Tracking with Graph Cuts: Treating Clutter with Adaptive Penalties

James Malcolm and Yogesh Rathi and Allen Tannenbaum

***Abstract*—Many techniques for tracking based on gradient descent cannot follow objects as they undergo large movements or deformation. On the other hand, multi-hypothesis trackers capable of handling such behavior are computationally expensive. The standard graph cut technique offers a middle ground, quickly capturing objects anywhere in the image; however, because of its global nature, it is prone to capturing outlying areas similar to the object of interest. This paper proposes a novel method to constrain the standard graph cut technique to regions of interest for tracking multiple interacting objects in near real-time. For each object, we introduce a penalty based upon distance from a region of interest. This results in a segmentation biased to this area. Also, we demonstrate the use of a track point filter for predicting the location of the object in each frame. The distance penalty is then centered at this location and adaptively scaled based on prediction confidence. We demonstrate tracking in gray-scale and color videos.**

***Index Terms*—Image segmentation, tracking, graph-theoretic methods**

I. 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 segmentations to leak out into the surrounding area, the object may be near other objects of similar intensity causing the tracking of unintended objects, or the object may suddenly move outside the algorithm's region of detection. Multi-object tracking raises additional concerns involving 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 [1]–[3]. When adjacent regions are similar to the object of interest, multiple hypothesis trackers can keep track of each region while for each frame determining the most likely region based on some criteria [4]–[6]. To simultaneously segment multiple objects, techniques have been developed to take into account the interaction among objects [7].

Methods based on gradient descent allow tracking highly deformable objects, but cannot track large movements since they search within a small region around the object [8], [9]. Such spatial motion can be reliably tracked using a finite dimensional state space, but the reduced state space representation then restricts the possible shape deformations. Recently, particle filters, typically used with finite dimensional state spaces, have been demonstrated in the space of infinite dimensional curves; however, this method is computationally complex and time consuming [10].

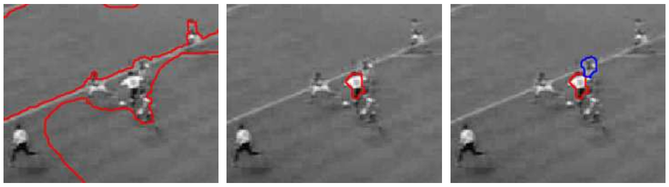

Fig. 1. Tracking two interacting soccer players among others of similar intensity: without distance penalty and applying distance penalty to track one or two players *(left to right)*. Without the distance penalty, multiple non-intended regions were captured.

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 finds regions with high likelihood given intensity priors [11]. Figure 1 shows an example where there are multiple regions of similar intensity. The standard graph cut algorithm captures all of these regions, and so post-processing is required 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 necessary.

Several techniques have used graph cuts for segmentation in visual tracking applications. In [12] 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 requires repeated cuts on this reduced domain. Furthermore, no motion model is assumed thus making the tracker highly dependent on the previous segmentation. In [13] the authors use one graph cut for each frame to estimate both the optical flow and object position despite changes in illumination. However, since

James Malcolm and Yogesh Rathi are with Brigham and Women's Hospital, Harvard Medical School. Allen Tannenbaum is with the School of Electrical and Computer Engineering, Georgia Institute of Technology.

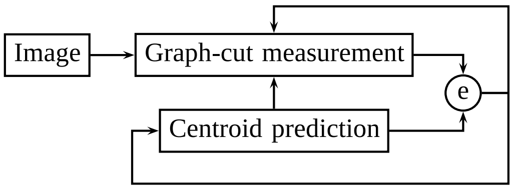

Fig. 2. The proposed closed-loop system. For each new image, we predict each object centroid and use graph-cut segmentation to measure each object centroid. The error between the two is then fed back both to the track-point filter and to the segmentation for adaptively scaling the basins of attraction.

optical flow requires the multi-label graph cut technique [14] and the graph proposed has such dense neighborhoods, the current approach requires about one minute per frame. Also, due to the local nature of optical flow, the technique cannot handle large movements.

