

Approximation of Salient Contours in Cluttered Scenes

Rui Huang , Nong Sang, Qiling Tang

*Institute for Pattern Recognition and Artificial Intelligence,
Huazhong University of Science and Technology, Wuhan 430074, PR China*
Ruihuang2008@gmail.com nsang@hust.edu.cn tglinn@sohu.com

Abstract

This paper proposes a new approach to describe the salient contours in cluttered scenes. No need to do the pre-processing, such as edge detection, we directly use a set of random straight line segments, as the intermediate level vision tokens, to approximate the salient contours. This line set is modeled by a stochastic framework, Marked Point Process, in which the point denotes the center of lines, and the marker denotes the orientation and length of lines. Generic Gestalt factors of proximity and collinear continuity are embedded to constraint the geometrical inter-relations between lines. Different data likelihoods are used on synthetic and real images. Optimization is done by simulated annealing using Reversible Jump Markov chain Monte Carlo. Our results not only have a good approximation to the salient contours, also make other post-processing application more robust.

1. Introduction

Finding the perceptually salient contours in a noisy and cluttered scene is central important to computer vision. To make the methods applied to more generic situations, high level prior information should not be included. Most methods in literatures usually work between the low level and the intermediate level vision in a bottom-up fashion, by firstly using the state-of-the-art edge detectors, e.g. Canny, to construct an edge map; secondly approximating the edge map with a set of non pixel-based tokens, such as line fragments [1-4] or small arcs[6]; and thirdly defining a grouping criterion and developing an optimizing algorithm to group the tokens into salient contours [1,4,6].

Under this three-step methodology, advantages and disadvantages coexist. From the view of implementation, each step mentioned above could be implemented independently, and the results of the previous step could be input to the later step. This makes it convenient to choose and evaluate different methods in one step if we fix the methods in other steps, such as in [7], Wang constructs a set of zero-length line fragments by uniformly sampling

the edge detection output, then these fragments are connected by interpolated arcs with G^1 -continuity, which approximating the detected edges are called detected arcs, which filling the gap between fragments are called gap-filling arcs. By this token representation, Wang uses an arc-pruning technique, and proposes a salient convex boundary detection framework. Under this framework, several previous contour grouping methods [1, 5, 6] with different grouping criterion could be used to detect salient convex contours, moreover, Wang compares the performances of these methods.

On the other side, yet the performance among these three steps is not independent, the performance of later step depends on that of previous one. As noted by Stahl and Wang in [4], that contour grouping in the third step may fail when the line segments could not be well detected from the input image in the first step. The state-of-the-art edge detectors usually process images based on pixel, and the detected results still contain uncertainty due to scale choice, image noise, and cluttered background *etc.* The grouping tokens constructed by this edge detection results usually contain many distracting ones, and leave the last grouping algorithm computationally complex, and the grouping results less robust, especially in noisy and cluttered situations.

In this paper, assuming the salient contour could be approximated by chains of small connected straight lines, we directly model the geometrical constraints between them, and their fitness to contours. Detection and grouping of these line tokens are simultaneously performed, and no need to do edge detection any more. The proposed method not only retains salient structures, also reduces many distracting edge elements due to noise, texture and cluttered background. We also put our approximate results to the method in [4] as intermediate input, and show that that method performs better than before.

2. Stochastic model for salient contour approximation

The ideal salient structures in images are usually long smooth simple curves, open or closed. To conveniently

describe these curves, we can approximately decompose them into a set of small straight line segments. Given an image, the problem of finding the salient contours can be considered as finding this approximate segment set. We assume that this segment set representation is, at least, perceptually comparable with the ideal one.

A line segment can be defined as $s_i = (p_i, l_i, \theta_i)$, where $p_i = (x_i, y_i) \in \Lambda \subset \mathbb{R}^2$ is the coordinate of its center, $l_i \in [l_{min}, l_{max}]$ is its length, and $\theta_i \in [0, \pi]$ is its orientation, Λ is the image lattice. The line segments set is $W = (N, \{(p_i, l_i, \theta_i)\}; i = 1, \dots, N)$, N is the number of segment, which is unknown. Hence, the solution space $\Omega = \bigcup_{N=0}^{\infty} \Omega_N$, is a union of many subspace Ω_N with different dimensions.

