Image-based Reconstruction and Synthesis of Dense Foliage

Derek Bradley[1] Derek Nowrouzezahrai[1,2] Paul Beardsley[1]
[1]Disney Research Zurich [2]Université de Montréal

Figure 1: *Multi-view stereo combined with a data-driven leaf synthesis approach can produce realistic reconstructions of dense foliage.*

Abstract

Flora is an element in many computer-generated scenes. But trees, bushes and plants have complex geometry and appearance, and are difficult to model manually. One way to address this is to capture models directly from the real world. Existing techniques have focused on extracting macro structure such as the branching structure of trees, or the structure of broad-leaved plants with a relatively small number of surfaces. This paper presents a finer scale technique to demonstrate for the first time the processing of densely leaved foliage - computation of 3D structure, plus extraction of statistics for leaf shape and the configuration of neighboring leaves. Our method starts with a mesh of a single exemplar leaf of the target foliage. Using a small number of images, point cloud data is obtained from multi-view stereo, and the exemplar leaf mesh is fitted non-rigidly to the point cloud over several iterations. In addition, our method learns a statistical model of leaf shape and appearance during the reconstruction phase, and a model of the transformations between neighboring leaves. This information is useful in two ways - to augment and increase leaf density in reconstructions of captured foliage, and to synthesize new foliage that conforms to a user-specified layout and density. The result of our technique is a dense set of captured leaves with realistic appearance, and a method for leaf synthesis. Our approach excels at reconstructing plants and bushes that are primarily defined by dense leaves and is demonstrated with multiple examples.

CR Categories: I.3.3 [COMPUTER GRAPHICS]: Picture/Image Generation—Digitizing and scanning;

Keywords: Flora reconstruction, image-based plant modeling, leaf synthesis.

Links: DL PDF WEB VIDEO

ACM Reference Format
Bradley, D., Nowrouzezahrai, D., Beardsley, P. 2013. Image-Based Reconstruction and Synthesis of Dense Foliage. ACM Trans. Graph. 32, 4, Article 74 (July 2013), 10 pages. DOI = 10.1145/2461912.2461952 http://doi.acm.org/10.1145/2461912.2461952.

DOI: http://doi.acm.org/10.1145/2461912.2461952

1 Introduction

This paper describes the 3D capture, modeling and synthesis of dense foliage. Digital modeling of the real world is everyday technology. For example, Google Street View is a standard way to browse cities online, Google Earth extends this to 3D, while a late-breaking concept like the IllumiRoom[1] uses a model of an indoor environment to achieve custom projection. But much of the existing work has focused on structured man-made environments, and the modeling of flora has had relatively little attention. Some reasons for this have been the lower commercial potential of modeling flora in the first waves of research on modeling the real world, and the acceptability of utilizing a small set of generic plant models for some applications e.g. for architectural design. There is additionally the technical hurdle - the complexity, diversity and non-rigidity of flora mean that there are fundamental research problems to address, and techniques that work for structured environments do not always readily extend.

Capture of flora enables multiple new applications. Firstly, games and movies continue to demand more sophisticated digital environments. In the games domain, *Pure*[2] has extensive, realistic natural environments. Content such as this is created manually which is time-consuming and costly. In the movie industry, *Avatar*[3] illustrated breathtaking flora environments but again required intensive and expensive manual work. Accurate flora models will enable new educational tools such as immersive reality [Wilson 2009] and provide an archiving method for natural environments. In the scientific community, tools to do quantitative analysis of flora imagery, such as estimation of carbon sequestration [Ahrends et al. 2009], have started to appear but there is great potential for future research plus crossover to the fields of vision and graphics. The concept of "Citizen's Observatories" [COBWEB 2013] proposes that citizens use regular hand-held devices to capture their natural environment, to build a massive online database which is both a public resource and a valuable source of data for researchers. Such databases of the biosphere require new ways to analyze, visualize and mine the data.

One of the ways in which the capture of flora differs from structured environments is that it poses a multi-scale problem. For example, a tree has interesting structure at the upper scale of its branches and at the lower scale of its leaves. This suggests that a collection of different algorithms is needed for scanning a flora environment. In keeping with this philosophy, this paper focuses on a specific goal

[1]http://research.microsoft.com/en-us/projects/illumiroom
[2]http://www.purevideogame.com
[3]http://www.avatarmovie.com

- modeling of dense foliage like that shown in Figure 1. The expectation is that it could be one component of a larger system, and can complement other work. Tree modeling has produced impressive results in recent years, but existing approaches have focused on capturing the high-level branching structure of a tree so that the 3D result looks similar to the real tree from a distance - the fine-scale twigs and leaves are often synthesized arbitrarily rather than being captured. Our work is intended to provide directly captured foliage, with the option of synthesizing foliage based on measured statistics of leaf shape and foliage configuration.

Our focus is on fully automatic capture of dense foliage. Other work has used image-based tools for semi-manual creation of leafy plant models. But a method requiring manual interaction is not applicable to dense foliage with at least hundreds of leaves. Our proposed solution is to take an exemplar leaf with known geometry and a small number of images of the flora to be reconstructed, and then run an iterative 3D recovery process to generate a dense reconstruction. The idea is to capture the dominant uppermost leaves in a first iteration, build a statistical model of leaf shape and appearance from these leaves, and then add more challenging leaves from the background on subsequent iterations with the statistical model as a guide. Since it is very challenging to recover extremely dense foliage due to occlusions, we augment our reconstruction algorithm with a novel leaf synthesis approach that takes advantage of the statistical model to generate new leaves which are indistinguishable from the directly captured flora. We also show that our synthesis method can be used to synthesize new plants from the captured species within a user-defined volume, providing a valuable approach to scene generation in video games and films. To our knowledge, this is the first method to focus on automatic capture of fine scale leaf details of flora. The key contributions can be summarized as follows:

- An iterative method for reconstructing dense leaves from a handful of photos.
- An approach to build statistical models of captured leaf shape and appearance, and leaf neighbor relationships.
- A new leaf synthesis technique that can be used to increase the density of a computed leaf reconstruction, or to incorporate flora into scenes under artistic direction.

2 Related Work

Producing high-quality flora models is a challenging task. Previous approaches have focused either on providing tools to alleviate the difficult modeling process, or on capture techniques to reconstruct flora from the the real world.

