

# AN INTERACTIVE METHOD FOR CURVE EXTRACTION

Ge Guo\*, Luoqi Liu\*, Zhebin Zhang\*, Yizhou Wang<sup>+</sup>, Wen Gao<sup>+</sup>

\*Inst. of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China

\*Graduate School, Chinese Academy of Sciences, Beijing, 100039, China

<sup>+</sup>Nat'l Engineering Lab for Video Technology,

<sup>+</sup>Key Lab. of Machine Perception (MoE), Sch'l of EECS, Peking University, Beijing, 100871, China

{gguo, lqliu, zbzhang}@jdl.ac.cn, {Yizhou.Wang, wgao}@pku.edu.cn

## ABSTRACT

We introduce a curve process framework to solve the challenging problem of curve extraction from “non-traceable” curve groups. We propose a comprehensive curve model, which consists of the geometric, photometric and topological sub-models. Two typical categories of the non-traceable curve groups are considered. First, for the interlaced curves with complex structures, we show how to use the proposed curve model especially the topological sub-model to extract curves from the group. Second, for the non-interlaced but over-dense or faint curves we leverage the curve group pattern priors in addition, and extract the whole pattern in a global optimization. Applications and experiments demonstrate the competence of our models and methods.

**Index Terms**— curve model, curve extraction

## 1. INTRODUCTION

Curves are ubiquitous in images and they play an important role in many computer vision applications such as image segmentation and parsing, shape perception and reconstruction, object detection and recognition. Without explicitly modeling curves and 1D curve objects, the performance of many vision-based methods suffer from degradation. As mentioned in [1], conventional region-based segmentation methods tend to produce “degenerated cluttered results” for thin-and-long regions. Also the state-of-the-art image segmentation methods based on the Graph-cut, has the “shrinking bias” of missing thin-and-long structures [2]. Thus the success of higher-level vision tasks will heavily depends on the modeling of curves and the performance of curve detection / localization.

In the literature a great deal of related work on curve modeling focuses on the geometric description of curves, such as the classical spline models, and the statistical curve models (e.g. the Active Shape Model [3] and the Elastic model [4]). Nevertheless, only geometrical information is insufficient especially in case of low image contrasts and appearance variations. Thus the Active Appearance Model [5], and Tu and

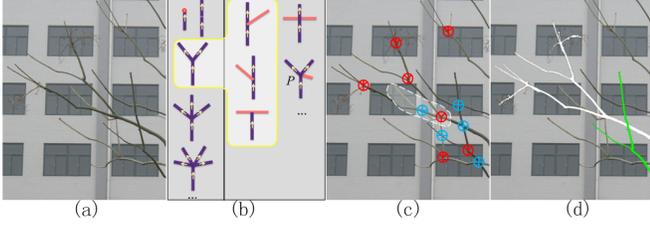
Zhu [1] incorporate both the geometric and photometric representations. However, in [1] it treats each subbranch of trees as a different curve, without modeling the underlying curve topologies. The resulted independent curves lose their semantic meaning of being a branch structure. We argue that the underlying model of topologies such as furcations should be part of a curve process. Another related work in [6] uses a predefined library of subtrees with different topologies to synthesize the invisible part of tree structures in graphics. Whereas our focus here is how to reliably extract curves from images.

In addition, many contour-based methods ([7][8], etc.) have explored the problems on contour continuities, junction detections. However they serves for the region-based interpretations. We argue that an advanced curve modeling are necessary for image interpretation besides region boundaries.

In this paper we provide a curve process framework, based on a comprehensive curve modeling integrating the geometric, photometric and topological properties of the curves (Fig.2). We combine the three aspects into a Markov tree process, and model their state transitions along the curve.

