Abstract—Modeling of real trees has attracted much attention in computer graphics in recent years. A new sketch-based method for fast tree modeling is presented in this paper. After sketching main branches and crown silhouette on two images of a tree, a realistic 3D tree model can be generated automatically. Freehand sketch provides necessary 2D shape information of the visible and hidden branches in input images. 3D main branches are built according to the structures of two 2D skeletons by depth retrieval. Small branches are generated within the crown silhouettes based on self-similarity. Experiments demonstrate that the main visual effects of the tree in images are well kept in the reconstructed 3D models.

I. INTRODUCTION

3D tree models are important for landscape simulation. In the past decades, various tree modeling methods have been developed. However, it is generally complex and time-consuming to model a real tree in the sense of data collection, reprocessing, and parameter adjustment. Technical challenges still exist.

In this paper, a new sketch-based approach for tree modeling is presented. It is designed for easy use and fast modeling, while the visual effect is specially considered. By necessary sketching, image segmentation is avoided, so that the system is robust and flexible. Using a simple reconstruction algorithm, the modeling process achieves a high efficiency. This method can be applied to various tree structures and the reconstructed models are faithful to their inputs.

Rule-based and image-based modeling: Plant modeling began early with rule-based methods[1][2][3]. They are generally used to simulate botanical organs or growing process rather than model a real tree. Image-based methods are newly developed. Calibrated images are normally used as inputs in classical works, such as Shlyakhter et al.[4], Quan et al.[5], Tan et al.[6] and Teng et al.[7]. Tan et al.[8] used a single image as input and provided an easy way for approximate modeling. Neubert et al.[9] adopt a particle-flow method for approximate tree modeling.

Sketch-based modeling: Sketch-based modeling constructs a 3D tree model from 2D strokes. Okabe et al.[10] did this by guaranteeing distances between branches as large as possible. Chen et al.[11] converted freehand sketches into 3D tree models by probabilistic optimization. Wither et al.[12] designed trees by inferring branches from silhouettes at multiple scales. These methods are flexible, whereas they need artistic 2D sketch. We use few photos without exact registration as guidance, as makes it easy for common users to obtain realistic 2D strokes by sketching.

II. SKELETON MODELING FOR MAIN BRANCHES

Two input images of our system are seen as photos of a tree taken from two orthogonal directions through parallel projection, as is consistent with the thought of [9] for approximate tree modeling. On each image, sketching can easily identify main branches immersed in leaves. From 2D skeletons A and B extracted from strokes, a 3D skeleton of main branches is built, which takes on similar visual effect to input 2D skeletons viewed from corresponding directions.

2D tree skeleton is a branched structure. Each branch contains a series of skeleton points from bud node to tip node. For a given height \( V \), all branches passing height \( V \) in a 2D skeleton are those branches containing at least one skeleton point whose Y-coordinate is \( V \). We define several qualities for each branch of 2D skeleton. Branch \( \lambda \) = \( \{P_\alpha = (x_\alpha^1, y_\alpha^1, r_\alpha^1); \alpha = 1, 2, \ldots, m\} \) is taken as a sample branch.

- Bud node \( P_1 \): The first skeleton point in branch \( \lambda \) connecting with \( \lambda \)'s father branch.
- Tip node \( P_m \): The last skeleton point in branch \( \lambda \).
- Begin height \( b(\lambda) \): The Y-coordinate of branch \( \lambda \)'s bud node, i.e., \( y_1 \).
- End height \( e(\lambda) \): The Y-coordinate of branch \( \lambda \)'s tip node, i.e., \( y_m \).
- Minimal height \( h(\lambda) \): Minimal Y-coordinate in all skeleton points of branch \( \lambda \).
- Maximal height \( H(\lambda) \): Maximal Y-coordinate in all skeleton points of branch \( \lambda \).
- Up-family \( u(\lambda) \): Branch \( \lambda \) and its offspring;
- Down-family \( d(\lambda) \): Branch \( \lambda \) and its forefathers;
- Maximal extension height \( M(\lambda) \): The maximal Y-coordinate of branch \( \lambda \)'s up-family;
- Down-path \( p(\lambda) \): A path in branch \( \lambda \)'s down-family from its branch tip to the tree root;

2D skeletons A and B are in two planes thought of as perpendicular to each other. The algorithm for 3D skeleton construction called Depth Retrieval is conducted in this way: for each branch in \( A \), such as \( \lambda^1 \), searching its corresponding branch \( \mu^3 \) in \( B \), so that branch \( \lambda^1 \) obtains its depth (Z-coordinate) from the X-coordinate of \( \mu^3 \). 3D construction
proceeds in all branches of skeleton A in the sequence of branch hierarchy, and we need to confirm that all constructed 3D branches are properly linked.

Depth retrieval algorithm for 3D skeleton modeling is illustrated in Fig. 2.

For any branch λ^i in skeleton A, all branches passing height V in skeleton B are searched and they constitute group G_i. As two 2D skeletons are consecutive and they have similar heights, G_i generally contains at least one branch. Branch μ^j ∈ G_i is selected as the corresponding branch of λ^i by feature matching and a new 3D branch is reconstructed from λ^i and μ^j. The process of feature matching and 3D branch reconstruction is discussed below with more details.

