Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Hierarchical retargetting of 2D motion fields to the animation of 3D plant models

Julien Diener[1], Lionel Reveret[1], Eugene Fiume[2]

[1] INRIA/GRAVIR, France
[2] DGP, Dept. of Computer Science, Univ. of Toronto, Canada

Abstract

The complexity of animating trees, shrubs and foliage is an impediment to the efficient and realistic depiction of natural environments. This paper presents an algorithm to extract, from a single video sequence, motion fields of real shrubs under the influence of wind, and to transfer this motion to the animation of complex, synthetic 3D plant models. The extracted motion is retargeted without requiring physical simulation. First, feature tracking is applied to the video footage, allowing the 2D position and velocity of automatically identified features to be clustered. A key contribution of the method is that the hierarchy obtained through statistical clustering can be used to synthesize a 2D hierarchical geometric structure of branches that terminates according to the cut-off threshold of a classification algorithm. This step extracts both the shape and the motion of a hierarchy of features groups that are identified as geometrical branches. The 2D hierarchy is then extended to three dimensions using the estimated spatial distribution of the features within each group. Another key contribution is that this 3D hierarchical structure can be efficiently used as a motion controller to animate any complex 3D model of similar but non-identical plants using a standard skinning algorithm. Thus, a single video source of a moving shrub becomes an input device for a large class of virtual shrubs. We illustrate the results on two examples of shrubs and one outdoor tree. Extensions to other outdoor plants are discussed.

1. Introduction

The realistic depiction of natural environments has long been a central problem in computer graphics. While the modeling and rendering of plants and trees has yielded convincing results in recent years, comparable progress in realistic plant motion under the influence of external forces such as wind remains problematic. Physical simulation is one possibility, but its computational complexity and the lack of direct control makes it desirable to find an alternative that may more readily operate at interactive rates, and that can allow direct control over the motion if desired. This paper demonstrates a method based on the observation of nature that provides visually convincing results without requiring the simulation of the exact motion of every branch and leaf. Currently, the standard method for real-time rendering and animation of moving plants and trees is to animate a hierarchy of billboards through the addition of *ad hoc* oscillatory or pseudo-random motion. We actually follow this approach structurally, but we show that more realistic motion and even geometrical structure can be automatically acquired from video input using careful statistical analysis.

The innovative idea of the paper is to illustrate that a sparse motion control structure of a tree can be automatically extracted from video footage and can be mapped to a complex 3D geometrical structure of a plant using a simple skinning framework. Physical phenomena such wind forces, structural elasticity, or inter-collision of the branches are statistically modeled from video, rather than explicitly simulated physically, which would be more expensive to compute, tune and control.

After reviewing previous work, we present an algorithm to extract a hierarchy of animated branches from a statistical clustering of features tracked from a single view in a video sequence. This step provides a set of results embedded in the video image plane. We employ a heuristic algorithm to project these structures into three dimensions, to achieve real-time rendering and animation of synthetic 3D

replicas of the images found on the input 2D video. We then show how the results can be used to animate similar, but non-identical, complex 3D models of shrubs. We finally discuss the limitations of our approach and conclude with future work.

2. Previous work

From the pioneering work of Prusinkiewicz and Lindenmayer [PL90], to the most recent advances (e.g., [RFL*05, WWD*05]), the modeling and rendering of plants has a long, rich literature. By contrast, the graphics literature on plant motion is much smaller, despite the fact that the problem is challenging and clearly essential to the depiction of natural environments.

Stam employs modal analysis to simulate the effect of wind as a load force on mechanical plant models. The wind force is numerically solved to account for turbulence effects [Sta97]. Perbet and Cani use a procedural wind model to animate different geometrical levels of plant representation (grass wisps and billboards) [PC01]. Through video analysis, we instead investigate how the visible effect of wind on plants can be modeled without making *a priori* assumptions about the wind force.

Beaudoin and Keyser quantify an accurate physical simulation of plants to provide a levels-of-detail system for real-time rendering of animated trees [BK04]. By using video analysis and statistical clustering, our approach provides an alternative to physical simulation. Because our algorithm is inherently hierarchical, our approach also offers a way to automatically generate an LOD structure.

