar{L}\} m_{AM}\{\bar{L}\}) \\ &= \frac{1}{1-K} m_{LR}\{L, \bar{L}\} m_{AM}\{\bar{L}\} \end{aligned} \quad (\text{A.4})$$

$$\begin{aligned} m_{LR \oplus AM}\{L, \bar{L}\} &= \frac{1}{1-K} \sum_{\substack{A, B \in \mathcal{P}(X) \\ A \cap B = \{L, \bar{L}\}}} m_{LR}(A) \cdot m_{AM}(B) \\ &= \frac{1}{1-K} m_{LR}\{L, \bar{L}\} m_{AM}\{L, \bar{L}\} \end{aligned} \quad (\text{A.5})$$

The joint distribution defines a couple of belief and plausibility of being a leaf pixel $bel(L)$ and $pl(L)$, the definition of which is given by:

$$\begin{aligned} bel_{LR \oplus AM}(L) &= \sum_{A|A \subseteq \{L\}} m_{LR \oplus AM}(A) = m_{LR \oplus AM}\{L\} \\ pl_{LR \oplus AM}(L) &= \sum_{A|A \cap \{L\} \neq \emptyset} m_{LR \oplus AM}(A) = m_{LR \oplus AM}\{L\} + m_{LR \oplus AM}\{L, \bar{L}\} \end{aligned} \quad (\text{A.6})$$

We consider then, as an approximation, that the actual probability that the pixel is a leaf pixel, which is somewhere between $bel_{LR \oplus AM}(L)$ and $pl_{LR \oplus AM}(L)$ to be right in the middle, and compute the dissimilarity term following this assumption. For normalization purposes, this term is only proportional to the probability that the pixel is not a leaf pixel

$$\begin{aligned} d_{Belief}(X) &\propto 1 - \frac{bel_{LR \oplus AM}(L) + pl_{LR \oplus AM}(L)}{2} \\ &\propto 1 - m_{LR \oplus AM}\{L\} - \frac{1}{2} m_{LR \oplus AM}\{L, \bar{L}\} \end{aligned} \quad (\text{A.7})$$

This dissimilarity, by definition of the rule of combination, and given that the basic belief assignments are affine functions, will generally be a rational function of degree 2 of d_{LR} and d_{AM} that takes into account more information than a regular linear combination, or a minimal value.

Appendix B. Segmentation results

To measure the performance of our segmentation algorithm, we performed measurements on 226 manually segmented natural background images. The value we measure is an overlap factor ρ between the final region we obtain after the evolution of the guided active contour Ω , and the manual ground truth Ω_T .

$$\rho(\Omega, \Omega_T) = \frac{\text{card}(\Omega \cap \Omega_T)}{\text{card}(\Omega \cup \Omega_T)} \quad (\text{B.1})$$

Though it is not the most accurate measure of how well the algorithm will perform afterwards, it is still a less tolerant measure than, for instance, the Dice index or other similar measures, and has the advantage of being scale invariant and always between 0 and 1.

For comparison purposes, we estimated this overlap factor for segmentation produced with various dissimilarity maps: with the linear regression alone [LR], with the adaptive local mean [AM], with the mean of those two, with their minimum, and finally with the evidence-based combination. The results of this comparison are presented in Fig. B.21 as cumulated histograms of the overlap, representing for each overlap value, the percentage of the images for which this value is reached. Such curves are obviously better as they tend to the upper-left corner. This underlines the benefit

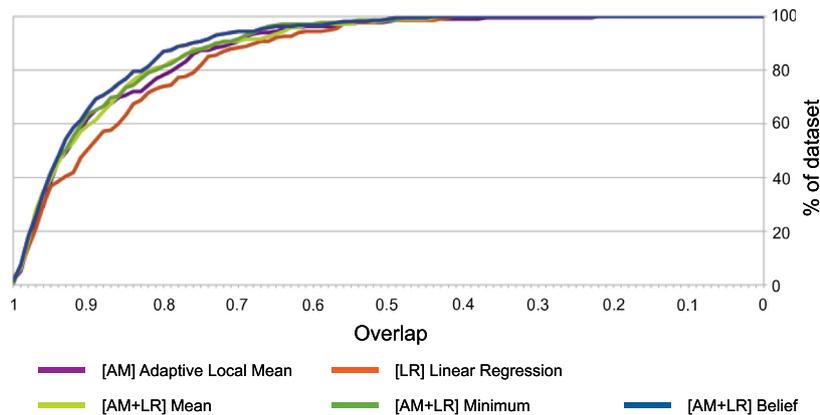


Fig. B.21. Overlap curves for different computations of the dissimilarity map.

of a combination of the global and local approaches, and also the interest of introducing evidence in this combination process.

