Modeling of Tree Branches by Bayesian Network Structure Inference

Wei Ma[1,2]
mawei@cis.pku.edu.cn

Yizhou Wang[1,2]
Yizhou.Wang@pku.edu.cn

Hongbin Zha[2]
zha@cis.pku.edu.cn

Wen Gao[1,2]
wgao@pku.edu.cn

[1]Nat'l Engineering Lab for Video Technology, Sch'l of EECS, Peking University
[2]Key Lab. of Machine Perception (MoE), Sch'l of EECS, Peking University

Abstract

In the paper, we present an approach to inferring 3D subtree structures from image pairs. The 3D structure is treated as a hidden Bayesian network, of which each node corresponds to an attributed skeleton point. The network structure is inferred in a bottom-up fashion. At the beginning, the root node of a subtree is manually specified in the images and then computed using stereo triangulation. Next, the subsequent computation automatically infers the child nodes stage by stage along the branches. At each stage, the child node states are sampled from a posterior distribution, which incorporates image observations in different viewpoints and pre-defined priors, such as smoothness. A trace-based stereo matching algorithm is introduced to propose the child node candidate states for computation efficiency. The experiments demonstrate that the proposed approach is competent in subtree construction.

1. Introduction

3D Modeling of real tree branches is important in enhancing the reality of virtual scenes. Reconstruction of a whole tree from only image information is extremely hard due to serious occlusions. Considering that the tree is composed of subtrees with a similar branching pattern, a widely accepted way is to synthesize intercrossed or occluded branches with reference to already-constructed subtrees [10] [8].

In this paper, we propose an approach to modeling subtree branches based on wide-baseline image pairs. Compared to other ordinary objects, modeling branch structures is much more challenging. First, stereo correspondences are ambiguous due to the large-disparity of most branches and the similar appearances among branches in color, texture and shape. The problem becomes more challenging when the input image pairs are wide-baseline. Second, branch intercrosses are inevitable. This is hard to be handled by traditional matching methods.

Stimulated by the similarity between branching structures and Bayesian Network, we represent the 3D branch structure by a directed graphical model, of which each node represents an attributed skeleton point on a branch. Our objective is to infer the graphical model structure satisfying the observations of different viewpoints and pre-defined priors using Bayesian inference. This is different from the other methods [4] [2], in which the structures of the graphical models were pre-constructed. Motivated by the bottom-up natural growing process of trees, we construct the hidden Bayesian network using one-pass bottom-up inference along the branches progressively. The one-pass inference demonstrates to be effective by the experimental results. Besides, branches constructed using our method have branch-unit and skeleton-form structures, which bring great convenience for editing without any further operation [6].

Given a pair of images about a subtree, we first calibrate them and remove the backgrounds. Then we perform the Bayesian inference to construct the graphical model progressively based on the calibrated foreground images. The first node, namely the root node of the subtree, is manually specified. Next, we find an MAP (Maximum a Posteriori) estimate of the child node states from their joint posterior distribution stage by stage (shown in Figure 1). The posterior distribution is computed by integrating the priors conditioned on the parent node states (such as structure smoothness constraints), and the likelihood of generating the foreground images. As a natural phenomenon, branches may grow, bifurcate or terminate at any stage throughout the modeling procedure. In another word, the number of graph nodes may be variable at different stages. To consider this, we adopt the stochastic algorithm MCMC (Markov Chain Monte Carlo) to sample the node states at each stage by reversible jumps and diffusions. In the end, all the branches terminate and form a complete subtree graphical model.

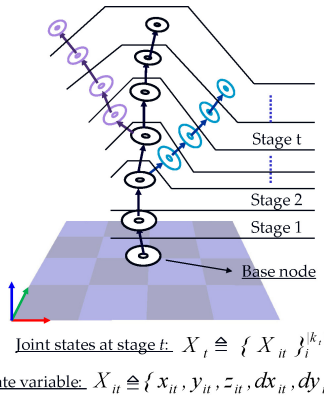

Figure 1. Progressive structure inference of a simple tree

In implementation, we adopt a data driven strategy to propose child node candidate states for growing and adding branches at each stage to accelerate the sampling. The candidate node states are obtained based on a 3D detection set, which is generated by matching the 2D branch segments detected in the image pair at that stage. To match the 2D segments, we introduce a reliable trace-based matching algorithm. The two segments in different images are considered to correspond to the same branch segment in 3D if any of their 2D traces are matched. Here, the 2D traces are defined as connected roads in the images starting from the segments and ending with branch tips.

In summary, our approach overcomes the difficulties in constructing branch models from wide-baseline and large-disparity image pairs. Observations in different viewpoints and pre-defined priors are integrated in a unified framework. The generated models are reliable in depth due to the use of wide-baseline image pairs. The branch intercross problem is also conquered thanks to the bottom-up inference strategy and the reliable stereo matching algorithm.

2. Related work

Recently, appears many approaches to modeling trees, specifically meaning branches, based on sensor data, including images and laser data. A big obstacle presenting ahead of this topic is occlusion and intercrossing structures of branches. For solvability, these approaches generally estimate a base structure of the tree and then synthesize the remaining parts.

