Multi-Object Tracking Through Clutter Using Graph Cuts

James Malcolm Yogesh Rathi Allen Tannenbaum
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332-0250
{malcolm,yogesh.rathi,tannenba}@bme.gatech.edu

Abstract

The standard graph cut technique is a robust method for globally optimal image segmentations. However, because of its global nature, it is prone to capture outlying areas similar to the object of interest. This paper proposes a novel method to constrain the standard graph cut technique for tracking anywhere from one to several objects in regions of interest. For each object, we introduce a pixel penalty based upon distance from a region of interest and so segmentation is biased to remain in this area. Also, we employ a filter predicting the location of the object. The distance penalty is then centered at this location and adaptively scaled based on prediction confidence. This method is capable of tracking multiple interacting objects of different intensity profiles in both gray-scale and color imagery.

1. Introduction

Tracking an object in video has been the focus of much research, and the problems accompanying this key task are well-known. For example, the object might have weak edges causing a given edge-based active contour segmentation method to leak out into the surrounding area, or the object may suddenly move outside the algorithm's region of detection, or the object may be near other objects of similar intensity causing unintended objects to be tracked. Multi-object tracking raises additional concerns such as the interaction among objects.

Various methods have been proposed to overcome these difficulties. To keep segmentations from spilling over object boundaries, learned shape priors constrain segmentation to a set of possible shapes [8, 9, 14]. To account for object movement, motion models can predict the likely location of the object in subsequent frames [7, 11]. When adjacent regions are similar to the object of interest, multiple hypothesis trackers can keep track of each region while determining the most likely in each frame based on some criteria [1, 12, 15, 19]. To segment multiple unique objects simultaneously, techniques have been developed to take into account the interaction among objects [23].

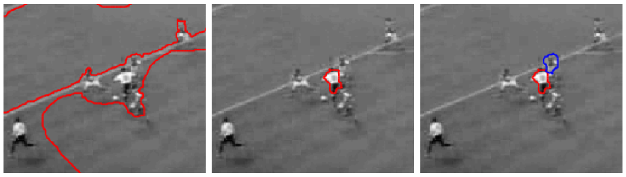

Figure 1. Tracking two interacting soccer players among others of similar intensity: no distance penalty and applying distance penalty to track one or two players *(left to right)*. Without the distance penalty, multiple non-intended regions were captured.

Graph cut techniques have received considerable attention as robust methods for energy minimization. Despite their success for such key vision tasks as image segmentation and stereo disparity, graph cuts have received little attention with respect to tracking. This is largely due to the global segmentations they produce which tend to catch unintended regions that are similar to the object of interest. For example, the standard graph cut technique for image segmentation [4] finds regions with high likelihood given intensity priors. Figure 1 shows an example where there are multiple regions of similar intensity. The standard graph cut algorithm captures such regions. Post-processing must be performed to filter out those regions that are not part of the object. However, this same feature, that of capturing such regions anywhere in the image, naturally addresses the problem of large object movements. The graph cut will find the object even if it moved far relative to its location in the previous frame. The problem is now one of constraining the graph cut to capture only the objects of interest, even if they made large movements yet ignoring other regions of similar intensity. Hence a spatial constraint is needed.

Several techniques have used graph cuts for segmentation in visual tracking applications. In [24] the segmentation is constrained to a narrow band. For each frame, successive graph cut segmentations converge on a final segmentation, each pass constrained to a narrow band around the cut boundary resulting from the previous pass. This method is dependent upon initial contour placement and re-

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

quires repeated cuts on this reduced domain. In [10] the authors use one graph cut for each frame to both estimate the optical flow and object position despite changes in illumination. However, since optical flow requires the multi-label graph cut technique [6] and the graph proposed has such dense neighborhoods, the current approach requires about a minute per frame. Also, due to the local nature of optical flow, the technique cannot handle large movements.

Besides tracking, work has been done to constrain segmentation based on a user selected region. The work of [20] begins with a rectangle bounding the object, while the work of [2] uses a narrow band. Both perform successive graph cut segmentations incorporating additional user interaction with each pass. Neither method is targeted towards tracking *per se*, but instead seeks a perfect segmentation. In these works, hard constraints confine the segmentation within a user-selected region and multiple graph cuts are performed. In our work, the object may be found a distance from the predicted centroid depending on the scale of the distance penalty, and segmentation is performed only once per frame. None of these methods has been generalized to simultaneously segment multiple unique objects.

