

OBJECT SHAPE DELINEATION DURING TRACKING PROCESS

Jean Gao

Department of Computer Science and Engineering
University of Texas, Arlington
Arlington, TX 76019
gao@cse.uta.edu

ABSTRACT

Object shape delineation during the tracking process plays important roles in correctly interpreting tracked results, providing visually meaningful outcomes, and furthermore assisting better motion estimation. For the majority of object tracking scenario, the emphasis has been put on achieving robust motion estimation in different situations; and object shape delineation, though critical, has not been paid enough attention due to its ill-posed nature. Approaches have been proposed by assuming the similarity of object pixels in the vicinities of the boundaries between the current frame and the previous one. Such an assumption is usually broken down when occlusion occurs; instead, our implementation is based on a stronger assumption: the local properties of object silhouette should be similar to those of the nearby object pixels. In this paper, we are going to address how to depict object boundary by a novel double-region growing and statistical pattern classification approach. Different from using a single point as a seed as which is a typical way for region growing, our seeds are segmented contours; also instead of growing outward in a single direction from the seed, we propose a two-directional region growing approach. Finally the best object boundary candidates are arbitrated from the dual-region growing results by a statistical classification approach.

1. INTRODUCTION

In this paper, we introduce an approach to the tracking of rigid objects with an emphasis on object shape delineation when self-occlusion occurs. Our approach is a feature-based one in the sense the object motion estimation is resulted from feature correspondences. To handle the problem of self-occlusion, we will present a multi-frame motion estimation scheme which is based on the genesis frames in which features are first registered. Our system also keeps extracting the object boundary during the tracking process.

As obvious as it is, some of the boundary points on an object undergoing motion – especially rotation – will be-

come occluded, while other points on the object surface will become the boundary points in the image captured by the camera. The goal of object shape recovery is to make a best attempt at extracting the silhouette of the object in the current frame.

The problem of silhouette extraction of a moving object undergoing rotation is highly ill-posed [1, 3, 8, 7, 9, 4]. All proposed solutions are based on the assumption that the object pixels in the vicinity of the boundary in the current frame possess texture and color properties similar to the object pixels in the vicinity of the boundary in the previous frame. But this assumption is often not satisfied by real-world objects when occlusion occurs.

For the boundary updating, while the implementations of other researchers' are based on the assumption that the local properties of the boundary pixels in the current frame should be similar to the local properties of boundary pixels in the previous frame, our implementation is based on a stronger assumption that the local properties of the boundary pixels in the current frame should be similar to the local properties of the nearby interior pixels in the *same* frame.

2. OBJECT MOTION ESTIMATION

Our tracking process begins with an automatic selection of feature points inside and on the boundary of the region of interest (ROI). The interior features contribute to the motion estimation, while the boundary ones delineate the tracked object. For initially registered interior feature points, we keep tracking them until they disappear due to self-occlusion or feature mis-match, and in the meantime, new features are registered as time goes on.

To estimate object motion when self-occlusion occurs, or in another word, when some of the initially selected features disappear, we first group the features in current frames according to their different starting frames. Let n be the index of current frame, 0 be the first frame of an image sequence, and k be the frame where feature i is first registered. Our goal is, of course, to estimate the object motion from

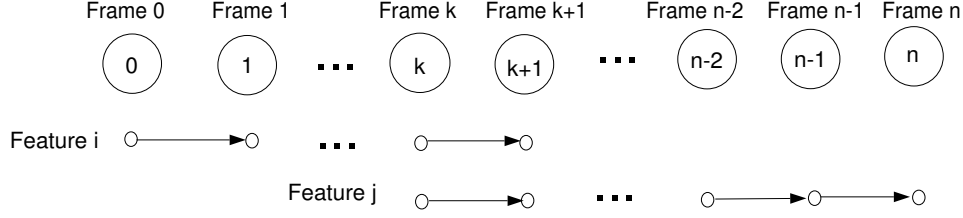


Fig. 1. Interior feature point flow, where feature i starts in frame 0 and disappears in frame $k + 2$, and feature j starts in frame k and is still active in current frame n

frame 0 to n . Toward that end, we will group the features in the current frame on the basis of the *genesis frame* for the features. The genesis frame for a given feature is that frame in which it first makes its appearance.