To estimate the base structures, Xu et al. [10] used laser depth images and compute the shortest path from every 3D point to the root. This method suffers from serious noises in the laser data especially near thin twigs. Among image-based approaches, the number of involved images ranges from one to tens. Han et al. [4] modeled branches from a single image using a Bayesian method. The information in a single image is not enough to accurately reconstruct 3D branches. Teng et al. [9] tried short-baseline stereo method to construct trunk models. Due to the short-baseline, the generated 3D models looks unnatural when observed from new viewpoints. Neubert et al. [7] used particle flows piloted by two or three images to simulate branches. Trees generated by this method deviate much from the input images. Tan et al. [8] recovered visible branches from point cloud obtained using Structure from Motion. This multi-view method suffer the same noise problem as the laser-based ones. Moreover, it involves much cost in extract information from a large number of images.

For easier construction, some works first extract 2D skeletons in images, such as [7] and [9]. 2D skeletons help much the reconstruction process, especially in generating skeleton-form 3D models. However, they require extra work to extract skeletons in the input 2D images.

Our approach is based on wide-baseline image pairs, and capable of generating reliable depth. Furthermore, we do not require 2D skeletons but can produce branches with branch-unit and skeleton-form 3D structures, which bring great conveniences for further editing. During the reconstruction, only the first node should be indicated manually.

3. Problem formulation

In a pre-processing step, we capture an image pair of a subtree. Then, we calibrate the image pair and remove the backgrounds. For background removal, we use the magic wand in *AdobePhotoshop*. Since only two images are involved, the pre-processing is fast and convenient.

The core modeling procedure is formulated as a Bayesian inference from the observed 2D profiles, denoted as Z, to a 3D subtree structure S, which is in the form of a set of attributed skeleton points

$$P(S|Z) \propto P(Z|S)P(S). \tag{1}$$

Motivated by the natural laws that trees grow from bottom to top and subbranches are generated by their lower-level parent branches, we adopt a Bayesian network as the 3D branch structure S. $S \triangleq (V, E), V \triangleq \{k_i, X_i\}_i$, where, V and E represent a set of attributed skeleton points, and edges connecting the points respectively. k_i denotes the identification number of X_i for explicit semantic structures of reconstructed models. $X_i \triangleq (x_i, y_i, z_i, dx_i, dy_i, dz_i, r_i, s_i)$ with (x_i, y_i, z_i) representing the 3D position of the skeleton point indicated by k_i, (dx_i, dy_i, dz_i) the growing direction, and r_i the radius at that point. Here, the position and direction variables are not independent, since the direction can be obtained by the position and that of its parent node. The variable s_i is the growing step length, i.e. the inter-distance between X_i and its child nodes. The prior $P(S)$ specifies the branch smoothness and growing step etc. Our objective is to find an optimal Bayesian network structure S by maximizing the posterior distribution given by formula 1.

The graphical model is constructed by using bottom-up inference, starting from the root and proceeding progressively with step length s_i at the node indicated by k_i. Here, we pack the variables according to different stages. For example, k_t denotes the indicator set of all the nodes at stage t, and $X_t \triangleq \{X_{it}\}_i^{|k_t|}$. As seen in Figure 1, once the configuration (k_{t-1}, X_{t-1}) at stage $t-1$ are determined, we infer the subsequential configuration (k_t, X_t) at stage t. The links between the nodes at stage t and those at stage $t-1$ reflects the natural growing phenomenon of a tree, i.e. new birth of sub-branches, and growth or termination of original branches. The initial node at the bottom, as the first node of the No.1 branch, i.e. $k_1 = \{1\}$, is given by users: first, manually specify initial growing directions and radius information in the image pairs; then, get the node state in 3D by stereo triangulation. At stage t, the configuration (k_t, X_t) is an MAP estimate obtained by sampling the posterior distribution, given by

$$\begin{aligned} &P(k_t, X_t|Z_t, k_{t-1}, X_{t-1}) \propto \\ &P(Z_t|k_t, X_t)P(k_t, X_t|k_{t-1}, X_{t-1}), \end{aligned} \tag{2}$$

where, $Z_t \triangleq \{Z_{it}\}_i$, and Z_{it} represents the measurement variable of X_{it}. The first term at the right-hand side of formula 2 is the likelihood model, which expresses the probability we would have observed the measurement Z_t given the states (k_t, X_t) at stage t. The observation of different branches are assumed to be independent

$$P(Z_t|k_t, X_t) = \prod_{i=1}^{|k_t|} P(Z_{it}|k_{it}, X_{it}). \tag{3}$$

The second term in formula 2, i.e. the growing prior, restricts the state space at stage t given the previous states at stage $t-1$. As we know, jointly inferring the child node states of the nodes at stage $t-1$ suffers from exponential complexity in the number of the nodes at stage $t-1$. Considering the computation efficiency, we assume the child node states inference of different nodes at stage $t-1$ are independent, i.e.

$$P(k_t, X_t|k_{t-1}, X_{t-1}) = \prod_{i=1}^{|k_{t-1}|} P(k_{t\cdot i}, X_{t\cdot i}|k_{i(t-1)}, X_{i(t-1)}), \tag{4}$$

where, $(k_{t\cdot i}, X_{t\cdot i})$ denotes the child nodes set of $(k_{i(t-1)}, X_{i(t-1)})$. The independent assumption is deduced by the proposed matching algorithm in section 4, which is used to propose child node state candidates.

