Real-Time Salient Closed Boundary Tracking using Perceptual Grouping and Shape Priors

Xuebin Qin
https://webdocs.cs.ualberta.ca/~xuebin/
Shida He
shida3@ualberta.ca
Zichen Zhang
http://webdocs.cs.ualberta.ca/~zichen2/
Masood Dehghan
masood1@ualberta.ca
Martin Jagersand
https://webdocs.cs.ualberta.ca/~jag/

Department of Computing Science
University of Alberta
Edmonton, Alberta, CA

Abstract

In this paper, we propose a real-time method for accurate salient closed boundary tracking via a combination of shape constraints and perceptual grouping on edge fragments. Particularly, we encode the Gestalt law of proximity and the prior shape constraint in a novel ratio-form grouping cost. The proximity and prior constraint are depicted by the relative gap length and average distance difference along the to-be-tracked boundary with respect to its area. We build a graph using the detected edge fragments and in-between gaps. The grouping problem is formulated as searching for a special cycle in this graph with a minimum grouping cost. To reduce the search space and achieve real-time performance, we propose a set of novel techniques for efficient edge fragments splitting and filtering. We evaluate this method on a public real-world video dataset against other methods. The average alignment errors of different sequences achieved by our method are mostly less than 1 pixel, an improvement over state-of-the-art methods.

1 Introduction

Real-time salient closed boundary tracking is an important yet challenging issue in the computer vision community. It is also an important technique for vision guided robot applications. Pertinent methods can be categorized into three main classes: 1) *template based methods*, which estimate the geometric transformations, such as affine and homography, of planar rigid target contours from one frame to the next one [8, 20], 2) *region segmentation based methods*, which determine boundaries on each frame by segmenting images into foreground and background using techniques like graph-cuts [6], level sets [21], etc. 3) *perceptual grouping based methods*, which search for a special cycle of low level primitives (e.g. edge fragments, line segments) forming the closed boundaries [11]. The proposed tracking method of this paper belongs to the third class.

Perceptual grouping algorithms have been widely used in contours completion [7, 10, 14] and salient closed boundaries extraction [1, 3, 9, 11, 16, 18]. Their basic principles draw from Gestalt laws [19] including proximity, good continuation, closure, etc. However, most of these methods focus on extracting contours by exploring grouping cues from the current image and cannot be used for closed boundary tracking directly.

Tracking a salient closed boundary through a video sequence can be formulated as a prior shape constrained perceptual grouping problem. Elder *et al*. [4] developed a framework of combining prior probabilistic knowledge of the target appearance with probabilistic models for contour grouping. Schoenemann and Cremers [15] chose the tangent angles of curves as a prior shape constraint term and combined it with pixel gradients, penalties for shape stretching and shrinking to formulate a pixel-wise elastic shape matching and grouping model. Neither of these methods run in real-time without GPU acceleration. Qin *et al*. [13] formulated a boundary tracking criterion by combining an area variation constraint and a grouping cost [16], which is defined as the ratio of gap length and region area enclosed by the boundary. They also developed a shortest path [2] based algorithm to search for the optimal boundary. However, the area variation constraint is weak and does not preserve fine structures. In addition, the line segments used in their methods can not fit curved boundaries well.