Sun *et al.* propose the use of VIDA, or "video-input driven animation", a system to estimate the wind velocity from videos of the motion of trees or other oscillating systems [SJF03]. They invert the equations of motion of a mechanical model of a moving plant to infer a representative wind field that would account for that motion. It allows the introduction of additional effects onto the original video, such as synthetic snow, leaves or dust, as well as new trees, all of which would be coherently controlled through interactions with the estimated wind field. In our case, a video of a plant is analyzed to directly build a model and an animation of a 3D synthesized plant without estimating explicitly the wind force.

The goal of extracting 3D information from a single monocular view has been extensively studied in the Computer Vision literature. In the domain of "shape from motion" approaches, factorization methods have proven that rank constrains allow to extract shape and motion of a collection of 3D rigid bodies [CK98] or of a 3D deformable body [TYAB01]. These approaches require to estimate in advance the rank constraint, which is related to the number of rigid groups in the case of multiple rigid bodies or the number of linear degrees of freedom in the case of deformable body. In the case of moving trees, motion of branches is not as rigid as motion of different part of an articulated object, making these rank methods not adapted to our case. The goal of our paper is not to recover the exact 3D shape and motion of the original plant, but rather to investigate how standard Computer Vision techniques can provide efficient cues to create motion controller for complex 3D model of plants.

Modeling trees from photographs has been proposed by [RMD04], but their volumetric representation cannot be easily extended to animation. Video-based animation has been explored to control character animation from cartoons [BLCD02], and the gait of animal locomotion from live video documents [FRDC04]. Our work addresses plants and presents a method to not only predict motion, but also to automatically build a hierarchical 3D structure.

3. Building a hierarchy of branches from a single video view

In this section, we describe our experimental conditions, and how the structure and motion of a hierarchy of branches can be automatically estimated from feature positions tracked in the video image plane.

3.1. Feature extraction

Our method starts with a single view video of the whole plant (in our case approximately 50 cm height). We use a standard color video camera at the typical scan rate and resolution (interleaved video format). The first experiments we describe were carried out on indoor shrubs under controlled lighting. Aperiodic, non-smooth "wind" was generated by waving a cardboard sheet near the plant. Using small shrubs and plants in a pot allowed an efficient background subtraction so that outliers are easily rejected from the video analysis. The steps in the feature extraction process are as follows.

1. A plant is filmed in front of a uniform white or blue background with a fixed camera.
2. A short sequence of the background is filmed without the plant so that the color of each pixel is modeled as a gaussian with known mean and standard deviation to characterize the background.
3. In the plant video, a pixel is removed (i.e., replaced with black) if its color is within 95% of the gaussian distribution of the background model.

More complex methods exist for background subtraction, but this simple approach has proven to be sufficient for our experiments to validate our appraoch.

Feature tracking is then applied on the video sequence of the plant. Feature location is initiated with the method from [ST94]. We use the pyramidal implementation of the Lucas-Kanade algorithm [LK81] by [Bou00] for feature tracking. The algorithm has been parameterized so that 200

features are correctly tracked over the whole sequence, initialized with an average distance of 15 pixels between neighboring features.

At the end of this process, 200 sequences of 2D feature trajectories are collected over the entire video sequence of the plant. On the two test sequences, 125 and 250 frames were analyzed.

3.2. Feature clustering

Our hypothesis is that leaf motion tends to be grouped by the underlying branch structure of the plant, despite their apparently random individual motions. This hypothesis is intuitively illustrated on the video accompanying the paper, in which only extracted features are displayed as green dots over a black background. Instead of a collection of dots with completely random motion, the overall structure of the plant plainly emerges. We investigated this hypothesis quantitatively using clustering analysis.

A first approach would be to cluster feature positions for each frame using an euclidean distance over each feature's image co-ordinates. Our experiments showed that this approach leads to incorrect feature clustering in the sense that the leaves from different branches can be grouped together, since the orientation of branches may place the leaves on different branches in close proximity. Figure 1 compares a ground truth clustering made manually and the results of an automatic clustering based on feature positions only.