The growing prior for each node at stage $t-1$ in formula 4 can be decomposed as the distribution of the indicator set $k_{t\cdot i}$ and that of $X_{t\cdot i}$ conditioned on $k_{t\cdot i}$

$$\begin{aligned} &P(k_{t\cdot i}, X_{t\cdot i}|k_{i(t-1)}, X_{i(t-1)}) = \\ &P(X_{t\cdot i}|k_{t\cdot i}, k_{i(t-1)}, X_{i(t-1)})P(k_{t\cdot i}|k_{i(t-1)}, X_{i(t-1)}), \end{aligned} \tag{5}$$

in which, $P(k_{t\cdot i}|k_{i(t-1)}, X_{i(t-1)})$ models which branch is likely to be continued, added, or terminated, and is realized based on a set of operations on a 3D detection set as we will explain in section 4. $P(X_{t\cdot i}|k_{t\cdot i}, k_{i(t-1)}, X_{i(t-1)})$ can be further decomposed as

$$P(X_{t\cdot i}|k_{t\cdot i}, k_{i(t-1)}, X_{i(t-1)}) = P(X_{t,v})P(X_{it}|X_{i(t-1)}), \tag{6}$$

where, $P(X_{t,v})$ is the prior for the branches whose states change, namely being newly added or terminated. $P(X_{it}|X_{i(t-1)})$ is the smoothness prior for the branch continually growing, and $X_{t\cdot i} = \{X_{t,v}, X_{it}\}$.

3.1. Prior for on-growing branches $P(X_{it}|X_{i(t-1)})$

The priors are given according to the specific characteristics of real trees. At stage t, we define priors for the on-growing branch respectively on its position $X_{it,p} \triangleq (x_{it}, y_{it}, z_{it})$, radius r_{it} and step length s_{it}. $X_{it,p}$ is constrained by its parent node $X_{i(t-1)}$ for smoothness of the recovered 3D model

$$X_{it,p} \sim \Phi(X_{it,p}|X_{i(t-1),p}) \prod_{\{i,j\}} \psi(X_{it,p}, X_{jt,p}). \tag{7}$$

Here, $\Phi(X_{it,p}|X_{i(t-1),p})$ is defined as a Gaussian distribution on a spherical surface with center $(x_{i(t-1)}, y_{i(t-1)}, z_{i(t-1)})$ and radius $r_{i(t-1)}$, namely, a truncated Gaussian distribution defined on $[-\pi, \pi]$

$$X_{it,p} \sim N(\mu_{gp}, \sigma_{gp}^2), \tag{8}$$

in which, μ_{gp} is 0 degree, showing the position on the spherical surface indicated by the direction of $X_{i(t-1)}$. σ_{gp} is given by observing the real target trees.

$\psi(X_{it}, X_{jt})$ models the interaction among the adjacent nodes i and j, and is defined as

$$\psi(X_{it}, X_{jt}) = e^{-g(X_{it}, X_{jt})}. \tag{9}$$

$g(X_{it}, X_{jt})$ is a penalty function, given by

$$g(X_{it}, X_{jt}) = \max(2(r_{it} + r_{jt}) - d_{ij}, 0). \tag{10}$$

Here, d_{ij} is the Euclidean distance of the adjacent nodes i and j.

The prior constraining the radius is also Gaussian with mean $r_{i(t-1)}$, and variance σ_{gr}^2 as

$$r_{it} \sim N(\mu_{gr}, \sigma_{gr}^2), \tag{11}$$

in which, $\mu_{gr} = r_{i(t-1)}$, and $\sigma_{gr} = r_{i(t-1)}/\gamma$. γ is also given by the target trees.

The distribution of the growing step s_{it} is given by

$$P(s_{it}|r_{it}) \sim N(\mu_{gs}, \sigma_{gs}^2), \tag{12}$$

where, $\mu_{gs} = cr_{it}$. c is a constant. In section 4.1, we will explain how to estimate it.

3.2. Prior for variable branches $P(X_{it,v})$

The prior for new-born branches plays an important role in filtering wrong state candidates. In the paper, the adding prior is modeled in three aspects. They are respectively that 1) the number of the new-born child nodes of a parent node is restricted with a maximum number N_a, by using an exponential model $exp(-\lambda \max(n_a - N_a, 0))$, where, n_a is the added child node number and $\lambda = 0.5$ in the paper; 2) to avoid conflicts with other branches, the interactions among adjacent nodes are modeled as done in section 3.1; 3) the angles for new-born branches are constrained by a Gaussian distribution $N(\mu_a, \sigma_a^2)$, where the mean μ_a and the standard deviation σ_a are also given by observing the modeling targets. The prior for terminating branches is simply assumed to be uniform in the paper.