In addition to tracking, work has been done to constrain segmentation based on a user selected region. The work of [15] begins with a rectangle bounding the object, while the work of [16] 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 the "best" segmentation. In these works, hard constraints confine the segmentation within a user-selected region and multiple graph cuts are performed. Additionally, none of these methods has been generalized to simultaneously segment multiple unique objects. In our work, the object may be found a given distance from the predicted centroid depending on the scale of the distance penalty, segmentation is performed only once per frame, and the formulation handles multiple interacting objects.

The method presented here makes several contributions to the field of visual tracking. First, we incorporate an adaptive distance penalty into the graph cut algorithm biasing segmentations to a region likely to contain the given object. Second, we adaptively scale the surface of this basin of attraction based upon performance error. Third, we demonstrate how the multi-label graph cut algorithm naturally handles multiple interacting objects. Further, we show how to naturally integrate a prediction filter in the proposed framework for robust tracking.

The basic algorithm is as follows. First, for each object, we incorporate a distance penalty into the graph cut algorithm to bias segmentations to a region likely to contain the object. Second, we use a 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 system is visualized in Figure 2.

The rest of the paper is organized as follows. Section II outlines the standard graph cut segmentation framework. Section III describes the distance penalty constraining segmentation. Section IV defines the filter used to predict the object centroid, and Section V integrates the filter prediction error with the distance penalty. Next, in Sections VI and VII, we present our algorithm and experiments. Finally, in Section VIII we summarize our work and outline some possible future research directions.

II. Graph cuts

In this section, we briefly outline the standard multi-label graph cut technique; for more details see [11], [14]–[17] and the references therein.

Taking advantage of efficient algorithms for global min-cut solutions, we cast the energy-based image segmentation formulation 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 region label $r \in \mathcal{R}$, such energies are typically constructed as the sum of a data dependent term and a term for smoothness. 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:

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

where p and q are pixels in the image domain Ω, and $\mathcal{N}$ is the set of 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 artifacts [18].

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 $r \in \mathcal{R}$, *e.g.* background, first object, second object. Figure 3 illustrates such a construction for a 3x3 image with two region labels. The data dependent term is implemented by connecting each pixel to these extra nodes with non-negative edge weights $R_p(r)$ representing the penalty for assigning pixel p to the region r. 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 this weighted graph represents the segmentation that best separates the regions. See [11], [14] for more details.

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 [11] use the negative log-likelihood of a pixel's fit into an intensity prior to compute the regional weights, while intensity contrast is used in the boundary term:

$$R_p(fg) = -\ln P(\mathcal{I}_p|fg), \qquad R_p(bg) = -\ln P(\mathcal{I}_p|bg),$$
$$B_{(p,q)} = \exp\left(\tfrac{-\|\mathcal{I}_p-\mathcal{I}_q\|^2}{2\sigma^2}\right)\tfrac{1}{\|p-q\|} \qquad (2)$$

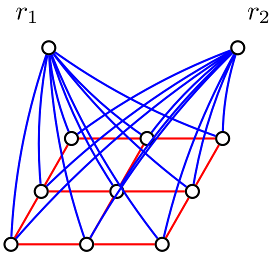

Fig. 3. Underlying graph construction of a 3x3 image for the case of two region labels, r_1 and r_b. Each grid node represents an image pixel and each label has its own additional node. Each pixel's node is connected to those of its neighbors with weight $B_{(p,q)}$ *(red)* and connected to the additional label nodes with weight R_p *(blue)*.

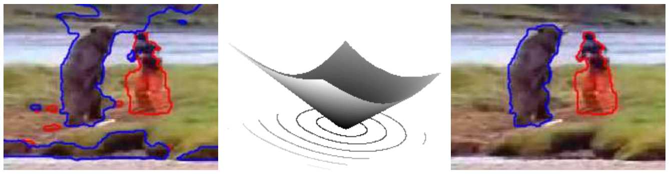

Fig. 4. Tracking a bear near other regions similar to its fur: no distance penalty, distance penalty ϕ with isocontours, and applying distance penalty *(left to right)*. Without the distance penalty, multiple non-intended regions were captured.

