Reconstructing Plants in 3D from a Single Image Using Analysis-by-Synthesis

Jérôme Guénard, Géraldine Morin, Frédéric Boudon, Vincent Charvillat

Reconstructing Plants in 3D from a Single Image using Analysis-by-Synthesis

Jérôme Guénard[1] Géraldine Morin[1] Frédéric Boudon[2] Vincent Charvillat[1]

[1] IRIT - VORTEX - University of Toulouse
[2] INRIA - Cirad - Montpellier

Abstract. Mature computer vision techniques allow the reconstruction of challenging 3D objects from images. However, due to high complexity of plant topology, dedicated methods for generating 3D plant models must be devised. We propose to generate a 3D model of a plant, using an analysis-by-synthesis method mixing information from a single image and a priori knowledge of the plant species.

First, our dedicated skeletonisation algorithm generates a possible branching structure from the foliage segmentation. Then, a 3D generative model, based on a parametric model of branching systems that takes into account botanical knowledge is built. The resulting skeleton follows the hierarchical organisation of natural branching structures. An instance of a 3D model can be generated. Moreover, varying parameter values of the generative model (main branching structure of the plant and foliage), we produce a series of candidate models. The reconstruction is improved by selecting the model among these proposals based on a matching criterion with the image. Realistic results obtained on different species of plants illustrate the performance of the proposed method.

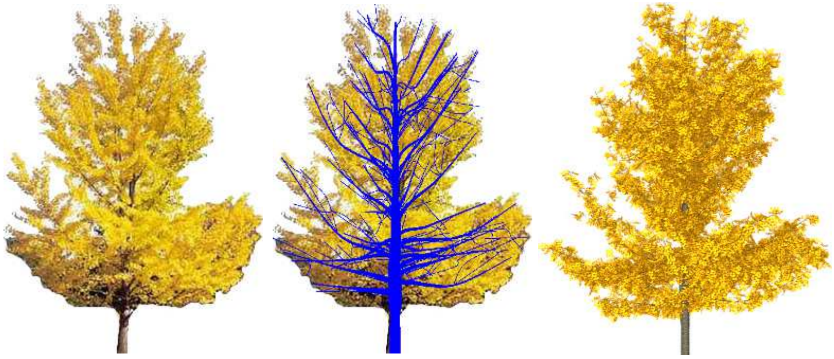

Fig. 1. On the left, an original image of Ginkgo tree. In the middle, a possible architecture of the branching extracted with our skeletonisation method. On the right, a 3D model of this tree with the same viewpoint.

1 Introduction

Procedural methods to generate plant models allow to build a complex plant architecture from few simple rules [1]. Lindenmayer first proposed the formalism of L-systems as a general framework in [2]. By carefully parameterising the rules, it is possible to achieve a large variety of realistic plant shapes [3, 4]. However, a strict recursive application of rules leads to self-similar structures and thus, to enhance realism, irregularities may be generated through probabilistic approaches [5, 1]. Adjusting stochastic parameters to achieve realistic models requires intensive botanical knowledge [6]. Another approach consists in modelling plant irregularities as a result of the competition for space between the different organs of the plants [7]. In this case, the volume of a plant is specified by the user and a generative process grows a branching structure with branches competing between each other. Competition can be biased to favor certain types of structures. However, automatic control of competition parameters to achieve a given shape is still complicated.

All these developments emerged from the computer graphics community. Other approaches use additional information provided by images to increase the degree of realism. Clearly, a plant should follow the biological property of its species and also resemble a picture of an existing instance. That is typically the subject of our work. Our idea is not of exactly reconstructing the plant from an image, including its hidden parts (which seems impracticable) but that of driving the instantiation of the plant 3D model by minimising the difference between its reprojection and the original plant in the image.

Unlike existing methods detailed in section 2, ours must be able to get a 3D model of a plant without any human interaction from images with possibly no visible branches. By integrating biological knowledge of the plant species, we propose a simple fully-automatic process to extract the structure of a plant from the shape of its foliage in a picture taken with as few restrictions as possible so the image may be of poor quality. We present a new skeletonisation algorithm in section 3 (middle image in Fig. 1). Then, an analysis-by-synthesis schemes generate multiple possible 3D model and selects the one insuring that the foliage model reprojection matches closely the original foliage like explained in section 4 (right on the Fig. 1). The last section shows results and validation comparing with data provided by experts.

