More of the Same: Synthesizing a Variety by Structural Layering

Jackie Assa and Daniel Cohen-Or ☆

The Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract

Highly detailed natural scenes and objects tend to be perceived as being realistic, while repeated parts and patterns decrease their realism. To avoid scenes with noticeable repeated elements, we introduce the notion of 'more of the same', which focuses on the task of generating additional similar instances from a small set of exemplars. The small number of exemplars, as well as their diversity and detailed structural texture, makes it difficult to apply statistical methods, or other machine learning tools, and thus more specialized tools need to be used. In this paper, we focus on generating a rich variation of highly detailed realistic leaves from just a handful set of examples.

The method that we present does use only minimal domain specific knowledge and requires only minimal user assistance applied on a single training leaf exemplar to extract and separate structural layers. The knowledge from one leaf is then transferred to the other exemplars by a novel color/spatial layer inducing algorithm. The premise of structural layering is that each set of layers is simple enough to be synthesized separately and then composed into a novel leaf structural texture. This composition also allows the synthesis of slightly modified layers from the set of examples, which can generate a large set of differently looking leaves. We demonstrate numerous results of realistically looking leaves produced by our method from a small set of leaves.

Keywords: Generating variations, Texture synthesis, Image generation

1. Introduction

One of the prominent tasks in computer graphics is to synthesize photorealistic scenes. One factor that directly affects the realism of a natural object or scene is richness of their details. Highly detailed objects tend to be perceived as being realistic, provided that there are no noticeable repetitions. For example, crowds, trees, and a brick wall, are objects that consist of a large number of similar parts. The realistic presentation of such objects increases as a function of the variety of their parts, while repeated parts and patterns decrease their realism.

In recent years considerable research effort has been directed at establishing techniques for generation of non-repetitive textures from a given example [1, 2, 3, 4, 5, 6, 7]. These techniques work well for stationary textures, but they do not handle structural variation coupled within the texture. For example, the leaf in Figure 1(a) contains an internal structure of its veins, which is tightly coupled to the contour of the leaf's shape. Furthermore, the leaf's texture is non-stationary as it is spatially coupled to the leaf's internal structure.

In this paper we introduce a technique that synthesizes a variety of structural textures from a given set of examples. In particular, we focus on the synthesis of leaves from a small set of examples. The method makes no special assumption about the nature of leaves and it is generic enough to be applied to other objects.

The challenging problem we address in this paper is in a larger problem context, which we name 'more of the same', focusing on generation of additional similar instances from a small set of exemplars. The small number of objects, as well as their diversity and delicate textures, makes it difficult to construct the detailed structural texture using statistical methods, or other machine learning tools, and thus more specialized tools need to be used. In our case, we focus on generating realistic texture and planar shape of leaves.

A recent work in this context is the study of Baxter and Anjyo [8] which is motivated by similar goals as ours. They introduce a Latent Doodle Space that al-

☆Corresponding author. Tel.:+972-3-6405368

Figure 1: *A rich variation of tree leaves synthesized from a small set of exemplars using structural layering.*

lows composing a variety of new simple line drawing examples consisting of a set of line scribbles. The focus of their work is the matching and mixing of the object-space lines. Extending their work to accommodate textures and structures in image-space is not trivial. Working in image-space, which is the focus of our work, allows dealing with real-world objects and synthesizing realistic looking entities. Most of the work on generative modeling has been applied in object-space. The model construction usually utilizes either domain specific knowledge or statistical knowledge generated by a sufficiently large and dense number of examples. A notable example of this type of work is introduced by Blanz and Vetter [9]. In their work, they derive a Morphable Face Model by transforming the shape and texture of over 200 examples into a vector space representation. New faces and expressions are modeled by forming linear combinations of the prototypes. Following studies have implemented similar solutions for different domains such as animal 3D skeletons and other objects [10, 11]. Contrary to this type of work, here, the detailed structural texture and its composition structure prohibits determining statistically valid texture model, especially without deep understanding of the domain knowledge. Instead, we learn from a small set of exemplars.

2. Related work

2.1. Texture synthesis

Texture synthesis is a common technique to generate textures which are similar to given exemplar [3, 1, 4]. Instead of modeling the texture, they fuse patches from the given example, allowing successful synthesis of rich and complex textures. The strength of these example based approaches is that they avoid the difficult task of analysis. These methods are mainly effective for the synthesis of stationary textures, and their generalization to spatially-varying textures is hard [12]. [13] includes some analysis and specifically aim at the generation of a variety of textures from a given example. They arrange the extracted texture patches into layers, and uses warping to generate more variability in the result. Rosenberger et al. [14] synthesize inhomogeneous textures by decomposing the texture of the exemplar into several layers, where the visible parts of each layer are occupied by a more homogeneous texture.