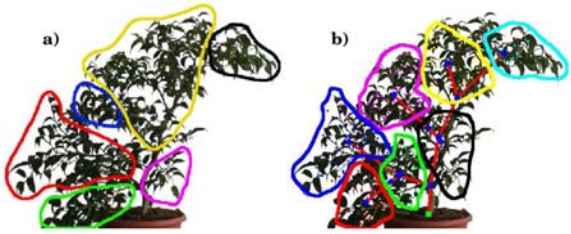

Figure 1: *(a) Ground truth obtained from a manual clustering, (b) Automatic clustering based on feature positions only.*

To solve this problem, we consider a composite distance for the clustering, which integrates both the position and the velocity of the features. The velocity is computed in 2D image co-ordinates as the difference between positions at two consecutive frames and thus corresponds to the velocity field of the optical flow algorithm used for feature tracking [Bou00]. Each velocity vector is normalized to have unit length. We use the product of the euclidean distance between positions and the angular distance between velocities as the new distance between two features. The distances on position and velocities are normalized by their respective standard deviation observed over all the features along the whole sequence.

As one might expect, our experiments showed that a per frame approach introduces instabilities, because incoherence between class composition occurs from frame to frame. To reduce this effect, a criterion is computed for each frame to evaluate its relevance to coherent motion. From a statistical perspective, we need to reject outlier frames that reduce the clustering quality. Such outliers arise due to two edge conditions as illustrated in Figure 2: (a) when there is only low-level "ambient" wind; and (b) when there is a considerable impulsive wind force. In both cases, there is diminished correlation between leaf motion and branch motion that is manifested in amplified tracking noise in the first, and overly homogeneous motion in the second. To characterize these two cases statistically, we first compute the average of distances between all pairs of points for each frame. Second, we compute the mean and standard deviation of this per frame value over the whole sequence. Frames with a value above or below two times the standard deviation with respect to the mean value are considered as outliers. On the two test sequences, containing respectively 125 and 250 frames, 16 and 22 are rejected. Finally, the metric for feature clustering is the average of composite distances between pairs of features computed over the selected inlier frames.

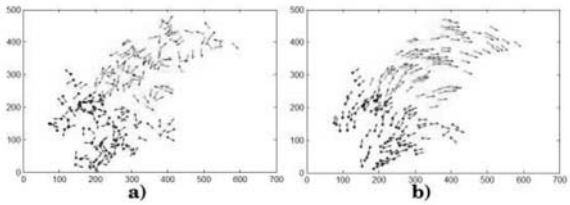

Figure 2: *(a) Unstructured motion, (b) Strong wind inducing a single overall motion.*

Finally, we obtain an automatic classification of features into groups with a solution closer to the manual clustering as illustrated in Figure 3.

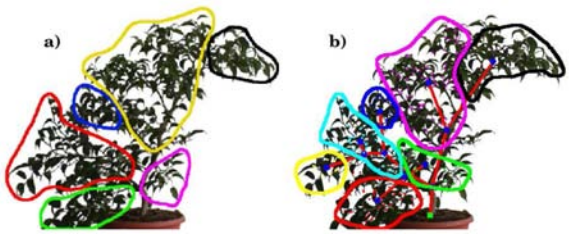

Figure 3: *(a) Ground truth manual clustering, (b) Automatic clustering based on distance combining feature position and velocity.*

3.3. Automatic selection of a hierarchy of branches

We now discuss details of the clustering algorithm and how it automatically proposes a hierarchy of branches. Most classification approaches fall into three main categories [Cor71, Gor87, HTF01]:

- *Partition*, where all individuals are clustered around representative data points (typically using *k*-means methods).
- *Division*, where an explicit ordering criterion iteratively splits individuals into hierarchical classes (leading to a hierarchical description of the data set–typically *kd*-tree methods).
- *Aggregation*, where individuals are iteratively compared and combined by closest pairs (leading also to a hierarchical representation).

Because we want to find a hierarchy, and because we have no inherent, explicit ordering criterion, we adopted the third choice with an aggregative method. At a first level, these methods require measuring the distance between two features and groups the closest pair. We use the distance described in the previous section.

The next step of the algorithm is: given three elements x, y, z, where x and y are already grouped into an aggregate group H, z is compared to the group H by evaluating a distance $d(H,z)$. Several choices are possible for this distance. The most common is the average distance:

$$d(H,z) = \frac{d(x,z)+d(y,z)}{2} \quad (1)$$

where $d(x,z)$ is the features distance described in the previous section.

By extension, if X and Y are subgroups gathered into a group H, and not just individual elements, the equivalent distance of z to the group H is given by

$$d(H,z) = \frac{n_X d(X,z)+n_Y d(Y,z)}{n_X+n_Y} \quad (2)$$

where n_X and n_Y are the number of leaf elements respectively below the subgroups X and Y.

