CORNER DETECTION OF CONTOUR IMAGES USING SPECTRAL CLUSTERING

Xi Li, Weiming Hu

National Laboratory of Pattern Recognition,
Institute of Automation, CAS, Beijing, China
Email: {lixi, wmhu}@nlpr.ia.ac.cn

Zhongfei Zhang

State University of New York,
Binghamton, NY 13902, USA
Email: {zhongfei@cs.binghamton.edu}

ABSTRACT

Corner detection plays an important role in object recognition and motion analysis. In this paper, we propose a hierarchical corner detection framework based on spectral clustering (SC). The framework consists of three stages: contour smoothing, corner cell extraction and corner localization. In the contour smoothing stage, wavelet decomposition is imposed on the raw contour to reduce noise. In the corner cell extraction stage, several atomic corner cells are obtained by SC. In the corner localization stage, the corner points of each corner cell are located by the corner locator based on the kernel-weighted cosine curvature measure. Experimental results demonstrate the superiority of our framework.

Index Terms**—** corner detection, spectral clustering, mean shift

1. INTRODUCTION

Corner detection is a common foundation in many applications of computer vision and image processing. Due to the robustness to scale, rotation, and translation variations, corners are very suitable for object matching and recognition. In this paper, we mainly focus on corner detection of contour images.

In recent years, much work has been done in corner detection of contour images. Rattarangsi and Chin [1] set up a Gaussian scale space to detect corner points on a 2D planar curve. Urdiales *et al.* [2] detect the corners by filtering the contour curvature function adaptively. Teh and Chin [3] propose a support region-based algorithm to find dominant points in a digital closed curve. A multi-scale corner detection method based on continuous wavelet transform is proposed in [4]. The corner points are located according to the local wavelet transform modulus maxima of the contour orientation. Gao *et al.* [5] present a dyadic wavelet transform-based method for corner detection. Accurate localization of corner points is achieved due to taking the local natural scale based strategy instead of the global one. Li and Chen [6] detect the corners using fuzzy reasoning such that the problem of corner detection is mapped to that of fuzzy classification. The common ground of the above methods is that they take a bottom-up strategy for corner detection. Corner localization is achieved according to the cue of local gradient-varying information. They all ignore the global shape information of the contour. As a result, corner detection results are usually sensitive to noise.

In this paper, we take a top-down strategy for corner detection. A hierarchical clustering-based corner detection framework is proposed. In the framework, an undirected graph is used for capturing the spatial and sequential relationships among contour points. We take advantage of a classic graph-theoretic clustering algorithm called spectral clustering (SC) [7] to partition the entire contour into several atomic cells. With efficiency and effectiveness in clustering data with complex structure, SC is very promising for multi-class data learning.

2. OUR CORNER DETECTION FRAMEWORK

2.1. Overview of the framework

Our corner detection framework includes three stages: contour smoothing, corner cell extraction, and corner localization. In the contour smoothing stage, two-level wavelet decomposition is used to smooth the raw contour. In the corner cell extraction stage, spectral clustering (SC) is exploited to decompose the smoothed contour into several atomic cells. Mean shift [8], a nonparametric kernel density estimation technique, is embedded in SC. And then the corner cells are selected according to the cue of our proposed kernel-weighted cosine curvature measure (KCCM). For each corner cell, the corner points are located by our KCCM-based corner locator in the corner localization stage.

2.2. Contour smoothing

In this paper, multi-resolution wavelet decomposition [9] is used to smooth contours. As we know, any square-integrable signal can be decomposed into a high-frequency signal and a low-frequency signal by wavelet decomposition. The high-frequency signal provides detailed information of the original signal while the low-frequency signal reserves the main structural characteristics. Consequently, the low-frequency signal can be viewed as a "smoothed" one of the original signal. The details of contour smoothing are described as follows.