Analysis of general textures is conceptually similar to the general segmentation problem, and similarly to segmentation, reasonable results requires the assistance of the user. Our analysis of the leaf textures is similar to the work of [15] on image segmentation by example. They constructed a non-parametric representation of the segmented example by patch-based representatives, allowing the segmentation of a similar image into complex semantic regions containing a large variety of colors and textures. Recent studies in this category includes the synthesis of urban street maps from a small given set of map examples, which preserve the urban global structure [16]. Another relevant segmentation by example is introduced by Borenstein and Ullman [17]. There, they segment an object in an image, based on a large set of pre-segmented images, all from the same family of specific objects. In our work, we use only on minimal domain specific knowledge for the determination of the layers, and are looking for a fine and detailed decomposition of an image into layers, without having a large number of training examples.

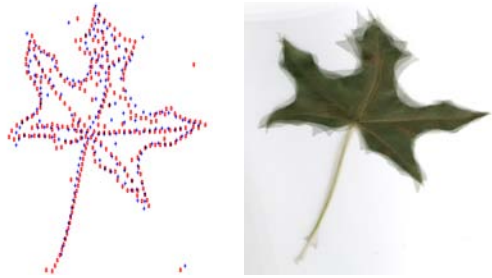
2.2. Leaves synthesis

During the last two decades a special research interest has been focused on generating realistically looking plants in 3D environments: A wide set of solutions where suggested for building tree models, which include the trunk, branches, and crown general shape, by tracking botanical models or other construction rules

[18, 19]. These techniques produce excellent results however generating realistic result for several tree types, requires labor intensive, manual effort in the preparation and configuration of the rule set. Moreover, it is hard to tune these methods to produce tree imperfections and to control the resulting tree specific shape. To overcome these obstacles, recent methods offer an easy example-based techniques for generating a realistically looking tree models, based on a small number of tree photographs [20, 21, 22, 23].

The generation of leaves patterns had been a specialized research field suggesting ways to model the leaves textures and shapes. Photorealistic results were demonstrated by techniques which offered a way to botanicaly model the evolution of the leaves venation [24]. However, those methods also suffer from the same problems as the tree modeling techniques, which are domain expertise and significant manual effort. Another venue for the generation of leaf textures, has been by using texture synthesis techniques, which aimed on generating non-repetitive textures from a given example [1, 3, 4, 5]. These techniques generate satisfactory results for stochastic textures but cannot deal with structural textures which contain global structures such as the ones in leaves. We introduce a method which address the synthesis of the structural textures and allows to produce many realistically looking leaves by using small number of examples and minimal user interaction. Our method is generic and almost does not rely on any botanic knowledge. In a sense, our method extends the work of Shen and his colleagues [13], where they synthesize general textures, by segmenting a sample texture to its main components. The segments generate a set of layers, one for each texture class, which are later combined. In our case, we use the similarity of the leaves to extract the structure information and to automatically infer the shape of these layers. The results of our method (as shown in Figure 1) assist in generating realistic complete tree models including their leaves details, based on real examples and with minimal user interactions.

Our approach is motivated by the observation that leaf structure consists of a small number of layers, each of which includes a texture with certain characteristics that can vary from being strongly structured, to being spatially varied, or even completely stochastic. These layers can be decomposed from a set of exemplars and mixed between instances to generate a richer set of variations. Following this notion, our method consists of two parts: decomposition and synthesis. The decomposition of a structural texture into simpler and controllable layers with a consistent characteristic is an extremely difficult task as it requires image understanding. A user guided approach with minimal effort on a single training exemplar leaf provides sufficient hints for decomposing these structures. Next, our novel automatic technique of inferring the layers of a given instance from the layers of an example generates a set of layers from all leaves. To synthesize novel instances, we generate convex combinations of the warping fields, warping the layers and composing them into a coherent novel instance. This composition method allows the generation of a diverse set of examples, which can then be used to enrich the realism of the generated scene.



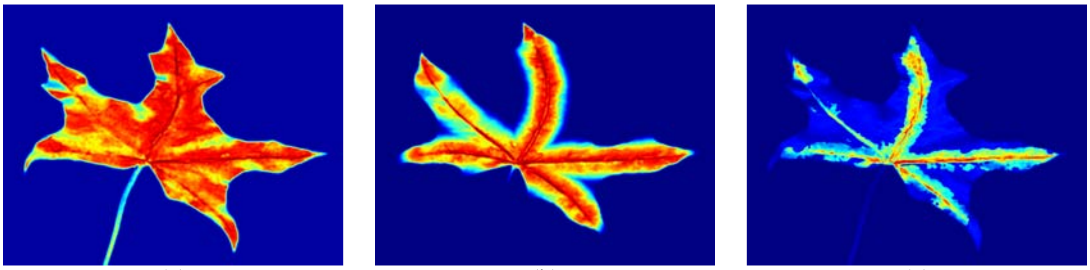
(a) (b)