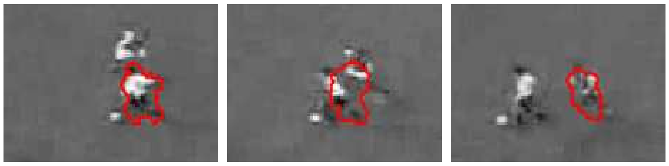

Fig. 5. Without location prediction, tracking can fail when the target makes sudden movements. Here the tracker catches a defender as the target passes *(left to right)*.

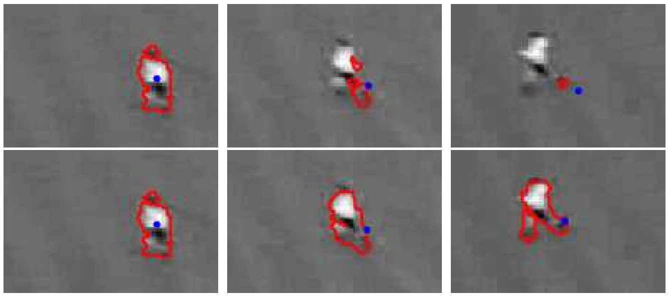

Fig. 6. Effect of adaptive α on tracking: non-adaptive α *(top, left to right)* and adaptive α using prediction error *(bottom, left to right)*. Tracking fails without using error feedback to adaptively scale distance penalty. Predicted centroid is shown as a blue dot.

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 [11] where the user marks regions of foreground and background to generate the intensity histograms for each region.

III. Distance penalty

The standard graph cut technique is capable of finding regions matching the object 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 4 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 which predicts the object shape. Centering that mask M at the predicted object location and assigning pixels within the mask zero penalty, each pixel x outside the mask is assigned its distance from the nearest masked pixel $m_x \in M$, *i.e.* $\phi(x) = \|x - m_x\|$. Such a construction can be quickly computed with the Fast Marching algorithm [19], [20].

A simple choice for the base mask M is the initial user segmentation from the first frame. For objects that quickly change shape from the initial segmentation, a moving average of the past few segmentations may be more suitable for the base mask (see Section VII and Figure 11). Several other methods can be used for representing deformable shape priors in graph cut segmentations [13], [21], [22].

IV. Location prediction

It is often the case that the object makes a large movement, at times large enough 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. This section describes integrating a general prediction filter into the proposed framework while the next section, Section V, details incorporating error feedback into the measurement.

To reveal 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 5 shows the failure to track after the object has made a sudden move. The movement placed the object too far outside of the basin of attraction.

Demonstrating the flexibility of filter choice, we employ two different filters, each motivated by the apparent motion models exhibited in our imagery. The first video we examined involved the camera following a soccer player across the field (see Figure 7). The camera alternates between a fixed view with the player moving across the screen and a relative view where the camera pans to catch up. Due to the lack of fixed image features, attempts at global motion estimation and camera stabilization performed poorly. Deciding not to model such transient camera motion, we used a simple linear filter where the predicted centroid $\tilde{c}_{t+1}$ is last centroid c_t projected forward by the average displacement in the past few frames:

$$\tilde{c}_{t+1} = c_t + \frac{1}{N}\sum_{j=0}^{N}(c_{t-j} - c_{t-j-1}). \tag{3}$$

The remaining videos feature relatively stationary targets, and so we employed a Kalman filter with an identity prediction model to compensate for slight camera motion.

V. 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 6 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

$$\alpha(x) = \exp\left(-x^2/\rho^2\right) \tag{4}$$

where ρ is a user defined model parameter that determines the amount of expected object motion between frames. The effect is that when the filter is off in its predictions of the object centroid, the distance penalty is lowered to still capture the object. After locking back onto the object, the prediction error decreases, and the α automatically raises the distance penalty back up to tighten around the object. Figure 6 shows how, despite incorrectly predicted centroids, the system is able to recover by adaptively widening the distance penalty.