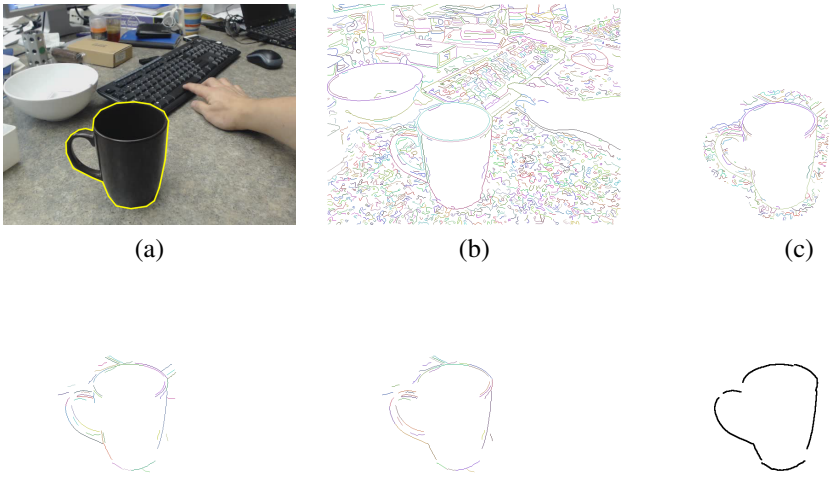
In this paper, we develop a perceptual grouping method that combines Gestalt saliency and prior shape information. We demonstrate a real-time CPU implementation. Contributions include: 1) We define a new grouping cost by adding a distance difference based prior shape constraint to the grouping cost. 2) We propose a novel technique for fast edge segments splitting to speed up the grouping process. It generates high quality edge fragments in less than one millisecond per frame. Redundant edge fragments are removed according to their lengths and average distance differences. 3) We implement this edge fragments based closed boundary tracking method and test it on a public real-world video dataset. It achieves state-of-the-art performance, outperforming a method adapted from [16] and the method proposed in [13].

2 Tracking via Prior Shape Constrained Grouping

Given a video sequence and a salient closed boundary defined in the first frame, as shown in Figure 1 (a), our goal is to track the closed boundary over the entire sequence by identifying and sequentially connecting a set of edge fragments in each frame, Figure 1 (f). Compared with grouping the salient closed boundaries from single image, the closed boundary tracking from video sequences can take advantage of both saliency properties derived from Gestalt laws and prior shape constraints from the tracked boundary on previous frame.

We divide prior shape constraints into three levels. First, intact boundary oriented coarse constraints, such as perimeter and area variations. Constraints of this level restrict significant shape variations other than fine structures. Second, primitives oriented middle level constraints such as curvature, direction, length and distance properties of edge fragments and line segments. These constraints provide relatively stronger restrictions to shape variations. Third, pixel-wise constraints that can even produce very fine and accurate pixel-by-pixel matching between to-be-tracked boundary and the prior boundary. In this paper, we use the first and second level constraints because the pixel-wise constraints are usually time consuming and hard to be solved in real-time.

To combine the Gestalt cues of the current frame and prior shape information, we first



(a) (b) (c)

(d) (e) (f)

Figure 1: Algorithm steps of salient closed boundary tracking. (a) current frame and the manually initialized boundary in yellow (or that tracked from the previous frame); (b) edge segments detected by Edge Drawing; (c) filtered edge segments pixels according to their distances to the prior boundary (yellow boundary in (a)); (d) edge fragments generated from edge segments splitting and length based filtering; (e) edge fragments filtered by the distance difference over length $\frac{DD_i}{L_i}$; (f) edge fragments selected for grouping the current salient closed boundary.

define a grouping cost $\Gamma(B)$ of a closed boundary B based on the middle level constraints as:

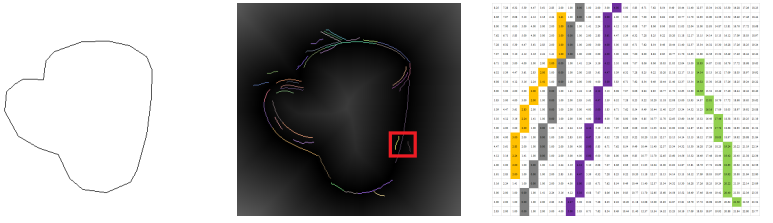
$$\Gamma(B) = \frac{|G_B| + |DD_B|}{\iint_{R(B)} dxdy} \tag{1}$$

where $\iint_{R(B)} dxdy$ is the area of the region $R(B)$ which is enclosed by the boundary. $|G_B|$ denotes the total length of the edge fragments' in-between gap segments along the boundary and $\frac{|G_B|}{\iint_{R(B)} dxdy}$ depicts the proximity (saliency) of the boundary [16]. $|DD_B| = \sum_{i \in EF} DD_i$ is the total absolute distance difference. Figure 2 illustrates the distance difference (DD_i) of an edge fragment EF_i which is:

$$DD_i = \sum_{j=1}^{p-1} |dist(P_{j+1}) - dist(P_j)| \tag{2}$$

where p is the number of the edge fragment pixels, P_j denotes the j-th pixel in the fragment pixel array and $dist(P_j)$ is its corresponding value on the distance transform map [5] of the prior shape.