Figure 2: *Automatic registration of the examples yields a global alignment, with several local mismatches. (a) shows the non-rigid registration of two leaves with their keypoints shown in red and blue. (b) shows 4 overlayed leaves after the alignment step.*

3. Analysis

Given a leaf image, the task of separating it into layers is closely related to the problem of image segmentation into parts. A typical image segmentation technique, aims at identifying homogeneous regions in an image that correspond to semantically meaningful regions in an image. Here, the layers are meant to differentiate the various structures and texture parts. Nevertheless, the fundamental difficulty remains extremely hard as, in both cases, a proper segmentation or layering requires image understanding and parts recognition.

Our method segments a set of example leaves into meaningful layers provided a single training leaf that is easily layered by the user. By learning the given training image, the rest of the examples are automatically analyzed and decomposed into meaningful layers by inferring the layers of the training image. This inference



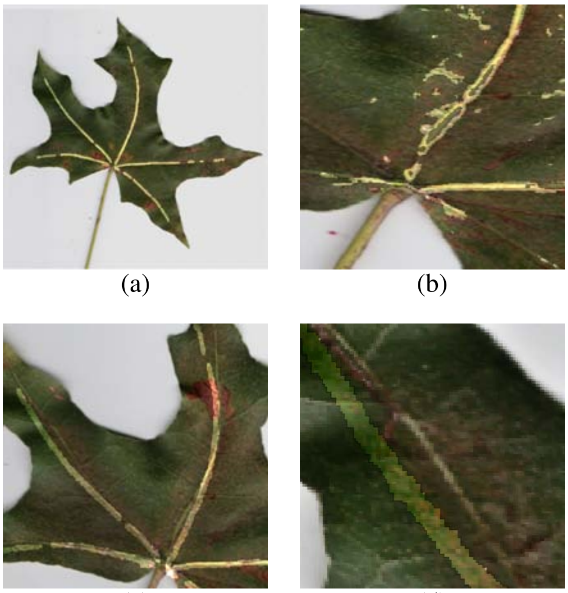
(a) (b) (c)

Figure 3: *Calculating layer similarity. High potential layer regions are marked in dark red. (a) Color histogram similarity heat map. (b) Structure distance similarity heat map. (c) Resulting similarity measurement for the veins.*

task is described in Section 4. The layers of the training leaf image are defined by the user with a set of scribbles. Next we apply the following steps on every leaf from the given exemplar set: We first align the exemplar leaf with the training leaf. The alignment is done by a non-rigid registration technique followed by a texture correction step. Following this, we perform a layer induction (Section 4) on one layer at a time, remove the layer pixels from the residual image, and generate the novel image by layers synthesis (Section 5).

To align the exemplar leaf to the training leaf, we apply the non-rigid unsupervised registration algorithm introduced by Chui and Rangarajan [25]. The registration is applied to a sparse subset of leaf keypoints. We use a sparse set of pixels from the shape strong edges as these keypoints, by applying a Sobel edge detector over a bilateral filtered image. In our setting, around 100 points were enough to convey the general shape in the leaf. An example of the resulting keypoint matching is shown in Figure 2 (a) . The resulting registration generates a global warp that aligns the example and training leaves and is also used later in the synthesis phase. The alignment is imperfect and may tolerate small (even up to 50 pixels) mismatches. Additional efforts to gain a more precise registration usually cause overfitting artifacts, and require labor-intensive and domain-specific efforts in defining the keypoints. This rough alignment is sufficient for our needs as it brings the examples close enough to apply a reasonable analysis. An example of several registered leaves are shown in Figure 2 (b).

To counteract the effects of the previous registration, we apply a texture synthesis which fixes the artifacts generated in the extremely warped regions. The technique is based on replacement of warped patches with similar patches taken from the original image, in the spirit of the technique of Fang and Hart [26].



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

Figure 4: *Naive transformations of the user layer induced to other leaves do not provide satisfactory results. (a) highlights the user vein layer selection. (b) highlights regions with similar color distribution. Note the level of false negatives, and missing regions in the induction. (c) highlights the user selection naively copied to another image demonstrating the registration mismatches, and (d) presents a zoom-in on the registration mismatch.*

4. Layer induction

Given a set of training leaf layers, and an aligned leaf example, the problem of inducing similar layers for the given example is difficult. Clearly, a similar layer cannot be based on color similarity alone, since the colors are not good descriptors to well define the layers. This fact is evident in Figure 4 (b). Note also that the structured layers in the two leaves after registration are spatially similar but definitely not the same, as shown in Figure 4 (c,d). We therefore introduce a technique for inducing the layers from the training leaf image to the example image by considering a fusion of *color* and *spatial* similarity.