To model W , we use one kind of Spatial Point Process, which is usually called Marked Point Process (MPP) [8]. It is an object-oriented stochastic framework and adaptive to globally describe the spatial geometrical relations of objects. The number of objects and their relations can be modified during the optimization process. Unlike Markov Random Fields, MPP does not have the restriction that the number of nodes and their relations must be fixed before optimization. In this paper, the modeling object or the ‘‘point’’ is the line segment s_i with p_i denoting the location of its center, and (l_i, θ_i) denoting its marker.

Gibbs Point Process [8,9] is one special form of MPP, which is frequently used in image processing [10]. The non normalized probability density of model is:

$$f(W) \propto \beta^n \exp(-E(W)) = \beta^n \exp(-(E_I(W) + E_D(W))) \quad (1)$$

In this paper, $E_I(W)$ is the interaction energy to describe the geometrical constraints between any two segments in W , and $E_D(W)$ is the data energy to test the fitness of each segment to the contour in image. The approximation to the ideal salient contour can be obtained by minimizing the energy function $E(W)$:

$$W^* = \arg \min \{E_I(W) + E_D(W) - n \log \beta\} \quad (2)$$

β is the intensity of point process, the term $-n \log \beta$ can be considered as the penalty on the total number of segments, and will be merged into $E_I(W)$.

To find the global minimum, W^* , simulated annealing scheme is used, the algorithm iteratively simulates the target function: $f(w, T) = [f(W)]^{\frac{1}{T}}$, while slowly decreasing the temperature T . When $T \rightarrow 0$, the simulating result converges in probability to the global minimum.

2.1. Interaction between segments

The term $E_I(W)$ defined in formula (1) is used to describe the geometrical relations between the segments. Considering for more general situations, we only embed two basic factors of Gestalt law, proximity and collinearity, into our model.

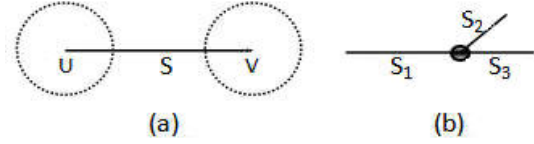


Figure 1. Segments and their connection. (a) Two end-points U, V of segment S with two dashed-line disks as attractive region. (b) s_1, s_2 and s_3 sharing a common end-point.

A segment has two end-points, U and V (see Figure 1 (a)), and its center p . Following the definition of segment types in [10], two segments are considered to be connected if two of their end-points are closer than a constant ϵ , and ϵ usually is small than $l_{min}/2$. One segment is called *free* if it is not connected, *single* if only one of its end-points is connected, and *double* if both of its end-points are connected. Under this setting, the energy function $E_I(W)$ is:

$$E_I(W) = \omega_0 n + \omega_1 n_f + \omega_2 n_s + \omega_a \sum_{\langle s_i, s_j \rangle} I_a(s_i, s_j) + \omega_r \sum_{\langle s_i, s_j \rangle} I_r(s_i, s_j) \quad (3)$$

Where n , n_f , and n_s are the number of total segments, free segments and single segments respectively. ω_0 , ω_1 and ω_2 are the corresponding penalty parameters. $\langle s_i, s_j \rangle$ stands for a pair of connected segments, where $i < j$. If we use C_o^r to denote the circle centered at o with radius r , then for each segment S , its two attractive regions can be defined as, $C_{U_S}^{l_S/4}$ and $C_{V_S}^{l_S/4}$ (see the dashed-line disks in figure 1(a)), while the attractive potential function $I_a(s_i, s_j)$ of two connected segments could be as:

$$I_a(s_i, s_j) = \sum_{o_1=U_{s_i}, V_{s_i}} \sum_{o_2=U_{s_j}, V_{s_j}} \frac{A(C_{o_1}^{l_{s_i}/4} \cap C_{o_2}^{l_{s_j}/4})}{\min(A(C_{o_1}^{l_{s_i}/4}), A(C_{o_2}^{l_{s_j}/4}))} \quad (4)$$