In addition to the area constraint used in [13], this paper introduces a perimeter variation



(a) (b) (c)

Figure 2: Distance Difference (*DD*): (a) prior shape; (b) detected edge fragments superimposed on the distance transform map of the prior shape; (c) zoom-in view of region enclosed by the red box in (b), gray pixels belong to the prior shape, colorized edges are extracted from the current frame, the value of each pixel is the Euclidean distance of the pixel to the closest pixel of the prior shape. The effect of *DD* is similar to the tangent angle [15] but easier to compute in real time.

constraint to reduce the search space and improve the tracking robustness as follows:

$$\begin{cases} v(P) = min(\frac{P_{prior}}{P_{cur}}, \frac{P_{cur}}{P_{prior}}) < e_P \\ v(A) = min(\frac{A_{prior}}{A_{cur}}, \frac{A_{cur}}{A_{prior}}) < e_A \end{cases} \tag{3}$$

where $v(P)$ and $v(A)$ are perimeter and area variations. P_{prior} and P_{cur} are the perimeters of prior and current closed boundaries. A_{prior} and A_{cur} are the areas of regions that enclosed by prior and current boundaries. e_P and e_A represent the perimeter and area variation constraints respectively.

3 Edge Fragments Detection

We use a real-time edge segment detector Edge Drawing (ED) [17] to extract high quality edge segments from incoming frames. Each of the resulting edge segment is a linear pixel chain with one-pixel width. The result of ED is outputted in vector form as an array of chain-wise edge segments, as shown in Figure 1 (b). These detected edge segments cannot be used for real-time salient closed boundary grouping directly due to the following reasons: 1) large number of redundant edge segments make the grouping search space huge; 2) foreground and background edge pixels are often improperly identified as one long edge segment. Hence, we propose a novel edge segments splitting technique combined with redundant edge pixels and fragments filtering to obtain a set of well-identified high quality edge fragments.

3.1 Edge Pixels Filtering

The edge segments detected by ED contains a lot of redundant edge pixels. We filter these pixels according to their distances to the prior shape. Pixels with absolute distances smaller than a threshold are retained, as shown in Figure 1 (c). The distance threshold (e.g. 30 pixels) depends on the relative motion speed and the frame rate (fps) of the video.

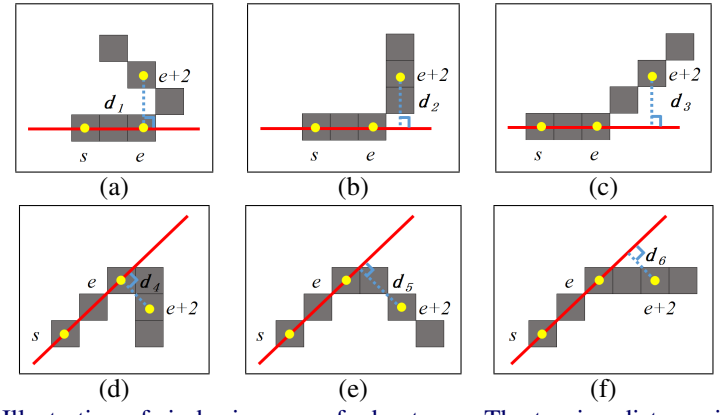


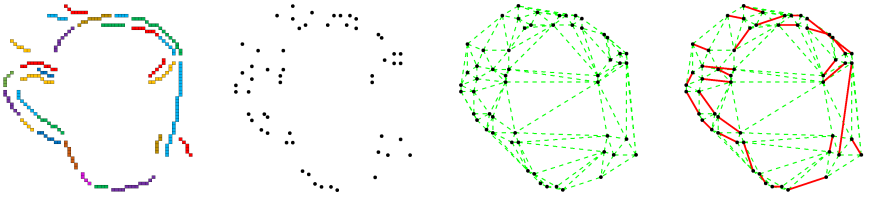
Figure 3: Illustration of six basic ways of edge turns. The turning distance is the distance from the extending pixel $(e+2)$ to the l_{se} line: $d_1 = d_2 = d_3 = 2$, $d_4 = d_6 = \sqrt{2}$, $d_5 = \frac{3\sqrt{2}}{2}$. According to these turning distances, we set the splitting threshold to $\sqrt{2} \approx 1.4$ pixel.