We denote with c the set of pixels marked by the user as representing a given layer over a training leaf image t. A color histogram of a region v is denoted with $H(v)$. Hence, the color histogram of the set c is denoted by $H(c)$. Now, given an example leaf image e, one can define an analogous layer in e with respect to given layer in t by simply measuring a distance $S(x, c)$ between the color distribution of a small neighborhood around each pixel x in e (denoted by $N^e(x)$) and the colors distribution of c:

$$S^e(x, c) = \|H(c), H(N^e(x))\|_m \quad (1)$$

where $\|\cdot\|_m$ is the normalized earth movers distance (EMD) between histograms (0 indicates a match, and 1 indicates maximally different histograms).

Such a histogram distance is not sufficient to classify the pixels of the example leaf image and create a reasonable analogous layer, since the colors solely do not provide sufficiently discriminative measures. This is demonstrated in Figure 4 (b), where colors similar to the ones defined by the user to describe the veins of the leaf, only partially defines the veins of the example leaf image, and also highlight regions which are clearly not related to the veins in the example leaf image. Therefore, the similarity measure needs to account for the spatial distance from the structure defined by the user:

$$D(x, c) = \lambda \cdot F(x, c) + S^e(x, c) \quad (2)$$

where $F(x, c)$ measures the spatial distance between a pixel x in e and the user selection set c. As such, the $F(x, c)$ is motivated by a normalized distance transform of the set c in the leaf image ($\|x, c\|_{L2}/\max(\|x, c\|_{L2})$).

Linear combination of the spatial and color similarity, where $F(x, c)$ is based solely on distance transform does not generate satisfactory results, as the spatial registration distance and color distribution distances do not change across the image in the same manner. The non-uniform relation of the colors and the spatial distances do not allow the use of constant measures across the image. In particular, if a global constant λ is used, often the generated layer includes large amount of details which are not related to the structure, and generates artifacts in the resulting composition.

To better express the structure in e, we would like to encourage regions with similar color distributions to c only when they are within a certain spatial falloff distance ε (related to the maximal registration error) in the example leaf image. Moreover, we would like also to consider the separability of the training leaf layer, and widen the falloff distance in regions where the training leaf did not had good color separation for the layer. Therefore the final spatial distance term is defined as:

$$F(x, c) = e^{-\frac{\|x,c\|_{L2}^2}{\varepsilon^2 \cdot (1 - S^t(x,c))}} \quad (3)$$

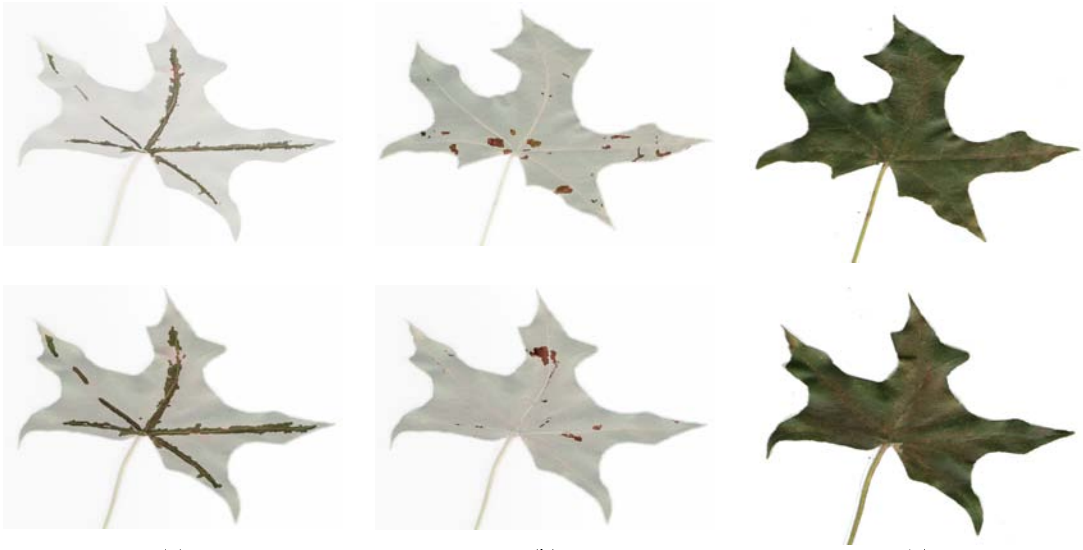
Note that the color distance term $S^t(x, c)$ indicates the color distance on the training leaf t. The effects of this term can be viewed as a heat map that expresses the likelihood of each pixel to be selected for the analogous layer. A heat map of this term is visualized in Figure 3(b), where red indicates high likelihood and blue a low likelihood for region similarity.