For the case of the linear filter (3), such a linear filter is unstable under large displacements causing the distance penalty to be driven to zero. To overcome this limitation, we decided to incorporate prior knowledge of typical object movement to limit the instability. Assuming objects to typically not move more than γ pixels, we saturate the error norm in (4) at γ:

$$\alpha(x) = \exp\left(\frac{-\min(x,\gamma)^2}{\rho^2}\right) \tag{5}$$

where ρ is empirically determined.

VI. 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 roughly marks the object and background in the first frame. This initialization defines both the intensity priors used in the regional edge weights (II) 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 $\|\tilde{c} - c\|$ using the form in (5).

We propose a new regional edge weight to augment the standard weight in (II). Given the image, our goal is to determine the best region assignment r for each pixel; in other words, our goal is to maximize $P(r|\mathcal{I})$. Now, Bayes rule tells us that $P(r|\mathcal{I}) \propto P(\mathcal{I}|r)P(r)$. If we assume $P(r)$ is uniform, then its negative log-likelihood is zero, and so it falls out of the expression and we have the standard regional term (II). Instead, we assume a non-uniform object prior with probability $P(r) \propto (\alpha(\|\tilde{c} - c\|)\phi)^\beta$ and hence: $-\ln P(r) \propto \alpha(\|\tilde{c}-c\|)\phi$. We assume the background to still be uniformly distributed and so its distance penalty prior disappears. The weight $\beta > 0$ defines relative distance penalty influence as compared to the intensity prior. We now have the new regional terms for object and background:

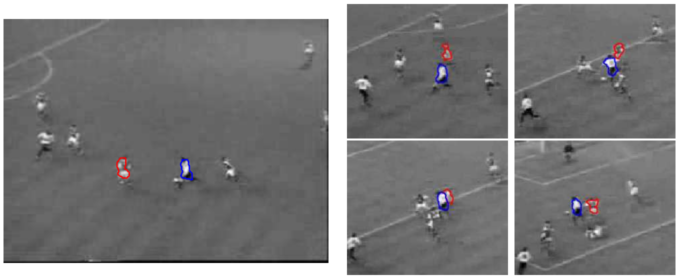

Fig. 7. Tracking two opposing players from the soccer sequence. Despite prolonged contact and occlusion, the technique is able to uniquely track the two targets. Full image *(left)* and selected cropped frames *(right)*.

$$\begin{aligned} R_p(r) &= -\ln P(\mathcal{I}_p|r) - \beta \, \ln P_p(r) \\ &= -\ln P(\mathcal{I}_p|r) + \beta \, \alpha(\|\tilde{c} - c\|)\phi(p) \end{aligned} \tag{6}$$

$$\begin{aligned} R_p(\text{bg}) &= -\ln P(\mathcal{I}_p|\text{bg}) - \beta \, \ln P_p(\text{bg}) \\ &= -\ln P(\mathcal{I}_p|\text{bg}) \end{aligned} \tag{7}$$

Finally, we take the min-cut of this graph to yield a multi-region segmentation.

VII. Experiments

Tracking was performed on gray-scale and color videos and representative frames were chosen to exhibit clutter with objects of similar intensity undergoing large movements. Full videos are included in the supplementary material.

The parameters were defined as follows. For all experiments, we set $\lambda = 10$ in (1). Also experiments involving the linear filter (3), we found $\gamma = 5$ and $\rho = \gamma/2$ to be quite robust. In (6), we set $\beta = 10$ for gray-scale imagery and $\beta = 2$ for color. Unless otherwise noted, the base mask M is taken to be the initial user segmentation.

On a standard workstation[1], depending on the image size, the current system tracks one object at roughly five frames per second and two objects at roughly two frames per second fluctuating slightly based on the chosen neighborhood. The choice of neighborhood also affects the smoothness of the segmentation with smaller neighborhoods tending to introduce irregular segmentations [18]. 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 first low-resolution, 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 7 shows tracking of a player from each team amidst occlusion and contact with several other players of similar intensity.