The major motivation is to solve the challenging problem **curve extraction from non-traceable curve groups**, which is seldom addressed in the curve detection literature. With simple user interactions, we extract curves of two typical cases that are non-traceable. (i) It is the *interlaced curve group pattern* with complex structures. Typically as the tree branches (Fig.1), the interlacement makes the conventional curve tracing methods confused at junctions – whether the junction is a branching point, or an intersection where several curves overlap after projection. We adopt the ancestral sampling method under the Bayesian paradigm. The underlying junction types and curve structures are inferred based on the curve model especially the topological modeling. (ii) The curves are not interlaced, but over-dense, too faint or blurred, so that they cannot be traced individually and reliably. We call them *non-interlaced curve group patterns*, e.g. the parallel, radial patterns (Fig. 3,5,6). we leverage prior knowledge on the whole pattern. A Linear Programming (LP) approach is adopted to approximate the solution of this NP hard problem with efficiency. Some useful applications – smart tree branch trimming, stroke extraction & zooming line drawings and 3D

Thanks to the research grant NSFC-60872077, the Doctoral Fund of Ministry of Education of China KEJ200900029, and the National Science Foundation for Young Scientists of China 60803069.



**Fig. 1.** An example of *interlaced curve group*. (a) The input image. (b) Illustration of junction types (the left collum) and possible intersection configurations (the others). The highlighted part shows the ambiguities of a 3-degree intersection. (c) The inferred junction types. (d) Two extracted curves.

reconstruction, demonstrate the advantages of our method in dealing with these challenging cases.

In the following we introduce the curve modeling in Section 2, and the curve extraction algorithms for the two curve group categories described above in Section 3. Section 4 demonstrates the applications and experimental results. Finally Section 5 summarizes this paper.

## 2. MODELING SINGLE CURVES

As shown in Fig.2, a curve is represented by a series of control points  $C = (c_1, c_2, \dots, c_n)$  on its middle axis. Each point  $c_i$  is associated with the geometric, photometric and topological attributes,  $c_i = (\mathcal{G}_i, \mathcal{A}_i, \mathcal{T}_i)$ . According to the connection degree  $k_i$ , the control points are divided into *curve segment control points* ( $k_i < 3$ ) and *junction points* ( $k_i \geq 3$ ). For *curve segment control points*, the attributes are extracted from a local patch/curve segment at  $c_i$ . The patch length is about twice the curve profile width.  $\mathcal{G}_i = (\mathbf{x}_i, w_i, \mathbf{sc}_i)$ , describing the point position, the curve segment profile width, and the shape context feature [9].  $\mathcal{A}_i = \mathbf{h}_i$ , where  $\mathbf{h}_i$  denotes the color/intensity histogram over the patch, or the average intensity profile across the curve inside the patch.  $\mathcal{T}_i = (k_i, \mathcal{N}_i, \Theta_i)$ , where  $\mathcal{N}_i$  is the set of neighbor control points of  $c_i$ ,  $\Theta_i$  denotes the angles between neighbor segments. The *junction points*  $\mathcal{J}$  inherit their ancestors' photometric and geometric properties as their own. The topological attributes are captured as that of the *curve segment points*.

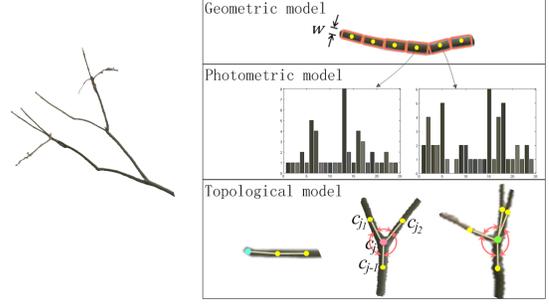
A general curve with branches is modeled by a Markov Tree. If a curve does not furcate, the model reduces to a Markov Chain model. Let  $c_{j-1}$  be the direct ancestor of  $c_j$ . The other neighboring points of  $c_j$  are the direct successors  $\mathcal{S}_j = \{c_{j_1}, \dots, c_{j_{k_j-1}}\}$ , we have

$$p(C) = p(c_1)p(c_2|c_1) \cdot \prod_{c_i \in C \setminus \mathcal{I}} p(c_i|c_{i-1}, c_{i-2}) \cdot \prod_{c_j \in \mathcal{J}} p(\mathcal{S}_j|c_j) \quad (1)$$

where  $\mathcal{I} = \cup_{j \in \mathcal{J}} \mathcal{S}_j$ . For the non-branching curve parts,

$$p(c_i|c_{i-1}, c_{i-2}) = p(k_i|k_{i-1})p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{x}_{i-2}) \cdot p(w_i|w_{i-1}, w_{i-2})p(\mathbf{h}_i|\mathbf{h}_{i-1}) \quad (2)$$

where  $p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{x}_{i-2})p(w_i|w_{i-1}, w_{i-2})$  is the *geometric sub-model*, being the second order linear regressions