To define the pixels participating in each layer, we set a threshold value δ automatically by analyzing the training leaf image and the user selected region c. In particular, δ is defined by searching for a value that best reconstructs the layer defined by the user. Let $L^t_\delta = \{x | D(x, c) < \delta\}$ on the training leaf t. Now, we look for the L^t_δ which maximizes the overlap with the user-defined layer C:

$$max \left(\frac{\left| L^t_\delta \cap C \right|}{\left| \overline{L^t_\delta} \cap C \right| \cdot \left| L^t_\delta \cap \overline{C} \right|} \right). \quad (4)$$

This expression yields a distinct maximum that characterizes the method ability to induce the user selection behavior. In most of the cases the maximal value was in the range of .7 - .8 which indicate that the layer can be induced successfully automatically.

In cases where the layer structure is not spatially coherent between the examples and training leaf image, such as the case of the leaf spots layer, the resulting layer exhibits a very low number of participating pixels. For such layers, we would like to reduce the spatial similarity term effect and thus use mainly standard color histogram similarity to induce the layer. To accommodate



(a) (b) (c)

Figure 5: *Inducing analogous layers from a training exemplar. The upper row, shows the training leaf layers defined by the user, whereas the lower shows the resulting induced layer. (a) show the veins structural layer. (b) shows the spots nonstructured layer (c) shows the remaining background after removal of the layers)*

this, we set the value of λ by examining the resulting layers overlap. More specifically, we use the percentage of overlap between layers, divided by the overall layer size to define its value:

$$\lambda = \kappa \cdot \bigcap_e \tilde{L}^e \Big/ \bigcup_e \tilde{L}^e \tag{5}$$

where κ is a constant set to 1.5 in all of our examples, and $\tilde{L}^e$ are the induced layer for exemplar e, using a liberal definition of:

$$\tilde{L}^e = \left\{ x | S^e(x, c) < 0.1 \text{ and } \|x, c\|_{L2}^2 < 2\varepsilon \right\}. \tag{6}$$

An example of the layers overlap is shown in Figure 6. An example of layer induction without structure coherence is shown in Figure 5 (b).

After the extraction of each layer, we apply a standard inpainting by example method, to complete the leaf texture, in pixels which participate in the layer. The completion of the layer allows to use the remaining leaf texture and structure for the following layers. An example of the remaining leaf is shown in Figure 7

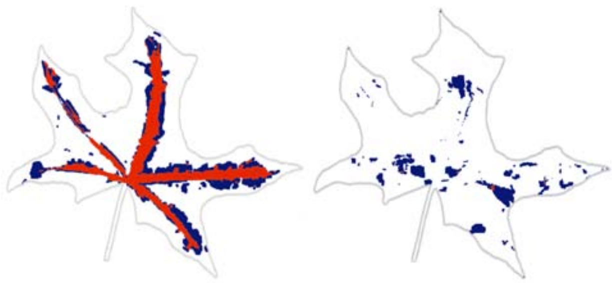


Figure 6: *Structural layers. The induced vein-layers (left) and spots-layer (right) overlap (blue indicates non overlapping areas, and the red marks regions which are not overlapping). The veins-layers overlap indicates it is a highly structural layer whereas the spots-layers non-overlap indicates it is non-structural. The overlap area is used to determine the value of λ.*

5. Synthesis

To generate a diverse set of leaves, we apply three techniques: (i) fusing layers from different examples, (ii) randomly warping each of the layers, and (iii) warping the complete composition back by selecting an inverse warp in the convex hull of the preliminary registration warps.

First, we select an inpainted background layer taken from one of the examples and apply to it a random warp as suggested in [4, 13]. The warp is defined to be sufficiently smooth and small, and should not exceed the

(a) (b) (c)

Figure 7: *Removing the layer participating pixels, generates a new image that can be analyzed in further layers. (a) shows the original leaf, (b) without its veins layer, and (c) after the removal of its spots layer.*

initial registration error. Next we fuse additional layers to the background layer using the similarity term D as a blending weight normalized per pixel across all the layers. Each layer is slightly warped in same manner as described above but with a different random warp. We use the layer spatial structure (shown in Figure 6) as an indicator, for the warp magnitude.