2 State of the art: generating plants from images

Realistic plants are challenging objects to model and recent advances in automatic modelling can be explained by the convergence of computer graphics and computer vision [8]. We start this state of the art with the first method of plant modelling from images. Then, we continue with the ones starting by reconstructing clouds of 3D points. After, we talk about other methods using several images to finish with approaches using a single image as ours.

A pioneering work on the reconstruction of trees from images was made by Shlyakhter *et al.* [9] who reconstruct the visual hull of the tree from silhouettes deduced from the images. A skeleton is computed from the hull using a Medial Axis Transform (MAT) and is used as main branches. Branchlets and leaves are then generated with an L-system. The skeleton determined from the MAT does not necessarily look like a realistic branching system. Also, the density of the original tree is not taken into account.

Quan *et al.* [10, 11] and Tan *et al.* [12] also use multiple images to reconstruct a 3D model of trees or plants. In order to avoid the features correspondences in different images, they use views close to each other (more than 20 images for any plants). They obtain a quasi-dense cloud of points by structure from motion. For simple plants, a parametric model is first fitted on each set of points representing a leaf. They then generate branches based on information given by the user. For trees, they start by reconstructing visible branches to create branch pattern that they combine in a fractal way until reaching leaves. Reche-Martinez *et al.* [13] propose another reconstruction from multiple images, based on billboards. Neubert *et al.* [14] construct a volume encompassing the plant in the form of voxels using image processing techniques and fill it with particles. Particles path toward the ground and a user given general skeleton are used as branching system.

Wang *et al.* [15] model different species of trees using images of tree samples from the real world which are analysed to extract similar elements. A stochastic model to assemble these element is also parameterized from the image and make it possible to generate many similar trees. The goal in this case is not necessarily to reconstruct a specific tree instance corresponding to an image. Similarly, Li *et al.* [16] propose a probabilistic approach to reconstruct a tree parameterized from videos. For these methods, the only source of information is the given images leading to template branching patterns. If the set of patterns is rich enough, it will produce aesthetically pleasing results, but without guarantee to be representative of its species. Additionally, user input are required to specify a draft of the structure on the image to avoid segmentation. Talton *et al.*, in [17], propose to fit a grammar-based procedural methods using MCMC technique to model objects from a 2D or 3D binary shape. Their results are aesthetically very convincing but optimization of their models requires long computation time.

Other approaches explore the use of a single image [18, 19]. In [18], the foliage of the plant is segmented by the user and visibles branches are extracted. A 3D representation of the skeleton is consequently deducted from visible parts and the encompassing volume, then the leaves are added. In [19], a graph topology is first extracted from a single image of a branching system (a tree without foliage). Then the 3D tree model is reconstructed by rotating the branches.

In general, methods of the literature, such as [12] and [18] require visible branches to learn about the structure of the skeleton. In our case, branches are derived directly from foliage structure. Indeed, the branching structure devised manually by experts from image show that the branches are deduced on one hand from the knowledge of a space filled by a branch and its attached leaves and on the other hand the silhouette of the foliage (see left of Fig. 9 on a vine example). We propose a generalised recursive skeletonisation algorithm together with an analysis-by-synthesis mechanism to determine the branches and their attached foliage that is the 3D model. Our approach is fully-automatic, that is, does not require any user interaction.

3 Analysis part: skeleton extraction

3.1 General field skeletonisation method

Skeletonisation is a classical topic in image processing. We followed the analysis of different approaches as proposed in [20]. In this article, they detail the different properties a skeleton may respect and analyse different approaches. For example in our case, the smoothness is very important to get realistic branches but we do not need a centred skeleton. Moreover, the connectivity is less important because it can be ensured by another way. We choose to adapt the general field method, and in particular the work of Cornea [21], since the properties of the derived skeleton best fits our needs.