Flora Modeling. A number of researchers have described interactive tools for flora modeling. Primarily, sketch-based and other interactive systems are popular due to their ease of use [Mundermann et al. 2003; Okabe et al. 2005; Anastacio et al. 2006; Chen et al. 2008; Tan et al. 2008; Pirk et al. 2012b]. These methods are typically used for generating single leaf structures or the overall shape of trees. Wither et al. [2009] design a sketch tool based on silhouettes that allows users to create several trees at various levels of detail. Another approach to simplify flora modeling is biologically-motivated L-systems [Lindenmayer 1968; Prusinkiewicz and Lindenmayer 1990; Prusinkiewicz et al. 1994] or procedural models [Greene 1989; Deussen and Lintermann 2005; Talton et al. 2011; Pirk et al. 2012a], which automatically generate botanical structures of plants through a set of rules. This approach can be extended to include interactive control [Lintermann and Deussen 1999; Palubicki et al. 2009], however with procedural approaches it can be very complex to design adequate rules in order to generate realistic plants, where image-based reconstruction techniques can capture specific flora species with a desired appearance. Procedural approaches are actually often used to complement reconstruction algorithms, to fill in missing details. In our paper we also use a form of synthesis to fill in missing details, however our technique is to learn the appropriate flora structure and color models from the captured data.

Flora Capture. Recently, a number of techniques have emerged for reconstructing the shape of trees and plants from images or laser scans of real foliage. Reche-Martinez et al. [2004] use a volumetric approach from a small number of images. Neubert et al. [2007] model trees from particle flows in a volume generated from images. Xu et al. [2007] and Livny et al. [2010] create tree models from laser-scanned point clouds, and Livny et al. [2011] designed a lobe-based representation to efficiently store the tree models. These methods all focus on capturing the large-scale branching structure or distant appearance of trees, while arbitrarily synthesizing the fine-scale leaf details. In contrast, our method aims to reconstruct plant-specific leaf geometry and appearance, allowing us to render these important details as they appear on the real tree. Our technique could complement the above methods, as they could be combined to form complete tree reconstructions. In the area of tree physiology, Sonohat et al. [2006] present a system for scanning partial flora and then synthesizing additional leaves using botanical rules. While their goal is to speed up the task of exhaustive leaf digitization for tree architecture analysis, our approach is similar in spirit although we require only images as input and learn a statistical model of the captured leaves for guiding a new synthesis algorithm.

Some methods attempt to model large-scale tree motion from video [Diener et al. 2006; Li et al. 2011]. Capturing foliage motion at the fine scale is outside the scope of our paper, however we consider this an interesting direction for future work.

Finally, image-based modeling approaches combine reconstruction with user interaction to produce realistic plants [Quan et al. 2006; Ma et al. 2008] and trees [Tan et al. 2007]. The method of Quan et al. [2006] is able to produce accurate leaf geometry, but does not easily scale to large, dense plants and bushes which would require a substantial amount of user-interaction. Our automatic approach of image-based reconstruction, learning a statistical model of the flora and then synthesizing additional leaves is better suited for large-scale data-driven modeling.

Leaf Appearance. In addition to reconstructing geometry, our method captures a statistical color model of the leaves. Other work has looked deeper into leaf appearance modeling. Baranoski and Rokne [2001] analyzed the reflectance of leaves and observed that they reflect light similarly to an ideal diffuse reflector, and Habel et al. [2007] leverage this fact when computing a compact, approximate shading model for real-time leaf relighting. While complex leaf appearance modeling can improve the visual quality of our results, the topic is beyond the scope of this paper.

Fitting Template Geometry to Points. Our reconstruction algorithm includes a technique to fit leaf geometry to 3D points in a RANSAC sampling framework. Schnabel et al. [2007] also use RANSAC to locate primitive shapes in range scan data. Papazov and Burschka [2011] extend this idea to find the pose of more complex but specific shapes in point clouds that contain little noise. In contrast, we fit arbitrary leaf shapes to noisy point clouds with a non-rigid alignment technique and demonstrate how a statistical model can be built and used as a fitting guide.

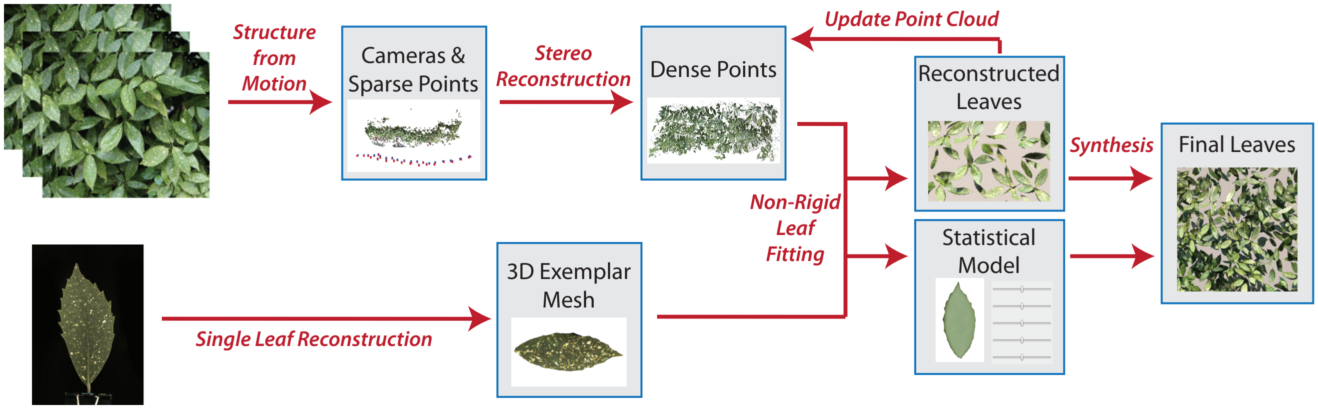

Figure 2: *Overview of our approach.*

3 Overview

The input to our method is a small number of overlapping images of the leaves, and at least one exemplar leaf that we reconstruct from a pruned example (Section 4). A standard structure-from-motion approach [Snavely et al. 2006] provides the camera parameters, and we develop a custom multi-view stereo reconstruction algorithm to generate a dense point cloud from the foliage (Section 5). The geometry of the flora is obtained from a novel non-rigid leaf fitting approach, which iteratively aligns the exemplar to the point cloud data in a RANSAC framework [Fischler and Bolles 1981] (Section 6). While fitting the leaves to the point cloud, we build up a statistical model of leaf shape and appearance based on the 3D morphable model of Blanz and Vetter [1999]. The model is trained on unoccluded leaves which are expected to dominate the fitting in the first iteration, and we use the model to guide fitting in subsequent iterations. Finally, the statistical model is combined with a novel synthesis algorithm to position new leaves in the scene with the realistic structure and appearance of the plant. We demonstrate two applications of leaf synthesis, one to increase the density of real captured flora, and the other to generate new foliage in a user-defined area with the same characteristics of the captured flora (Section 7). Figure 2 illustrates our system.