Where $A(\cdot)$ denotes the area. This quantification favors the segment pairs with good proximity. However, as shown in figure 1(b), while segments s_1, s_2 and s_3 share a common point, $\langle s_1, s_2 \rangle$ and $\langle s_1, s_3 \rangle$ have the same attractive potentials, we prefer $\langle s_1, s_3 \rangle$ for collinear continuity. Hence, for segment S , we define its repulsive region by $C_{P_S}^{l_S/2}$ (see the dashed-line circles in figure 2), where P_S is the center of segment. If two

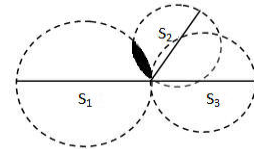


Figure 2. The repulsive region of segments

segments s_i and s_j are connected, the repulsive potential function $I_r(s_i, s_j)$ is defined by:

$$I_r(s_i, s_j) = \frac{A\left(C_{P_{s_i}}^{l_{s_i}/2} \cap C_{P_{s_j}}^{l_{s_j}/2}\right)}{\min\left(A\left(C_{P_{s_i}}^{l_{s_i}/2}\right), A\left(C_{P_{s_j}}^{l_{s_j}/2}\right)\right)} \quad (5)$$

From figure 2, we see that $I_r(s_1, s_2)$ shown in shadow area is larger than $I_r(s_1, s_3)$ which is zero. For the parameters, ω_1 , ω_2 and ω_r are positive, ω_a is negative, and ω_0 is positive, which is the term $-\log\beta$ in formula(2).

2.2. The data term of segments

The data term $E_D(W)$ defined in formula (1), is used to check how well the segment is fitted to the image data, and to locate the segment on image. Given a segment s_i , we define its fitting goodness by $\mu(s_i)$, whose concrete formulation depends on the type of images, such as binary, gray or color images. In section 4, we give two formulations of $\mu(s_i)$ on synthetic binary image and real gray images respectively. Whatever the formulation of $\mu(s_i)$ is, in our implementation, we all threshold it into penal part and encouraging part:

$$\mu'(s_i) = \begin{cases} \infty & \text{if } \mu(s_i) < t_1 \\ 1 - 2\frac{\mu(s_i) - t_1}{t_2 - t_1} & \text{if } t_1 \leq \mu(s_i) < t_2 \\ -1 & \text{if } \mu(s_i) \geq t_2 \end{cases} \quad (6)$$

If $\mu(s_i)$ is too small, it is absolutely penalized, otherwise, it is casted into range $[-1, 1]$ with both penal and encouraging effect, and this range is compatible with the value computed in formula (5) and (6). t_1 and t_2 are constants, and their choice depends on image type. Hence, the data term is given by:

$$E_D(W) = \omega_d \sum_{s_i \in W} \mu'(s_i) \quad (7)$$

The parameter ω_d is a positive constant.

3. Optimization

Our solution W^* is a set of line segments with expected configuration which satisfies the formula (2). As the solution space consists of many subspaces with different dimension, to search for a global minimum, we use Reversible Jump Markov chain Monte Carlo (RJMCMC) algorithm with the Metropolis-Hastings-Green dynamics [11] and Simulated Annealing technique. At each temperature T , the density $[f(W)]^{\frac{1}{T}}$ is simulated by RJMCMC. To ensure the ergodicity of Markov chain, in our implementation, the jump dynamics, which move between subspaces with different dimensions, contain segments birth and death, and the diffuse dynamics, which move within a subspace with fixed dimension, contain

modification of the length and orientation of segments. These three kinds of dynamics, birth, death and modification, are used at random with probabilities q_b , q_d and q_m respectively.

Given the current state ϖ of Markov chain, the new proposed transition state ϖ' is accepted with a probability, $\alpha(\varpi, \varpi') = \min\{1, R(\varpi, \varpi')\}$, where $R(\varpi, \varpi')$ is the Green's ratio:

$$R(\varpi, \varpi') = \frac{P(\varpi', \varpi)f(\varpi')^{\frac{1}{T}}}{P(\varpi, \varpi')f(\varpi)^{\frac{1}{T}}} \quad (8)$$

where $P(\cdot)$ is the transition kernel.