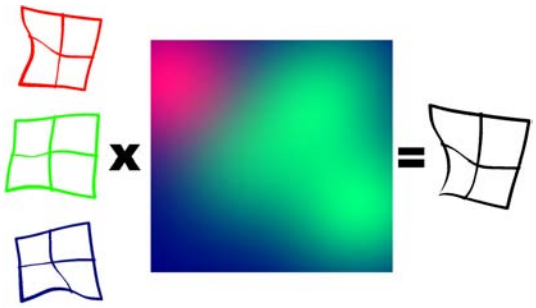


Figure 8: *A visualization of the warping process. left: three exemplar warp fields $W^{-1}(x)$ (here each in different color channels). Middle: The warp coefficients map, pixel colors represents the corresponding exemplar warp weight. Right: The resulting warp field.*

The composition of the layers yields a brand new structural texture aligned to the training leaf image. This result then needs to be warped back toward the shape of the example leaf images to produce a variety of shapes rather than of texture. Each of the exemplar leaf i is associated with the aligning field W_i that initially aligned it to the training leaf image. To generate a random shape, we use a linear combination of the inverse of these warps, denoted by W_i^{-1}. To blend these fields, we define a discrete map $a_i(x)$ of coefficients vectors such that $\sum_i a_i = 1$ to guarantee a convex combination. The blend of warps W^{-1} is defined by:

$$W^{-1}(x) = \sum_i a_i(x) \cdot W_i^{-1}(x). \quad (7)$$

The discrete map $a_i(x)$ of the coefficients is constructed by a small set of random coefficients, blurred with a low pass Gaussian. The coefficients are normalized for each pixel x. The generated map for three examples is illustrated in Figure 8, where each of the RGB channels shows the values of different coefficients in the map. As a final step to the synthesis, we apply the texture correction step, described in Section 3, which alleviates warping artifacts in the image.

6. Results and Limitations

We've implemented structural layering using Matlab and applied it on various leaf classes. The execution time of our method for a leaf class with 4 exemplar leaves (Each with 2M pixels images) takes several minutes on a 2GHz Intel Pentium M CPU with 1GB of memory where the time consuming are non-rigid registration and the layer inpainting. Defining the example leaf layers, is done manually by applying around 10-15 short scribbles per layer for each leaf class, using a standard lazy snapping based interface [27]. Our results are shown in Figure 9.

In all our experiments the user segmented only a single training leaf, to define the layers. In particular in the leaves set, we defined vein and 2-3 colored spots layers, some of which were found to be structural layers. The color histograms were calculated over a small neighborhood $N^e(x)$ of 8X8 pixels and ε used in all of the examples was 40 pixels.

Correct order for the layers, is important for the success of our method, as it removes and induces the layers one at a time. Therefore, the results of leaves in which the selected order is not correct, or when layers are intermixed, may reduce the realism of the resulting leaves. The large structure scale variance holds an additional challenge for our method. The problem of detecting and inducing the structure is hierarchical in nature - Currently, the user scribbles highlight only the larger veins, whereas selecting each and every vein may prove to be non feasible. Therefore we actually infer layers which include the structure up to a certain scale and may cause artifacts in the finer details of the result. A possible direction extending our method would be to first extend the user selections, in a hierarchical manner, and to consider this extension during the inference of the layers. Such a method may also require stronger assumptions on the automatic registration misalignments.

Our method does requires some domain knowledge for determining the number of layers and their types. Nevertheless, this knowledge is not the result of botanic rules, and is only guided by visual examination of the leaves, and determinating which structural textures should be used. The composition of multiple layers, and objects which are constructed from several component introduce new challenges that are currently not handled. While this method can be applied to each component, the alignment as well as the integration between components introduce new challenges in the composition of the results. Another limitation of our method, is the conservative warping we apply to the shape. A better understanding on the shape global structure, can allow better and more extreme warping to be applied while still preserving the exemplar shape theme.

7. Conclusions and future research

In this work we presented a technique that deals with the synthesis of new leaves given a small set of scanned exemplar leaves. This simple technique can be used to enhance the realistic look of modeled trees and plants, in 3D models and images. Our method, does not apply any plant-life domain specific assumptions and as such can be later extended to other cases which include textures which interlocks with natural structures. The key point of our technique is the decomposition of the structural texture into a series of simplified meaningful layers and their synthesis.

We believe that the decomposition of structural textures into layers has more potential beyond the one demonstrated here. However, such decomposition is a challenging task that requires more research. The approach that we took in our work, namely, using user guidance over a single image has proved to be effective. Further research is required to extend this principle to a broader spectrum of images, shapes, and applications, such as dynamic creation of photorealistic objects to enrich 3D virtual worlds, methods for blending various structures, not only textures, and in being able to better blend different classes of complex objects to generate a variety novel instances of specific classes. This technique makes a an additional step towards solving the general problem of creating a large and rich variety of similar instances from a given small set of examples.

References

[1] A. A. Efros, W. T. Freeman, Image quilting for texture synthesis and transfer, in: ACM Transactions on Graphics, (SIGGRAPH), ACM Press, New York, NY, USA, 2001, pp. 341–346.