4 Acquisition

There are two phases to the acquisition process. We first record images of the flora to be reconstructed, and then we capture at least one leaf exemplar.

Scene Acquisition. The flora is photographed from a small number of viewpoints, which contain partially overlapping views of the leaves. In all of our experiments we capture between 8 and 20 views with a handheld camera (Canon 500D), while simply walking by the plant at a distance of approximately 1 meter. Since we require only hand-held, uncalibrated camera images, our approach can be used for flora modeling in both urban and remote natural environments. Approaches that rely on laser scanners, in contrast, require more expensive hardware that is less portable.

Constructing an Exemplar Leaf. In order to reconstruct leaves that are faithful to the geometry of the plant, we require an exemplar leaf of the specific species. To construct an exemplar, a leaf is pruned from the plant and scanned to get the 3D geometry. For this step, we use a multi-view stereo approach [Bradley et al. 2008] initialized by the calibration method described by Beeler et al. [2010], although any other suitable scanning method could be applied. Some of the exemplars we prepared for our datasets are shown in Figure 3. Once an exemplar is constructed for a species, it can be used for all instances of that plant. Note that we require only a single exemplar leaf, however multiple exemplars could also be scanned if the foliage exhibits large variations in leaf shape.

Figure 3: *Leaf exemplars for our datasets.*

5 Initial Reconstruction

Reconstruction starts by estimating the internal parameters and relative locations of the camera images. We use a freely-available structure-from-motion software package[4], developed for the Photo-Tourism work by Snavely et al. [2006]. The structure-from-motion algorithm also generates a sparse set of 3D points for visible features in the images, which can be used for computing initial depth estimates for the cameras. We call these points P_S.

The goal of the initial reconstruction step is to generate a dense point cloud representation of the leaves, which we will call P_0. Our reconstruction approach differs from many standard multi-view reconstruction algorithms (see Seitz et al. [2006] for a survey), which try to find a single 3D surface that best describes the images. In our case, a dense set of leaves is actually many small surfaces of varying shape and size, often containing many self-occlusions and depth discontinuities. For this reason, we require an approach that decouples the estimation of the 3D structure from the surface reconstruction. Depth map approaches [Goesele et al. 2006; Strecha et al. 2006; Bradley et al. 2008], which compute per-pixel depth estimates for each camera, are well-suited for our problem. The depth maps can then be easily combined into a single dense point cloud.

We develop a simple correlation-based depth map reconstruction algorithm, but note that several other multi-view stereo methods could be used instead. For each camera C_P, we identify a neighboring reference camera C_R with significant view overlap. The depth map for C_P is computed by searching for each pixel ρ_P of C_P in the image of C_R, and then triangulating to get depth values. We use a sliding window-match technique, by shifting a local 15×15 pixel window centered on ρ_P along the corresponding epipolar line

[4]http://phototour.cs.washington.edu/bundler

in C_R, and choosing the sub-pixel match with the highest normalized cross-correlation score. The search is constrained to points that lie within the bounding box of the sparse points P_S. Depth values are computed independently for each pixel in a greedy fashion. As we have mentioned, the leaves form a chaotic array of tiny surfaces, and this tends to hinder global solutions such as optimizing over the entire unknown depth map [Szeliski 1999]. While our local reconstruction algorithm can lead to significantly more outliers in the combined point cloud, our leaf-fitting approach is designed to robustly handle a large number of outlying points.

6 Leaf Fitting

Our leaf fitting approach identifies multiple occurrences of fits between the exemplar leaf mesh and the dense point cloud computed in Section 5. The technique is iterative, continually aligning new leaves and then updating the point cloud by removing those points assigned to reconstructed leaves. We start by describing how a single leaf is aligned, and then discuss our framework for global leaf fitting.

6.1 Single leaf alignment

We wish to fit our leaf exemplar to points in the point cloud that correspond to a real leaf. We initialize the exemplar with a random initial pose somewhere in the volume spanned by the points.

Rigid alignment. The iterative closest point (ICP) algorithm [Besl and McKay 1992] is an efficient approach for rigid alignment of geometry, and we extend this technique to perform non-rigid alignment of the leaf to a subset of points in the vicinity of the initial pose. Depending on this initialization, the nearby points may by inliers or outliers, however at this stage the alignment proceeds optimistically. We perform n ICP steps, while computing distances for p percent of the mesh vertices each step (refer to Table 1 for a list of all parameter values used in the algorithm). Distances are computed efficiently using a kd-tree. After rigid alignment, we compute a score for this transformation t as

$$S_t = \sum_{i=1}^{N_v} \|v_i - c(v_i)\|^2, \tag{1}$$

where v_i are the vertices of the leaf mesh and $c(v_i)$ are the closest points in the point cloud.

It is well-known that ICP requires a good starting location, so if the initial pose was close to a real leaf then S_t will be lower than if the leaf ended up aligning to outlier points. We empirically determined a threshold τ, which indicates if the exemplar has been rigidly fit to the pose of a real leaf. When $S_t < \tau$, we proceed with non-rigid alignment, in order to obtain a better fit of the leaf to the points.