3.2 Edge Segments Splitting

Curvature [18], turning angle [14] and line fitting [16] based approaches have been proposed to split detected edge segments into multiple fragments. The curvature based method used in [18] requires spline fitting which is time consuming. Although the line fitting based method [16] is more efficient, it often misses short edge fragments. We develop a novel splitting method based on an analysis of the six basic ways of edge turns, as shown in Figure 3. Given an edge segment, we split it into one or multiple fragments by traversing it with a step size of two pixels. Particularly, we start a fragment searching by sampling two pixels with indices of s (starting pixel) and $e = s+2$ (ending pixel). Then, we compute the distance from the extending pixel $(e+2)$ to the line l_{se} to decide whether to split or not. Based on Figure 3 if the distance is greater than a threshold (1.4 pixels), we split the segment. Another splitting criterion is the middle pixel deviation. If the distance from the middle pixel $(m = \frac{s+e}{2})$ to l_{se} is greater than a threshold (5 pixels) we also split the segment. Otherwise, we extend the current fragment by $e = e+2$. This splitting algorithm runs in less than one millisecond per frame.

3.3 Edge Fragments Filtering

Edge segments splitting produces a large number of edge fragments. However, some of them are redundant. To reduce grouping search space and achieve real-time performance, we propose to use their length L_i and average distance difference $\frac{DD_i}{L_i}$ to filter redundant edge fragments. The length based filtering is aiming at excluding those tiny spurious edge fragments, as shown in Figure 1 (d). On the other hand, we assume that the to-be-tracked boundary in the current frame is almost parallel to the prior one. The average distance difference depicts the parallelism between fragments and the prior boundary. It filters those edge fragments which are more likely to be perpendicular to the prior boundary, as shown in Figure 1 (e).



(a) (b) (c) (d)

Figure 4: Illustration of graph modeling using detected edge fragments: (a) detected and filtered edge fragments. There are no intersections and common endpoints; (b) endpoints of those detected edge fragments; (c) Delaunay Triangulation (DT) of the endpoints; (d) graph structure constructed by the union of DT edges (c) and the corresponding degenerated edges from (a).

4 Graph Modeling and Optimization

In this section, we construct an undirected graph $G = (V,E)$ [13] data structure of the remaining edge fragments and assign weights to the graph edges according to our tracking cost function. First, to model the graph structure, we take the endpoints of those disconnected edge fragments as graph vertices, as illustrated in Figure 4 (b). Then, Delaunay Triangualtion (DT) [12] is used to generate gap filling segments and connect filled vertices, Figure 4 (c). Figure 4 (d) shows the graph structure whose edges are the union of the green dotted lines (DT edges) and the red solid lines (edges corresponding to the edge fragments in Figure 4 (a)).

Each graph edge is assigned two weights $w_L(e_i)$ and $w_{DD}(e_i)$ as follows:

$$w_L(e_i) = \begin{cases} 0 & e_i \text{ corresponds to an edge fragment} \\ |V_1^i V_2^i| & e_i \text{ corresponds to a DT edge} \end{cases} \tag{4}$$

where $|V_1^i V_2^i|$ is the length of corresponding gap filling line segment $V_1^i V_2^i$ of the graph edge e_i.

$$w_{DD}(e_i) = DD_i \tag{5}$$

where DD_i is the distance difference of the edge e_i from (2).

The grouping cost in (1) is computed as:

$$\Gamma(B) = \frac{\sum_{i\in E(\mathbf{C})} w_L^i + \sum_{i\in E(\mathbf{C})} w_{DD}^i}{Area(\mathbf{C})} \tag{6}$$

where $\mathbf{C}$ is a graph cycle that corresponds to a closed boundary B, $E(\mathbf{C})$ is the set of edges of $\mathbf{C}$. $Area(\mathbf{C})$ is the area of the corresponding polygon of the graph cycle and it approximates the closed boundary area $\iint_{R(B)} dxdy$ in (1) to simplify the area computation.