[2] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, D. H. Salesin, Image analogies, in: ACM Transactions on Graphics, (SIGGRAPH), ACM Press, New York, NY, USA, 2001, pp. 327–340.

[3] L.-Y. Wei, M. Levoy, Fast texture synthesis using tree-structured vector quantization, in: ACM Transactions on Graphics, (SIGGRAPH), ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000, pp. 479–488.

[4] V. Kwatra, A. Schödl, I. Essa, G. Turk, A. Bobick, Graphcut textures: image and video synthesis using graph cuts, in: ACM Transactions on Graphics, (SIGGRAPH), ACM Press, New York, NY, USA, 2003, pp. 277–286.

[5] Y. Liu, W.-C. Lin, J. Hays, Near-regular texture analysis and manipulation, in: ACM Transactions on Graphics, (SIGGRAPH), ACM Press, New York, NY, USA, 2004, pp. 368–376.

[6] W. Matusik, M. Zwicker, F. Durand, Texture design using a simplicial complex of morphable textures, in: ACM Transactions on Graphics, (SIGGRAPH), ACM Press, New York, NY, USA, 2005, pp. 787–794.

[7] E. Risser, C. Han, R. Dahyot, E. Grinspun, Synthesizing structured image hybrids, ACM Trans. Graph. 29 (4) (2010) 85:1–85:6.

[8] W. Baxter, K.-i. Anjyo, Latent Doodle Space, Computer Graphics Forum 25 (3) (2006) 477–485.

[9] V. Blanz, T. Vetter, A morphable model for the synthesis of 3-d faces, in: ACM Transactions on Graphics, (SIGGRAPH), 1999, pp. 187–194.

[10] L. Reveret, L. Favreau, C. Depraz, M.-P. Cani, Morphable model of quadrupeds skeletons for animating 3d animals, in: SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, ACM Press, New York, NY, USA, 2005, pp. 135–142.

[11] M. J. Jones, T. Poggio, Multidimensional morphable models: A framework for representing and matching object classes, International Journal of Computer Vision 29 (2) (1998) 107–131.

[12] J. Zhang, K. Zhou, L. Velho, B. Guo, H.-Y. Shum, Synthesis of progressively-variant textures on arbitrary surfaces, in: ACM Transactions on Graphics, (SIGGRAPH), ACM Press, New York, NY, USA, 2003, pp. 295–302.