We also compared our results with the segmentation obtained with the same dissimilarity map (evidence-based combination) but without the first step of computing the polygonal model. The region is initialized as an initial polygon, but then it simply tries to evolve and minimize the energy in which the weight ω_{II} corresponding to the shape constraint is set 0 (there is anyway no shape to constrain it in the first place). The resulting overlap curves are shown in Fig. B.22 where it appears that, if the polygonal constraint seem to lower performance for very easy leaves (overlap over 0.95) it has a significant influence to produce good results on more complicated images.

To illustrate the procedure of our segmentation algorithm, Fig. B.23 provides examples of leaf segmentation of various images under different conditions. It details the different steps of both the active parametric polygon computation and the final guided active contour evolution.

Appendix C. A distance for classification

We consider a class Φ in a N -dimensional parameter space to be represented by its centroid $(\mu_i)_{i=1..N}$ and its covariance matrix Σ . Given a parameter vector $(x_i)_{i=1..N}$, we want to compute its distance to the centroid while taking into account the variability of the class. A standard measure would be the Mahalanobis distance:

$$\|x - \mu\|_M = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \quad (C.1)$$

This distance tends however to stretch or contract the parameter space in different ways for the different classes, which is a disadvantage in our case where the species may have very different variabilities over different parameters. The risk is that a point which is very close (in terms of Euclidean distance) from a centroid may appear closer from a more distant one if the class it represents has a lot of variability. Very variable classes tend to swallow up more restricted ones, that will always appear more distant.

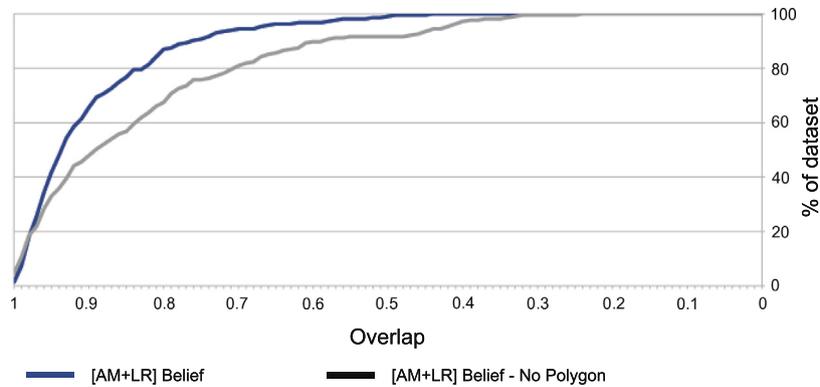


Fig. B.22. Overlap curves for segmentation with and without polygonal approximation and constraint.