Recently, we began work to constrain the standard graph cut segmentation for tracking by predicting object location and forming a basin of attraction at the predicted location [18]. Using the binary graph cut method, the proposed technique was able to track one object among background clutter. We also demonstrated tracking multiple objects, however, this worked only if the objects were all of similar intensity and did not interact. This present work extends our preliminary results by generalizing the technique to capture an arbitrary number of objects. Considering each region to have its own label, the multi-label graph cut lends itself naturally to segmenting multiple interacting objects, each with unique intensity profile.

In this present work, the basic algorithm is as follows. First, for each object, we first incorporate a distance penalty into the graph cut algorithm to bias segmentations to a region likely to contain the object. Second, we present a simple filter to predict the location of that object based on the location of the previous segmentation and a moving average of the object's velocity. The distance penalty is then centered at the predicted object centroid and extends outward forming a basin of attraction. Third, to further integrate the filter with the distance penalty, the scale of this distance penalty, and hence the slope of its surface, is adaptively set based on the prediction error. Finally, the interaction among objects is naturally handled as segmentation is performed in one cut using the standard multi-label graph cut algorithm.

The method presented here represents several useful contributions to the field of visual tracking. First, to bias graph cut segmentations to regions of interest, the distance penalty introduces a per-object spatial prior based on predicated location. Second, the graph cut edge weights are adaptively determined by the prediction error. And lastly, the multi-label graph cut technique uniquely captures multiple interacting objects in one cut.

The rest of the paper is organized as follows. Section 2 outlines the standard graph cut segmentation framework. Section 3 describes the distance penalty constraining segmentation. Section 4 defines the filter used to predict the object centroid. Section 5 integrates the filter prediction error with the distance penalty. Next, in Sections 6 and 7, we present our algorithm and results. Finally, in Section 8 we summarize our work and describe some possible future research directions.

2. Graph cuts

In this section, we briefly outline the standard multi-label graph cut technique; for more details see [2, 3, 4, 6, 20] and the references therein.

Taking advantage of efficient algorithms for global min-cut solutions, we cast the energy-based image segmentation problem in a graph structure of which the min-cut corresponds to a globally optimal segmentation. Evaluated for an assignment A of each pixel to a label $f \in \mathcal{F}$, such energies are designed as a data dependent term and a smoothness term. The data dependent term evaluates the penalty for assigning a particular pixel to a given label. The smoothness term evaluates the penalty for assigning two neighboring pixels to different regions, *i.e.* a boundary discontinuity. These two terms may be thought of as a regional term and a boundary term, often weighted by $\lambda \geq 0$ for relative influence:

$$E(A) = \sum_{p \in I} R_p(A_p) + \lambda \sum_{\substack{(p,q)\in\mathcal{N} \\ A_p \neq A_q}} B_{(p,q)} \tag{1}$$

where I represents all image pixels, $\mathcal{N}$ all unordered neighborhood pixel pairs. The choice of neighborhood size and structure has a large influence on the solution as smaller neighborhoods tend to introduce metrication artifacts [5].

To construct the graph representing this energy, each pixel is considered as a graph node in addition to an extra node for each region label $f \in \mathcal{F}$, *e.g.* background, first object, second object. The data dependent term is implemented by connecting each pixel to these extra nodes with non-negative edge weights $R_p(f)$ representing the penalty for assigning pixel p to the region f. Lastly, the smoothness term is implemented by connecting each pairwise combination of neighboring pixels (p,q) with a non-negative edge weight $B_{(p,q)}$ representing the penalty for assigning pixels p and q to different regions. The min-cut of the weighted graph represents the segmentation that best separates the regions. See [4, 6] for more details.

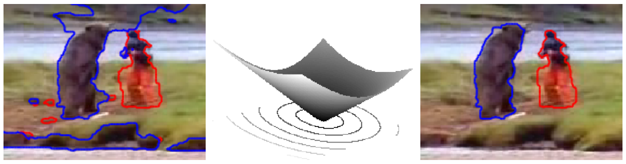

Figure 2. Tracking a bear near other regions similar to its fur: no distance penalty, distance penalty ϕ with isocontours, and applying distance penalty *(left to right)*. Without the distance penalty, multiple non-intended regions were captured.