Let the S groups of features formed according to their genesis indices be denoted $\vec{G}_i, i = 0, \dots, S - 1$. For features within a group \vec{G}_i whose genesis index is k , we do feature extraction from frame k to n based on the normalized cross-correlation and motion estimation of ${}_n\mathbf{U}_k : ({}_n\bar{\mathbf{p}}_{k,n} \Sigma_k)$ by multi-visit of extended Kalman filtering (EKF) [5]. Here ${}_n\mathbf{U}_k : ({}_n\bar{\mathbf{p}}_{k,n} \Sigma_k)$ is the motion uncertainty from frame k to n of which the first element is the mean motion vector, meaning the 3D pose change from frame k to frame n , and the second element is the covariance associated with this pose change.

Given motion estimates ${}_n\mathbf{U}_k$ in the current frame n for different values of genesis frame index k , our goal now is to estimate ${}_n\mathbf{U}_0$. In terms of the genesis frame index k , the motion transformations \mathbf{H} 's among initial frame 0, genesis frame k and current frame n are related by

$${}_n\mathbf{H}_0 = {}_n\mathbf{H}_k {}_k\mathbf{H}_0 \quad (1)$$

Since ${}_k\mathbf{H}_0$ was already estimated when the current frame index was k , and since we have available to us ${}_n\mathbf{H}_k$ from the two-frame motion estimation, we can use Eq.(1) to update the uncertainty associated with ${}_n\mathbf{H}_0$. To integrate all estimates for different values of k between 0 and the current frame index n . We proposed a Kalman framework to bring about this integration. In this framework, we treat ${}_n\bar{\mathbf{p}}_0$ as the “state vector” to be estimates from the “measurements” ${}_n\bar{\mathbf{p}}_k$ and ${}_k\bar{\mathbf{p}}_0$ for different values of k . A set of constraint functions expanded from Eq. (1) are linearized for the updating. Mathematical details will not be provided here [6].

3. DUAL REGION-GROWING FOR BOUNDARY POINT DETECTION

Once we obtained the motion estimation from frame 0 to frame n , as described in the previous section, the motion parameters between the two adjacent frames $n - 1$ and n

can be calculated through:

$${}_n\mathbf{H}_{n-1} = {}_n\mathbf{H}_0 ({}_{n-1}\mathbf{H}_0)^{-1} \quad (2)$$

The initially predicted object boundary \mathcal{B}_n^{\sim} in frame n can be obtained by perspective transformation using the motion transform ${}_n\mathbf{H}_{n-1}$. Based on the predicted boundary \mathcal{B}_n^{\sim} , we define \mathcal{B}^{out} as its dilation, and \mathcal{B}^{in} as its erosion. In Fig. 2(a), the contours in yellow, blue, and green show \mathcal{B}_n^{\sim} , \mathcal{B}^{in} and \mathcal{B}^{out} respectively. The uncertainty field is defined as the region between \mathcal{B}^{out} and \mathcal{B}^{in} . We will now present a framework to locate the real object boundary \mathcal{B}_n within this field.

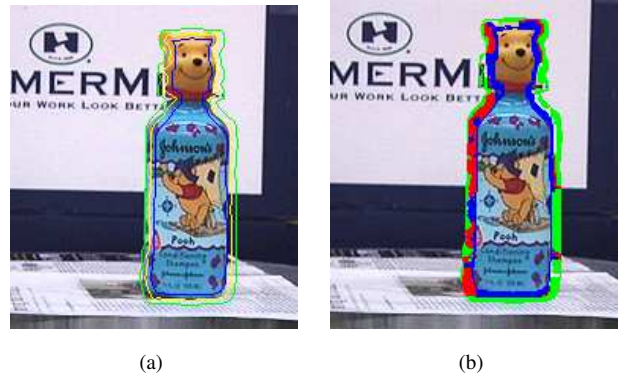


Fig. 2. (a) Boundary potential field definition. (b) Dual region growing result where the red pixels show the overlap of inward and outward region growing, and blue and green are outward and inward grown points separately.

3.1. Outward Region-Growing

We first introduce the outward region-growing from the eroded ROI boundary \mathcal{B}^{in} . Outward region growing here means the eroded ROI boundary \mathcal{B}^{in} is expanded outside into the uncertainty field based on certain similarity and discontinuity measures.

Seed Segment Selection:

We start the outward region-growing from the selection of a

set of seeds on the contour \mathcal{B}^{in} . Instead of choosing single points as seeds, the seeds here are chosen as the segmented m segments $\mathcal{C}_i^{in}, i = 0, \dots, m - 1$ on the contour \mathcal{B}^{in} . The squared Fisher distance is used as the similarity criterion to segment the contour:

$$D_{Fisher}^2 = \frac{(n_1 + n_2)(\hat{\mu}_1 - \hat{\mu}_2)^2}{n_1\hat{\sigma}_1^2 + n_2\hat{\sigma}_2^2} \quad (3)$$

where $n_1, n_2, \hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1^2, \hat{\sigma}_2^2$ are the sizes, means, and variances of the two adjacent contour segments on \mathcal{B}^{in} . In Fig. 3, the segmented seed contours on \mathcal{B}^{in} are represented in blue and white “x”s.