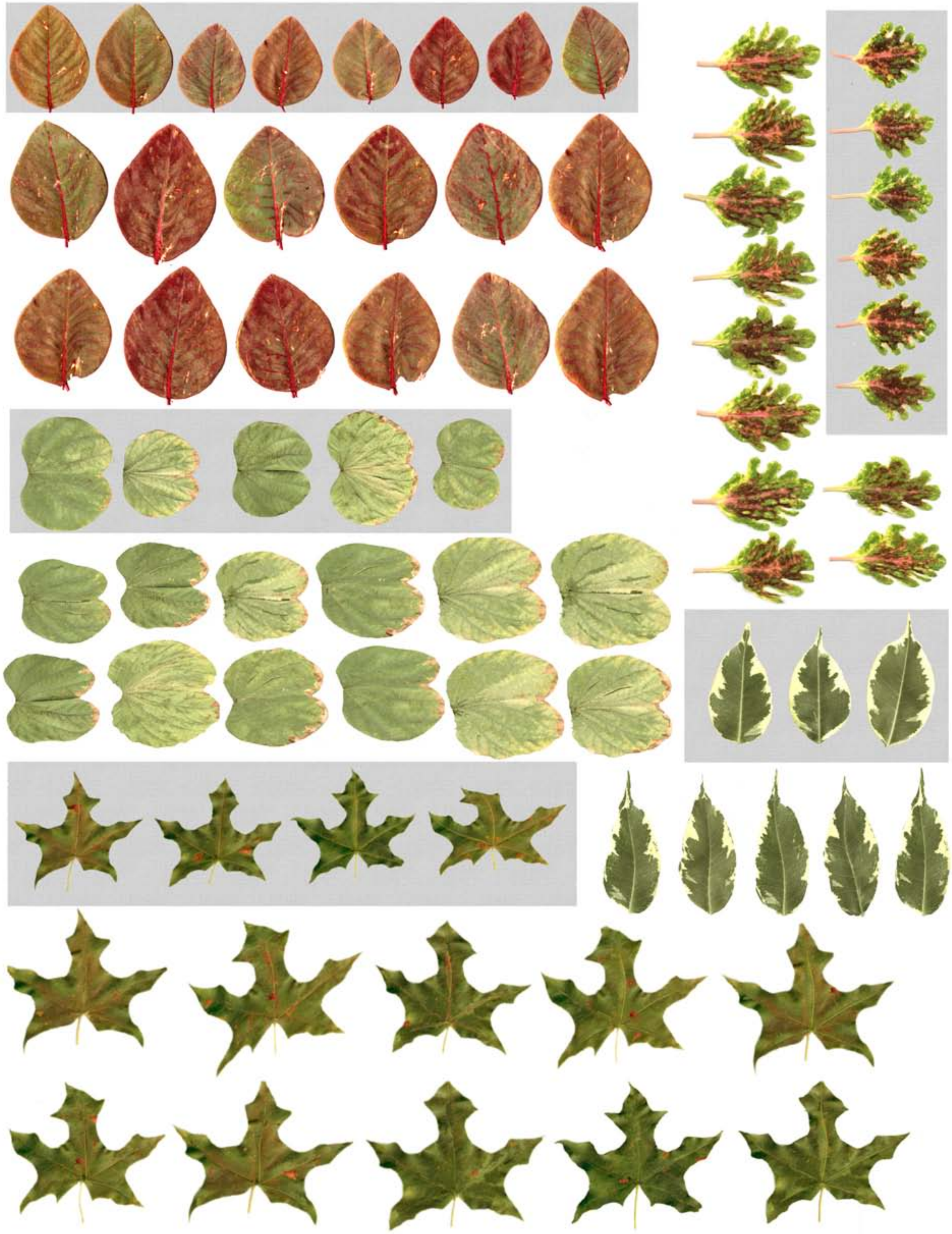


Figure 9: *Structural layering results. Structural layering results. Each group of leaves show the exemplar leaves at the top row and the synthesized leaves below it. All examples used vein layers and 2-3 color spots/regions layers.*

[13] J. Shen, X. Jin, X. Mao, J. Feng, Completion-based texture design using deformation, Vis. Comput. 22 (9) (2006) 936–945.

[14] A. Rosenberger, D. Cohen-Or, D. Lischinski, Layered shape synthesis: automatic generation of control maps for non-stationary textures, in: ACM SIGGRAPH Asia 2009 papers, SIGGRAPH Asia '09, ACM, New York, NY, USA, 2009, pp. 107:1–107:9.

[15] Y. Schnitman, Y. Caspi, D. Cohen-Or, D. Lischinski, Inducing semantic segmentation from an example, in: Asian Conference on Computer Vision, 2006, pp. 373–384.

[16] D. G. Aliaga, C. A. Vanegas, B. Benes, Interactive example-based urban layout synthesis., ACM Trans. Graph. (2008) –1–1.

[17] E. Borenstein, S. Ullman, Class-specific, top-down segmentation, in: European Conference on Computer Vision, 2002, pp. 109–124.

[18] J. Weber, J. Penn, Creation and rendering of realistic trees, ACM Transactions on Graphics, (SIGGRAPH) (1995) 119–128.

[19] P. de Reffye, C. Edelin, J. Françon, M. Jaeger, C. Puech, Plant models faithful to botanical structure and developmentr, ACM Transactions on Graphics, (SIGGRAPH) (1988) 151–158.

[20] A. Reche, I. Martin, G. Drettakis, Volumetric reconstruction and interactive rendering of trees from photographs, ACM Transactions on Graphics, (SIGGRAPH) (3).

[21] M. Okabe, S. Owada, T. Igarashi, Interactive design of botanical trees using freehand sketches and example-based editing, Comput. Graph. Forum 24 (3) (2005) 487–496.

[22] X. Chen, B. Neubert, Y.-Q. Xu, O. Deussen, S. B. Kang, Sketch-based tree modeling using markov random field, ACM Transactions on Graphics, (SIGGRAPH) 27 (5) (2008) 1–9.

[23] P. Tan, T. Fang, J. Xiao, P. Zhao, L. Quan, Single image tree modeling, ACM Transactions on Graphics, (SIGGRAPH) 27 (5) (2008) 108.

[24] A. Runions, M. Fuhrer, B. Lane, P. Federl, A.-G. Rolland-Lagan, P. Prusinkiewicz, Modeling and visualization of leaf venation patterns, ACM Transactions on Graphics, (SIGGRAPH) (3) (2005) 702.

[25] H. Chui, A. Rangarajan, A new point matching algorithm for non-rigid registration, Computer Vision and Image Understanding 89 (2-3) (2003) 114–141.

[26] H. Fang, J. C. Hart, Detail preserving shape deformation in image editing, ACM Trans. Graph. 26 (3) (2007) 12.

[27] Y. Li, J. Sun, C.-K. Tang, H.-Y. Shum, Lazy snapping, ACM Trans. Graph. 23 (2004) 303–308.