1-4244-1437-7/07/$20.00 ©2007 IEEE

A contour is first unwrapped at a randomly selected contour point. As a result, the contour is represented as two 1D signals which are x-coordinate and y-coordinate respectively. Due to the closure property of a contour, we should employ a periodical signal to capture its spatial and sequential information. To simplify the computation of contour smoothing, we only consider three periods. Thus, the contour is represented as:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \left(\begin{matrix} \overbrace{x_1 \ x_2 \ \cdots \ x_{n-1} \ x_n}^{\times 3} \\ \underbrace{y_1 \ y_2 \ \cdots \ y_{n-1} \ y_n}_{\times 3} \end{matrix} \right) \tag{1}$$

where n is the number of points on that contour. We impose a two-level wavelet decomposition respectively on X and Y, and then obtain the position expression of the smoothed contour:

$$\begin{pmatrix} X_e^{s^*} \\ Y_e^{s^*} \end{pmatrix} = \left(\begin{matrix} x_1^{s^*} \cdots x_n^{s^*} & \overbrace{x_{n+1}^{s^*} \cdots x_{2n}^{s^*}}^{\text{Truncated}} & x_{2n+1}^{s^*} \cdots x_{3n}^{s^*} \\ y_1^{s^*} \cdots y_n^{s^*} & \underbrace{y_{n+1}^{s^*} \cdots y_{2n}^{s^*}}_{\text{Truncated}} & y_{2n+1}^{s^*} \cdots y_{3n}^{s^*} \end{matrix} \right) \tag{2}$$

where $X_e^{s^*}$ and $Y_e^{s^*}$ respectively denote the smoothed signals of the original X and Y at the resolution 2^{-2}. In this way, we obtain the final smoothed contour by truncating $X_e^{s^*}$ and $Y_e^{s^*}$ respectively from the $(n+1)$th element to the $(2n)$th one. Consequently, the final smoothed contour can be represented as:

$$\begin{pmatrix} X^s \\ Y^s \end{pmatrix} = \begin{pmatrix} x_{n+1}^{s^*} & x_{n+2}^{s^*} & \cdots & x_{2n}^{s^*} \\ y_{n+1}^{s^*} & y_{n+2}^{s^*} & \cdots & y_{2n}^{s^*} \end{pmatrix} \tag{3}$$

2.3. Corner cell extraction

We take a divisive clustering structure for corner cell extraction. The entire contour is decomposed by SC into several atomic cells. Then the corner cells are selected according to the KCCM-based corner discrimination criterion function. Fig. 1 shows the architecture of corner cell extraction. The details of corner cell extraction are described in the following three subsections.

2.3.1. Mean shift clustering

Mean shift [8], a nonparametric kernel density estimator, seeks the nearest mode of a point sample density function. It is an instance of gradient ascent with an adaptive step size. Consequently, point samples can be learned efficiently and adaptively by the mean shift given the kernel and the bandwidth matrix. The core of the mean shift is to calculate an offset according to the mean shift vector represented as:

$$m(x) = \frac{\sum_{i=1}^{n} x_i g(\|\frac{x-x_i}{h}\|^2)}{\sum_{i=1}^{n} g(\|\frac{x-x_i}{h}\|^2)} - x \tag{4}$$

where $g(x) = -k'(\cdot)$ which denotes the first derivative of the kernel profile $k(x)$, h is a bandwidth parameter, and $\|\cdot\|$ denotes the L_2 norm. In our framework, the kernel profile $k(\cdot)$ is chosen as:

$$k(x) = \exp(-\frac{1}{2}x) \qquad x \geq 0 \tag{5}$$

In this way, we have the standard mean shift procedure which takes iteratively the following steps:

- computing the mean shift vector $m(x^k)$,
- updating the current position $x^{k+1} = x^k + m(x^k)$.

The mean shift clustering algorithm [8] is just a practical application of the mode finding procedure:

- run the mean shift procedure to find the stationary points of the density function,
- prune these points by retaining only the local maxima.

The set of data points converging to the same mode is defined as the basin of attraction of that mode. Each basin of attraction corresponds to a cluster.

2.3.2. Spectral clustering

Spectral clustering is a graph-theoretic unsupervised leaning technique. It is efficient in handling multi-class learning. Many different versions exist for spectral clustering. In our framework, we propose an improved version based on [7]. The algorithm applies the mean shift to clustering the data instead of the KMeans used in [7]. The specific procedure of our algorithm is detailed as follows.