Suppose the current line segments set is W , then $\varpi = W$. The transition kernels of three dynamics are computed as follows: For the segment birth dynamics, $\varpi' = W \cup \{s_i\}$, a segment s_i is born and added to W , then the transition kernel is computed by $P(\varpi, \varpi') = q_b q(p_i) q(l_i | p_i) q(\theta_i | p_i)$, it firstly choose the center p_i of s_i from the image lattice with probability $q(p_i)$, then choose the length l_i and orientation θ_i with the uniform distribution $q(l_i | p_i)$ and $q(\theta_i | p_i)$ respectively. The reverse kernel is computed by $P(\varpi', \varpi) = q_d \frac{1}{\text{card}(W)+1}$, where $\text{card}(W)$ is the number of segments set W ; For the segment death dynamics, $\varpi' = W \setminus s_i$, a segment s_i is deleted from W , as death is a reverse jump of birth, its two transition kernels are similar with that in birth dynamics; For the segment modification dynamics, $\varpi' = (W \setminus s_i) \cup \{s_i'\}$, a segment s_i is chosen, and modified to $s_i' = (p_i, l_i \pm \Delta l, \theta_i \pm \Delta\theta)$, the length and orientation each is either added or subtracted by a small value Δ , or kept for no change, all these actions are taken at random with equal probability. As no segment birth or death, the two modification transition kernels are equal, the Green's ratio in (8) becomes to the Metropolis's ratio:

$$R(\varpi, \varpi') = \frac{f(\varpi')^{\frac{1}{T}}}{f(\varpi)^{\frac{1}{T}}} \quad (9)$$

In our implementation, the initial line segment set W_0 is empty.

4. Experimental results and Conclusion

We implement our method in C++ language, and run it on PC with 1.73Hz CPU and 1G memory. Both synthetic and real images are tested. Here, we show four challenging images. The two synthetic binary images with size 128×128 , shown in figure 3(a)(b), one is a "S-shape" curve, and the other is a circle, both have a cluttered background. The two real gray images with sizes 265×175 and 280×165 are shown in figure 3(c)(d). Figure 3(e)-(h) show our results, the approximation to the salient contours in figure 3(a)-(d) respectively. No post-processing is used here, e.g. edge-linking, or gap-filling.

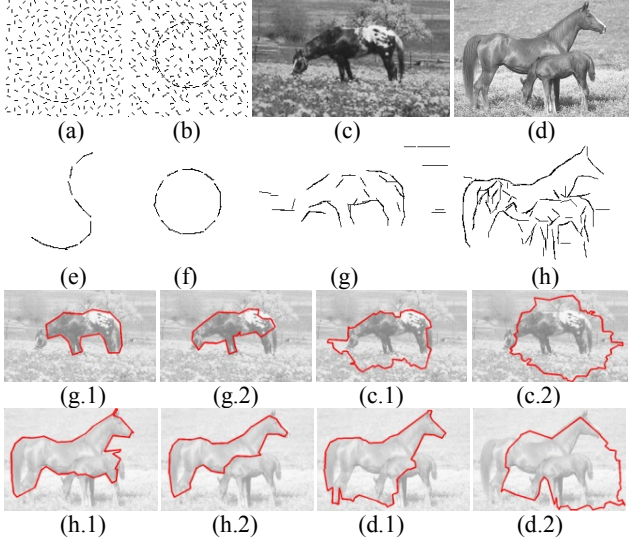


Figure 3.(a)-(d), the input images. (e)-(h), our results of line segments set approximation for salient contours in(a)-(d) respectively. (g.1)(g.2) / (h.1)(h.2), the first and second salient closed contour in (c) / (d) obtained by grouping method in [4] using (g) / (h) as its intermediate input. (c.1)(c.2) / (d.1)(d.2), the first and second salient closed contour in (c) / (d) obtained by [4] directly.