3.3. Likelihood model $P(Z_{it}|X_{it})$

The likelihood for each node models the observations in the image pair. In the previous formulas, for clarity, the projection parameters are omitted. Here, we explicitly express them in the likelihood distribution

$$P(Z_{it}|k_{it}, X_{it}; M) = \prod_{j=1}^{m} P(Z_{it}|k_{it}, X_{it}; M_j), \qquad (13)$$

where, $m = 2$, representing the number of images, and M_j is the projection matrix corresponding to image j. The appearance model of X_{it} in image j takes the form

$$P(Z_{it}|k_{it}, X_{it}; M_j) \propto e^{-f(t_p, g_e)}, \qquad (14)$$

where,

$$f(t_p, g_e) = \frac{1}{N_p} \sum_{p_i \in p} t_{p_i} + \frac{1}{N_e} \sum_{e_i \in e} (1 - g_{e_i}). \qquad (15)$$

t_{p_i} is the transparency of pixel p_i, and g_{e_i} is the gradient of e_i in the outward direction. The set p (different from the one in subsection 3.1) consists of the pixels lying on the T shape in Figure 2a as the supporting points of the branch segment, and N_p is the number of pixels in set p. The set $e \triangleq \{e_1, e_2\}$ is composed of the two terminals as shown in Figure 2a, and $N_e = 2$.

4. Bottom-up inference

The 3D structure S, in the form of a graphical model, is obtained by a one-pass bottom-up inference progressively. At each stage, we compute the child node states by finding an MAP estimate from the posterior distribution given by formula 2. The MAP estimate is produced by using RJMCMC (Reversible Jump Markov Chain Monte Carlo) [3] due to the following considerations. First, our branch growing problem is a little bit similar with the tracking problem in [5], in which, the authors tracked a variable number of interacting targets and proved the computation efficiency of

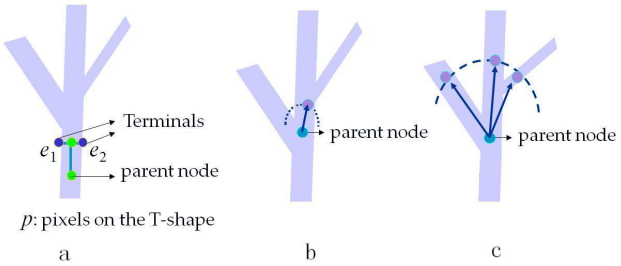

Figure 2. Demonstrations: a. definition of the likelihood model; b and c. the effects of over small and large step length in detecting 2D subbranches.

RJMCMC. Second, in this paper we aim at establishing a general framework of stereo-based inference, in which any independent assumptions can be relaxed and any forms of botanical priors or observations can be freely integrated. Therefore, RJMCMC is adopted due to its ability in sampling complex distributions and its efficiency in traversing spaces of high dimensionality.

As done in [4] [5], to speed up the computation, we utilize a data-driven strategy to propose child node state candidates. As shown in Figure 3, for each node at stage $t-1$, we detect 2D candidates (the squares in Figure 3) in the image pair and store a set of 3D candidates (the plates in Figure 3) computed by all corresponding subbranches in the image pair. To prune the wrong 3D candidates appearing due to ambiguous stereo correspondences, we develop a trace-based stereo matching algorithm. After the matching, the remaining candidates form a detection set k_d. k_d is used to propose node state candidates to accelerate RJMCMC sampling. The reliable matching procedure promotes us to make an assumption of the independent child node states inference of the nodes at stage $t-1$ for consideration of computation efficiency, since the detection set for different nodes at stage $t-1$ are separated after the matching procedure. Therefore, we can determine the configuration (k_t, X_t) by growing, dividing or terminating the nodes at stage $t-1$ one by one.

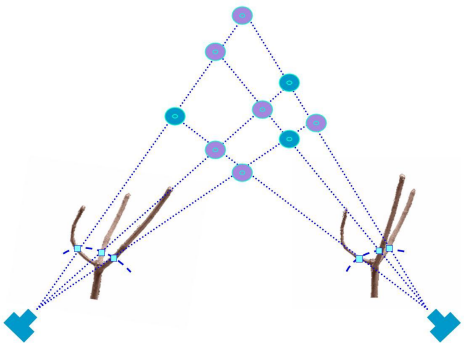

Figure 3. Ambiguous stereo correspondences

For each node $X_{i(t-1)}$ at stage $t-1$, we perform the Metropolis-Hastings algorithm to find its child nodes $(k_{t\cdot i}, X_{t\cdot i})$ at stage t. Given a configuration $(k_{t\cdot i}, X_{t\cdot i})$, we select a move type from predefined dynamics, with probability q_a, q_d, q_g, q_s, q_l respectively for adding, deleting, growing, terminating, and location diffusion ($q_a+q_d+q_g+q_s+q_l=1$), and then propose a new state $(k'_{t\cdot i}, X'_{t\cdot i})$ with acceptance ratio

$$A(k'_{t\cdot i}, X'_{t\cdot i}; k_{t\cdot i}, X_{t\cdot i}) = \min(1, \alpha) \tag{16}$$

in which, α is given by

$$\alpha = \frac{P(k'_{t\cdot i}, X'_{t\cdot i}|Z_{t\cdot i})}{P(k_{t\cdot i}, X_{t\cdot i}|Z_{t\cdot i})} \frac{Q(k_{t\cdot i}, X_{t\cdot i}; k'_{t\cdot i}, X'_{t\cdot i})}{Q(k'_{t\cdot i}, X'_{t\cdot i}; k_{t\cdot i}, X_{t\cdot i})}. \tag{17}$$

If $\alpha > 1$, the state is certainly accepted. Otherwise, it is accepted with probability α.