Given the data set $X = \{p_i : 1 \leq i \leq n\}$ where $p_i = (x_i, y_i)$, we cluster the data set X into a number of classes using the following procedure:

1. Compute the similarity matrix A, where A_{ij} is defined as: $A_{ij} = exp[-(d_{ij}^2/2\sigma_d^2 + g_{ij}^2/2\sigma_g^2)]$ in which d_{ij} and g_{ij} respectively denote the Euclidean distance and the shortest contour geodesic distance between points i and j.
2. Construct the matrix $L = D^{-1/2}AD^{-1/2}$, where D is a diagonal matrix with its ith diagonal element equal to the sum of the elements in the ith row of A.
3. Obtain a new matrix $E = [e_1, e_2, \ldots, e_K]$ formed by the normalized K largest eigenvectors of L.
4. Apply the mean shift clustering algorithm to classifying row vectors of E into a number of clusters automatically.

2.3.3. Corner discrimination

Now, we are ready to discuss corner discrimination in details. A KCCM is developed to capture the gradient-varying information of the contour points. The details of the KCCM are described as follows. Given an n-point contour cell $C = \{p_l : 1 \leq l \leq n\}$ where $p_l = (x_l, y_l)$, the KCCM is defined as:

$$KCCM(k) = \frac{\sum_{i<k} \sum_{j>k} exp(-g_{ij}^2/\sigma_g^2)\theta_{\widehat{ikj}}}{\sum_{i<k} \sum_{j>k} exp(-g_{ij}^2/\sigma_g^2)} \tag{6}$$

where $2 \leq k \leq n-1$, g_{ij} denotes the geodesic distance between points i and j, σ_g is a scaling factor, and the angle $\theta_{\widehat{ikj}}$ between $\overrightarrow{ki}$ and $\overrightarrow{kj}$ is calculated as:

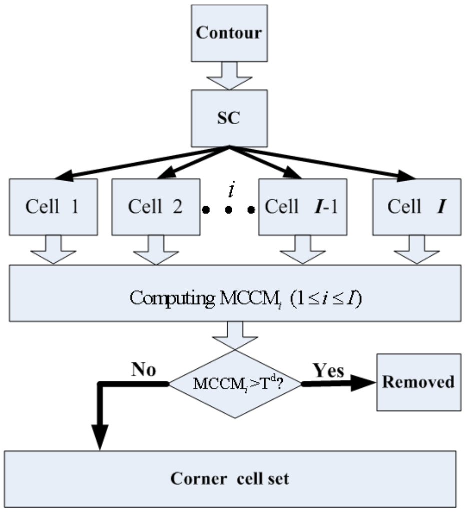


Fig. 1. The architecture of corner cell extraction.

$$\theta_{\widehat{ikj}} = \arccos\left(\frac{d_{ik}^2 + d_{kj}^2 - d_{ij}^2}{2d_{ik}d_{kj}}\right) \tag{7}$$

where d_{mn} denotes the Euclidean distance between points m and n. Let $MCCM = \min\limits_{2\leq k\leq n-1}[KCCM(k)]$. Consequently, the criterion function for corner discrimination is defined as:

$$C = \begin{cases} \text{non-corner cell} & \text{if } MCCM > T^d \\ \text{corner cell} & \text{otherwise.} \end{cases} \tag{8}$$

2.4. Corner localization

We propose a KCCM-based corner locator, which detects the corner point by finding the local minima of the KCCM. Let C_p be the corner points we expect to detect; p_k be the kth point of an n-point contour. Consequently, C_p is located by the following rule:

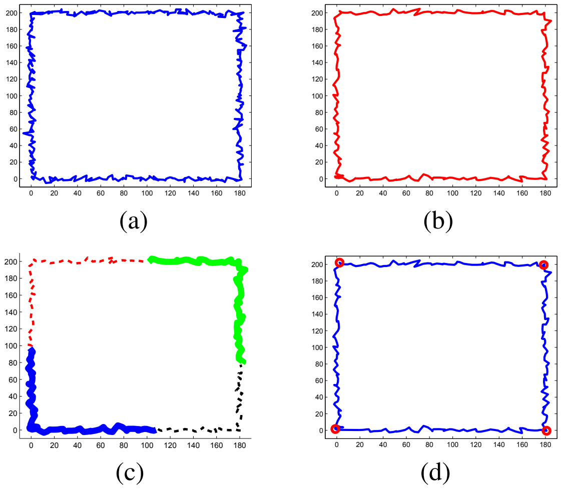
$$C_p = \{p_k | KCCM(k) < T^l, p_k \in \Omega\} \tag{9}$$

where $KCCM(\cdot)$ is defined in (6), and Ω corresponds to the set consisting of all the local minima of KCCM.

