OBJECT CONTOUR TRACKING USING GRAPH CUTS BASED ACTIVE CONTOURS

Ning Xu and Narendra Ahuja

Beckman Institute and ECE Department
University of Illinois at Urbana-Champaign
{ningxu,ahuja}@vision.ai.uiuc.edu

ABSTRACT

In this paper, we present an object contour tracking approach using graph cuts based active contours (GCBAC). Our proposed algorithm does not need any *a priori* global shape model, which makes it useful for tracking objects with deformable shapes and appearances. GCBAC are not sensitive to initial conditions and always converge to the optimal contour within the dilated neighborhood of itself. Given an initial boundary near the object in the first frame, GCBAC can iteratively converge to an optimal object boundary. In each frame thereafter, the resulting contour in the previous frame is taken as initialization and the algorithm consists of two steps. In the first step, GCBAC are applied to the difference between this frame and its previous one. The resulting contour is taken as initialization of the second step, which applies GCBAC to current frame directly. To evaluate the tracking performance, we apply the algorithm to several real world video sequences. Experimental results are provided.

1. INTRODUCTION

Object contour tracking is useful in various real world applications, such as video surveillance and video conferencing. All these applications require robust and efficient visual tracking which is still an open problem. In most of the applications, the camera keeps still so that the background is unchanged while the objects are moving within the field of view. This makes the tracking problem easier than the one with a changing background. However, there are still many difficulties even in this simplified scenario, such as the presence of other moving objects.

Various contour tracking approaches have been proposed in computer vision literature. Snakes [1, 2], the active contour models, have proved to be very effective in extracting object contours in images and tracking them in video. Snakes depend upon external and internal energies to pull the contours towards desired image features while keeping the contour smooth. Many methods, such as dynamic programming [3, 4] and level sets [5] have been proposed to drive the snakes gradually to the desired place. However, the contour evolution may easily get stuck in local minima and is thus sensitive to initial conditions. Parametric models, where the deforming contour is described using a few parameters [6], have been successfully used for the cases where the topology of the extracted contour is simple. The condensation algorithm [7] explores the prior knowledge of shape and motion by using a stochastic framework and propagates the conditional probability densities over time. However, it requires accurate models for both shape and motion dynamics and the required number of samples grow exponentially with the dimension of the state space. The optimal radial contour method of [8] uses a radial contour representation together with some *a priori* global shape model which allows the global optimal solution to be found using dynamic programming.

In this paper, we present an approach which uses graph cuts based active contours (GCBAC) [9] to track object contours in video sequences captured from a single still camera. Unlike most of the other active contour methods, GCBAC do not get stuck in local minima and are not sensitive to initial conditions. Given an initial boundary near the object, GCBAC can iteratively deform to the desired object boundary. In each frame, the resulting contour in the previous frame is taken as initialization and our approach takes advantage of both the intensity information within this frame and the difference between this frame and its previous one to find the new object contour using GCBAC.

In Section 2, we review the GCBAC approach and present our tracking algorithm in detail. Experimental results are provided in Section 3, and Section 4 presents some concluding remarks.

2. OUR APPROACH

2.1. Graph cuts based active contours

The graph cuts based active contours(GCBAC) approach was first proposed in [9] for iterative object segmentation. In each iteration, the object segmentation problem is formulated as a multi-source multi-sink s-t minimum cut problem. Graph-theoretic description of single $s-t$ minimum cut can be found in many graph theory textbooks [10, 11]. The minimum cut considered in this paper is required to separate multiple source nodes from multiple sink nodes. An operation on a graph G called *node identification* identifies a set of nodes $\{v_1, v_2, ..., v_n\}$ as a single new node v, deleting self loops, if any, and merging parallel edges, as shown in Fig.1. *Node identification* allows us to use single $s-t$ minimum cut algorithms to solve the multi-source multi-sink minimum cut problem by merging the multi-source into a single source and the multi-sink into a single sink respectively.

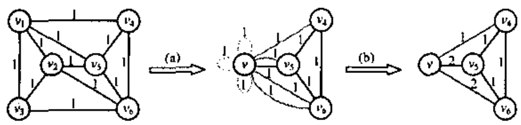


Fig. 1. Node identification. (a) Node v_1, v_2, v_3 are merged into a new node v. (b) Self loops are deleted and parallel edges are replaced by a single edge.