Finding a graph cycle that minimizes the grouping cost $\Gamma(B)$ (6) is a minimum-ratio-cycle problem and can be solved by algorithms proposed in [18] and [16] in polynomial time. However, we have to integrate the length and area variation constraints (3) into the optimization problem. To this end, we use a bidirectional shortest path based search strategy of [13] to generate number of graph cycle candidates and find the optimal one according to our tracking measure defined in (3) and (6).

5 Experimental Results

To evaluate the performance of our tracking method, we tested it on a public dataset [13] (*https://github.com/NathanUA/SalientClosedBoundaryTrackingDataset*) which has nine real-world video sequences and 9598 frames in total. Each video sequence is about 30-45 seconds (30 fps, size: 640× 480) and contains a single moving salient closed boundary with different types of motions including translation, rotation, zooming in/out and even out of plane rotation. These to-be-tracked targets includes planar/non-planar, non-Lambertian and irregular closed boundaries with/without clustered backgrounds, see Figures 6 and 7.

5.1 Evaluation

To evaluate our tracking method quantitatively, we introduce the maximal average cross alignment error $aveE\_AL = max\{B_i \otimes Dist_{B_{gt}}/P_{B_i}, B_{gt} \otimes Dist_{B_i}/P_{B_{gt}}\}$ where B_i and B_{gt} represent the binary images of the tracked boundary and the corresponding ground truth. $Dist_{B_i}$ and $Dist_{B_{gt}}$ are the distance transform maps of B_i and B_{gt}. $\otimes$ denotes the summation of element(pixel)-wise multiplication. P_{B_i} and $P_{B_{gt}}$ are the number of pixels in the tracked boundary and the corresponding ground truth.

We compared our *Edge Fragments Grouping* (EFG) based method with two others: a real-time tracker which is adapted from the regional information combined ratio contour (RRC) algorithm [16] and a state-of-the-art closed boundary tracking method (BDSP) which only uses a saliency measure and an area constraint [13].

Figure 5 plots the alignment error *aveE_AL* of each frame achieved by these three methods. Table 1 shows the average *aveE_AL* of each sequence. As illustrated in Figure 5 and Table 1, the RRC tracker fails quickly on sequences **MCC** (failure starts from frame: 285/total frame number: 1226), **BSC** (676/899), **NBR** (416/1454), **MCR** (750/859), **TBR** (200/1135), **MCRP** (213/971) and **GBR** (426/924) and produces large average alignment errors. These failures are mainly caused by cluttered backgrounds, see the first two rows of Figure 6, and relatively complex boundaries, as shown in the last two rows of Figure 6. Compared with RRC, the BDSP tracker is more robust because of its area variation constraint, as shown in Table 1. However, boundaries tracked by BDSP often have some erroneous wiggles, as illustrated in the last two rows of Figure 6. The reason for these wiggles is that the area variation constraint is too weak to restricts fine structures.

Our EFG tracker suppresses accidental wiggles effectively through the distance term. As shown in Figure 5 and Table 1, almost all of the errors of our EFG tracker on different sequences are smaller than those of RRC and BDSP. Because both RRC and BDSP operate on detected line segments while our EFG splits edge segments into smaller smooth fragments. Figure 7 illustrates the qualitative difference of representing boundaries via line segments and edge fragments. Compared with RRC and BDSP, our EFG tracker fits curved boundaries better and produces higher accuracy.

5.2 Run Time

We implemented our tracking method in C++ using OpenCV and Boost library on Ubuntu 14.04 64-bit OS. The time costs are collected by running our tracking method on a quad core 3.10 GHz and 16 GB RAM computer without GPU acceleration. To demonstrate our tracking method runs in real-time, the average tracking time costs of each video sequence are illustrated in Table 2. The average per frame time cost of RRC is in the range of 15 to