[1] Pentium IV 3 GHz, 2 GB RAM

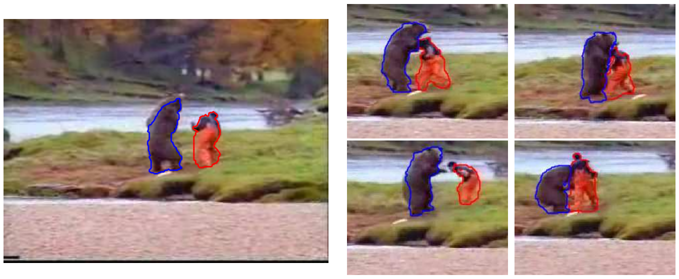

Fig. 8. Tracking the bear and man in color. Due to large movements and changes in shape, at several points the tracker is partly thrown off, yet it recovers fully. Full image *(left)* and selected cropped frames *(right)*.

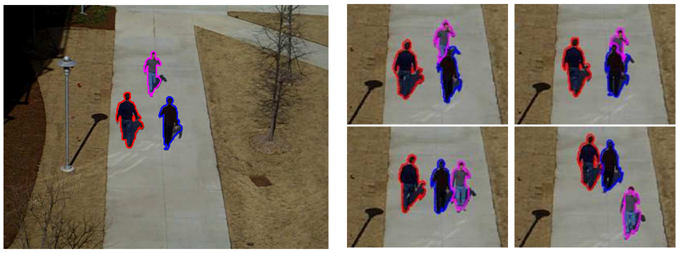

Fig. 9. Tracking three people in color despite severe occlusion and the similar intensity profiles of the dark figures. Full image *(left)* and selected cropped frames *(right)*.

The second video sequence is a color television commercial staging a fight between a bear and a man. Figure 8 shows tracking of the bear and man as they make sudden movements or change shape. These sudden changes throw the tracker slightly off but in all cases the tracker recovers fully in a few frames.

The third sequence is a color video demonstrating occlusion as two people walk past a third. Figure 9 shows that the tracker is able to maintain track of all three figures. Notice also that despite the similar intensity profiles of the two dark figures, the tracker keeps them separated throughout the video.

The fourth sequence is a color video of a stuffed animal as the camera rotates with egomotion. As the stuffed animal rotates in the picture, its shape changes significantly and so, as Figure 10 shows, the initial segmentation is a poor choice for forming the base mask M; the basin is too narrow and so clips the head and tail as the object rotates. Figure 11 demonstrates that simply forming the base mask from the average of the past few segmentations captures this change and provides an appropriately shaped basin to capture the full object. Here we used the average of the last five segmentations to form the base mask. Figure 12 shows the underlying averaged mask for these frames.

The last and perhaps most challenging video sequence involves a helicopter circling overhead a van. This low-resolution video combines highly unstable camera motion with a low frame rate making for large movements as the helicopter struggles to keep the camera centered on the van. Figure 13 shows selected frames. Notice that the segmentation captures the bright white van top, the range of intensities most distinguishing it from the background. To highlight the camera instability, Figure 14 shows several frames as the helicopter struggles to maintain a stable view. For several frames the tracker fails to find the van at all. Notice that the camera even loses sight of the van all together in one frame. Although several regions similar to the van are captured, the tracker is ultimately able to resume tracking the van as the distance penalty adaptively scales.