Typical applications of graph cuts to image segmentation differ only in the definitions of R_p and $B_{(p,q)}$. For example, in the case of the binary foreground/background segmentation problem, the authors of [4] use the negative log-likelihood of a pixel's intensity to compute the regional weights while intensity contrast is used in the boundary term:

$$R_p(fg) = -\ln P(I_p|fg), \qquad R_p(bg) = -\ln P(I_p|bg),$$
$$B_{(p,q)} = \exp(\frac{-\|I_p - I_q\|^2}{2\sigma^2})\frac{1}{\|p-q\|} \qquad (2)$$

where $\|p-q\|$ is the standard L_2 Euclidean pixel distance in the image and $\sigma^2 = \langle \|I_p - I_q\|^2/\|p-q\|^2 \rangle$, the average contrast over all $(p,q) \in \mathcal{N}$. Initialization proceeds as in [4] where the user marks regions of foreground and background to generate the intensity histograms for each region.

3. Distance penalty

The standard graph cut technique is capable of finding regions matching the object intensity located anywhere in the image. However, by penalizing pixels based on their distance from the expected location, a potential well is formed biasing segmentation to a region of interest. Figure 2 shows segmentation with and without such a penalty in the presence of unintended regions similar to the object.

The distance penalty ϕ is formed from a base mask M predicting the object shape. Centering that mask M at the predicted object location and assigning it zero penalty, each pixel x outside the mask is assigned its distance from the nearest masked pixel $m_x \in M$, *i.e.* $\phi(x) = \|x - m_x\|$ or zero if $x \in M$. Such a construction can be quickly computed with the Fast Marching algorithm [21, 25]. In this work, we used the initial user segmentation as the base mask M; however, several methods are available for representing more deformable shapes [10, 17, 22].

4. Location prediction

It is often the case that the object makes a large movement, large enough at times to place it in an area of high distance penalty. To overcome this problem, we predict the location of the object in each frame based on its previous location and center the distance penalty at this predicted location.

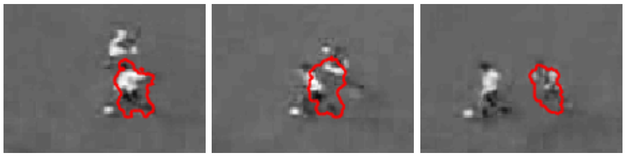

Figure 3. Without location prediction, tracking can fail when the target makes sudden movements. Here the tracker catches a defender as the target passes *(left to right)*.

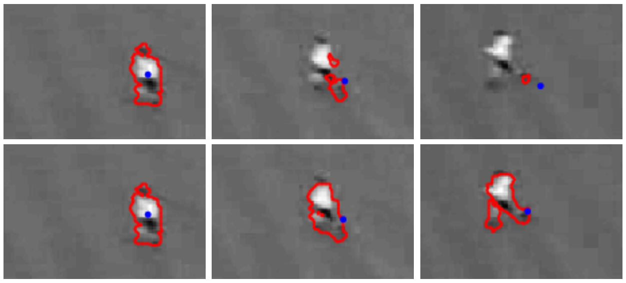

Figure 4. Effect of adaptive α on tracking: non-adaptive alpha (assume zero error) *(top, left to right)*, alpha with prediction error *(bottom, left to right)*. Tracking fails without using error feedback to scale distance penalty. Centroid is shown as blue dot.

To demonstrate the need for some form of prediction, we experimented with the assumption that the object has not moved: the distance penalty is centered at the last known object position. Figure 3 shows the failure to track after the object has made a sudden move, despite the use of adaptive α scaling described in Section 5. The movement placed the object too far outside of the basin of attraction.

Introducing simple prediction, we assume the object is traveling with continuous velocity, hence we predict the next object location $\tilde{c}_{t+1}$ based on projecting forward by the average displacement in the past few frames.

5. Error feedback

We now have the distance penalty constraining segmentation and the filter predicting where to center this distance penalty, but what if the filter is wrong? Figure 4 shows just such a case. The object has made a sudden move outside the predicted basin of attraction.

What is needed is a way of adaptively scaling the distance penalty based on the prediction error. In this work, we take the error in prediction to be the distance between the predicted $\tilde{c}$ and actual c centroids. The distance map is then scaled by $\alpha(\|\tilde{c} - c\|)$ taken from an exponential distribution of the prediction error $\alpha(x) = \exp(-x^2/\rho^2)$, where ρ is user specified based on empirical motion. The effect is that when the filter is off in its predictions of the object centroid, the distance penalty is lowered to hopefully still capture the object. After locking back onto the object, the α

automatically raises the distance penalty back up to tighten around the object as the prediction error decreases. Figure 4 shows how, despite incorrectly predicted centroids, the system is able to recover by adaptively widening the distance penalty.