Given an initial boundary, the GCBAC algorithm consists of the following steps:

1. Dilate current boundary into an area of interest with an inner boundary and an outer boundary, as shown in Fig.2.
2. Represent the data within the area of interest using an adjacency graph of edge connectivity. Edge weights are defined as in [9].
3. Identify all the nodes on the inner boundary as a single source s and identify all the nodes on the outer boundary as a single sink t.
4. Compute the $s-t$ minimum cut to identify a new boundary that optimally separates the inner boundary from the outer boundary.
5. Return to step 1 until the algorithm converges.

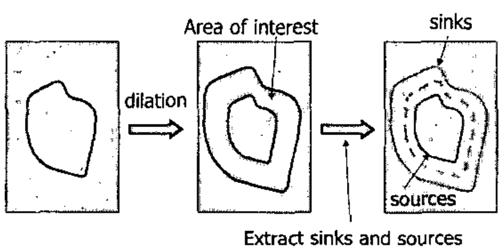


Fig. 2. Using dilation to get sources, sinks and area of interest.

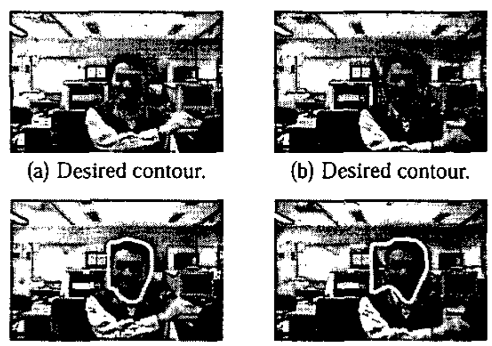
Dilation used in step 1 leads to several advantages. First, it generates an area of interest in which the resulting boundary is globally optimal. Second, if the data is homogeneous within the area of interest, since the cost function is the summation of the edge weights along the cut, the contour will shrink. This is useful when the object lies within a simple background and the initial contour is much bigger than the real object contour. In that case, the GCBAC will contract until it hits the object boundary. Third, dilation generates an inner boundary which is grouped as multiple sources and is always contained in the S part of the resulting $s-t$ minimum cut. Since the min-cut is prone to yield a small region, the use of *node identification* ensures that the resulting boundary in each step should be bigger than the inner boundary of previous one.

2.2. Object contour tracking

The resulting object contour in the previous frame is a good initialization for our algorithm in current frame of the video, if we assume that the object doesn't move too fast and the position of the object doesn't change a lot in consecutive frames.

In this section, we propose to incorporate both the intensity information of current frame and the difference between current frame and the previous one to track the object contour, with the aim of improving results over the case when either information is used alone.

2.2.1. Tracking using only the intensity information of current frame



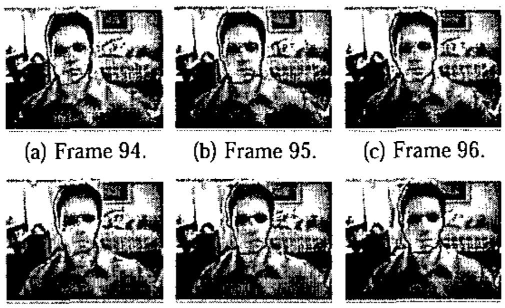
(a) Desired contour. (b) Desired contour.

(c) Resulting contour. (d) Resulting contour.

Fig. 3. Errors when single frame intensity information is used. (a, b) Desired object contours. (c) Resulting contour is distracted by the cluttered background. (d) The error accumulates with frames.

We first apply the GCBAC approach to the video sequences using only the intensity information. Taking the resulting contour in the previous frame as initialization, the algorithm works well for the cases where the background is simple. However, it fails when the background together with the object provides some false contour which is the optimal one within its neighborhood. It becomes even worse when the error accumulates. Since we do not have an *a priori* global shape model to control tracking, the false contour may wander away from the desired contour and finally get lost. Fig.3(a) shows an example of the desired contour. Since there is a strong vertical edge to the left and above, the contour shown in Fig.3(c) is the result we might get in this frame. When the object is moving to the right, the error accumulates and the contour will be distracted far away, as in Fig.3(d), from the desired one shown in Fig.3(b).