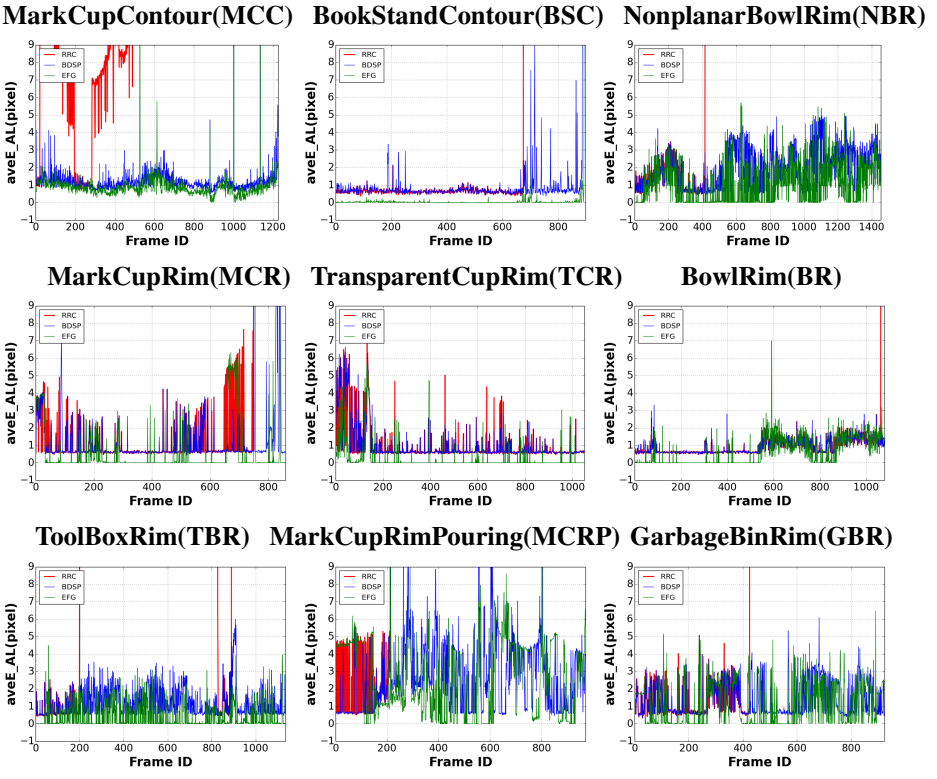


Figure 5: Average alignment error *aveE_AL* of the closed boundary tracking achieved by trackers RRC, BDSP and our EFG.

Video	MCC	BSC	NBR	MCR	TCR	BR	TBR	MCRP	GBR
RRC	56.09	58.06	120.48	19.29	1.01	3.37	74.09	58.18	108.28
BDSP	1.31	1.24	1.99	1.18	1.03	0.94	1.21	**2.73**	1.38
EFG	**0.99**	**0.18**	**1.08**	**0.68**	**0.26**	**0.61**	**0.46**	2.80	**1.22**

Table 1: Average *aveE_AL* (pixel) of each sequence.

Video	MCC	BSC	NBR	MCR	TCR	BR	TBR	MCRP	GBR
$aveEFs$	27	23	25	36	30	25	29	35	28
$aveGFs$	145	120	137	200	162	136	155	192	154
$T(ms)$	**28.11**	**20.37**	**21.60**	**36.39**	**26.45**	**22.31**	**26.39**	**35.08**	**30.92**

Table 2: Average tracking time costs for each video sequence: $aveEFs$ and $aveGFs$ are the average numbers of Edge Fragments and Gap Filling segments. $2 \times aveEFs$ is the number of graph vertices and $aveEFs + aveGFs$ is the number of graph edges. T is the average time cost of each frame tracking in milliseconds. It includes edge segments detection, splitting, filtering, graph construction and optimization. The frame loading time is not included because this process can be implemented in parallel and will not influence the tracking speed significantly.