M(x) is the maximal extension height of branch x. A linear discriminant function F_i(x) is defined in (1) to find a branch in G_i, whose maximal extension height is more similar to M(λ^i), viz. to find a branch making F_i(x) minimal, such as μ^j.

\[ F_i(x) = |M(x) - M(\lambda^i)| \]  (1)

\[ \mu^j = \arg \{ \min \{ F_i(x); x \in G_i \} \} \]  (2)

From branch λ^i and μ^j, a 3D branch η^k is reconstructed. Each point of η^k is built from two 2D points at the same height - one in branch λ^i and the other in μ^j according to the rules below.

Supposing 2D point \( P^i_\alpha = \{ x^i_\alpha, y^i_\alpha, r^i_\alpha \} \) in λ^i and \( Q^j_\beta = \{ x^j_\beta, y^j_\beta, r^j_\beta \} \) in μ^j are at the same height viz. \( y^i_\alpha = y^j_\beta \). 3D point \( S^k_\gamma = \{ x^k_\gamma, y^k_\gamma, z^k_\gamma, r^k_\gamma \} \) in η^k is constructed from \( P^i_\alpha \) and \( Q^j_\beta \) accordingly:

\[ x^3_{k,\gamma} = x^1_{k,\alpha} \]  (3)

\[ y^3_{k,\gamma} = y^3_{k,\gamma} \]  (4)

\[ z^3_{k,\gamma} = x^2_{k,\beta} \]  (5)

\[ r^3_{k,\gamma} = (r^1_{k,\alpha} + r^2_{k,\beta})/2 \]  (6)

If λ^i is lower than μ^j (\( H(\lambda^i) < H(\mu^j) \)), all points of λ^i can be converted into 3D points. Then we go to the next branch of skeleton A. If branch λ^i is higher than branch μ^j (viz. \( H(\lambda^i) > H(\mu^j) \)), construction stops at a point of λ^i (such as \( P^i_\alpha = \{ x^i_\alpha, y^i_\alpha, r^i_\alpha \} \), where \( y^i_\alpha = H(\mu^j) + 1 \)), we search a new branch in skeleton B passing height V (\( V = y^i_\alpha \)) by feature matching and use it as the corresponding branch for the remainder points of λ^i.

To make sure the reconstructed 3D skeleton connective, if η^k are not linked with former built branches, branch λ^i and μ^j should be prolonged backward along their down-paths before 3D branch construction.

When 3D construction in skeleton A finishes, generally, there are some branches left in skeleton B - complete or part of them have not been used as corresponding branches. These unused parts are put into a group L. Similar 3D construction is conducted in group L by seeking corresponding branches in skeleton A. Finally, a 3D skeleton of main branches is obtained and it resembles two input 2D skeletons.

In order to verify the consistency of the 3D model with its 2D input strokes, the 3D skeleton is projected to images. Fig. 3 shows one result of 3D skeleton reconstruction. It can be seen that the projection of 3D skeleton overlaps all input strokes.

III. CROWN MODELING THROUGH BRANCH PROPAGATION

After constructing main branches, small branches are added to complete the whole tree while guarantee its crown shape.
consistent with 2D crown silhouettes. Based on the principle of self-similarity, small branches are inferred from the structure of main branches. They are modeled by copying part of their farther branch, turning to appropriate directions and connecting in a plausible way. In this way, new generations keep the feature of the main branches.

Radii of the branches are determined by the rules discovered by Da Vinci: \( r^2 = a \cdot \sum r_i^2 \), which describes the relationship between the radius of a branch and the radius of its children [9]. A set of cones are placed around the 3D skeleton to construct branch geometry.

Finally, tree crown is enriched with leaves by attaching quadrilaterals mapped by a leaf texture along skeleton nodes of small branches, while the angle between leaf and branch can be adjusted.

IV. RESULTS AND DISCUSSION

Three trees are modeled in this paper, a London Plane, a peach and an apple tree. Fig. 1 shows an example of London Plane modeling from two photos. In Fig. 4, 3D models of a peach are reconstructed from strokes of different amount. We can see that the final effects of these models are similar to the input images. It is still possible to use only one image as input, such as the case in Fig. 5, where an apple tree is created from one photo downloaded from internet. Forests could be built from these tree models, as is shown in Fig. 6.

The system works at interactive rates. Sketching on images creates strokes interactively. The construction of 3D main branches from strokes needs about 10 seconds for increasing 2000 branches. Total time of tree modeling is typically less than one minute on a standard PC with 2G CPU.

Limitations of the method: Although this is a fast method to generate complex real trees from freehand sketch, our method has some limitations. The spatial occupation of branches are mainly decided by branch strokes and parameters, so that the distribution of branches could be uneven in the tree crown space. It is possible to control branch distribution by adding global control, but the cost could be longer running time. In addition, although we have defined several groups of constant parameters for easy use, the parameters lack direct link to the prior knowledge of tree species.

V. CONCLUSIONS AND FUTURE WORK

A new technique is presented in this paper for real tree modeling. It provides a convenient and fast way to model different species of trees. This method runs at interactive rates, and it keeps well the main structure and crown shape of a real tree in images. We have tested it with several sketched image sets.

In the further, a mixed method based on sketch and image could be developed for 2D shape extraction, and more parameters can be processed to be self-adaptive according to image information. In this way, the modeling speed and effect should be well balanced.

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China with projects No. 60970093, No. 60902078 and No. 60872120; in part by the National Key Technology R&D Program of China under Grant No. 2009BAK43B18; in part by China Postdoctoral Science Foundation funded project No. 200801127 and No. 20080430584; in part by the National High Technology Development 863 Program of China under Grant No. 2008AA01Z301.
Fig. 6. Forests made up of two peach models and an apple tree model. Each forest contains 100 trees in the area of 60m*60m.

REFERENCES