Non-rigid alignment. The goal is to deform the leaf in a way that it best fits the points, but also remains faithful to the plant species in terms of leaf shape. At the beginning, we have only one template leaf and so we wish to stay as close as possible to this leaf shape while globally deforming the mesh. As-rigid-as-possible mesh deformation is a popular field of study, and we employ the common Laplacian surface editing method of Sorkine et al. [2004], in an iterative closest point framework. Starting from the transformed exemplar mesh, we iteratively compute new vertex positions by minimizing the following global energy functional over the set of vertices V:

$$E_L(V) = \sum_{i=1}^{N_v} (\|v_i - c(v_i)\|^2 + \lambda \|L(v_i) - \delta_i\|^2), \tag{2}$$

where $L(v_i)$ is the Laplacian operator, δ_i are the Laplacian coordinates of the un-deformed exemplar mesh, and λ defines the rigidity of the leaf. In each iteration, $c(v_i)$ re-computes the closest points to the mesh vertices. As with rigid ICP, we take n steps of our non-rigid alignment procedure. In practice, n is chosen large enough that the deformation converges (see Table 1). The process of aligning a single leaf to a subset of a point cloud is shown in Figure 4, first illustrating the result of rigid alignment and then the final deformed non-rigid fit.

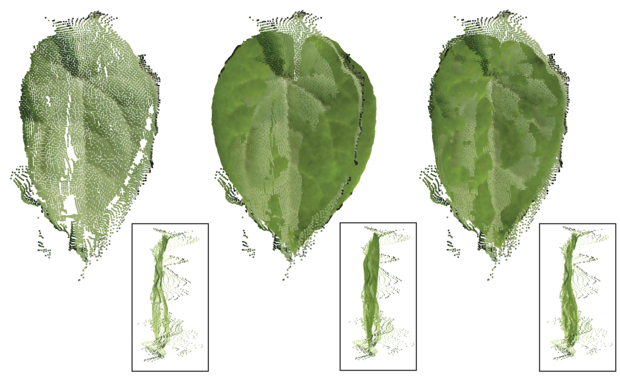

Figure 4: *Aligning the leaf exemplar mesh to points. The intermediate result after rigid alignment is shown in the center, and the result after non-rigid alignment is on the right. The inset shows a side view.*

When deforming the leaf to match the shape of the points, it is important to take special care of the leaf boundary. Even when the overall leaf shape matches perfectly, if the boundary does not align to a real leaf then artifacts may appear at the fringe, which we illustrate on the left of Figure 5. To address this problem, we propose an extension to the non-rigid alignment method which loosely incorporates knowledge of leaf boundaries. In many cases, edges in the input images correspond well to leaf boundaries, in particular when the edges are combined into 3D and a consensus is formed from multiple views. We take advantage of this by automatically marking point cloud samples that were triangulated from edge pixels, and then selecting only from this set when choosing the closest points $c(v_i)$ for any vertex v_i that lies on the boundary of a leaf. Since the rigidly deformed leaf is already close to its final position, it is safe to assume that nearby points marked as edges will correspond to a boundary. As a result, boundary vertices are pulled into a better position during the optimization, which we demonstrate on the right side of Figure 5.

Figure 5: *Explicitly handling leaf boundaries during non-rigid alignment can improve leaf reconstruction.*

Statistical model. The Laplacian deformation method above is useful when only a single leaf exemplar is available. However, if we assume for now that we have many example leaves of the species, then we can build a statistical model of the shape variance and perform non-rigid alignment in a *shape-space* spanned by the model parameters. We use the 3D morphable model of Blanz and Vetter [1999] since it is easy to compute and the model parameters translate directly to the deformation modes of the leaf, which allows us to optimize directly for model parameters during non-rigid alignment. The model is defined as follows:

$$S_{model} = \bar{S} + \sum_{j=1}^{m-1} \alpha_j s_j, \tag{3}$$

where $\bar{S}$ is the average of the m input shapes and s_j are the eigenvectors of the covariance matrix, defining the shape space of the leaf. The model is built using Principal Components Analysis (PCA) of the set of exemplar leaves. Given such a model, non-rigid alignment proceeds as before, except that we minimize the following energy over the set of model parameters $a_j \in A$:

$$E_S(A) = \|\bar{S} + \sum_{j=1}^{m-1} \alpha_j s_j - C\|^2, \tag{4}$$

where C is the set of all closest points $c(v_i)$. The benefit of this approach is that the optimization affects the entire leaf, rather than on a per-vertex basis as in Equation 2 which can lead to highly irregular leaves in poorly sampled regions of the point cloud. Performing non-rigid alignment using the statistical model helps us fit leaves to noisy and incomplete point clouds while remaining within the space of example leaf shapes. This approach assumes we have several exemplar leaves, which we will construct in a global framework.

6.2 Global RANSAC framework

The quality of single leaf fitting depends highly on the initial pose of the exemplar, since it is based on local alignment of points. For this reason, we take a Random Sample and Consensus (RANSAC) [Fischler and Bolles 1981] approach to robustly locate true leaves in the noisy point cloud. RANSAC can be described as the process of randomly choosing a solution, and then computing how well the data supports the solution. If the process is repeated many times, a consensus can be formed. We use RANSAC to obtain an initialization for the leaf fitting method described above, which allows us to robustly reconstruct leaves even in the presence of many outlying points.

Our RANSAC method begins with a large number of copies (N_L) of the exemplar and starts the single leaf fitting procedure in parallel. After the rigid alignment step, we sort the N_L leaves by their fitting score (Equation 1) and perform the non-rigid alignment sequentially starting with the most confident rigid fit. After N_M fits using Laplacian deformation alignment (Equation 2) we build a statistical model of the fit leaves (Equation 3) and perform subsequent alignments using the model (Equation 4). Each time a leaf is aligned to the points we extract its surface texture from the original cameras through projection, and then add it to the final list of leaves. All nearby points (within a distance of ϵ from the leaf) are then removed from further processing, and any other leaf that had also fit to those points is ignored. Note that we also build a statistical model of leaf appearance in addition to shape, and both models will be used in our leaf synthesis algorithm described in Section 7.

Updating the point cloud. From the N_L candidate leaves, typically only a small percentage are stored as true leaf fits, and many are discarded. However, from these leaves we gain valuable knowledge of the partial structure of the foliage. We use this knowledge to update the point cloud in an additional pass of multi-view stereo. In the first execution of stereo reconstruction, we used the sparse points resulting from the structure-from-motion algorithm to provide very approximate depth estimates to constrain the stereo search (refer to Section 5). We now formulate more reliable depth estimates by projecting the current set of reconstructed leaves onto the camera images and measuring the per-pixel depth. In-between leaves, depth values are interpolated, providing a complete target depth map for each camera. Stereo reconstruction is then performed exactly as before, building a dense point cloud, except that this time we omit pixels that are associated with an existing leaf fit, and we search for points only at the target depths (plus/minus an offset α). This approach improves the point cloud data for leaves that we could not fit confidently in the previous iteration.