Region Growing by Point Aggregation:

The above selected contour segments then grow outside into the uncertainty field. A neighbor pixel (x, y) is joined to the current contour segment region with satisfying of the following intensity and chromacity criteria:

$$\left\{ \begin{array}{l} |I(x, y) - \hat{\mu}_i^I| \leq 2\hat{\sigma}_i^I \\ |r(x, y) - \hat{\mu}_i^r| \leq 2\hat{\sigma}_i^r \\ |g(x, y) - \hat{\mu}_i^g| \leq 2\hat{\sigma}_i^g \end{array} \right. \quad (4)$$

where $I, r,$ and g refer to the image intensity, normalized red, and green colors.

All the seed segments grow simultaneously during each iteration and act like water front. Fig. 3 shows a visual demonstration of the seed selection and the point aggregation process. All the blue solid dots depict outward region-grown points.

3.2. Inward Region-Growing

The inward region-growing performs a similar operation as the above outward one except with a different growing direction.

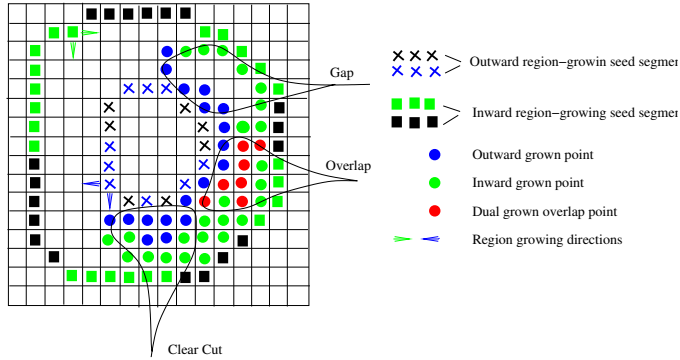


Fig. 3. Dual region growing.

3.3. Boundary Points Arbitrating from Results of Dual Region-Growing

During the dual region-growing, under the assumption that the intensity and chromacity of the eroded boundary are similar to those of the actual object boundary, the object boundary can always be flooded by the growing of eroded contour. The ideal situation after dual region-growing is that the grown regions from the two directions, outward and inward, meet each other and the object boundary is at the meeting point. But the actual reality is not the case. As we can see from Fig. 3, after dual region-growing, there will be three different situations:

Clear cut:

In this case, the inward growing region and outward growing one confront with each other. This cut provides the object boundary we pursue.

Gap:

When the contours \mathcal{B}_{in} or \mathcal{B}_{out} don't represent all pixel predicates within the uncertainty field, there will be gaps between two grown regions. Under the assumption that predicates of \mathcal{B}_{in} represent those of the boundary, the true object boundary can always be grown from \mathcal{B}_{in} . In this case, it is plausible to take the most outward grown points as boundary points.

Overlap:

When a pixel satisfies the predicates of both inside eroded contour \mathcal{B}_{in} and outside dilated contour \mathcal{B}_{out} , it will be labeled as the grown point from both directions.

For the overlap region, starting point A is where the inward growing part first meets outward growing parts, and point B is where the meeting is over as shown in Fig. 4(a). To locate object boundary pixels within overlap region, it is equal to trace boundary points starting from point A .

Let's say point s is the previous boundary point, and point p is current boundary point. The next possible boundary candidate q is chosen within the 8 possible neighborhood of point p . And the candidate is chosen by producing the maximum difference or separability between a local inside window and a local outside window cut by the link from p to q . The local inside and outside windows are represented as In and Out in Fig. 4(b). If we choose both two local windows the same size, the difference on two sides of the link pq can be measured by squared Fisher distance which in this case would be,

$$\arg \max_q = \arg \max D_{Fisher}^2 = \frac{(\hat{\mu}_1 - \hat{\mu}_2)^2}{\hat{\sigma}_1^2 + \hat{\sigma}_2^2} \quad (5)$$

where as before $\hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1,$ and $\hat{\sigma}_2$ are estimated local statistic parameters of chosen windows.