2.2.2. Tracking using differences between two consecutive frames



(a) Frame 94. (b) Frame 95. (c) Frame 96.

(d) Frame 97. (e) Frame 98. (f) Frame 99.

Fig. 4. Errors of only using differences between consecutive frames. (a)-(f) show 6 consecutive frames of a video sequence and an object is moving from left to right. In each frame, the resulting contour shows a lag on the side opposite to the direction of movement.

Since in most cases, the camera keeps still while capturing the video sequences, we can take advantage of this and use differences between two consecutive frames to eliminate the effect of complex background. The difference between current frame n, and previous frame $n-1$, is computed as $\delta I = \mathrm{abs}(I_n - I_{n-1})$. This difference information is used to compute edge weights in the corresponding graphs in GCBAC algorithm and accordingly we can expect that the resulting contour will not be distracted by the background since the background is cancelled. However, the tracking result is not accurate since the difference information between two frames does not provide accurate information about where the real boundary is. As in Fig.4, the result of each frame shows a lag on the side opposite to the direction of movement. Fig.4(a)-(f) are 6 consecutive frames and the person is moving from left to right. As we can see in the figure, the resulting contours are not accurate at the left side of the boundary—some additional areas are included in the resulting contours. It is because these areas are within the object boundaries in the previous frames. Also, if an object keeps still for a while, the tracker will get lost since no difference is found between two consecutive frames.

2.2.3. A two-step tracking algorithm

Our tracking algorithm is a combination of the two processes mentioned above. The flowchart of the algorithm is shown in Fig.5. First, we apply the GCBAC on the difference data which is defined as $\delta I = \mathrm{abs}(I_n - I_{n-1})$. Second, using the resulting contour from the first step as initialization, we apply the GCBAC again to the image data of current frame to get a final result. We use a heuristic technique in the first step to prevent the error caused by the low difference between two consecutive frames. If the amount of difference within a neighbor area of the initial contour is less than a preset threshold, we consider that the object is not moving and the initial contour is sent directly to the second step. So the initialization of the second step will be either the initial contour, or the resulting contour of the first step. The key advantage is that the errors caused by the background no longer accumulate, and at the same time, the tracked contour is more accurate than when only difference information is used.

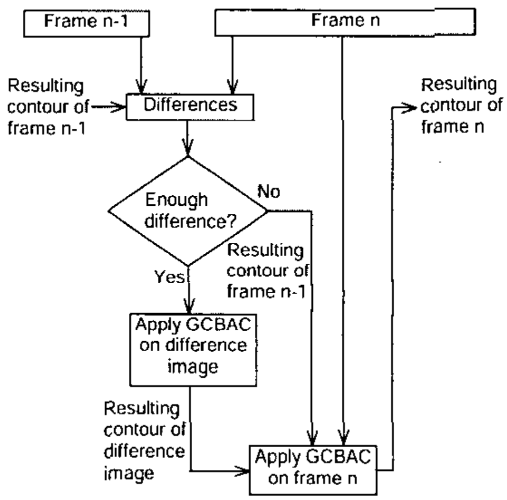
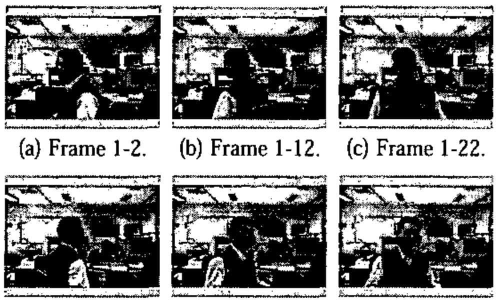


Fig. 5. Sketch of our algorithm using GCBAC.