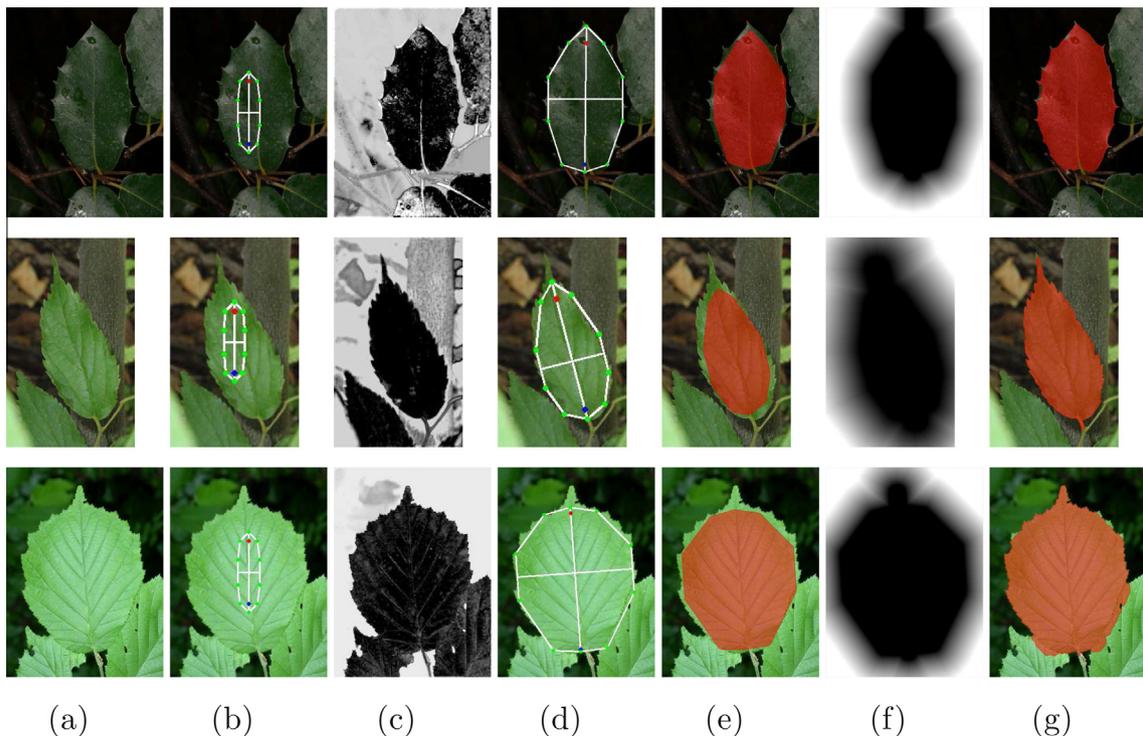


Fig. B.23. Segmentation process: image (a), polygonal model initialization (b), color dissimilarity map (c), final polygonal model (d), active contour initialization (e), guiding distance map (f) and final segmentation (g).

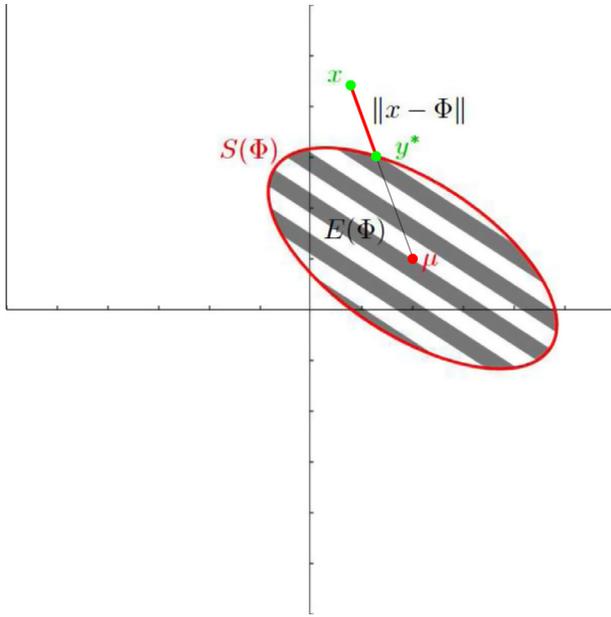


Fig. C.24. Distance of x to the surface of class Φ in a 2-dimensional case.

To avoid this distortion, we want to always compute distances with the same scale, as does the Euclidean distance. To include the width of each class in the computation, we represent a class by the hyper-ellipsoid centered at μ and with axes and dimensions determined by the covariance matrix Σ . This volume is then defined as:

$$E(\Phi) = \{y, \sqrt{(y - \mu)^T \Sigma^{-1} (y - \mu)} \leq 1\} \quad (C.2)$$

If the point x lies inside this ellipsoid, we can consider to be within the class Φ , and then we should have $\forall x \in E(\Phi), \|x - \Phi\| = 0$. Outside of it, we want to estimate the distance of x to the surface of $E(\Phi)$ given by:

$$S(\Phi) = \{y, \sqrt{(y - \mu)^T \Sigma^{-1} (y - \mu)} = 1\} \quad (C.3)$$

The distance we chose to compute is the distance from x to the point y^* of $S(\Phi)$ that is in the same direction as x from the centroid μ . Fig. C.24 illustrates this definition in a simple 2-dimensional case. y^* verifies then the two following properties:

$$\begin{aligned} \sqrt{(y^* - \mu)^T \Sigma^{-1} (y^* - \mu)} &= 1 \\ \exists k \in \mathbb{R}^+, (x - \mu) &= k(y^* - \mu) \end{aligned} \quad (C.4)$$

Consequently, we can write that:

$$\begin{aligned} \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} &= \sqrt{(k(y^* - \mu))^T \Sigma^{-1} (k(y^* - \mu))} \\ &= k \sqrt{(y^* - \mu)^T \Sigma^{-1} (y^* - \mu)} = k \end{aligned} \quad (C.5)$$

This equation leads to:

$$y^* = \mu + \frac{x - \mu}{\sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}} \quad (C.6)$$