Cornea *et al.* original method [21] consists in computing the skeleton (Fig. 2 (c)) from a vector field (Fig. 2 (b)). For each interior pixel $\mathbf{p_i}$ of the binary shape $\mathcal{B}$, a force vector $\overrightarrow{\mathbf{f_i}}$ is computed as a weighted average of unit vectors to the boundary pixels: $\overrightarrow{\mathbf{f}}_i = \sum_{\mathbf{m_j} \in \Omega} \frac{1}{||\overrightarrow{\mathbf{m_j p_i}}||^2} \frac{\overrightarrow{\mathbf{m_j p_i}}}{||\overrightarrow{\mathbf{m_j p_i}}||}$ where Ω contains the contour pixels $\mathbf{m_j}$ of $\mathcal{B}$ (Fig. 2 (a)). Then, points where the magnitude of the force vector vanishes, so-called *critical points* (Fig. 2 (b)), are connected by following the force direction pixel by pixel. The result of this method can be seen in Fig. 2 (c). This original method have some drawbacks for our particular application that we address:

- it is not robust to holes in the binary shape;
- the structure of the branching system need to look realistic (for example, plants are organised around a main trunk in monopodial case);
- the number of branches need to be increased in large areas.

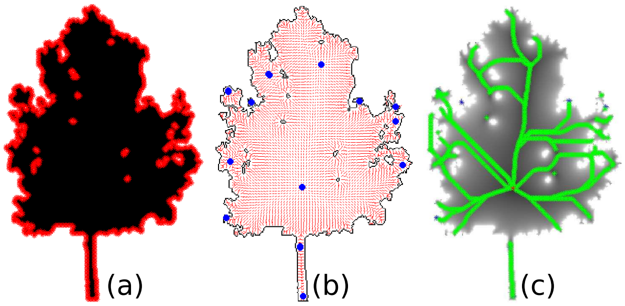

Fig. 2. Cornea *et al.* original method. (a) shape $\mathcal{B}$ with contours pixels $\in \Omega$ represented in red. (b) vector field with critical points in blue. (c) extracted skeleton in green.

3.2 A new computation of the vector field

We adapt Cornea's vector field method to get a realistic skeleton in 2D. Here, we explain the method with monopodial plants (i.e. plants organised around a main trunk).

Based on botanical expertise, we assume that different branches of relatively similar size coexist and share the space of the crown of a plant. A large convex silhouette may in fact be the sum of all this branching system. For the skeleton reflects this hierarchy of branches, we propose a strategy to partition silhouette space into subsets by positioning artificial *contour points* in the shape (see Fig. 3).

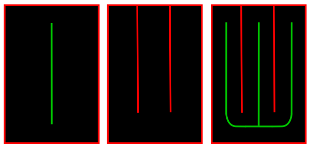

Fig. 3. On the left, the skeleton (in green) extracted with Cornea's original method. Adding artificial contour points (red lines) constraints the skeleton computation to have more branches. On the right, the resulting skeleton when all red points are considered as contour points.

3.3 Definition of the probability map

Because we want to have a non deterministic model, we compute a probability map $\mathcal{P}$ on $\mathcal{B}$. For each interior point $\mathbf{p_i}$, $\mathcal{P}_i$ is the probability to be considered as a *contour point*. The new force vector $\overrightarrow{\mathbf{f_i}}$ now depends on all points $\mathbf{m_j} \in \Omega$:

$$\overrightarrow{\mathbf{f}}_i = \sum_{\mathbf{m_j} \in \Omega} \frac{1}{||\overrightarrow{\mathbf{m_j p_i}}||^2} \frac{\overrightarrow{\mathbf{m_j p_i}}}{||\overrightarrow{\mathbf{m_j p_i}}||} + \sum_{\substack{\mathbf{p_j} \in \mathcal{B} \setminus \Omega \\ j \neq i}} \frac{\mathcal{P}_j}{||\overrightarrow{\mathbf{p_j p_i}}||^2} \frac{\overrightarrow{\mathbf{p_j p_i}}}{||\overrightarrow{\mathbf{p_j p_i}}||} \qquad (1)$$

We assume here that n the number of branches is given. We compute the *probability map* $\mathcal{P}$ with an iterative algorithm. The first step is the choice of cuts in $\mathcal{B}$. The cuts are segments with one *starting point* and one *ending point* and represent the possible positions of the separations between the n branches in the shape. Assuming that the shoots grow from the vertical trunk, we propose to place trivially the *ending points* $\mathbf{e_i}, i = 1..n-1$ of the cuts uniformly in the middle of $\mathcal{B}$ (Fig. 4 (b)). Then, the *starting points* are computed one by one. To do that, we compute the DCE (*Discrete Curve Evolution*) of Ω as in [22]. It provides a simplified polygonal boundary composed of N vertices $(\mathbf{s_l})_{,l=1..N}$ (Fig. 4 (a)). Usually, we choose $N = 2n$. An angle α_l can be associated with each vertex, representing clockwise angle between the 2 segments around the vertex. A set of points $(\mathbf{c_k})_{,k=1..K}$ uniformly discretises the polygon.