**Fig. 2.** Curve modeling.

with Gaussian perturbations. The *photometric sub-model*  $p(\mathbf{h}_i|\mathbf{h}_{i-1}) = G(\mathbf{h}_i; \mathbf{h}_{i-1}, \Sigma_h)$  ( $G$  denotes the Gaussian models), constraining the appearance coherence. The *topological sub-model* is modeled by a transition matrix.  $p(k_j|k_{j-1})$  is the transition probability.  $p(c_2|c_1)$  is the first order Markov Chain, of which the sub-models are similar to the second order ones.  $p(c_1)$  is assumed to be uniform and initialized by user interactions.  $\mathbf{sc}$  is not introduced here but explored in the curve group patterns (Section 3.2). For the junction model,

$$p(\mathcal{S}_j|c_j) = p(\Theta_j) \cdot \prod_{m=1}^{k_j-1} p(w_{j_m}|w_j) \cdot \prod_{m=1}^{k_j-1} p(\mathbf{h}_{j_m}|\mathbf{h}_j) \quad (3)$$

where the three terms are the *topological*, *geometric*, and *photometric sub-model* for a junction respectively.  $p(w_{j_m}|w_j) = G(w_{j_m}; w_j - \delta_w, \sigma_w^2)$ ,  $p(\mathbf{h}_{j_m}|\mathbf{h}_j) = G(\mathbf{h}_{j_m}; \mathbf{h}_j, \Sigma_h)$ ,  $p(\Theta_j)$  is the angle distribution modeled by a truncated Gaussian.

## 3. CURVE EXTRACTION FROM NON-TRACEABLE CURVE GROUPS

In this section we provide the curve extraction framework of the two typical curve group patterns – the *interlaced curve groups* and the *non-interlaced curve group patterns*. Simple interactions are used to provide initialization information and relieve the computational burden.

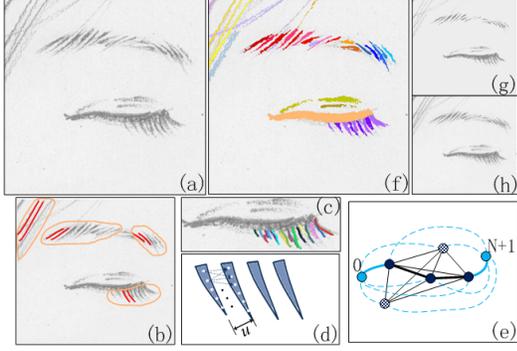
### 3.1. The interlaced curve groups

To extract curves such as the branches in Fig.1 and 4, we assume each curve in the group is independently distributed. Due to the interlacement and various junction configurations, the global method such as in [1] will be hard to design, and of rather high computational cost. Instead we adopt a greedy method based on the *ancestral sampling* for efficiency. Moreover, the curve structure is recovered based our comprehensive curve model including the topological modeling.

According to the Bayesian rule  $p(C|\mathbf{I}) = p(C)p(\mathbf{I}|C)$ ,

$$p(\mathbf{I}|C) = \prod_{c_i \in C} p(\mathbf{I}_{c_i}|c_i) = \prod_{c_i \in C} G(\mathbf{I}_{c_i} - \mathbf{J}_{c_i}; \mathbf{0}, \sigma_I^2), \quad (4)$$

where  $\mathbf{I}, \mathbf{J}$  is the local original image patch and the synthesized patch respectively. The sampling starts from an end point  $c_1$  (initialized by the user) to its descenders. At each step  $t$ , the state of  $c_t$  is selected from a library of candidate