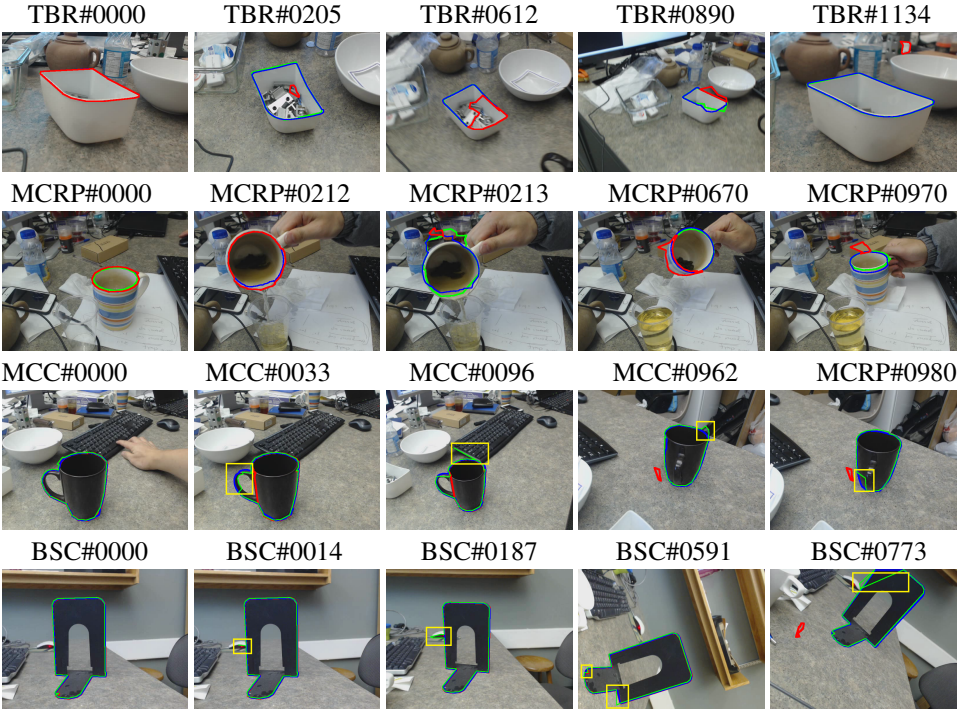


Figure 6: Failures of RRC and wiggles of BDSP: Red, green and blue boundaries are results of RRC, BDSP and EFG respectively. All of the four rows demonstrate the failure of RRC tracking. The last two rows show the wiggles (within yellow boxes) produced by BDSP.

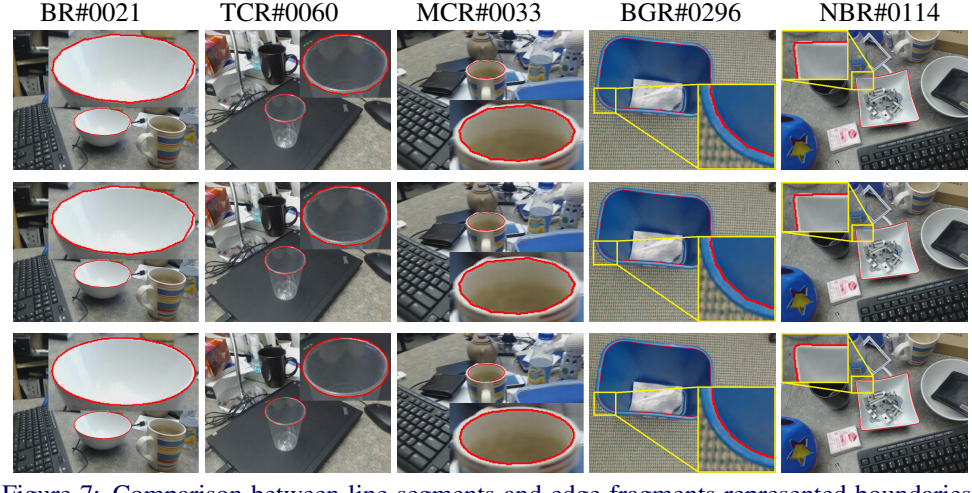


Figure 7: Comparison between line segments and edge fragments represented boundaries. From top row to bottom row are results of RRC, BDSP and our EFG respectively.

30 ms but it fails quickly in most of the sequences. The average time cost of our method is 27.51 ms which is slightly (4.64 ms) more than that of BDSP 22.87 ms, while our method improved the accuracy from 1.45 (BDSP) to 0.92 (see Tables 1&2). The 0.53 difference per pixel is usually the result of large wiggles in the boundary (Figure 6, 3rd and 4th rows, yellow boxes), which is suppressed in our method. The results show that our method takes close to or less than 30 ms per frame which means its speed is acceptable for real-time applications.