4.1. Detection for data-driven sampling

The detection set used to propose child node states for a node, whose configuration is $(k_{i(t-1)}, X_{i(t-1)})$, in stage $t-1$ is formed as follows: first, we define a part of spherical surface with center $(x_{i(t-1)}, y_{i(t-1)}, z_{i(t-1)})$, radius $r_{i(t-1)}$ and a pre-defined detection scope in the form of inter-angle with $(dx_{i(t-1)}, dy_{i(t-1)}, dz_{i(t-1)})$. Then, we project the maximum circle corresponding to each image in the spherical surface part to each image (the dash arcs in Figure 3). Third, we find the disconnected arc segments as 2D detections (the squares in Figure 3); Fourth, the 3D detection set k_d is computed using those matched 2D detections. The stereo matching algorithm will be introduced in section 4.2.

It should be noticed that for effective 2D detections, a proper step length s_{it} is essential. A small step cannot generate a valid arc for identifying the 2D candidates (the left branch in Figure 2b is missed) and takes a risk of losing branches due to asynchronous detection (as we will explain in section 5). While a large one will bring deviations from correct junctions, like the right subbranch in Figure 2c. Besides, a large step can not produce smooth 3D models. To handle the above problems, we should adapt step variable to the global subtree structures. However, the adaptive step distribution is hard to model in our current framework since branches are constructed stage by stage without any information about their following structures. It will be considered in future work. Here, we simply set the step variable to being conditioned on the radius variable as given in formula 12, in which the constant c is estimated by balancing the above factors.

4.2. Trace-based matching

As we know, stereo matching is a difficult task especially for large-disparity and wide-baseline image pairs. As shown in Figure 3, the 2D detections in the image pair form nine 3D candidates. It is hard to find the true three using traditional color-based methods, such as NCC (Normalized Cross Correlation) and SSD (Sum of Squared Difference) etc., due to the similar appearances among branches. In view of the branching structures, we propose to use traces for matching. Given a 3D candidate, we grow it step by step along the proposed roads computed by 2D detections in the image pair as done in section 4.1. The detections near junctions will provide multiple 3D roads, in which the one making the trace of the candidate straightest is preferred first for efficiency, since branches are generally growing straightly. The extension along a 3D road terminates until there is no detection in one or two images. The first kind of termination means the 3D candidate is correct. In the second termination, another road is selected and traversed as described above. The road searching iteration ends until the 3D candidate is proved to be correct or all the possible roads are traversed. Figure 4 shows some matching results, in which lines with the same colors (and the same number) in the two images are the trace projections of the same 3D candidate.

As other matching criterions, the trace-based matching algorithm does not always work well. Although not a common phenomenon, sometimes two 2D traces of a false 3D candidate in the image pair may have the very same length. In this case, they will be matched. For such cases, the growing priors in section 3 take effect to filter unnatural 3D candidates. As we have noted, the potential limitation of the matching algorithm is its efficiency, especially for those bent branches and those confusedly interacting with other branches. As shown in Figure 4, the No.2 trace ever tried a wrong 3D road and then found the right one.

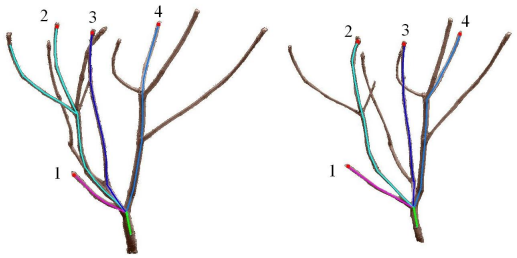

Figure 4. Trace-based matching of four subbranches as divisions of the branch at the bottommost

4.3. Five Dynamics

Since the branches at stage $t-1$ are assumed to be independent, we determine their states at stage t one by one. Given a node $X_{i(t-1)}$, its child nodes, denoted as $(k_{t\cdot i}, X_{t\cdot i})$ at stage t are inferred by using Markov chain jumps to traverse the solution space in different dimensions. Five dynamics are designed to drive five types of proposals, namely continuous growth, division and its reverse jump, terminate and its reverse jump.

4.3.1 Dynamic 1. Location diffusion

This dynamic aims to find the best location for the node X_{it}, as a continuation of node $X_{i(t-1)}$. For efficiency, the diffusion is performed by data-driven. A 3D candidate X_d is chosen from the set $k_{d\cdot i}$ with uniform probability. Then, a new state $X_{it}^{'}$ is proposed from a Gaussian distribution defined on a spherical surface with center (x_d, y_d, z_d) and radius r_d. The new state is accepted with

$$\alpha_l = \frac{P(Z_{it}|X_{it}^{'})P(X_{it}^{'}|X_{i(t-1)})}{P(Z_{it}|X_{it})P(X_{it}|X_{i(t-1)})}\frac{1}{Q(X_{it}^{'};k_{d\cdot i})}. \tag{18}$$

Here,

$$Q(X_{it}^{'};k_{d\cdot i}) = N(\mu_l, \sigma_l^2), \tag{19}$$

where, μ_l is 0 degree, showing the position on the spherical surface indicated by (dx_d, dy_d, dz_d), and σ_l is given by $\arctan((r_d + \sigma_c)/r_{i(t-1)})$. σ_c is a spatial length near X_d, which corresponds to a pixel in images under perspective projection. Apparently, for short-baseline image pairs, σ_c will be very large, thereby resulting in large searching space for location diffusion and risking an optimal location.