3. EXPERIMENTAL RESULTS

In our experiments, a total number of 50 complex contours are tested by the corner detection framework. Their corners are pre-labeled manually in order to test the performance of our framework effectively. In the corner cell extraction stage, the threshold T^d used for corner cell discrimination is set as 2.6. Moreover, the settings of σ_d and σ_g mentioned in the first step of SC are obtained from experiments. In the third step of SC, the criterion for selecting K is defined as:

$$K = \min\left(\{k | \sum_{i=1}^{k}\lambda_i / \sum_{j=1}^{n}\lambda_j > 0.95, 1 \leq k \leq n\}\right) \tag{10}$$



(a) (b)

(c) (d)

Fig. 2. An example of corner detection. (a) A noisy contour. (b)The smoothed contour. (c) The results of corner cell extraction. (d) The results of corner localization.

where $\lambda_l (1 \leq l \leq n)$ denotes the lth eigenvalue of the matrix L referred to in the second step of SC. The bandwidth parameter h of the mean shift procedure is set as 0.5. The setting of σ_g defined in (6) is also obtained from experiments. In the corner localization stage, the threshold T^l is set as 1.6. We take the noisy rectangle-like contour shown in Fig. 2(a) for example to illustrate the entire process of the corner detection. The contour after contour smoothing is shown in Fig. 2(b). Then it is decomposed by SC into four atomic corner cells. Subsequently, the corner localization is achieved by the KCCM-based corner locator detailed in Section 2.4. The final corner detection results are shown in Fig. 2(d). For better visualization, we just choose to display twelve typical contours after the corner detection. They are shown in Fig.3 from which we see that our framework is able to find the corner points accurately with the disturbances of noise.

In order to evaluate the performance of the corner detection framework quantitatively, we define a criterion function for measuring the corner detection accuracy. Given an N-point contour denoted as $C = \{p_i | 1 \leq i \leq N, p_i = (x_i, y_i)\}$, we introduce an indicator vector $L_C = \{s_i | 1 \leq i \leq N\}$ to represent C's corner label information. The ith element s_i of L_C is defined as:

$$s_i = \begin{cases} 1 & \text{if } p_i \text{ is detected as a corner point} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Then we divide the interval $\{1, 2, \ldots, N\}$ into J equal subintervals which are denoted as: $I_1, I_2, \ldots, I_J$. All the points in the contour are mapped to their corresponding subintervals. We calculate $r_i = \frac{N_i^c}{N_i}$ where N_i represents the number of contour points in I_i, and $N_i^c = \sum_{j\in I_i} s_j$ which denotes the number of detected corner points in I_i. As a result, we obtain C's corner label distribution vector (CLD) denoted as: $(r_1, \ldots, r_J)$. The Euclidean distance is used to measure the similarity of two CLDs. Let CLD^d be the CLD obtained by

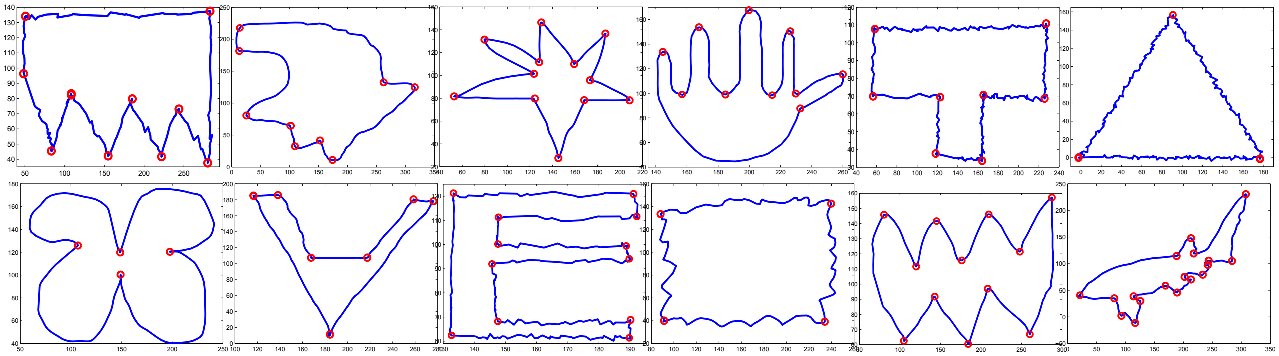


Fig. 3. The results of corner detection. Each circle corresponds to a corner point.