The leaf fitting and point updating method iterates until no more leaves are located. The final result is a set of captured leaves with per-leaf textures that closely represent the true geometry and appearance of the real plant. Figure 6 shows an example point cloud and set of leaves after one iteration of the algorithm, along with the final set of leaves after eight iterations.

Parameter	Value
n - ICP steps	100
p - percent of vertices	5%
τ - rigid score threshold	0.3 - 0.5
λ - leaf rigidity	100
N_L - leaves per iteration	10,000
N_M - statistical model training size	100
ϵ - point removal distance	5 mm
α - target depth offset	10 cm

Table 1: *Parameter values used in our fitting algorithm.*

Parameters. A number of parameters control the behavior of our leaf fitting algorithm (Table 1). All parameters are empirically set once and used for all datasets, with the exception of the rigid score threshold τ, which varies depending on the size of the leaf and the number of exemplar mesh vertices.

7 Synthesis

Due to heavy occlusion, it may not be possible to purely reconstruct every leaf of a plant or bush, especially if it contains many depth layers (for example, the bush in Figure 1). In this case a user may wish to extend leaves into the unseen area. Similarly, an artist may wish to extend foliage into new areas when building virtual scenes for movies and video games. Scene creation can be a very time consuming task when flora is required. To address these issues, we propose a novel leaf synthesis method that can automatically generate new foliage that resembles the captured plant. Our algorithm attempts to preserve the structure of captured flora at two different scales: the individual leaf shape and color, and the global placement and orientation of all leaves.

Our synthesis approach is to iteratively create new leaves until a desired resolution is reached. The process has two steps. First a leaf is generated, and then it is placed in the scene. Both steps aim to maintain the global structure and appearance of the flora that was captured.

Figure 6: *Iterative leaf fitting. Point cloud after 1 iteration (left), fit leaves after 1 iteration (middle), fit leaves after 8 iterations (right).*

7.1 Leaf Generation

In order to generate individual leaves that are faithful to the captured species we employ the statistical model that we built during the fitting phase (Section 6.1). Recall from Equation 3 that we have a model of leaf shape parameterized over the coefficients a_j. New leaf shapes can easily be generated by varying the coefficients. In each iteration of synthesis, we create a new leaf by randomly selecting values of a_j within the range of the training set that we used to construct the model. Similarly, we can use a statistical color model of the captured data and randomly choose a texture for the leaf. We illustrate the statistical model for one of our datasets in Figure 7, which shows the mean leaf shape ($\bar{S}$ in Equation 3) with mean color, and several random permutations of the model coefficients to generate new synthetic leaves.

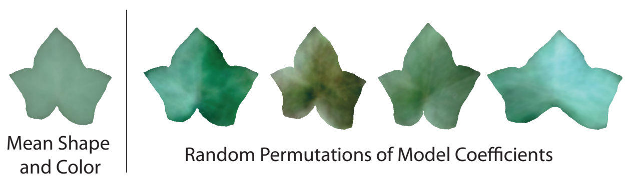

Figure 7: *Statistical model of leaf shape and color for the plant in Figure 10. Varying the model parameters randomly generates new synthetic leaves.*

7.2 Leaf Placement

Placing new synthetic leaves is a difficult task. On the one hand, the arrangement should reflect the random structure of nature, while on the other hand purely random placement is usually not faithful to the growing structure of the captured flora (see Figure 13). We propose a data-driven method for leaf placement, using the captured leaves to guide the position and orientation of new leaves. Our algorithm is based on the following theory: if you choose any two leaves in the captured leaf set, that pair exhibits a valid spatial relationship for two leaves of the given species. In other words, the transformation between those two leaves represents natural growth, and we can therefore use this transformation to place new leaves. Consider the example shown in Figure 8. On the top we have several leaves from the captured set, and we examine the pair L_1 and L_2 shown in red. If we compute the transformation T that registers L_1 and L_2 (in a least-squares sense), we could argue that this transformation could exist again between a different pair of leaves. Therefore, it is safe to choose a third leaf, L_3 shown in magenta, and place a new leaf at $L_3 + T$ shown in blue.

Based on this theory, our algorithm for synthesis starts by computing transformations between all pairs of captured leaves, and then ordering the transformations by the Euclidean distance of the leaf centers (from closest to farthest), since we find that the transformation between neighboring leaves is a better indicator of plant structure than that of distant leaf pairs. To place a new leaf that we

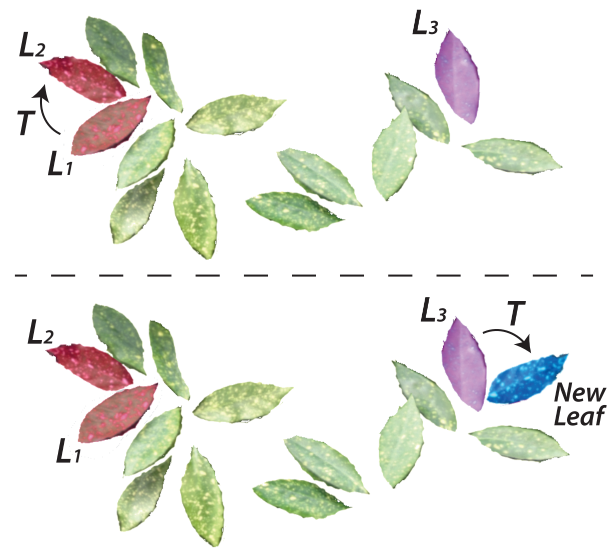

Figure 8: *Placement of synthesized leaves uses captured data in order to remain faithful to the foliage structure.*

generated in the previous section, we randomly choose a transformation T_i from the set of all N ordered transformations. The index i is chosen to favor leaves that are close together (i.e. favor a low index) by sampling the positive side of a Gaussian distribution centered at index zero with variance $\frac{N}{d_1}$. The parameter d_1 allows the user to control how strictly we favor a low index, in other words how closely the synthetic leaves will match the structure of the captured flora.