4.3.2 Dynamic 2. Adding branches

This dynamic is used to give birth to subbranches. First, uniformly select a candidate a from $k_{d\cdot i} \setminus k_{t\cdot i}$, which means branches in $k_{d\cdot i}$ while not in $k_{t\cdot i}$. Then, add a to $k_{t\cdot i}$ with

$$\alpha_a = P(Z_a|X_a)\frac{P(k_{t\cdot i}^{'}, X_{t\cdot i}^{'}|k_{i(t-1)}, X_{i(t-1)})}{P(k_{t\cdot i}, X_{t\cdot i}|k_{i(t-1)}, X_{i(t-1)})}\frac{q_d|k_{d\cdot i} \setminus k_{t\cdot i}|}{q_a|k_{d\cdot i} \cap k_{t\cdot i}^{'}|}, \tag{20}$$

where, $|k_{d\cdot i} \cap k_{t\cdot i}^{'}|$ denotes the intersection of $k_{d\cdot i}$ and $k_{t\cdot i}^{'}$.

4.3.3 Dynamic 3. Deleting branches

This is the reversible jump of the adding operation. First, select a branch d from the intersection of $k_{d\cdot i}$ and $k_{t\cdot i}$, i.e. $|k_{d\cdot i} \cap k_{t\cdot i}|$ with uniform distribution. Then, delete d from $k_{t\cdot i}$ with

$$\alpha_d = \frac{1}{P(Z_d|X_d)}\frac{P(k_{t\cdot i}^{'}, X_{t\cdot i}^{'}|k_{i(t-1)}, X_{i(t-1)})}{P(k_{t\cdot i}, X_{t\cdot i}|k_{i(t-1)}, X_{i(t-1)})}\frac{q_a|k_{d\cdot i} \cap k_{t\cdot i}|}{q_d|k_{d\cdot i} \setminus k_{t\cdot i}^{'}|}. \tag{21}$$

4.3.4 Dynamic 4. Growing the branch

The following two reverse dynamics are used to terminate the branch with tip node $(k_{i(t-1)}, X_{i(t-1)})$. If the growth is terminated, i.e. X_{it} is invalid, we regrow the branch by sampling a distribution $Q(X_{it}^{'};k_{d\cdot i})$ as done in Dynamic 1 and add the sample to $k_{t\cdot i}$ with

$$\alpha_g = P(Z_{it}|X_{it}^{'})\frac{P(k_{t\cdot i}^{'}, X_{t\cdot i}^{'}|k_{i(t-1)}, X_{i(t-1)})}{P(k_{t\cdot i}, X_{t\cdot i}|k_{i(t-1)}, X_{i(t-1)})}\frac{q_s}{q_g Q(X_{it}^{'};k_{d\cdot i})}. \tag{22}$$

4.3.5 Dynamic 5. Terminating the branch

To terminate the branch, we delete it from $k_{t\cdot i}$ with

$$\alpha_s = \frac{1}{P(Z_{it}|X_{it})}\frac{P(k_{t\cdot i}^{'}, X_{t\cdot i}^{'}|k_{i(t-1)}, X_{i(t-1)})}{P(k_{t\cdot i}, X_{t\cdot i}|k_{i(t-1)}, X_{i(t-1)})}\frac{q_g Q(X_{it};k_{d\cdot i})}{q_s}. \tag{23}$$

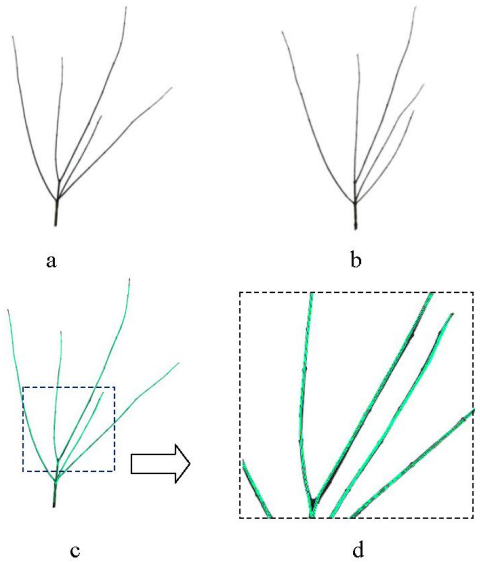

Figure 5. Subtree 1: a and b. the input images; c. the back projections of the 3D reconstructed model on one input image; d. close-up of the back projection

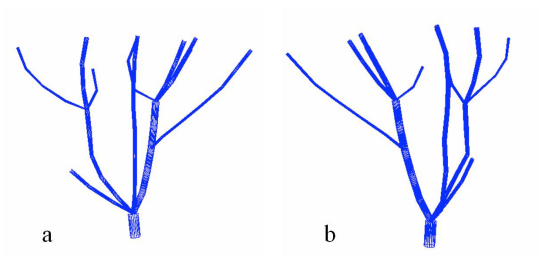

Figure 6. Subtree 2: the reconstructed model with the input images in Figure 4; a. the 3D model observed from an original viewpoint; b. a novel viewpoint

4.4. Post-processing

It should be noted that the graphical model is constructed stage by stage, during which only conflicts at the same stage are considered. To avoid branch conflicts at different stages, a post-processing step is adopted to prune constructed 3D tree models by taking subbranch as a unit. Next, for practical applications, we convert the skeleton forms into mesh models simply by triangulating the surface vertices, which

are sampled from the shapes determined by two consecutive skeleton points and their corresponding radii.