And finally:

$$\begin{aligned} \|x - \Phi\| &= \|x - y^*\|_2 = \|x - \mu - \frac{x - \mu}{\sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}}\|_2 \\ &= \|(x - \mu) \left(1 - \frac{1}{\sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}}\right)\|_2 = \left(1 - \frac{1}{\|x - \mu\|_M}\right) \|x - \mu\|_2 \end{aligned} \quad (C.7)$$

When x is inside $E(\Phi)$, the distance $\sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \leq 1$, leading the distance to be negative. If we want $\|x - \Phi\|$ to be equal to 0 inside the ellipsoid, the distance can be written as:

$$\|x - \Phi\| = \|x - \mu\|_2 \max\left(1 - \frac{1}{\|x - \mu\|_M}, 0\right) \quad (C.8)$$

If we simplify our initial problem to the case where the covariance matrix is diagonal, with variances $(\sigma_i^2)_{i=1..N}$, the ellipsoid is aligned on the axes, and the Mahalanobis distance becomes a normalized Euclidean distance. The expression for the distance to its surface becomes then:

$$\|x - \Phi\| = \left(\sqrt{\sum_{i=1}^N (x_i - \mu_i)^2}\right) \max\left(1 - \frac{1}{\sqrt{\sum_{i=1}^N \frac{(x_i - \mu_i)^2}{\sigma_i^2}}}, 0\right) \quad (C.9)$$

This distance is proportional to the Euclidean distance between a point and the centroid, avoiding risky distortions of the parameter space, and still gives an advantage to highly variable classes. However points that are very distant from the centroid in the first place remain far from the surface, which makes sure that they can be recognized as closer to other centroids, even with smaller variances.

References

- [1] P. Belhumeur, D. Chen, S. Feiner, D. Jacobs, W. Kress, H. Ling, I. Lopez, R. Ramamoorthi, S. Sheorey, S. White, L. Zhang, Searching the World's Herbaria: a system for visual identification of plant species, in: European Conference on Computer Vision, 2008.
- [2] H. Goëau, P. Bonnet, A. Joly, N. Boujemaa, D. Barthelemy, J.-F. Molino, P. Bimbaum, E. Mouysset, M. Picard, The CLEF 2011 Plant Images Classification Task, 2011a.
- [3] X.F. Wang, D.S. Huang, J.X. Du, X. Huan, L. Heutte, Classification of plant leaf images with complicated background, *Applied Mathematics and Computation* 205 (2) (2008) 916–926.
- [4] M.-E. Nilsback, A. Zisserman, Delving into the whorl of flower segmentation, in: *British Machine Vision Conference*, vol. 1, 2007, pp. 570–579.
- [5] H. Goëau, A. Joly, I. Yahiaoui, P. Bonnet, E. Mouysset, Participation of INRIA & Pl@ntNet to ImageCLEF 2011 plant images classification task, in: *CLEF (Notebook Papers/Labs/Workshop)*, 2011b.
- [6] F. Mokhtarian, S. Abbasi, Matching shapes with self-intersections: application to leaf classification, *IEEE Transactions on Image Processing* 13 (5) (2004) 653–661.
- [7] Z. Wang, Z. Chi, D. Feng, Q. Wang, Leaf image retrieval with shape features, in: *Advances in Visual Information Systems*, vol. 1929 of Lecture Notes in Computer Science, 2000, pp. 41–52.
- [8] C.-H. Teng, Y.-T. Kuo, Y.-S. Chen, Leaf segmentation, its 3D position estimation and leaf classification from a few images with very close viewpoints, in: *Proceedings of the 6th International Conference on Image Analysis and Recognition, ICIAR '09*, 2009, pp. 937–946.
- [9] T. Saitoh, T. Kaneko, Automatic recognition of wild flowers, in: *International Conference on Pattern Recognition*, 2000, pp. 2507–2510.
- [10] J.X. Du, X.F. Wang, G.-J. Zhang, Leaf shape based plant species recognition, *Applied Mathematics and Computation* 185 (2007) 883–893.
- [11] C. Im, H. Nishida, T.L. Kunii, A hierarchical method of recognizing plant species by leaf shapes, in: *MVA*, 1998, pp. 158–161.
- [12] T. Saitoh, T. Kaneko, Automatic recognition of blooming flowers, in: *International Conference on Pattern Recognition*, vol. 1, 2004, pp. 27–30.
- [13] A.-G. Manh, G. Rabatel, L. Assemat, M.-J. Aldon, Weed leaf image segmentation by deformable templates, *Journal of Agricultural Engineering Research* 80 (2) (2001) 139–146.
- [14] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, *International Journal of Computer Vision* 1 (4) (1988) 321–331.
- [15] T. Chan, L. Vese, Active contours without edges, *IEEE Transactions on Image Processing* 10 (2) (2001) 266–277.
- [16] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, *Journal of Computational Physics* 79 (1) (1988) 12–49.
- [17] G. Unal, A. Yezzi, H. Krim, Information-theoretic active polygons for unsupervised texture segmentation, *International Journal of Computer Vision* 62 (3) (2005) 199–220.
- [18] A. Yuille, P. Hallinan, D. Cohen, Feature extraction from faces using deformable templates, *International Journal of Computer Vision* 8 (2) (1992) 99–111.
- [19] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models – their training and application, *Computer Vision and Image Understanding* 61 (1) (1995) 38–59.
- [20] P. Felzenszwalb, Representation and detection of deformable shapes, *PAMI* 27 (2004) 208–220.
- [21] D. Cremers, F. Tischhäuser, J. Weickert, C. Schnörr, Diffusion snakes: introducing statistical shape knowledge into the Mumford–Shah functional, *International Journal of Computer Vision* 50 (2002) 295–313.