The parameters are, $\omega_0 = 20, \omega_1 = 220, \omega_2 = 40, \omega_d = -10000, \omega_r = 10000, \omega_d = 10, l_{min} = 9, l_{max} = 17, \Delta l = 2, \Delta\theta = \pi/12, q_b = q_d = q_m = 1/3$. These parameters depend on the resolution and types of images. Currently, they are fixed empirically. All line segments are of one-pixel width. For the synthetic binary image (pixel value is 0 or 255), the fitness $\mu(s_i)$ of one line segment to the actual contour, is computed by the mean of the pixel value on the segment; For real image, a segment is expected on the boundaries. We consider the contrast between two rectangle regions which locate to the right and left of segment respectively, and align with the segment. The segment is a sharing side of these two rectangles. Both regions have the same size, $l \times \Delta w$, where l is length of segment chosen randomly, Δw is fixed to 4. The fitness of a segment to the actual contour is computed by $\mu(s_i) = |m_R - m_L| / \sqrt{\sigma_R^2/n_R + \sigma_L^2/n_L}$, where m_R, m_L are the mean, σ_R, σ_L are standard deviation, corresponding to the pixel gray values in right and left rectangle respectively, n_R and n_L , the number of pixels in each rectangle.

The distribution $q(p_i)$ in section 3, which the center of segment is sampled according to, can be uniform on image lattice. To accelerate sampling, we make $q(p_i)$ proportional to normalized pixel value in figure 3(a)(b), and proportional to normalized pixel gradient value in figure 3(c) (d). We use a logarithmic annealing schedule in the sampling procedure. For most images, our algorithm converges in 15-40 sweeps (1sweep = 10000 iterations in

our implementation). The computational time for one image depends on its resolution and complexity. The larger the size of image is, and the more complex the scene of image is, the much time needs to take. In our experiment, the computational time taken by figure 3(a)-(d) to get the results of figure 3(e)-(h) is around 5,7,12 and 10 minutes respectively.

Our line segment set also can be used as intermediate input to the grouping method [4] for extracting two most salient closed contours in each of figure 3(c) and (d), the comparison of the results in the last two rows of figure 3, shows our line segment representation improves the robustness of grouping method due to less distracting tokens.

On conclusion, this paper introduces a new approximate representation to the salient contours in noisy and cluttered images. Our method directly models non-pixel line tokens, and makes the tokens detection and grouping completed together in a stochastic framework. Compared with those methods of bottom-up fashion in literature, our method reduces more false positives in the approximate representation of contours by avoiding the low-level edge detection.

Acknowledgement

This work was supported by a Chinese National 863 grant, No.2007AA01Z166 .

References

- [1] J. H. Elder, and S. W. Zucker, "Computing contour closure", in *Proc. ECCV*, pp.399-412, 1996.
- [2] F. Estrada, and A. Jepson, "Perceptual grouping for contour extraction", in *Proc. ICPR*, 2:32-35, 2004
- [3] D. Jacobs, "Robust and efficient detection of convex groups", *IEEE Trans. PAMI*, 18(1):23-37, 1996.
- [4] J. S. Stahl, and S. Wang. "Edge Grouping Combining Boundary and Region Information", *IEEE Trans. Image Process.*, 16(10):377-384, oct.2007.
- [5] L. Williams, and K. K. Thornber, "A comparison measures for detecting natural shapes in cluttered background", *IJCV*, 34(2/3):81-96, 2000.
- [6] S. Wang, T. Kubota, J. Siskind, and J. Wang, "Salient closed boundary extraction with ratio contour", *IEEE Trans. PAMI*, 27(4):546-561, 2005.
- [7] S. Wang, J. S. Stahl, A. Bailey and M. Dropps, "Global detection of salient convex boundaries", *IJCV*, 71(3):337-359, 2007.
- [8] A. J. Baddeley and M. N. M. van Lieshout, "Stochastic geometry models in high-level vision", *Statistics and Images*, 1:233-258, 1993.
- [9] M. N. M. van Lieshout, *Markov Point Processes and Their Applications*, Imperial College Press, Londres , 2000.
- [10] R. Stoica, X. Descombes, and J. Zerubia, "A Gibbs point process for road extraction in remotely sensed images", *IJCV*, 57(2):121-136, 2004.
- [11] P. Green, "Reversible Jump Markov Chain Monte-Carlo Computation and Bayesian Model Determination", *Biometrika*, 82(4): 711-732, 1995.