5. Results

We tested the proposed algorithm in four different forms of subtrees (as shown in Figure 5, Figure 6, Figure 7, and Figure 8). To capture their photos, the subtrees can be first cut from large trees and then photographed indoor (as we did for subtree 1). This way is more reasonable for seriously occluded trees and has controllable shooting conditions. Alternatively, they can be captured directly as part of large trees (as we did for subtree 3 and 4). The captured images are first segmented to remove the background. Since the proposed method can discriminate intercrosses, there is no need to treat all unusable branches as background during the segmentation.

Figure 5 shows a subtree cut from a large tree as a typical pattern and pictured indoor with inter-angle about 45^0. The subtree is more than 40cm high but the radii of its twigs are less than 2mm. It is hard to match the line-shape twigs in the wide-baseline image pair (Figure 5a and b) even using our eyes. The proposed approach successfully constructs a 3D model, simply requiring us to specify an initial node in 2D forms in the images. From the back projections of the reconstructed model (Figure 5c and d), we can see that the model is consistent with the observations in the input images.

Another highlight of the method is its ability in generating 3D models with semantic structure, which is unavailable using traditional stereo-based method [1]. Figure 7 shows a type of subtree captured directly outdoor, with inter-angle around 40^0. Once the starting node is indicated in the image(as shown in Figure 7a and b), the proposed approach construct the model branch by branch. Each branch own its unique label(as shown in Figure 7f). This structure with semantic meaning provides great convenience for further editing, such as animation.

The wide-baseline stereo can generate reliable 3D models. As shown in Figure 8, the subtree is first captured in three different viewpoints(Figure 8a, b and e). The first two images with inter-angle around 24^0 are adopted to construct the 3D model. Then we back-projected to the third image(Figure 8e), with inter-angle around 35^0 to the second image. Figure 8g shows a close-up to the back projection part in Figure 8f. From this picture, we can see that the 3D model is consistent with the new viewpoint image.

Figure 6 gives an indoor plant without leaves, captured with about 30^0 inter-angle. Such branches take more chances of having almost the same length. Therefore, few branches are wrongly matched and added as real ones. However, they do not influence the natural appearance of the 3D model as observed from Figure 6a and Figure 6b, an original viewpoint and a new viewpoint result respectively.

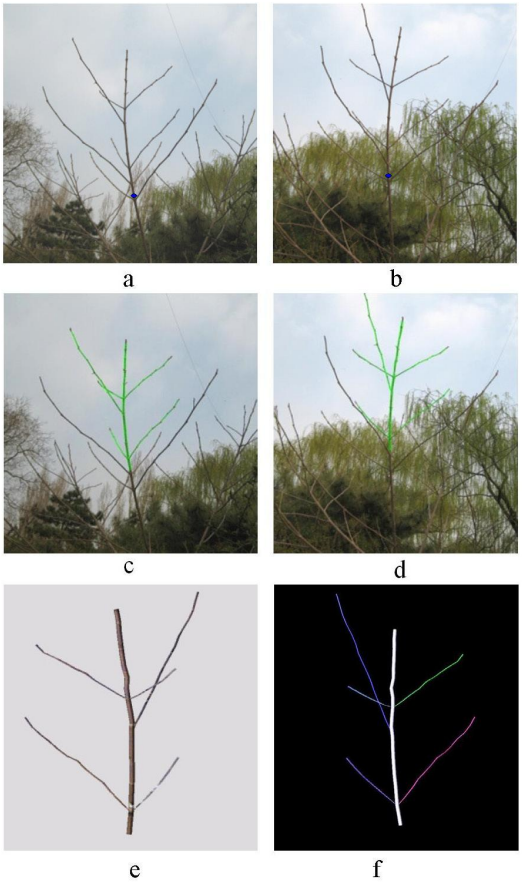

Figure 7. Subtree 3: a and b. original images; b and c. back projection onto the original images; c. the reconstructed textured model; d. the model with branch-unit materials.

The reason for this is that they passed the filtering of our adding priors (section 3.2). This example demonstrates that our approach can reconstruct not only subtree patterns but also branches of simple structured whole trees. Since our algorithm aims at recovering natural branches based on stereo matching, those branches being large-partially occluded in any one of the image pair are certainly omitted.