6. Proposed algorithm

For each new frame and for each object, the algorithm predicts the object location, determines the distance penalty scaling based on prediction error, computes edge weights for the graph, and performs a graph cut segmentation. For initialization, the user is required to roughly mark in the first frame the object and background. This initialization defines both the intensity priors used in the regional edge weights (2) as well as the base mask M for each object.

In the prediction step, the centroid from the previous frame's segmentation is used as a measurement c. The filter predicts the object centroid location in this new frame $\tilde{c}$ from a moving average of displacements.

The $\alpha(\cdot)$ scaling function for the distance penalty is calculated from an exponential distribution of error $\|\tilde{c}-c\|$. Since the proposed simple filter is unstable against large displacements, we found the need to limit this distance in practice to a user-defined γ so that the distance penalty is not driven completely to zero. The $\alpha(\cdot)$ used is then:

$$\alpha(x) = \exp\left(-\min(x,\gamma)^2/\rho^2\right). \tag{3}$$

We propose a new regional edge weight to augment the standard weight in (2). Our goal is to determine $P(f|I)$ for each pixel, and Bayes rule tells us that $P(f|I) \propto P(I|f)P(f)$. If we assume $P(f)$ is uniform, then its negative log-likelihood is zero, and so it falls out of the expression as in the standard regional term (2). Here, we assume a non-uniform object prior $P(f)$ and claim: $-\ln P(f) \propto \alpha(\|\tilde{c}-c\|)\phi$. We assume the background region to still be uniformly distributed and so it does not have a distance penalty prior. Introducing a weight $\beta > 0$ for relative distance penalty influence, we have a new regional term:

$$\begin{aligned} R_p(f) &= -\ln P(I_p|f) - \beta \ln P_p(f) \\ &= -\ln P(I_p|f) + \beta\,\alpha(\|\tilde{c}-c\|)\phi(p) \end{aligned} \tag{4}$$

For all experiments, we use the standard intensity contrast smoothness term $B_{(p,q)}$ in (2). Finally, we take the min-cut of this graph to yield a multi-region segmentation.

7. Results

Tracking was performed on color and gray-scale videos and representative frames chosen to exhibit clutter with objects of similar intensity.

Parameters are defined as follows. For all experiments, we set $\lambda = 10$ in (1). Also for all experiments, objects are assumed to not move more than 5 pixels between frames so $\gamma = 5$ in (3) and in practice $\rho = \gamma/2$ is quite robust. In (4), we set $\beta = 10$ for gray-scale imagery and $\beta = 2$ for color.

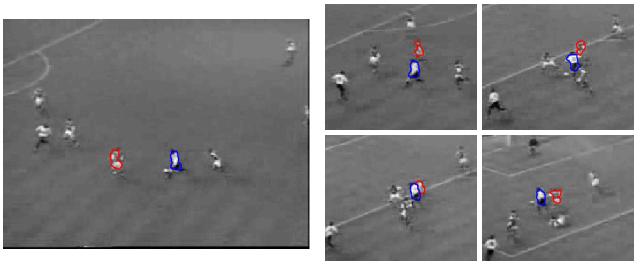

Figure 5. Tracking two opposing players from the soccer sequence. Despite prolonged contact and occlusion, the technique is able to uniquely track the two targets. Full image *(left)* and selected cropped frames *(right)*.

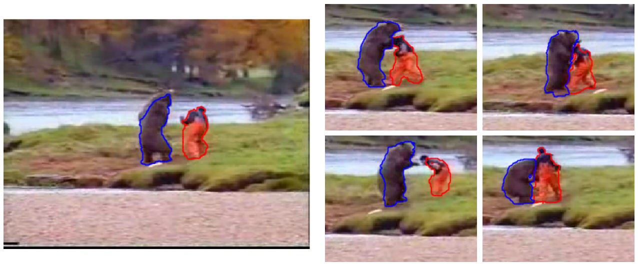

Figure 6. Tracking the bear and man in color. Due to large movements and changes in shape, at several points the tracker is partly thrown off, yet it recovers fully. Full image *(left)* and selected cropped frames *(right)*.

On a standard workstation, the current system tracks two objects at roughly two frames per second fluctuating little based on the neighborhood chosen. The choice of neighborhood also affects the smoothness of the segmentation with smaller neighborhoods tending to introduce irregular segmentations [5]. It is important to note that, since the segmentations for sizes 4 and 8 were not as smooth, they introduced larger variations in the calculated centroid and hence larger prediction errors. Increased smoothing (λ) was required to maintain track with smaller neighborhoods. Tracking with size 4 or 8 was therefore not as robust as size 16. Unless otherwise noted, results are shown with a neighborhood of size 16.