Once a transformation T_i is chosen, the algorithm proceeds to randomly select a current leaf in the output volume and place the new leaf with a pose transformed by T_i. If the new leaf location falls outside the user-defined volume, or is with a distance of d_2 to an existing leaf (from center to center), then it is discarded and the process repeats. This second parameter, d_2, controls the final density of the synthesized flora. In practice, it can be helpful to set d_2 quite high for spreading the foliage sparsely into large new areas, and then decrease it for filling in the leaves densely. Once a leaf is placed we can optionally update its texture. Some flora have a distinctive spatially-varying color profile, for instance if parts of the plant get more sun than others. In general we found that neighboring leaves are often more similar in appearance than distant ones. Since new synthesized leaves start off with a random texture from the statistical color model (Section 7.1), the appearance of a positioned leaf may seem out of place amongst its local neighbors. For this reason, we generate a new texture from the leaf color model, but this time using the set of neighboring leaves as a guide. Specifically, we randomly choose model parameters again, but this time *within* the range of the parameters of the leaves that are within a distance of d_3 from the new leaf. This final parameter controls the variation in synthesized leaf appearance.

We propose two applications for leaf synthesis. The first is to increase the overall density of a captured plant, for instance if heavy occlusions prevented a full reconstruction. The second is more artistic in nature, to grow foliage in new scenes as defined by an arbitrary volume. In the next section we will demonstrate both examples.

8 Results

We demonstrate our leaf reconstruction and synthesis algorithms on four different flora species with dense leaf structure.

Figure 9 shows a reconstruction for a bush with broad leaves present at varying depths. The initial reconstruction (2nd image) captures the first few depth layers of leaves. Given this information about the structure and appearance of the bush, we use our data-driven synthesis approach to fill in the deeper layers. A simple volume (3rd image) specifies the area we wish to fill in. The final result (4th image) is a dense bush with foliage that closely resembles the real plant. Figure 1 shows a close-up render of the result. Please refer to the accompanying video for more results.

In Figure 10 we show the result of reconstructing dense ground leaf cover. This dataset contains many partial occlusions, however our method successfully reconstructs many leaves, as shown at top center. Using our synthesis approach within the bounding box of the captured flora we can controllably increase the density of the leaves. Here we demonstrate several different densification levels (100% more leaves, 200%, 300% and 400%).

A third species result is shown in Figure 11. Here we demonstrate the ability to use multiple exemplars, by pruning and scanning three different leaves (shown in the center). Densification is performed in a similar way to Figure 9. The final result closely resembles the real foliage and represents the general appearance of the captured species.

Our most challenging dataset is the plant in Figure 12, which contains thousands of leaves with many occlusions. Our reconstruction algorithm captures a dense sampling of the leaves, and we increase the density by an additional 50% using the synthesis algorithm.

To highlight the importance of utilizing learned leaf structure and configuration during synthesis, we compare our data-driven synthesis technique to a synthesis approach that randomly places and orients leaves within the volume. Figure 13 shows the result, where the captured bush leaves from Figure 9 are densified by a factor of 50%. Our data-driven approach synthesizes leaves that more naturally blend with the captured flora. In particular, note that the synthesized leaves in the data-driven approach have a similar frontal configuration to the captured leaves. While this effect could be achieved using a heuristic, we achieve the result using a physically realistic learned model of leaf configuration.

Finally, Figure 14 demonstrates artistic use of the algorithm. Here an artist is constructing a scene and has identified two volumes where two different species of flora should grow. Our method automatically fills these volumes with foliage that resembles the captured leaves both in structure and appearance. In front of the archway we use the model learned from the bush dataset (Figures 1 and 9) and inside the archway we synthesize ivy-like vegetation from our ground-based leaves dataset (Figure 10). The result is natural-looking flora at the cost of only a few images of the real-world example.

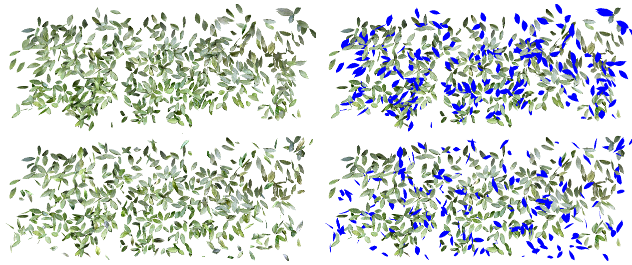

Figure 13: *The captured leaves from Figure 9 are increased in density by 50% using two different schemes. Top row: densification using our data-driven leaf synthesis. Bottom row: densification using randomly placed leaves. For both rows, the left and right images have identical structure, but the synthesized leaves are shown in blue at right.*

All our photo-realistic results were generated with the Mitsuba renderer [2012] and a custom *leaf BSDF* plug-in that implements a simple phenomenological leaf appearance model and its Monte Carlo importance-sampling routines. Our appearance model is composed of three components: front-facing diffuse reflection, modulated by the captured leaf reflectance; back-facing diffuse transmission, similarly modulated by a de-saturated filtering of the captured leaf reflectance; and front-facing specular reflection based on Blinn's [1977] formulation.

9 Conclusion

This paper describes the 3D reconstruction of dense flora. The problem has been little investigated so far and represents a difficult scene type for 3D reconstruction because of repeated structure, low variation in color, and a multiplicity of small overlapping surface patches. Nevertheless, our capture algorithm demonstrates good quality reconstructions for varied examples of foliage, and our novel approach for data-driven leaf synthesis can be used to increase the density of captured flora or to design foliage elements of a scene under artistic control.

Limitations. A limitation in the current system is that we capture only one species of leaf at a time. Dense foliage often consists of mixed and closely adjacent species, and correct handling requires extension to our method. For example, advance segmentation could be done as a preprocess, or a joint process could simultaneously apply leaf exemplars for the multiple species that are present. A further challenge is that capturing in direct sunlight results in shadows baked into the leaf textures. The result in Figure 12 is an example. Thus capture is best done under diffuse ambient illumination, such as on a cloudy day or in the shade. It is also preferable to capture on a calm day, so that the foliage remains still during acquisition. Finally, our method is more suited for foliage that contains a certain amount of natural randomness rather than well-structured and regular patterns.