Finally, when all individuals are already clustered into a group, further aggregation requires the evaluation of the distance between groups. The following formula is thus used:

$$d(X,Y) = \frac{1}{n_X}\frac{1}{n_Y}\sum_{x_i\in X}\sum_{y_j\in Y} d(x_i,y_j) \quad (3)$$

where X and Y are groups containing respectively n_X and n_Y individual elements.

Elements and subgroups are iteratively aggregated into groups by choosing the smallest distance. This minimum distance criterion provides a basic approach for aggregative clustering. However, it turned from our experiments that this approach lacked of robustness and provided unbalanced groups. To overcome this limitation, we have used a refined approach which considers the concept of inertia, known as the Ward criterion. This criterion uses the inertia of the system:

$$I = I_{inter} + I_{intra} = \sum_q n_q d(H_q,G) + \sum_q \sum_{x_i\in H_q} d(H_q,x_i) \quad (4)$$

where G is the set of all points, H_q is the subset of points in a subgroup, n_q is the number of individual elements of H_q.

The value of I, the total inertia of the system, is constant as it is independent of the clustering. At the initial stage all groups g_q contain only one feature, and thus the *intra* group inertia, I_{intra}, is null. Then, at each stage of the clustering process, the inertia of the system will be transferred from the *inter* group inertia, I_{inter}, to the *intra* group inertia by an increment Δ_i depending on which elements will be aggregated together. The principle of the Ward criterion is to aggregate, at each stage, the pair of elements which will minimize the increment Δ_i. This criterion has proven to be more robust for our problem than the minimal distance criterion, at the expense of an increase in calculation time.

These methods induce a hierarchical representation known as *dendrogram*. Given a required number of classes, the final clustering is obtained by adjusting a cut-off line on the dendrogram representation as illustrated in Figure 4.

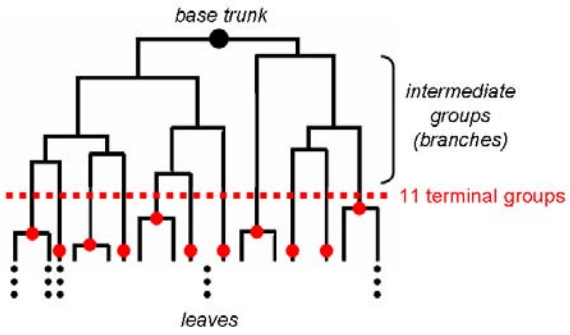

Figure 4: *Hierarchical clustering and cut-off line for the determination of the number of classes identified as terminal groups.*

From Figure 4, once a number of classes is set, all the leaf nodes below the cut-off are grouped as *terminal groups*. Groups above the cut-off line automatically provides a hierarchical structure of the shrub with *intermediate groups*, "up" to the base trunk. A branch is identified as joining two groups. By changing the location of the cut-off line, a different choice of branches is coherently selected. This approach automatically produces an LOD mechanism based on the statistical distribution of data only.

4. Creating 3D shape and motion

The algorithm of the previous section yields a topological hierarchy of branches that is valid over the entire video se-

quence. This section presents a method to automatically deduce 3D shape and motion from this structure. The shape and motion of the branches (terminal and intermediate groups) are first defined in the 2D video image plane and then converted to 3D.

4.1. Creating motion of the terminal groups

For each frame of the video sequence, we consider the distribution of the features within a terminal group to deduce the 2D parameters of an ellipse. This ellipse corresponds to the 95% isocurve of a gaussian distribution defined through the covariance matrix of the data set. A third axis is created perpendicular to the image view plane, with a length equal to the shortest axis of the 2D ellipse. This generates 3D ellipsoids for each frame and each terminal group. The center of each ellipsoid provides the position of the group in the image plane co-ordinate system.

This approach is inspired by methods for inflating 3D shape from 2D structure, typically illustrated by the "Teddy" system [IMT99], where 2D contours drawn by hand are automatically converted into 3D volumes. In our case, we do not create complete continuous polygonal surfaces, but instead we project depth values away from the image plane to the tree nodes.

4.2. Propagating motion to intermediate groups

The terminal groups are now geometrically positioned at each frame. For each frame, the position of an intermediate group is firstly computed as the average of its two child groups (each group has exactly two children, recalling Figure 4). This initialization process starts with the terminal groups as their location is already known from the previous section. Spatial locations are propagated down to the root node, providing the position of the intermediate group in the image-plane co-ordinate system. An additional node is introduced at the base of the trunk for visualization only.