**Fig. 3.** Stroke extraction & zooming. (a) The original image. (b) User interactions: example strokes (red lines), pattern ranges (orange contours). (c) The candidate curves. (d) Curve matching. (e) The graph model for all the candidate curves (the deep blue nodes – curves in the pattern, the dotted nodes – spurious curves, 0 – virtual start,  $N + 1$  – virtual end). (f) The stroke extraction result. Four patterned groups, each of similar colors, are detected in the eyelashes, eyebrows and hairs. The zooming-out results by our method in (g) and down-sampling after smoothing in (h).

set as illustrated in Fig.1.(b). We compute each candidate configuration’s posterior probability,  $p(c_t | \mathbf{I}_{c_t}, c_{t-1}, c_{t-2}) \propto p(\mathbf{I}_{c_t} | c_t) p(c_t | c_{t-1}, c_{t-2})$ , and draw a sample from them. (Points of different topological attributes will be differentiated, e.g. for  $P$  in Fig.1.(b), if the Y-junction model fits the data much better, the extra curve segment is interpreted as another curve. The sub-branches of a curve can overlap each other as in the highlighted region of Fig.1.(c).) When a non-branching control point is sampled, the curve process continues ahead. Otherwise if a branching point, we sample each sub-branch along its direction as an independent thread. It stops if a termination is sampled. Once a whole curve  $\hat{C}$  is sampled, we evaluate it by  $p(\hat{C} | \mathbf{I}_{\hat{C}})$ . If it is above a predefined threshold  $\epsilon$  the sample is kept, and otherwise discarded. This process continues until no new curve can be extracted.

### 3.2. The non-interlaced curve group pattern

We consider the group of similar curves that are orderly distributed in the image space without interlacement, but form perceptually salient patterns. We discuss the parallel and radial pattern here, in which curves are organized in a chain model. However, there are always spurious curves detected (Fig.3.(c)), which break the real neighboring relation in-between the patterned curves. So a fully-connected graph is built, of which each node represents a candidate curve (Fig.3.(e)). Let  $x_{ij} \in \{0, 1\}$  associate to each graph edge.  $x_{ij} = 1$  indicates  $C_i$  and  $C_j$  are true neighbors in the pattern; otherwise  $x_{ij} = 0$ . Now the problem is formulated as a constrained linear optimization:

$$\begin{aligned} \min \sum_{(i,j)} (E(C_i, C_j) + \frac{1}{2}(E(C_i | \mathbf{I}) + E(C_j | \mathbf{I})) - \lambda) x_{ij} \quad (5) \\ \text{s.t. } \begin{cases} 0 \leq \sum_{i \neq k} x_{ik} = \sum_{j \neq k} x_{kj} \leq 1 (1 \leq k \leq N); \\ \sum_{j=1}^N x_{0j} = 1; \quad \sum_{j=1}^N x_{jN+1} = 1; \quad x_{ij} \in \{0, 1\}. \end{cases} \end{aligned}$$

where the constraints ensure a chain model to be found;  $\lambda$  is positive encouraging more patterned curves. The weights on the links to the virtual start and end nodes of the chain (“0” and “ $N+1$ ” respectively) are zeros. The single curve potential  $E(C_i | \mathbf{I}) = -\log p(C_i | \mathbf{I})$ . The edge potential  $E(C_i, C_j) = E_D(C_i, C_j) + E_M(C_i, C_j)$ , measured by the spatial distribution and similarity of the two curves. By using any LP solver for Eq.(5), the patterned curves will be extracted from spurious ones, and simultaneously arranged in order.

$$E_D(C_i, C_j) = \frac{1}{2\sigma_u^2} \sum_{m(s)} (\|\mathbf{x}_{m(s)} - \mathbf{x}_{m'(s)}\| - u(s))^2 \quad (6)$$

where  $u(s) = \alpha s + \beta$  is a distance function with respect to the normalized arc length  $s \in [0, 1]$ .  $\alpha = 0$  for parallel patterns,  $\alpha \neq 0$  for radial patterns.  $m(s)$ ,  $m'(s)$  denote the control point index at  $s$  on  $C_i$  and  $C_j$  respectively.  $\sigma_u^2$  is the variance.