The gray-scale video sequence involves several soccer players of similar intensity, yet the intensity profile of each team differs enough that opposing players can be distinguished. Figure 5 shows tracking of a player from each team amidst occlusion and contact with several other players of similar intensity.

The color video sequence is a commercial faking a fight between a bear and a man. Figure 6 shows tracking of the bear and man as they make sudden movements or change shape. These sudden changes throw the tracker off but in all cases the tracker recovers fully in a few frames.

8. Conclusion

This paper demonstrates a distance penalty to constrain the standard graph cut segmentation to regions of interest. An observer is proposed to predict object locations while the prediction error is used to scale the distance penalties forming basins of attraction that are adaptively sized. The multi-label graph cut algorithm is then used to find the objects in one pass.

There are several future directions of research. Anisotropic distance penalties may be used to bias certain directions based on expected object trajectory. Instead of rebuilding the graph from scratch for each frame as in the current system, speed can be enhanced by updating the graph in place from frame to frame [13]. Furthermore, segmentation may be made more robust for a larger class of imagery by tracking in a feature space with more information than simple intensity [16].

9. Acknowledgments

This work was supported in part by grants from NSF, AFOSR, ARO, MURI, MRI-HEL as well as by a grant from NIH (NAC P41 RR-13218) through Brigham and Women's Hospital. This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. Information on the National Centers for Biomedical Computing can be obtained from `http://nihroadmap.nih.gov/bioinformatics`.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear non-gaussian bayesian tracking. *Trans. Signal Processing*, 55(2):174–188, 2002.

[2] A. Blake, C. Rother, M. Brown, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, volume 3021, pages 428–441, 2004.

[3] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *IJCV*, 70:109–131, 2006.

[4] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICIP*, pages 105–112, 2001.

[5] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, pages 26–33, 2003.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23:1222–1239, 2001.

[7] D. Cremers. Dynamical statistical shape priors for level set based tracking. *PAMI*, 28(8):1262–1273, 2006.

[8] D. Cremers, T. Kohlberger, and C. Schnorr. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36:1929–1943, 2003.

[9] S. Dambreville, Y. Rathi, and A. Tannenbaum. Shape-based approach to robust image segmentation using kernel PCA. In *CVPR*, pages 977–984, 2006.

[10] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, pages 755–762, 2005.

[11] R. Frezza, G. Picci, and S. Soatto. *A Lagrangian formulation of nonholonomic path following*, pages 118–133. The Confluence of Vision and Control. Springer Verlag, 1998.

[12] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[13] P. Kohli and P. Torr. Effciently solving dynamic markov random fields using graph cuts. In *ICCV*, pages 922–929, 2005.

[14] M. Leventon, E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *CVPR*, pages 1316–1324, 2000.

[15] E. Maggio and A. Cavallaro. Hybrid particle filter and mean shift tracker with adaptive transition model. In *ICASSP*, pages 221–224, 2005.

[16] J. Malcolm, Y. Rathi, and A. Tannenbaum. A graph cut approach to image segmentation in tensor space. In *Workshop on Component Analysis Methods (CVPR)*, pages 18–25, 2007.

[17] J. Malcolm, Y. Rathi, and A. Tannenbaum. Graph cut segmentation with nonlinear shape priors. In *ICIP*, 2007. (To appear).

[18] J. Malcolm, Y. Rathi, and A. Tannenbaum. Tracking through clutter using graph cuts. In *BMVC*, 2007. (To appear).

[19] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Particle filtering for geometric active contours with application to tracking moving and deforming objects. In *CVPR*, pages 2–9, 1997.

[20] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. In *ACM Trans. on Graphics (SIGGRAPH)*, 2004.

[21] J. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proc. Nat. Acad. Sci.*, volume 93, pages 1591–1595, 1996.

[22] G. Slabaugh and G. Unal. Graph cuts segmentation using an elliptical shape prior. In *ICIP*, pages 1222–5, 2005.

[23] L. Vese and T. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *IJCV*, 50:271–293, 2002.

[24] N. Xu, R. Bansal, and N. Ahuja. Object segmentation using graph cuts based active contours. In *CVPR*, pages 46–53, 2003.

[25] L. Yatziv, A. Bartesaghi, and G. Sapiro. O(N) implementation of the fast marching algorithm. *J. of Computational Physics*, 212:393–399, 2006.