This first step results in unrealistic T-junctions for branches. To create more realistic tree shapes, intermediate groups initially at T-junctions are shifted toward their parent group. It is realistic to expect that branch lengths do not change over the duration of the animation. We therefore estimate this *shifting coefficient* α by minimizing the standard deviation over all sequences of the distance between the node P_i to update and its two child nodes P_n and P_m (see Figure 5):

$$\alpha_i = argmin_\alpha [\sigma(\| P_i(\alpha) - P_n \|) + \sigma(\| P_i(\alpha) - P_m \|)] \quad (5)$$

To keep a coherent shape, this optimization process is started from the root node and propagated toward the terminal groups.

The position of every group (intermediate and terminal) is then transformed from the global co-ordinate system of the image plane to a set of local co-ordinate systems along the branch using the topology of the hierarchy. Group positions are converted into local polar co-ordinates. The length of each branch is fixed at its mean to keep fixed-length branches. The motion of each branch is expressed as a rotational animation curve in the 2D image plane. The clustering and 2D geometrical hierarchy is illustrated in Figure 6.

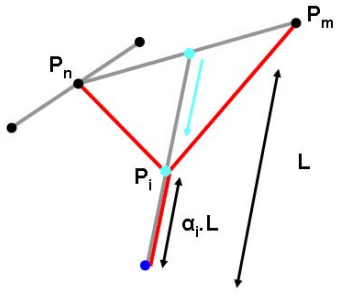

Figure 5: *Optimizing intermediate group locations.*

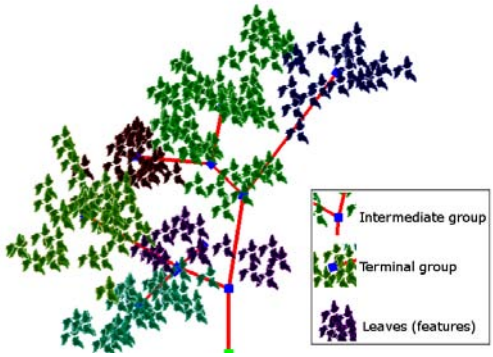

Figure 6: *Final geometrical hierarchy of groups.*

4.3. Extending groups to 3D shape and motion

So far, the shape and motion of the branch structure is embedded in the 2D image plane (only leaves are shaped in 3D). We propose two methods to finally extend them to 3D: one method relies on user interaction, the other one is fully automatic.

The first method keeps the geometrical hierarchy obtained from the previous section as is and is presented to the user to manually edit the tree structure. To make this process intuitive in 3D, for each branch node, axes are oriented so that:

- the branch joining the parent to child group corresponds to the x-axis,
- the z-axis is perpendicular to the image plane - this plane is assumed to carry the main motion of the branch,

- the y-axis is obtained by cross-product of the x and z axes.

Motion of the branch is assumed to be mainly contained in the image plane, which is orthogonal to the z-axis, resulting in a 2D rotational motion oriented around the z axis for the branch. An animation curve as a rotation around this z-axis is kept as the whole motion of the branch. The motion is thus embedded into the image plane. To create a non-planar 3D distribution of branches, while keeping the motion of the branch in a plane, it is proposed to the user to simply rotate this plane by changing the y-axis only. By changing only one degree of freedom, a wide variety of orientations can be intuitively created. This method was used to implement the rendering where each leaf is assigned a small texture (Figure 7). This shape editing has to be done for a single frame only, the animation staying in a predefined plane. Note that by grouping all the leaves of the same terminal group into a display list, real-time rendering is easily achieved in this case. Intermediate groups are animated as a standard hierarchy of rigid co-ordinate systems.