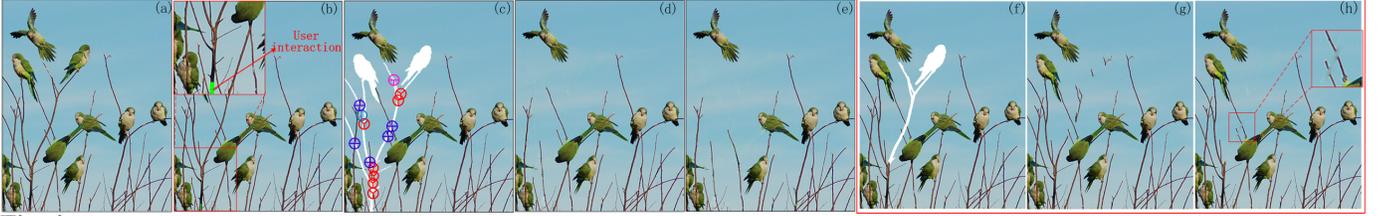
The similarity potential  $E_M(C_i, C_j)$  is measured by the matching cost of the two sets of curve control points (Fig.3.(d)). We adopt the algorithm proposed in [10] to compute the curve matching cost, utilizing the geometric and photometric attributes at control points. The feature matching cost ( $c$  in [10]) is computed as  $d(c_m, c_{m'}) = \lambda_1 \|w_m - w_{m'}\| + \lambda_2 \|s c_m - s c_{m'}\| + \lambda_3 \|\mathbf{h}_m - \mathbf{h}_{m'}\|$ ,  $c_m \in C_i$ ,  $c_{m'} \in C_j$ . And the spatial matching cost ( $g$  in [10]) is designed similarly.

User interactions are illustrated in Fig.3.(b), including designation of the pattern type (parallel or radial). The interactions help to (1) learn the single curve model from the example curve; (2) acquire candidate curves by sampling along the directions approximately vertical to the example curve; and discard some candidates too dissimilar to the example  $C_{exp}$  in advance according to  $E_M(C_i, C_{exp})$ ; (3) obtain the spatial distribution parameter  $u(s)$  (if two example curves are labeled). The single curve potential  $E(C_i | \mathbf{I})$  can be improved to  $-\log p(C_i | \mathbf{I}) + E_M(C_i, C_{exp})$ .

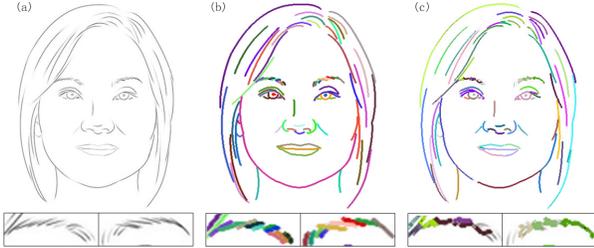
## 4. APPLICATIONS AND EXPERIMENTS

**Smart tree branch trimming.** Fig.4 shows an application of interactively extracting and removing tree branches. The whole piece of a branch with its actual structure topologies is traced out, and the attached objects (such as resting birds, hanging fruits) are together erased. Because some branch segments and junctions may be occluded or missed, some possible connections are proposed at first according to the smoothness, distance of the connection segments, and the reversibility constraint of the connection. The parameters e.g. in the junction modeling, are set  $\delta_w = 0.2w_j, \sigma_w = 2$ ,  $\Sigma_h$  an identity matrix. And symmetric angle distributions are encouraged.

We provide a layered image inpainting method adapted based on [12] to remove branches from the image. We first inpaint the the empty regions only in the background layer, with all the other foreground branches being fixed. Then we inpaint the originally occluded branch regions by the branch layer information. Our inpainting results is better than those of simply applying traditional inpainting methods. It can be



**Fig. 4.** Smart branch trimming. (a) The input image. (b) User interaction to specify a branch to be removed. (c) Inferred junction types and the extracted branch. (d)-(e) The inpainting results after removing branches one by one. The inpainting results for the mask in (f) by using the example-based method[11], and that of the PDE-based method[12] are shown in (g) and (h) respectively.