a corner detection method, CLD^m be the CLD corresponding to the ground truth derived from manual labelling, and E_{dm} be the Euclidean distance between CLD^d and CLD^m. The similarity S_{dm} of CLD^d and CLD^m is defined as:

$$S_{dm} = \frac{e^{2(1-\frac{1}{\sqrt{J}}E_{dm})}-1}{e^2-1} \in [0,1] \qquad (12)$$

The larger the S_{dm}, the better the corner detection result. Consequently, we evaluate the performance of our framework by determining the similarity S_{dm}. In the experiments, J is set as 50. We make a comparison among three corner detection methods which are Rattarangsi and Chin's [1], Gao *et al.*'s [5], and ours. Fig. 4 shows the corner detection accuracy of the above three methods. The x-axis and the y-axis respectively correspond to contour ID number and the accuracy S_{dm} described above. It is seen from Fig. 4 that our framework performs best.

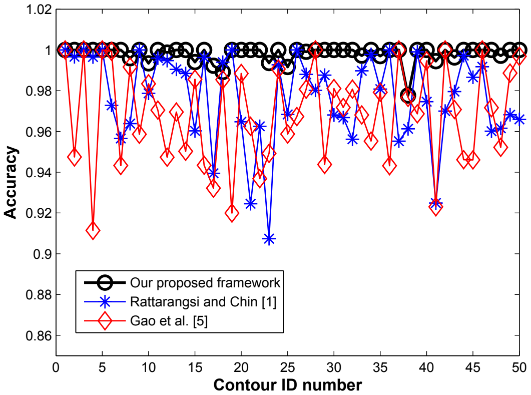


Fig. 4. Corner detection accuracy.

4. CONCLUSION

We have proposed a hierarchical corner detection framework based on the spectral clustering in which the mean shift is embedded. Wavelet decomposition has been used for smoothing contours. Meanwhile, a novel kernel weighted cosine curvature measure has been proposed. Experimental results have demonstrated the effectiveness and the promise of our framework.

5. ACKNOWLEDGMENT

This work is partly supported by NSFC (Grant No. 60520120099 and 60672040) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453). ZZ is partly supported by NSF (IIS-0535162), AFRL (FA8750-05-2-0284), and AFOSR (FA9550-06-1-0327).

6. REFERENCES

[1] A. Rattarangsi and R.T. Chin, "Scale-based Detection of Corners of Planar Curves," *IEEE Trans. PAMI.*, Vol. 14, Iss. 4, pp. 430 - 449, April 1992.

[2] C. Urdiales, C. Trazegnies, A. Bandera, and F. Sandoval, "Corner Detection Based on Adaptively Filtered Curvature Function," *Electronics Letters*, Vol. 39, Iss. 5, pp.426 - 428, 6 March 2003.

[3] C.-H. Teh and R.T. Chin, "On the Detection of Dominant Points on Digital Curves," *IEEE Trans. PAMI.*, Vol. 11, Iss. 8, pp.859 - 872, Aug. 1989.

[4] X. Gao, F. Sattar, A. Quddus, and R. Venkateswarlu, "Comer Detection of Contour Images Using Continuous Wavelet Transform,", *IEEE ICICS-PCM'03*, 2003.

[5] X. Gao, F. Sattar, A. Quddus, and R. Venkateswarlu, "Local Natural Scale Based Contour Corner Detection Using Wavelet Transform," *IEEE ICICS*, pp.329 - 333, 2005.

[6] L. Li and W. Chen, "Corner Detection and Interpretation on Planar Curves Using Fuzzy Reasoning," *IEEE Trans. PAMI.*, Vol. 21, Iss. 11, pp.1204 - 1210, Nov. 1999.

[7] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and An Algorithm," *NIPS'01*, 2001.

[8] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. PAMI.*, Vol. 24, Iss. 5, pp. 603-619, May 2002.

[9] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Trans. PAMI.*, Vol.11, Iss. 7, pp.674 - 693, July 1989.