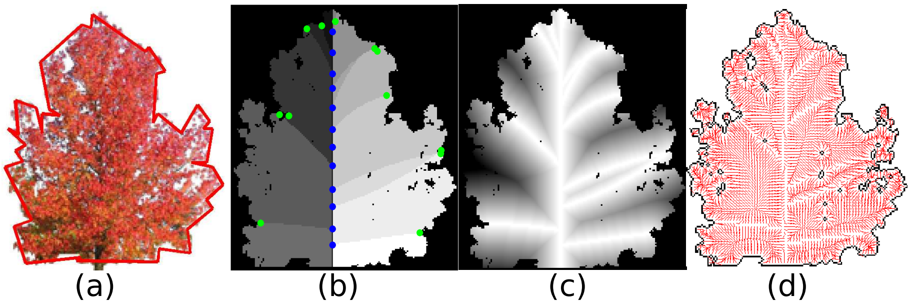

Fig. 4. On the left, an original image and in red, DCE result with 42 vertices. In the middle, an example of cuts with $n = 14$ branches and the probability map. On the right, the new vector field.

Then a new probability ρ_k to be a *starting point* is computed for each point $\mathbf{c_k}$ taking into account two values:

- the proximity to an inward angle $\rho_k^1 \sim \sum_{l=1}^{N} \frac{1}{d(\mathbf{c_k},\mathbf{s_l})}(1 - \frac{\alpha_l}{2\pi})$,
- the distance along a boundary to the set $\mathcal{H}$ of already chosen *starting points* $\rho_k^2 \sim \min_{\mathbf{c} \in \mathcal{H}} d(\mathbf{c_k}, \mathbf{c})$.

The mix probability ρ_k is proportional to $\phi(\rho_k^1, 1, \sigma) + \phi(\rho_k^2, 1, \sigma)$ where $\phi(., 1, \sigma)$ represents the gaussian function with a mean equals to 1 and a standard deviation equals to σ (here, $\sigma = 0.4$). For each ending point, a starting point $\mathbf{c_k}$ is selected according to the probability ρ_k. The cut is accepted if the angle between the cut and the trunk is coherent (for the Liquidambar example, around $\frac{\pi}{2}$ in the bottom of the tree, $\frac{\pi}{6}$ in the top and with an angle computed linearly between these two values for an intermediate cut). The cuts are quadratic curves for which we fix tangent direction.

When all the cuts have been accepted, the probability map (Fig. 4 (c)) is computed. $\mathcal{P}_i = 1$ on the boundary and $\frac{1}{s\sqrt{2\pi}} \exp \frac{-\delta(\mathbf{p_i})}{2s^2}$ elsewhere ($\delta(\mathbf{p_i})$ is the euclidean distance between $\mathbf{p_i}$ and its projection on the closest cut). Finally the new vector field is computed (Fig. 4 (d)).

We now have a vector field coherent with the n branches assumption. We want to extract branches from this vector field. For each row i of the image and each area p of the partition (as we can see Fig. 4 (b)), we extract the attracting point $\mathbf{a_i^P}$ which is the point with the smallest vector norm. Each branch $\mathbf{b_p}$ is a Catmull-Rom curve adjusted on the attracting points $\mathbf{a_i^P}$, using a least square criterion. Extracted skeletons are shown on the left of Fig. 9.

3.4 Iterative skeletonisation algorithm

The branches are extracted using our algorithm presented above. This algorithm is applied recursively to get second order branches for each partition. We can see an example of cuts with a *Liquidambar tree* in Fig. 5.