6 Conclusions and Discussions

In this paper, we proposed a novel real-time edge fragments grouping based method for salient closed boundary tracking. Our tracker encodes the prior shape constraint into the distance difference of deliberately split edge fragments and combines it with the boundary salient measure of relative gap length. It suppresses most of the small erroneous wiggles on the boundaries and improves the tracking accuracy. We validated our method on real-world video sequences and achieve the state-of-the-art results both qualitatively and quantitatively.

Currently, complex shape priors and/or shapes with tiny clustered structures (e.g. comb) are hard to track as the grouping process may ignore the tiny structures. Also, the shape prior based algorithm can not handle the large motions in-between frames. We will try to address these problems by introducing shape templates and more robust shape descriptors in our future work.

References

[1] T. D. Alter and Ronen Basri. Extracting salient curves from images: An analysis of the saliency network. *International Journal of Computer Vision*, 27(1):51–69, 1998.

[2] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[3] James H. Elder and Steven W. Zucker. Computing contour closure. In *Computer Vision - ECCV'96, 4th European Conference on Computer Vision, Cambridge, UK, April 15-18, 1996, Proceedings, Volume I*, pages 399–412, 1996.

[4] James H. Elder, Amnon Krupnik, and Leigh A. Johnston. Contour grouping with prior models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):661–674, 2003.

[5] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8(1):415–428, 2012.

[6] Martin Godec, Peter M. Roth, and Horst Bischof. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding*, 117(10):1245–1256, 2013.

[7] Ryan Kennedy, Jean H. Gallier, and Jianbo Shi. Contour cut: Identifying salient contours in images by solving a hermitian eigenvalue problem. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 2065–2072, 2011.

[8] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on*

Artificial Intelligence, IJCAI '81, Vancouver, BC, Canada, August 24-28, 1981, pages 674–679, 1981.

[9] Shyjan Mahamud, Lance R. Williams, Karvel K. Thornber, and Kanglin Xu. Segmentation of multiple salient closed contours from real images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(4):433–444, 2003.

[10] Yansheng Ming, Hongdong Li, and Xuming He. Connected contours: A new contour completion model that respects the closure effect. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 829–836, 2012.

[11] Vida Movahedi and James H. Elder. Combining local and global cues for closed contour extraction. In *British Machine Vision Conference, BMVC 2013, Bristol, UK, September 9-13, 2013*, 2013.

[12] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry - An Introduction*. Texts and Monographs in Computer Science. Springer, 1985.

[13] Xuebin Qin, Shida He, Camilo Perez Quintero, Abhineet Singh, Masood Dehghan, and Martin Jagersand. Real-time salient closed boundary tracking via line segments perceptual grouping. *arXiv preprint arXiv:1705.00360*, 2017.

[14] Xiaofeng Ren, Charless C. Fowlkes, and Jitendra Malik. Scale-invariant contour completion using conditional random fields. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, pages 1214–1221, 2005.

[15] Thomas Schoenemann and Daniel Cremers. A combinatorial solution for model-based image segmentation and real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(7):1153–1164, 2010.

[16] Joachim S. Stahl and Song Wang. Edge grouping combining boundary and region information. *IEEE Trans. Image Processing*, 16(10):2590–2606, 2007.

[17] Cihan Topal and Cuneyt Akinlar. Edge drawing: A combined real-time edge and segment detector. *J. Visual Communication and Image Representation*, 23(6):862–872, 2012.

[18] Song Wang, Toshiro Kubota, Jeffrey Mark Siskind, and Jun Wang. Salient closed boundary extraction with ratio contour. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4): 546–561, 2005.

[19] Max Wertheimer. Untersuchungen zur lehre von der gestalt. *Psychological Research*, 1(1):47–58, 1922.

[20] T. Wu, X. Ding, and S. Wang. Video tracking using improved chamfer matching and particle filter. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, volume 3, pages 169–173, 2007.

[21] Alper Yilmaz, Xin Li, and Mubarak Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11):1531–1536, 2004.