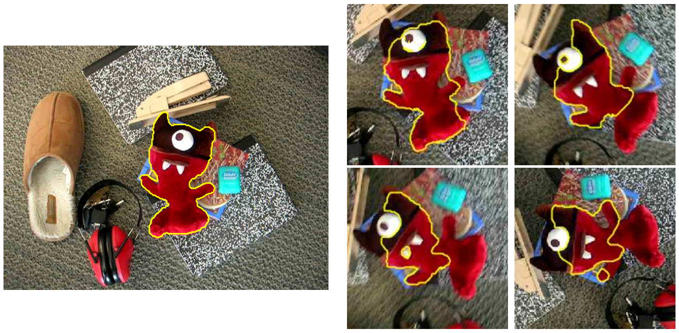

Fig. 10. Unsuccessful tracking using a static base mask. If the base mask is formed from initial upright segmentation *(left)*, the basin of attraction is too narrow and so tracking fails to fully capture the object as it rotates. Full image *(left)* and selected cropped frames showing failure to capture the full object *(right)*.

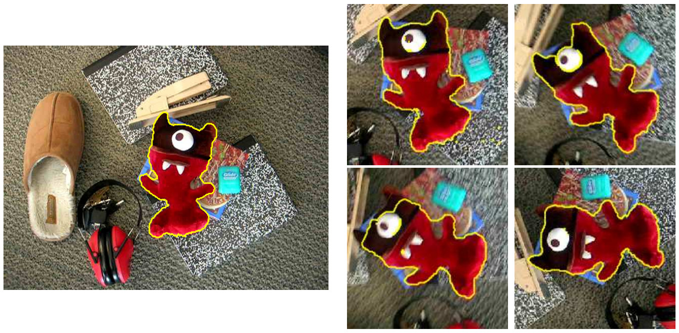

Fig. 11. Forming the base mask from an average of the past few segmentations provides a basin of attraction suitable to capture the full object. Full image *(left)* and selected cropped frames *(right)*.

To highlight the advantages of the global graph cut segmentation, we compared against the more local level set method. Figure 15 shows the standard level set method failing to track the van. The level set technique performs local gradient descent, and so it only looks for the van in the small region immediately surrounding the curve. Thus, due to a large displacement, it makes an incorrect segmentation from which the system is unable to recover. Particle filters have been used in conjunction with the level set method [23]. However, the computational burden of computing the segmentation separately for each particle makes this approach impractical in most applications where speed is critical.

VIII. 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.

From here, there are several future directions. In the experiments presented, we used intensity alone as a discriminating feature, yet more robust feature spaces can be exploited [24].

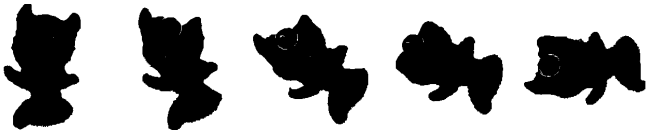

Fig. 12. The base masks corresponding to the frames in Figure 11 formed from an average of the past few segmentations.

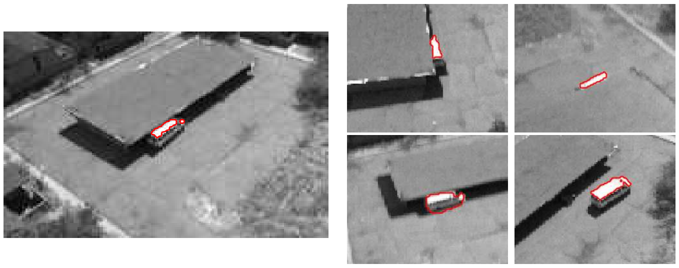

Fig. 13. Tracking a van from a helicopter-mounted camera circling overhead. Full image *(left)* and selected cropped frames *(right)*.

Also, at present the system is not at full real-time rates for multiple objects, and so faster graph solution methods should be examined and extended for multi-label energies [25]. In many cases, the distance penalty inhibits capturing the entire object, something that more principled shape priors may overcome [21].

References

[1] D. Cremers, T. Kohlberger, and C. Schnorr, "Shape statistics in kernel space for variational image segmentation," *Pattern Recognition*, vol. 36, 2003.

[2] S. Dambreville, Y. Rathi, and A. Tannenbaum, "A shape-based approach to robust image segmentation," in *Int. Conf. on Image Analysis and Recognition*, 2006.

[3] M. Leventon, E. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Computer Vision and Pattern Recognition (CVPR)*, 2000.