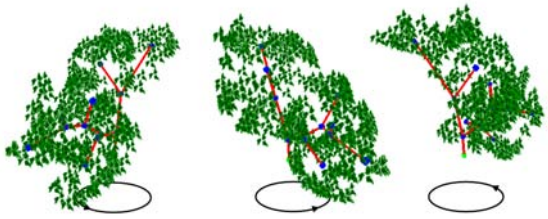

Figure 7: *3D geometrical structure with texture sprites for leaves.*

The second method provides 3D location and motion of groups automatically. The algorithm in section 4.1 is slightly modified: indeed, for each frame, the distribution of the leaves is now estimated over the entire plant data set, and not on a per-group basis any more. This creates a global 3D ellipsoid for the plant. The location of the center of the terminal groups is therefore estimated in 3D by projecting their 2D positions perpendicularly to the image plane. It thus provides a depth value to each terminal group. The algorithm described in section 4.2 remains valid in 3D and is applied to propagate motion to the intermediate group and compute the *shifting coefficient* α. Now that the terminal groups are initially distributed in 3D, so too are the intermediate groups. 3D positions are also converted into local co-ordinates of the hierarchy. Motion is now converted into spherical co-ordinates, instead of polar co-ordinates as above. Note that depth is inflated into one direction only. To cover the other half part of 3D space, the results are simply mirrored with respect to the image plane. This method provides denser 3D space coverage of the branches and was used in the animation method described in the next section and illustrated by Figures 11 and 9.

5. Controlling the animation of complex 3D plant models

The previous section gave a method to extract an animated 3D hierarchical structure, and renders in real-time a 3D replica of the input video footage. In this section, we present a surprising result in which a similar but different structure of a more complex 3D plant model can be animated using the same extracted 3D tree structure.

5.1. 3D animation of a plant by skinning

The leaves attached to a terminal group only contribute to the rendering and can be removed without altering the tree structure. The key idea is to use this tree structure as a control skeleton for animating a 3D model using skinning algorithms, just as it is currently done for character animation. The complex 3D model of a shrub is bound to the control skeleton using the standard algorithm for smooth skinning implemented in the *Maya* software. As the branches of the skeleton move, so do those of the shrub (Figure 8).

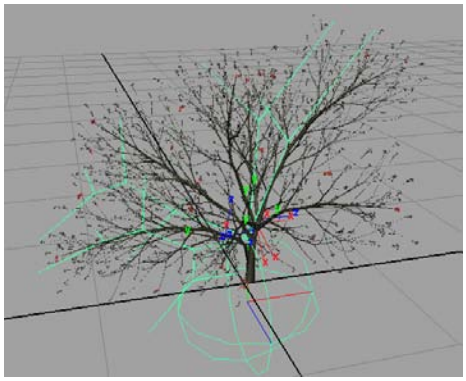

Figure 8: *3D geometrical tree structure used as a control skeleton for animation by skinning.*

Although the geometrical structure of the control skeleton does not exactly reflect the structure of the target 3D model, the results are surprisingly convincing. We believe that such results are achieved due to the fact that the 3D model used as a target has been generated by a computer program, and as such, it has a rather isotropic structure: in any direction pointed by the control branches for motion, dense population of leaves can be found. The video illustrates in detail the example displayed on Figure 9.

5.2. Interactive 3D modeling

The isotropic assumption may be too strong to be respected in some cases, such as the one on the shrub displayed on Figure 11. Indeed, the structure of this shrub is quite unbalanced. As in the previous case, the skeleton structure is first scaled to match the target 3D model. The user then assembles 3D branches with foliage to construct a final tree, guided

Figure 9: *Original footage and synthesis target for animation of a complex 3D model of shrub*

by the location and orientation of the control skeleton. Based on the 3D hierarchical structure provided by the video analysis, this modeling process could be automated. Figure 10 illustrates an example of this interactive modeling. In this case, six branches are copied from an existing 3D model and composed together in a new configuration.

Figure 10: *Three main steps of the interactive modeling of shrubs that fit the control skeleton.*

Figure 11: *Video footage controlling 3D animation of a shrub (motion blur corresponds to a* $1/50$*s shutter speed).*

6. Limitations and discussion

Indoor experiments on shrubs have greatly facilitated the background subtraction of the tree foliage, as well as the robustness of the feature tracking. We have include in the video a preliminary test on a real outdoor tree with strong wind (Figure 12). This test mostly challenges the feature tracking as leaves are rapidly appearing and disappearing, while the standard technique for feature tracking we used relies on intensity consistency. Typically, uncontrolled outdoor conditions may necessitate the use of more robust feature-tracking algorithms from the Computer Vision literature ([MS05]). However, the goal of this paper has been to focus on animation and to demonstrate the basic idea that statistical modeling can be used to replicate realistic motion of plants from video without resorting to physical simulation.