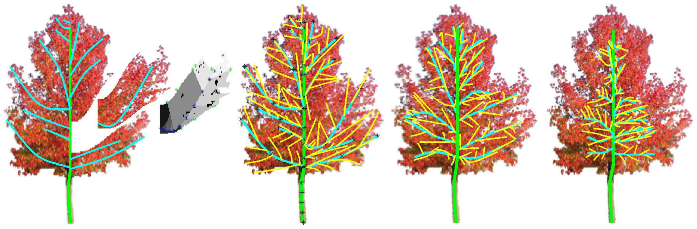

Fig. 5. On the left, first order branches and on the second image, second order branches. On the right, the images represent our skeletonisation algorithm applied on eroded binary shapes of the Liquidambar tree.

3.5 Depth information

To generate 3D information, we drew inspiration from Zeng *et al.* [19] and Okabe *et al.* [23]. The goal is to deduce depth information for the branches in the 2D skeleton to make a realistic plant from other views, preserving the appearance from the original viewpoint as it is shown in Fig. 6. First, we compute the convex hull of our 2D skeleton. Then, revolving this convex hull around the line passing through the trunk, we obtain a encompassing volume of the plant. Considering an orthographic projection onto the ground, for each branch which does not touch the 2D convex hull, we change depth information for that the end of this branch touches the boundary of the bounding volume. We have two possibilities, in the front or in the back. We choose the one which maximises the angles between the projections of all the branches to the ground, adding the branch one by one.

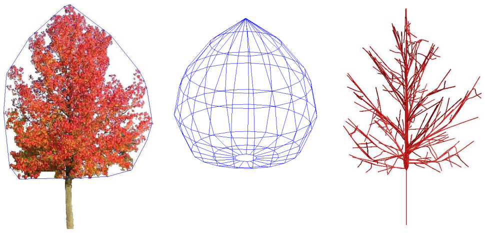

Fig. 6. On the left, we can see the 2D convex hull of the foliage, in the middle, thebounding volume and on the right, the final 3D skeleton of a Liquidambar tree.

To get enough branches in all the directions, we apply our skeletonisation algorithm on an eroded binary shape of the plant like shown in Fig. 5. These new branches do not touch the convex hull and so, they are mapped into a front or back plane like explained above.

4 Analysis-by-Synthesis Scheme

From a binary image $\mathcal{B}$ representing a segmented foliage and knowledge of the plant, we explained how to extract a possible branching structure. Now, we generate a plant model $\mathcal{M}$ in the L-systems modeller L-Py [24]. L-systems rules will create a branching structure from the estimated skeleton and populate it with leaves and branchlets. The analysis-by-synthesis step aims at improving the quality of the 3D reconstructed model $\mathcal{M}$ knowing $\mathcal{B}$, i.e. maximising the probability $p(\mathcal{M}|\mathcal{B})$. Each parameter (like the number of branches, the position of the cuts or the leaves densities) is a random variable. We generate multiple models $\mathcal{M}_i$ according to these random variables and reproject them 3D in images with the same viewpoint as the original image. We obtain multiple binary shapes $\mathcal{I}_i$ (1 if foliage, 0 elsewhere). Using bayesian formula, we select the best candidate proposed by the generative model:

$$\mathcal{M}_{i_0} = \underset{\mathcal{M}_i}{\operatorname{argmax}}\, p(\mathcal{M}_i|\mathcal{B}) = \underset{\mathcal{M}_i}{\operatorname{argmax}}\, p(\mathcal{M}_i)p(\mathcal{B}|\mathcal{M}_i) \tag{2}$$

where $p(\mathcal{M})$ (a priori law) is a product of terms which are probabilities function of all the knowledge of the plant and

$$p(\mathcal{B}|\mathcal{M}_i) = \frac{\#((\mathcal{I}_i - \mathcal{B})^2 == 1)}{\#pixels(\mathcal{B})} \tag{3}$$

(posterior probability) evaluates the difference between $\mathcal{B}$ and $\mathcal{I}_i$. For example, one of the term of $p(\mathcal{M}_i)$ is a gaussian representing the probability of the number of branches. Fig. 7 shows different error maps where $p(\mathcal{B}|\mathcal{M}_i)$ is the number of gray pixels divided by the total number of pixels.