6. Conclusion

The paper described a framework to conveniently construct branching structures from wide-baseline image pairs. The models, in the form of Bayesian networks were constructed by progressively inferring child node states stage by stage. At each stage, an MAP estimate of the child node states was computed by sampling a posterior distribution, which integrates observations in different images and multiple pre-defined priors. A trace-based matching algorithm was introduced to propose child node state candidates for computation efficiency of the sampling. The prior parameters involved can be easily given by observing the real target

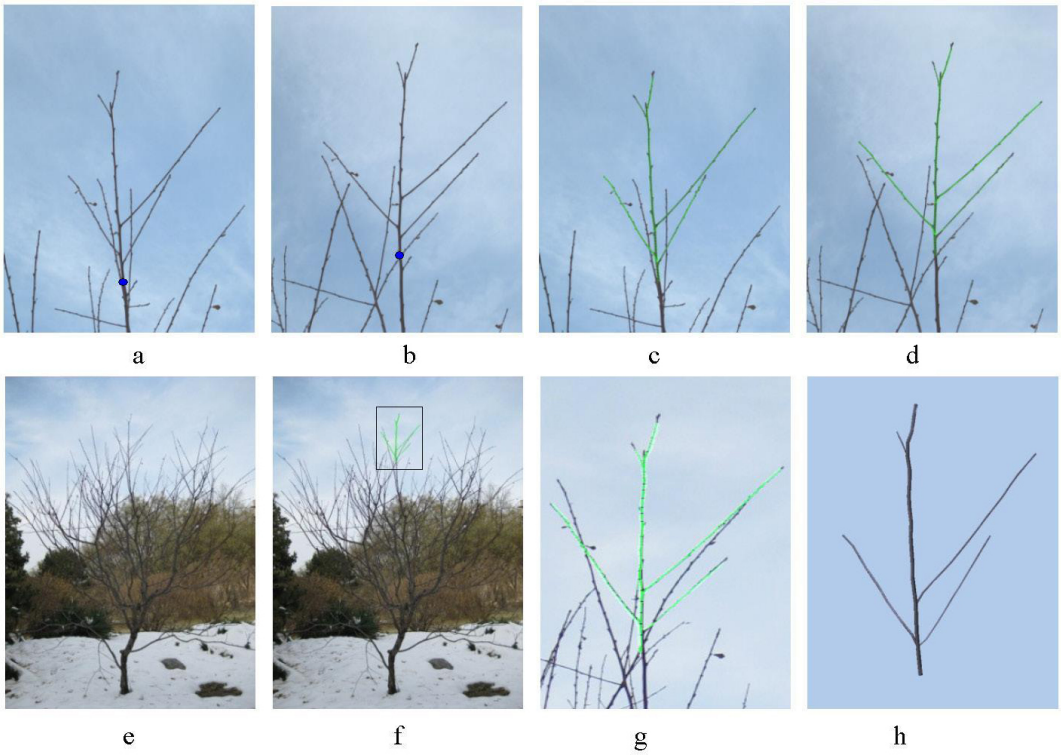

Figure 8. Subtree 4: a and b. the used image pair; h. the textured 3D model; c and d. back projection of the 3D model onto image a and b; e. another image not involving in the reconstruction; f: back projection of the 3D model onto image e; g. close-up of the projected part in image f.

trees. Besides, it is straightforward to integrate any knowledge, such as the spatial distribution prior defined in [4], into our framework. Future work will focus on the adapted step size as we mentioned on section 4.1 and large tree reconstruction according to its images and its subtree patterns.

7. Acknowledgement

Great thanks to Yangfeng Ji and Tingting Jiang for revising the paper, Yang Liu for texturing the branches, and Lei Zhang for his work on calibration and segmentation. This research was supported in part by NSFC (No. 60872077), Doctoral Fund of Ministry of Education of China (No. KEJ200900029), and NHTRDP 863 Grant (No. 2007AA01Z315).

References

[1] A. Barbu and S.-C. Zhu. Incorporating visual knowledge representation in stereo reconstruction. In *10th IEEE International Conference on Computer Vision*, pages 572–579, 2005. 7

[2] X. Chen, B. Neubert, Y. Q. Xu, O. Deussen, and S. B. Kang. Sketch-based tree modeling using markov random field. *ACM Transactions on Graphics*, 27(5):109:1–9, 2008. 1

[3] P. J. Green. Reversible jump Markov Chain Monte Carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995. 4

[4] F. Han and S. C. Zhu. Bayesian reconstruction of 3D shapes and scenes from a single image. In *Proceedings of IEEE Workshop on Higher-Level Knowledge in 3D modeling and Motion Analysis*, pages 12–20, 2003. 1, 2, 4, 8

[5] A. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1918, 2005. 4

[6] W. Ma, B. Xiang, X. Zhang, and H. Zha. Decomposition of branching volume data by tip detection. In *IEEE International Conference on Image Processing*, pages 1948–1951, 2008. 1

[7] B. Neubert, T. Franken, and O. Deussen. Approximate image-based tree modeling using particle flows. *ACM Transactions on Graphics*, 26(3):88:1–8, 2007. 2

[8] P. Tan, G. Zeng, J. D. Wang, S. B. Wang, and L. Quan. Image-based tree modeling. *ACM Transactions on Graphics*, 26(3):87:1–7, 2007. 1, 2

[9] C. H. Teng, Y. S. Chen, and W. H. Hsu. Constructing a 3D trunk model from two images. *Graphical Models*, 69(1):33–56, 2007. 2

[10] H. Xu, N. Gossett, and B. Chen. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics*, 26(4):19:1–13, 2007. 1, 2