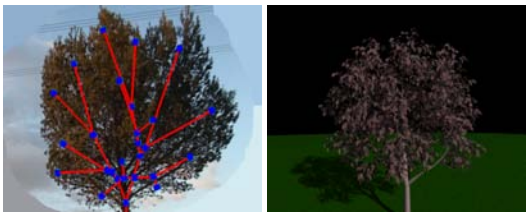

Figure 12: *Original footage and synthesis target for an outdoor tree*

Another key aspect of our approach is that it is based on single-view video footage. Although our method generates 3D structure, working with a single view naturally implies that all the relevant motion of the plant is extracted from a single viewing plane. To take maximal advantage of this configuration, our methodology has been to film the plants so that the wind flow is parallel to the view plane. It turned out that most visually pertinent motion was observed using this approach. Using multiple cameras to effect stereoscopic reconstruction would permit extracting fully 3D plant motion. However, this would raise other problems such as feature correspondence, which are beyond the scope of this paper.

Finally, the flickering appearance of rapid motion of leaves are natural part of a realistic aspect of a tree under the influence of strong wing (see the outdoor example on the video). This point is not addressed by our method. Our approach has been to provide a final result which can be easily integrated into a standard computer animation pipeline, based on an animated hierarchy of textured rigid objects. However, the rendering part of our technique can be easily extended using advanced techniques for per pixel shading in order to incorporate more realistic aspect of the leaves. Based on our approach which provides a stabilized tracking of leaves location, capturing this subtle variation of the leaves aspect from video will be an interesting future work.

7. Conclusion

This paper has demonstrated that subtle and complex plant motion can be visually modeled and mapped onto an animated 3D model with motion extracted from a single-view video. The coupling of video analysis and hierarchical clustering offers an appealing alternative to physical simulation, which is known to be difficult to control. An unusual application of skinning combined with our method indicated that a wide range of complex 3D models may also be animated.

Our short term plans for future work include automatic modeling of rich 3D plant model based on the 3D hierarchical tree structure extracted from video by our method. The

validation of the method on outdoor trees is also an important direction.

More broadly, we are working on an extension of our method in which wind amplitude and direction are used as control parameters for the generated 3D animation. As presented here, video and animation are essentially played back from the original footage. If quantitative measurement of the wind amplitude and orientation could be captured along with the video sequence, it would allow us to extend our method into a motion synthesis framework that would permit direct control by wind parameters. A learning phase would formalize correspondences between these wind parameters and the 3D motion of plants generated by our method. The resulting model would provide a method to control realistic plant motion from intuitive wind parameters.

References

[BK04] BEAUDOIN J., KEYSER J.: Simulation levels of detail for plant motion. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA) 2004* (2004). 2

[BLCD02] BREGLER C., LOEB L., CHUANG E., DESHPANDE H.: Turning to the masters: motion capturing cartoons. In *Proc. SIGGRAPH'02* (2002). 2

[Bou00] BOUGUET J.-Y.: Pyramidal implementation of the Lucas-Kanade feature tracker. Intel Corporation, Microprocessor Research Labs report, 2000. 2, 3

[CK98] COSTEIRA J., KANADE T.: A multibody factorization method for independently moving objects. *IJCV 29*, 3 (1998), 159–179. 2

[Cor71] CORMACK R.: A review of classification. *J. of the Royal Statistical Society, Series A 134*, 3 (1971), 321–367. 4

[FRDC04] FAVREAU L., REVERET L., DEPRAZ C., CANI M.-P.: Animal gaits from video. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA) 2004* (2004). 2

[Gor87] GORDON A.: A review of hierarchical classification. *J. of the Royal Statistical Society, Series A 150* (1987), 119–137. 4

[HTF01] HASTIE T., TIBSHIRANI R., FRIEDMAN J.: *The Elements of Statistical Leaning*. Springer-Verlag, 2001. 4

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform design. In *Proc. of ACM SIGGRAPH'99* (1999), pp. 409–416. 5

[LK81] LUCAS B., KANADE T.: Iterative image registration technique with an application to stereo vision. In *Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)* (1981), pp. 674–679. 2

[MS05] MIKOLAJCZYK K., SCHMID C.: A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence 27*, 10 (2005), 1615–1630. 7

[PC01] PERBET F., CANI M.-P.: Animating prairies in real-time. In *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics* (New York, NY, USA, 2001), ACM Press, pp. 103–110. 2

[PL90] PRUSINKIEWICZ P., LINDENMAYER A.: *The Algorithmic Beauty of Plants*. Springer-Verlag, 1990. 2