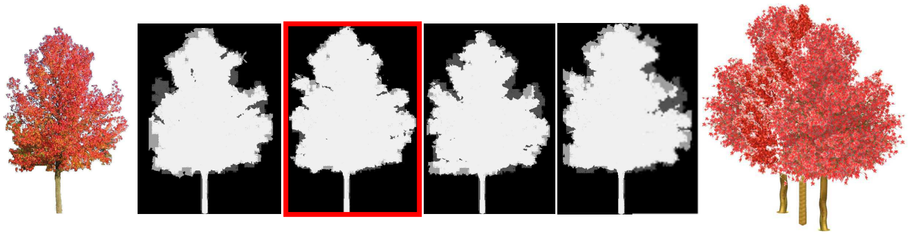

Fig. 7. On the left, the original image. On the right, reprojected models. In the middle, different errors maps with four models with varying numbers of branches, distributions of leaves and densities. White pixels correspond to pixels where the original image and the reprojected one are superposed and gray pixels are *wrong* pixels. The map outlined in red is the error map of the selected 3D model.

In the next session, we show that increasing the number of proposal does decrease the posterior criterion. Then, we evaluate our result in the particular setting of vines.

5 Results and Validation

Reprojection errors

Our method has been tested on a large number of images. Some results are shown in Fig. 1, 9 and 8. We measure the reprojection error by the posterior probability given by the equation (3). The average error for the case of vines is 6.9%, 7.6% for the Ginkgo , 7.2% and 7.5% for the Liquidambar. The picture are taken in arbitrary conditions and may be of poor quality (for example, in the vine case, the image is degraded after a metric rectification due to the assumption that all the principal branches are in a plane). We showed that the greater is the number of tested models, the lower is the reprojection error. This proves the effectiveness of our skeletonisation method which restricts significantly the search space.

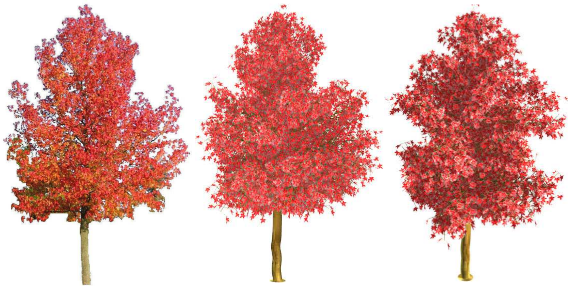

Fig. 8. Liquidambar example. On the left, the original image. In the middle, the 3D model with the same viewpoint. On the right, the 3D model with another viewpoint.

Vines modelling

We used the reconstruction of the models for computing significant parameters for the wine culture. In that context, our models can be compared to the reconstruction done by hand by viticulture experts (Fig. 9). We adapt our method by positioning the cuts vertically from a cane which is attached horizontally by the winegrower. It seems difficult to find a significant measure by comparing the ground truth to our skeletons. So, we used our algorithm on the drawn ground truth skeletons with different leaves distributions and different leaves densities to find the best 3D model. The improvement of the reprojection criterion in comparison to our automatically generated skeleton is only 0.2% in average. This small difference proves the performance of our method which does not require human intervention.

In Tab. 1, we can see the number of shoots drawn by two differents experts from vines images. The last row shows the number of shoots of the 3D models generated with our method from the same images. Our method is able to reconstruct a 3D model with a number of shoots similar with the number estimated by one of the two experts.

Image	1	2	3	4	5	6	7	8	9
First expert estimation	3	6	5	5	6	5	3	7	7
Second expert estimation	2	6	4	5	4	4	2	5	4
Our method	2	5	4	5	6	6	3	5	6

Table 1. The first row represents the number of the vine image. The second and the third rows represent the numbers of shoots drawn by the experts. The last row represent the number of shoots of the 3D models generated with our method from these images.

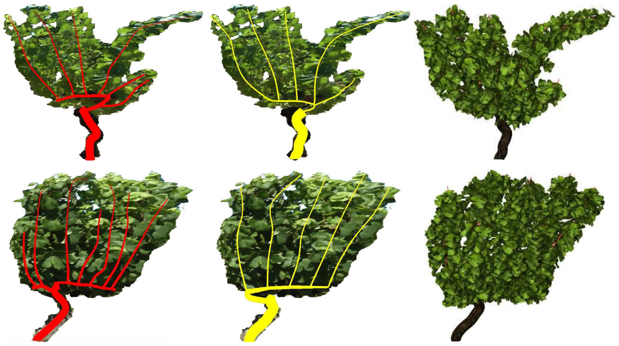

Fig. 9. On the left, a viticulture expert has drawn skeletons on vine images (in red). In the middle, the projections of the skeletons of our method (in yellow). On the right, renderings of automatically generated vine models using our approach.