Future work. We identify several directions for future work. One possibility is to detect leaves in the images as a pre-filter to guide the leaf fitting. Another idea is to infer stem layout from captured leaves in order to produce a complete plant. Furthermore, the combination of our leaf reconstruction and synthesis methods together with large-scale tree capture techniques can lead to complete multi-scale tree models.

Flora modeling is an early-stage research area, but we believe that it has applications across games and movies, education, scientific

Figure 9: *Reconstruction and synthesis for a bush with varying depth. From left to right: a reference image, the captured leaves, user-specified volume for extending the leaves deeper, and final result for capture + synthesis.*

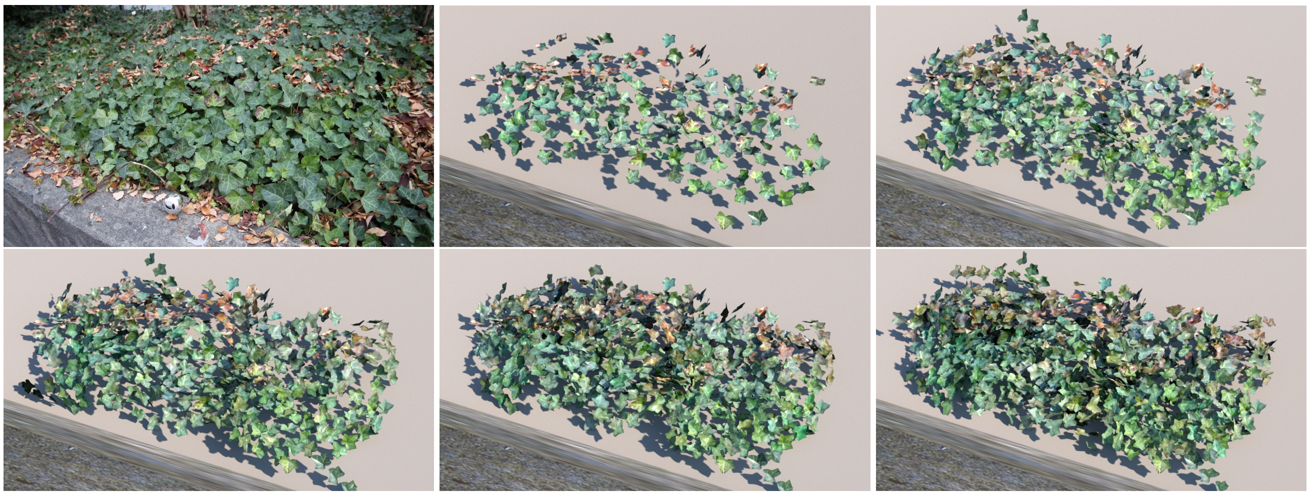

Figure 10: *Reconstruction of dense ground leaves. Top row: the reference image, captured leaves and addition of synthesized leaves by 100%. Bottom row: addition of 200%, 300% and 400% respectively.*

measurement, and archiving of natural environments. As far as we are aware, this work represents a first attack on automatically reconstructing dense foliage from images. Coupled with the method for data-driven synthesis, our technique provides an invaluable tool to model complex flora environments.

Acknowledgements

We thank Maurizio Nitti for modeling the archway scene.

References

AHRENDS, H. E., ETZOLD, S., EUGSTER, W., BUCHMANN, N., JEANNERET, F., AND WANNER, H. 2009. Use of digital images to observe forest phenology and drought stress. In *EGU General Assembly Conference Abstracts*, vol. 11, 10886.

ANASTACIO, F., SOUSA, M. C., SAMAVATI, F., AND JORGE, J. A. 2006. Modeling plant structures using concept sketches. In *Proceedings of NPAR*, 105–113.

BARANOSKI, G. V. G., AND ROKNE, J. G. 2001. Efficiently simulating scattering of light by leaves. *The Visual Computer 17*, 8, 491–505.

BEELER, T., BICKEL, B., SUMNER, R., BEARDSLEY, P., AND GROSS, M. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. Graphics (Proc. SIGGRAPH 98)*.

BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. on PAMI 14*, 2, 239–256.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 187–194.

BLINN, J. F. 1977. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph. 11*, 2 (July), 192–198.

BRADLEY, D., BOUBEKEUR, T., AND HEIDRICH, W. 2008. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *Proc. CVPR*.

CHEN, X., NEUBERT, B., XU, Y.-Q., DEUSSEN, O., AND KANG, S. B. 2008. Sketch-based tree modeling using markov random field. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 27*, 109.

COBWEB. 2013. *Citizen Observatory Web,edina.ac.uk*.

DEUSSEN, O., AND LINTERMANN, B. 2005. *Digital Design of Nature: Computer Generated Plants and Organics*. Springer-Verlag.

DIENER, J., REVERET, L., AND EUGENE, F. 2006. Hierarchical retargetting of 2d motion fields to the animation of 3d plant models. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 187–195.

FISCHLER, M. A., AND BOLLES, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6, 381–395.

GOESELE, M., CURLESS, B., AND SEITZ, S. M. 2006. Multi-view stereo revisited. In *CVPR*.

Figure 11: *A dense bush reconstruction. The reconstructed and densified result (right) closely resembles the target foliage (left). This result is generated using three different exemplars (center).*

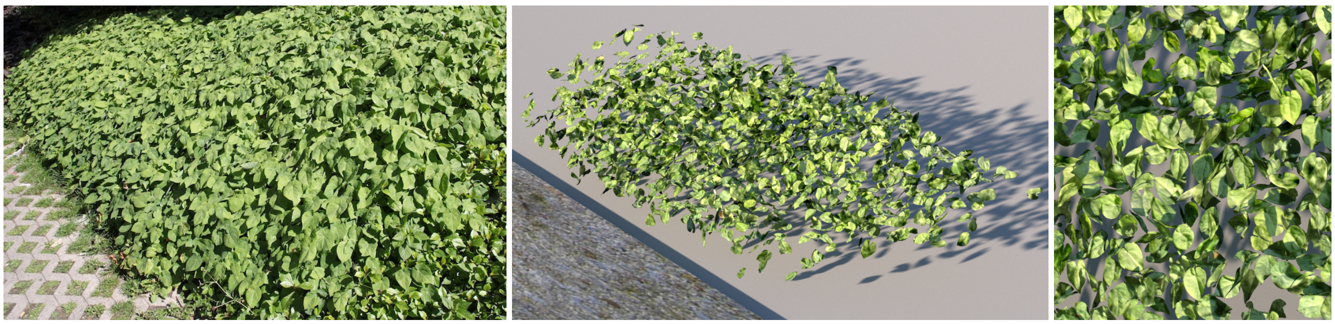

Figure 12: *We reconstruct a very dense plant with many partial occlusions. From left to right: a reference image, the reconstructed leaves with 50% densification, and a zoom-in render.*

GREENE, N. 1989. Voxel space automata: modeling with stochastic growth processes in voxel space. *SIGGRAPH Comput. Graph. 23*, 3, 175–184.