3. EXPERIMENTAL RESULTS

To validate the robustness and efficacy of our GCBAC based contour tracking algorithm, we test it on several video sequences with cluttered background. In the test sequences, the object motions include translation and rotation. The translation could be in any direction so that the object size might be changing. The rotation of the object changes its shape and appearance. Also, in our test sequences, the object may be occluded by another moving object. Fig.6 shows that our algorithm can handle the rotation of the head, where the appearance and shape are changing. Fig.7 shows a different rotation and translation of the head. Fig.8 shows the

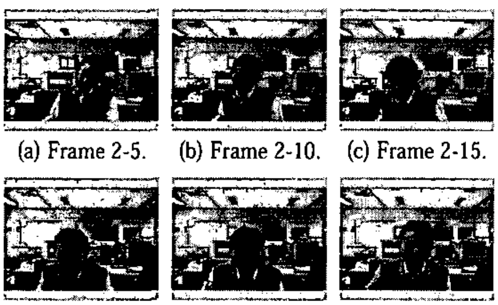


(a) Frame 1-2. (b) Frame 1-12. (c) Frame 1-22.

(d) Frame 1-32. (e) Frame 1-42. (f) Frame 1-52.

Fig. 6. In this sequence, the head is rotating so that the tracked object is changing its appearance and shape.

robustness of our algorithm when a moving hand occludes the face in part of the sequence.[1]



(a) Frame 2-5. (b) Frame 2-10. (c) Frame 2-15.

(d) Frame 2-20. (e) Frame 2-25. (f) Frame 2-30.

Fig. 7. In this sequence, the head is rotating and translating.

4. DISCUSSION

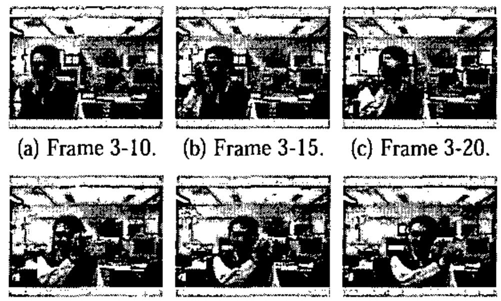
In this paper, we proposed a two step contour tracking approach using graph cuts based active contours (GCBAC). This approach takes advantages of both the intensity information of a frame and the difference between consecutive frames. The approach does not require an *a priori* global shape model and the resulting contours are globally optimal within their neighborhoods. The experimental results are promising.

The most time consuming part of the proposed approach is the $s-t$ minimum cut algorithm, which runs in polynomial time. The ongoing work includes implementing faster $s-t$ minimum cut algorithm using a divide and conquer strategy.

5. REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, pp. 321–331, 1988.

[1] Video results available at http://vision.ai.uiuc.edu/~ningxu/tracking.



(a) Frame 3-10. (b) Frame 3-15. (c) Frame 3-20.

(d) Frame 3-25. (e) Frame 3-30. (f) Frame 3-35.

Fig. 8. The head is occluded by a moving hand while it is translating from left to right.

[2] D. Terzopoulos and R. Szeliski, "Tracking with kalman snakes," *Active Vision*, pp. 3–20, 1992, A. Blake and A. Yuille, Eds.

[3] Amir Amini et al., "Using dynamic programming for solving variational problems in vision," *IEEE Trans. on PAMI*, vol. 12, no. 9, September 1990.

[4] Davi Geiger, Alok Gupta, Luiz A. Costa, and John Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. on PAMI*, vol. 17, no. 3, March 1995.

[5] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, 1995.

[6] L. H. Staib and J. S. Duncan, "Boundary finding with parametrically deformable models," , *IEEE Trans on PAMI*, vol. 14, no. 11, pp. 90–104, 1992.

[7] Michael. Isard and Andrew Blake, "Condensation—conditional density propagation for visual tracking," *Int. J. Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[8] Yunqiang Chen, Yong Rui, and Thomas S. Huang, "Optimal radial contour tracking by dynamic programming," *ICIP01*, 2001.

[9] Ning Xu, Ravi Bansal, and Narendra Ahuja, "Object boundary segmentation using graph cuts based active contours," *CVPR2001 Technical Sketches*, pp. 87–90, Dec. 2001.

[10] Ravindra K. Ahuja, Thomas Magnanti, and James Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, 1993.

[11] Thomas Cormen, Charles Leiserson, and Ronald Rivest, *Introduction to Algorithms*, McGraw-Hill Companies, 1990.