[RFL*05] RUNIONS A., FUHRER M., LAN B., FEDERL P., ROLLAND-LAGAN A.-G., PRUSINKIEWICZ P.: Modeling and visualization of leaf venation patterns. *ACM Transactions on Graphics (SIGGRAPH'05 Proc.) 24*, 3 (2005), 702–711. 2

[RMD04] RECHE A., MARTIN I., DRETTAKIS G.: Volumetric reconstruction and interactive rendering of trees from photographs. *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings) 23*, 3 (2004). 2

[SJF03] SUN M., JEPSON A., FIUME E.: Video input driven animation (vida). In *International Conference on Computer Vision (ICCV) 2003* (2003). 2

[ST94] SHI J., TOMASI C.: Good features to track. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (1994), pp. 593–600. 2

[Sta97] STAM J.: Stochastics dynamics: Simulating the effects of turbulence on flexible structures. In *Proc. of Eurographics'97* (1997), vol. 16, pp. 159–164. 2

[TYAB01] TORRESANI L., YANG D., ALEXANDER G., BREGLER C.: Tracking and modelling non-rigid objects with rank constraints. In *Proc. IEEE CVPR 2001* (2001). 2

[WWD*05] WANG L., WANG W., DORSEY J., YANG X., GUO B., SHUM H.-Y.: Real-time rendering of plant leaves. *ACM Transactions on Graphics (SIGGRAPH'05 Proc.) 24*, 3 (2005), 712–719. 2

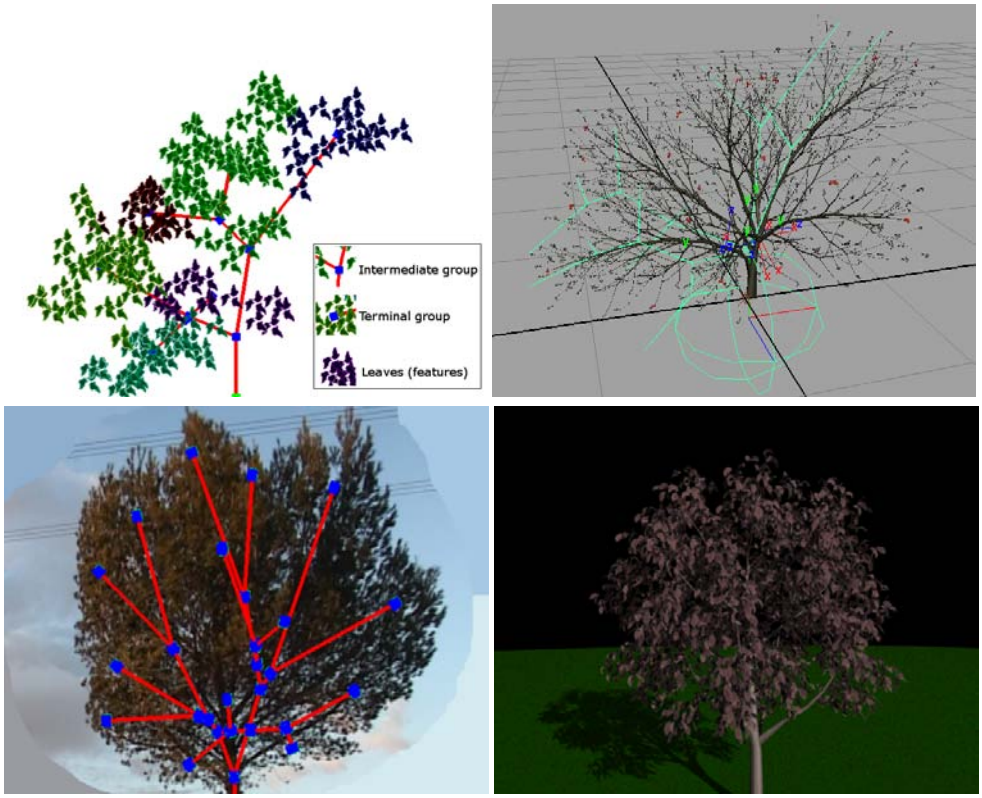

Figure 13: *Results of hierarchical clustering (top left); Control skeleton for 3D animation of a complex 3D model(top right); Original footage with hierarchical clustering and synthesis target for outdoor tree (bottom).*