HABEL, R., KUSTERNIG, A., AND WIMMER, M. 2007. Physically based real-time translucency for leaves. In *Proc. Eurographics Symposium on Rendering*, 253–263.

JAKOB, W., 2012. Mitsuba renderer. www.mitsuba-renderer.org.

LI, C., DEUSSEN, O., SONG, Y.-Z., WILLIS, P., AND HALL, P. 2011. Modeling and generating moving trees from video. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 30*, 127.

LINDENMAYER, A. 1968. Mathematical models for cellular interactions in development ii. simple and branching filaments with two-sided inputs. *Journal of Theoretical Biology 18*, 3, 300–315.

LINTERMANN, B., AND DEUSSEN, O. 1999. Interactive modeling of plants. *IEEE Comput. Graph. Appl. 19*, 56–65.

LIVNY, Y., YAN, F., OLSON, M., CHEN, B., ZHANG, H., AND EL-SANA, J. 2010. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 29*, 151.

LIVNY, Y., PIRK, S., CHENG, Z., YAN, F., DEUSSEN, O., COHEN-OR, D., AND CHEN, B. 2011. Texture-lobes for tree modelling. *ACM Trans. Graphics (Proc. SIGGRAPH) 30*, 53.

MA, W., ZHA, H., LIU, J., ZHANG, X., AND XIANG, B. 2008. Image-based plant modeling by knowing leaves from their apexes. In *Proc. ICPR*.

MUNDERMANN, L., MACMURCHY, P., PIVOVAROV, J., AND PRUSINKIEWICZ, P. 2003. Modeling lobed leaves. In *Computer Graphics International*.

NEUBERT, B., FRANKEN, T., AND DEUSSEN, O. 2007. Approximate image-based tree-modeling using particle flows. *ACM Trans. Graphics (Proc. SIGGRAPH) 26*, 88.

OKABE, M., OWADA, S., AND IGARASHI, T. 2005. Interactive design of botanical trees using freehand sketches and example-based editing. *Computer Graphics Forum 24*, 3, 487–496.

PALUBICKI, W., HOREL, K., LONGAY, S., RUNIONS, A., LANE, B., MĚCH, R., AND PRUSINKIEWICZ, P. 2009. Self-organizing tree models for image synthesis. *ACM Trans. Graph. (Proc. SIGGRAPH) 28*, 58.

PAPAZOV, C., AND BURSCHKA, D. 2011. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *Proceedings of the 10th Asian conference on Computer vision*, 135–148.

PIRK, S., NIESE, T., DEUSSEN, O., AND NEUBERT, B. 2012. Capturing and animating the morphogenesis of polygonal tree models. *ACM Trans. Graph. 31*, 6, 169:1–169:10.

PIRK, S., STAVA, O., KRATT, J., SAID, M. A. M., NEUBERT, B., MĚCH, R., BENES, B., AND DEUSSEN, O. 2012. Plastic trees: interactive self-adapting botanical tree models. *ACM Trans. Graph. 31*, 4, 50:1–50:10.

PRUSINKIEWICZ, P., AND LINDENMAYER, A. 1990. *The algorithmic beauty of plants*. Springer-Verlag.

PRUSINKIEWICZ, P., JAMES, M., AND MĚCH, R. 1994. Synthetic topiary. In *SIGGRAPH '94*, 351–358.

QUAN, L., TAN, P., ZENG, G., YUAN, L., WANG, J., AND KANG, S. B. 2006. Image-based plant modeling. *ACM Trans. Graphics (Proc. SIGGRAPH) 25*, 599–604.

RECHE-MARTINEZ, A., MARTIN, I., AND DRETTAKIS, G. 2004. Volumetric reconstruction and interactive rendering of trees from

Figure 14: *Artistic synthesis of flora for scene modeling based on two of our captured datasets.*

photographs. *ACM Trans. Graphics (Proc. SIGGRAPH) 23*, 720–727.

SCHNABEL, R., WAHL, R., , AND KLEIN, R. 2007. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum 26*, 2, 214–226.

SEITZ, S. M., CURLESS, B., DIEBEL, J., SCHARSTEIN, D., AND SZELISKI, R. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: Exploring image collections in 3d. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 25*, 3, 835–846.

SONOHAT, G., SINOQUET, H., KULANDAIVELU, V., COMBES, D., AND LESCOURRET, F. 2006. Three-dimensional reconstruction of partially 3d-digitized peach tree canopies. *Tree Physiology 26*, 3, 337–351.

SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 175–184.

STRECHA, C., FRANSENS, R., AND GOOL, L. V. 2006. Combined depth and outlier estimation in multi-view stereo. In *CVPR*.

SZELISKI, R. 1999. A multi-view approach to motion and stereo. In *CVPR*.

TALTON, J. O., LOU, Y., LESSER, S., DUKE, J., MĚCH, R., AND KOLTUN, V. 2011. Metropolis procedural modeling. *ACM Trans. Graph. 30*, 11:1–11:14.

TAN, P., ZENG, G., WANG, J., KANG, S. B., AND QUAN, L. 2007. Image-based tree modeling. *ACM Trans. Graphics (Proc. SIGGRAPH) 26*, 87.

TAN, P., FANG, T., XIAO, J., ZHAO, P., AND QUAN, L. 2008. Single image tree modeling. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 27*, 108.

WILSON, E. 2009. *Ant Lovers Unite,www.npr.org*.

WITHER, J., BOUDON, F., CANI, M.-P., AND GODIN, C. 2009. Structure from silhouettes: a new paradigm for fast sketch-based design of trees. *Computer Graphics Forum 28*, 2, 541–550.

XU, H., GOSSETT, N., AND CHEN, B. 2007. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Trans. Graphics 26*, 19–31.