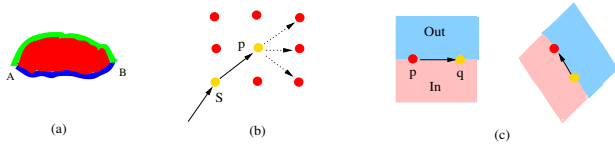


Fig. 4. Boundary point tracking within overlap regions. (a) Red part shows the overlapped region. (b) Possible boundary candidate q is selected within 8 neighborhood of current boundary point p . (c) Two possible directions for candidate selection.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Object silhouette delineation during tracking is a challenging problem in the sense of unpredictability and ill-posedness especially when occlusion occurs. In this paper, we presented a computational scheme to tackle this problem by a novel region growing approach. Fig. 5 demonstrates an example to show the performance of our algorithms. The object is a Pooh shampoo container sitting on a rotating table. The table rotates counter-clockwise which causes the container turn around from left to right. We can see the self-occlusion from the position of Pooh's nose and the texture content change of the container. In spite of the rotation, our algorithm can still follow the object and well depict the contour of the object.

5. REFERENCES

[1] I. Celasun, A. Tekalp, M. Gokcetekin, and D.M. Harmandi, "2-D Mesh-Based Video Object Segmentation and Tracking with Occlusion Resolution," *Signal Processing: Image Communication*, Vol. 16, pp. 949-962, 2001.

[2] Y. Yao and R. Chellappa, Tracking a Dynamic Set of Feature Points, *IEEE Transactions on Image Processing*, pp. 1382-1395, Vol. 4, No. 10, 1995.

[3] C. Toklu, A. Tekalp, and A. Erdem, "Semi-Automatic Video Object Segmentation in the Presence of Occlusion," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 10, No. 4, pp. 624-629, 2000.

[4] Yue Fu, Erdem AT, and Tekalp AM, "Tracking visible boundary of objects using occlusion adaptive motion snake," *IEEE Transactions of Image Processing*, pp. 2051-60, vol.9, no.12, Dec. 2000.

[5] J. Gao and A. Kosaka, "Circularly Establishing Feature Correspondences and Motion Estimation for Object Tracking via Multi-Visit of Kalman Filtering," *Proceedings of SPIE Electronic Imaging*, Jan, 2002.

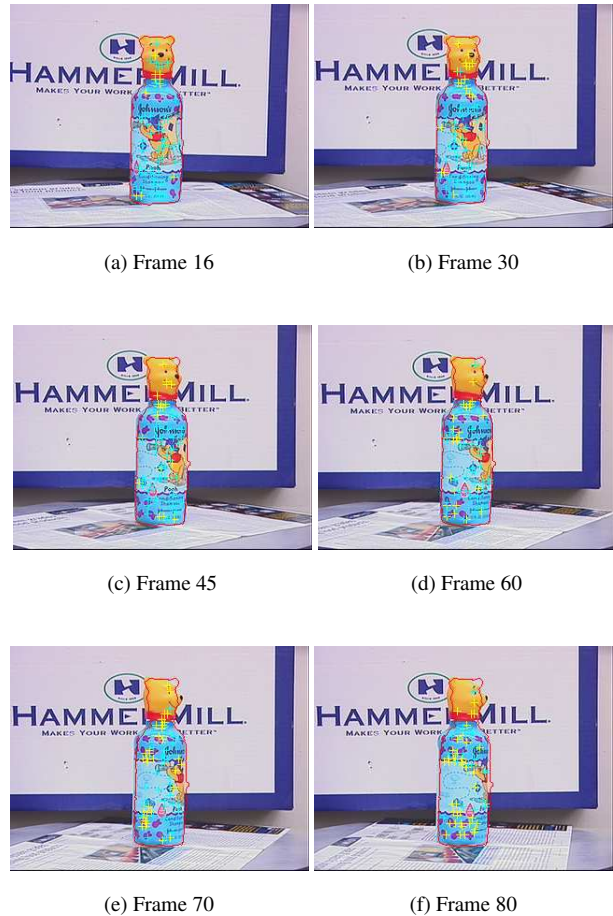


Fig. 5. The result of dual-region growing for rotating object tracking.

[6] J. Gao, "Self-Occlusion Immune Video Tracking of Objects in Cluttered Environments," *Proceedings of IEEE Advanced Video and Signal Processing based Surveillance*, Miami, FL, 2003.

[7] N. Peterfreund, "Robust tracking of position and velocity with Kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.21, No.6, pp.564-9, June, 1999.

[8] M. Kim, J.G. Jeon, J.S. Kwak, M.H. Lee, and C. Ahn, "Moving Object Segmentation in Video Sequences by User Interaction and Automatic Object Tracking," *Image and Vision Computing*, 245-260, Vol.19, No.5, 2001.

[9] Yu Zhong, A.K Jain, Dubuisson-Jolly M-P, "Object tracking using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.22, No.5, pp. 544-9, May 2000.