- [22] G. Cerutti, L. Tougne, A. Vacavant, D. Coquin, A parametric active polygon for leaf segmentation and shape estimation, in: 7th International Symposium on Visual Computing, 2011a.
- [23] L. Cheng, T. Caelli, A. Sanchez-Azofeifa, Component optimization for image understanding: a bayesian approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(5) (2006).
- [24] A.P. Dempster, A generalization of bayesian inference, *Journal of the Royal Statistical Society* 30 (1968) 205–247.
- [25] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.
- [26] G. Cerutti, L. Tougne, J. Mille, A. Vacavant, D. Coquin, Guiding active contours for tree leaf segmentation and identification, in: CLEF (Notebook Papers/Labs/Workshop), 2011b.
- [27] H. Coste, *Flore descriptive et illustrée de la France de la Corse et des contrées limitrophes*, 1906.
- [28] Y. Shi, W.C. Karl, A real-time algorithm for the approximation of level-set-based curve evolution, *IEEE Transactions on Image Processing* 17 (5) (2008) 645–656.
- [29] R. Malladi, J.A. Sethian, B.C. Vemuri, Shape modeling with front propagation: a level set approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (2) (1995) 158–175.
- [30] C.H. Teh, R.T. Chin, On the detection of dominant points on digital curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (8) (1989) 859–872.
- [31] S.C. Pei, C.N. Lin, The detection of dominant points on digital curves by scale-space filtering, *Pattern Recognition* 25 (11) (1992) 1307–1314.



Guillaume Cerutti was born in 1986. He received the Engineer degree from the Ecole Nationale Supérieure des Télécommunications (Telecom ParisTech), Paris, France and the M.S. degree in Computer Science from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2009. Since 2010, he is Ph.D. student at the University of Lyon 2 and works at the LIRIS Laboratory, Lyon, France.



Pr. Laure Tougne was born in 1971. She received her Ph.D. in 1997 in Computer Science from the Ecole Normale Supérieure of Lyon, France. From 1997 to 2005, she was Associated Professor at the University of Lyon 2. Since 2005, she is Professor in Computer Science at the University of Lyon 2 and works at the LIRIS Laboratory, Lyon, France. Her research topics concern image analysis, video analysis and discrete geometry.



Julien Mille received the M.S. degree and Ph.D. degree in computer science from Université de Tours, France, in 2004 and 2007, respectively. In 2008, he was a post-doctoral researcher at Université Paris Dauphine. Since 2009, he is an associate professor in Lyon, where he teaches computer science at Université Lyon 1, France and do his research in the LIRIS laboratory. He is interested in image and video analysis, especially variational and discrete optimization methods for segmentation and tracking.



Antoine Vacavant obtained the M.S. degree from the Université Lyon 1, France, in 2005, and the Ph.D. degree in Computer Science from the Université Lyon 2 in 2008. He is now associate professor at the Université d'Auvergne Clermont 1, in the IUT of Le Puy en Velay. He is member of the ISIT lab, and his main research topics are discrete and computational geometry, image analysis and processing.



Pr. Didier Coquin received the Ph.D. degree in Signal Processing and Telecommunication from University of Rennes I, France, in 1991. From 1992 to 2012, he was Associate Professor of telecommunication and network engineering at the Savoie University (Technological Academic Institute). Since 2012, he is Professor at the Savoie University and works in the Informatics, Systems, Information and Knowledge Porocessing Laboratory (LISTIC), Annecy, France. His research interests include image processing, pattern recognition, image and video indexation, data fusion and fuzzy logic.