[4] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear non-Gaussian Bayesian tracking," *Trans. on Signal Processing*, vol. 55, no. 2, 2002.

[5] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *Int. J. of Computer Vision*, vol. 29, no. 1, 1998.

[6] E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Int. Conf. on Acoustics, Speech, and Signal Processing*, 2005.

[7] L. Vese and T. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *Int. J. of Computer Vision*, vol. 50, 2002.

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, 2003.

[9] N. Peterfreund, "Robust tracking of position and velocity with kalman snakes," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, 1999.

[10] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, 2007.

[11] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," in *Int. Conf. on Computer Vision*, 2001.

[12] N. Xu, R. Bansal, and N. Ahuja, "Object segmentation using graph cuts based active contours," in *Computer Vision and Pattern Recognition (CVPR)*, 2003.

[13] D. Freedman and T. Zhang, "Interactive graph cut based segmentation with shape priors," in *Computer Vision and Pattern Recognition (CVPR)*, 2005.

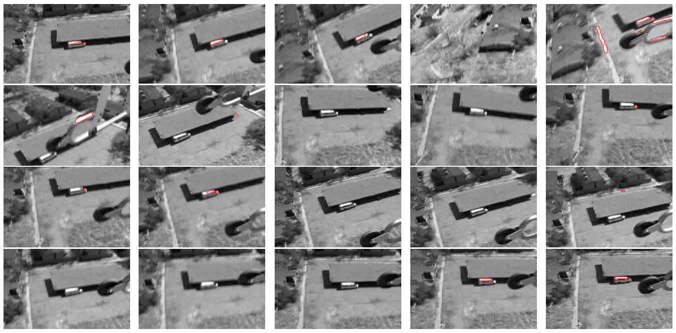

Fig. 14. Twenty consecutive frames from the aerial van sequence demonstrating the significant camera motion. Over this span, this unpredictable camera motion leads to increased prediction error and wider search region as the distance penalty α is lowered. The system shortly regains track *(left to right, then top to bottom)*.

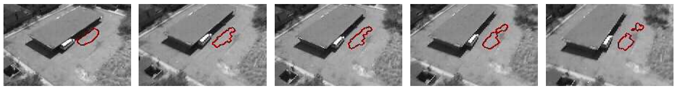

Fig. 15. Selected consecutive frames from the aerial van sequence demonstrating the failure of the level set method to track under a large displacement. Since this technique is more local, once one frame is significantly off, it is unable to search far enough to recover the van and so remains off.

[14] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, 2001.

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

[16] A. Blake, C. Rother, M. Brown, and P. Torr, "Interactive image segmentation using an adaptive GMMRF model," in *European Conf. on Computer Vision*, vol. 3021, 2004.

[17] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *Int. J. of Computer Vision*, vol. 70, 2006.

[18] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *Int. Conf. on Computer Vision*, 2003.

[19] J. Sethian, "A fast marching level set method for monotonically advancing fronts," in *Proc. Nat. Acad. Sci.*, vol. 93, 1996.

[20] L. Yatziv, A. Bartesaghi, and G. Sapiro, "O(N) implementation of the fast marching algorithm," *J. of Computational Physics*, vol. 212, 2006.

[21] J. Malcolm, Y. Rathi, and A. Tannenbaum, "Graph cut segmentation with nonlinear shape priors," in *Int. Conf. on Image Processing (ICIP)*, 2007.

[22] G. Slabaugh and G. Unal, "Graph cuts segmentation using an elliptical shape prior," in *Int. Conf. on Image Processing (ICIP)*, 2005.

[23] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Particle filtering for geometric active contours with application to tracking moving and deforming objects," in *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[24] J. Malcolm, Y. Rathi, and A. Tannenbaum, "A graph cut approach to image segmentation in tensor space," in *Component Analysis Methods (in CVPR)*, 2007.

[25] P. Kohli and P. Torr, "Efficiently solving dynamic markov random fields using graph cuts," in *Int. Conf. on Computer Vision*, 2005.