6 Conclusion and Perspective

Combining analysis and synthesis, we have proposed a new fully-automatic method of plant modelling from a single low resolution image without any branching pattern.

Next, we would like to extend our method to non monopodial trees. The derivation of a priori knowledge on a species could be improved from the models instantiation. Such a learning process could avoid injecting too much knowledge into the system.

References

1. Prusinkiewicz, P., Lindenmayer, A.: The algorithmic beauty of plants. Springer Verlag (1990)
2. Lindenmayer, A.: Mathematical models for cellular interaction in development: Parts i and ii. Journal of Theoretical Biology **18** (1968)
3. Weber, J., Penn, J.: Creation and rendering of realistic trees. SIGGRAPH, ACM (1995) 119–128
4. Deussen, O., Lintermann, B.: Digital Design of Nature: Computer Generated Plants and Organics. Springer-Verlag (2005)
5. de Reffye, P., Edelin, C., Françon, J., Jaeger, M., Puech, C.: Plant models faithful to botanical structure and development. SIGGRAPH, ACM (1988) 151–158

6. Chaubert-Pereira, F., Guédon, Y., Lavergne, C., Trottier, C.: Markov and semi-markov switching linear mixed models used to identify forest tree growth components. Biometrics (2009)
7. Palubicki, W., Horel, K., Longay, S., Runions, A., Lane, B., Měch, R., Prusinkiewicz, P.: Self-organizing tree models for image synthesis. SIGGRAPH (2009) 1–10
8. Quan, L.: Image-based Plant Modeling. Springer (2010)
9. Shlyakhter, I., Rozenoer, M., Dorsey, J., Teller, S.: Reconstructing 3d tree models from instrumented photographs. IEEE Comput. Graph. Appl. (2001) 53–61
10. Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J., Kang, S.B.: Image-based plant modeling. ACM TOG (2006) 599–604
11. Quan, L., Wang, J., Tan, P., Yuan, L.: Image-based modeling by joint segmentation. IJCV (2007) 135–150
12. Tan, P., Zeng, G., Wang, J., Kang, S.B., Quan, L.: Image-based tree modeling. ACM TOG (2007) 87
13. Reche-Martinez, A., Martin, I., Drettakis, G.: Volumetric reconstruction and interactive rendering of trees from photographs. ACM TOG (2004) 720–727
14. Neubert, B., Franken, T., Deussen, O.: Approximate image-based tree-modeling using particle flows. ACM TOG (Proc. of SIGGRAPH) (2007)
15. Wang, R., Hua, W., Dong, Z., Peng, Q., Bao, H.: Synthesizing trees by plantons. Vis. Comput. **22** (2006) 238–248
16. Li, C., Deussen, O., Song, Y.Z., Willis, P., Hall, P.: Modeling and generating moving trees from video. ACM Trans. Graph. **30** (2011) 127:1–127:12
17. Talton, J.O., Lou, Y., Lesser, S., Duke, J., Měch, R., Koltun, V.: Metropolis procedural modeling. ACM Trans. Graph. **30** (2011) 11:1–11:14
18. Tan, P., Fang, T., Xiao, J., Zhao, P., Quan, L.: Single image tree modeling. ACM SIGGRAPH (2008) 1–7
19. Zeng, J., Zhang, Y., Zhan, S.: 3d tree models reconstruction from a single image. ISDA (2006) 445–450
20. Cornea, N.D., Silver, D., Min, P.: Curve-skeleton properties, applications, and algorithms. TVCG (2007) 530–548
21. Cornea, N.D., Silver, D., Yuan, X., Balasubramanian, R.: Computing hierarchical curve-skeletons of 3d objects. The Visual Computer (2005)
22. Latecki, L.J., Lakmper, R.: Shape similarity measure based on correspondence of visual parts. PAMI (2000) 1185–1190
23. Okabe, M., Owada, S., Igarashi, T.: Interactive design of botanical trees using freehand sketches and example-based editing. Comp. Graph. Forum (2005) 487–496
24. Boudon, F., Pradal, C., Cokelaer, T., Prusinkiewicz, P., Godin, C.: L-py: an l-system simulation framework for modeling plant development based on a dynamic language. Frontiers in Plant Science **3** (2012)