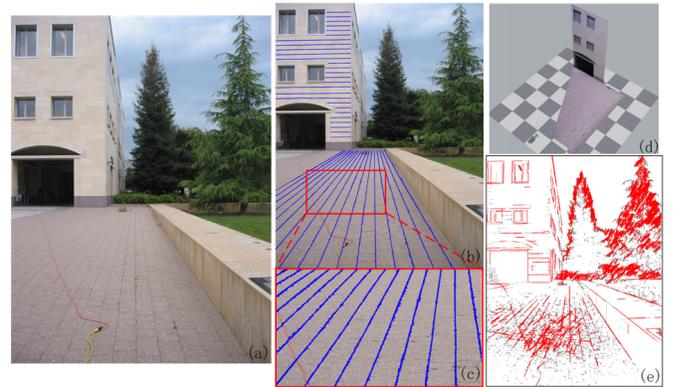
**Fig. 5.** Stroke extraction of a sketch image (a). Results of ours and the traditional edge-tracing are shown in (b), (c) respectively.

seen that the example-based methods intend to recover the erased branch using the similar branch examples remaining in the image; the PDE-based method is hard to recover the broken branches.(Fig. 4.(f)-(h).)

**Applications of parallel/radial curve patterns.** In Fig.3,5, we first extract the patterned curves and then the remaining free curves. We can see that the over-dense strokes can be separated nicely. While using the traditional curve tracing methods such as edge linking/tracing, there are wrongly linked spurious curves and missing ones. Based on the extracted strokes, each one is zoomed out separately as a vectorized stroke (Fig.3.(g)). It leads to aliasing distortions by directly image down-sampling, whereas blurring by smoothing the image before down-sampling (Fig.3.(h)). In Fig.6, we utilize the extracted curve groups (on the assumption that each group lines are equally spaced and parallel in real scenes) to calculate the vanishing lines by [13]. The scene planes are reconstructed with the mapped textures. We can see that the Canny edge detection and Hough line detection find few lines in the low contrast areas and return cluttered results on the texture regions. The parameters are set *e.g.*  $\lambda_1 = \lambda_3 = 0.2, \lambda_2 = 0.6$  for Fig. 3,5, and  $\lambda_1 = \lambda_3 = 0.4, \lambda_2 = 0.2$  for Fig.6. Notice that each term is normalized to the range of [0,1].

## 5. CONCLUSION AND FUTURE WORK

In this paper we have introduced the curve processes based on a comprehensive curve model. We propose solutions to extracting curves from the interlaced, over-dense or faint curve groups. The interesting applications demonstrate the competence of our models and methods. In the future, we shall enrich the curve model, especially to augment the topological model library and curve statistical features. In addition we shall develop automatic methods to learn the free parameters.



**Fig. 6.** 3D reconstruction based on curve group patterns. (a) The input image. (b) Two groups of patterned lines extracted (details in (c)). (d) The reconstructed planes in 3D. (e) Canny edges (in black) and Hough lines (in red).

## References

- [1] Z. Tu and S.-C. Zhu, "Parsing images into regions, curves, and curve groups," *IJCV*, 2006.
- [2] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *CVPR 2008*.
- [3] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active shape models - their training and application," *CVIU*, 1995.
- [4] D.B. Mumford, "Elastica and computer vision," *Algebraic Geometry and Its Applications*, 1994.
- [5] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE TPAMI*, 2001.
- [6] P. Tan, T. Fang, J. Xiao, P. Zhao, and L. Quan, "Single image tree modeling," *ACM Trans. Graph.*, 2008.
- [7] X. Ren, C. Fowlkes, and J. Malik, "Learning probabilistic models for contour completion in natural images," *IJCV*, 2008.
- [8] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," in *CVPR*, 2008.
- [9] S. Belongie, J. Malik, and J. Puzicha, "Shape context: a new descriptor for shape matching and object recognition," *NIPS2000*.
- [10] Hao Jiang and Stella X. Yu, "Linear solution to scale and rotation invariant object matching," in *CVPR*, 2009.
- [11] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *CVPR*, 2003.
- [12] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graphics Tools*, 2004.
- [13] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2000.