Integrating Region and Boundary Information for Improved Spatial Coherence in Object Tracking

Desmond Chung†, W. James MacLean† & Sven Dickinson‡
(†) Department of Electrical & Computer Engineering
(‡) Department of Computer Science
10 King's College Road
University of Toronto
Toronto, Canada M5S 3G4
{dchung,maclean}@eecg.utoronto.ca, sven@cs.toronto.edu

Abstract

This paper describes a novel method for performing spatially coherent motion estimation by integrating region and boundary information. The method begins with a layered, parametric flow model. Since the resulting flow estimates are typically sparse, we use the computed motion in a novel way to compare intensity values between images, thereby providing improved spatial coherence of a moving region. This dense set of intensity constraints is then used to initialize an active contour, which is influenced by both motion and intensity data to track the object's boundary.

The active contour, in turn, provides additional spatial coherence by identifying motion constraints within the object boundary and using them exclusively in subsequent motion estimation for that object. The active contour is therefore automatically initialized once and, in subsequent frames, is warped forward based on the motion model. The spatial coherence constraints provided by both the motion and the boundary information act together to overcome their individual limitations. Furthermore, the approach is general, and makes no assumptions about a static background and/or a static camera. We apply the method to image sequences in which both the object and the background are moving.

1. Introduction

The motion estimation community has typically focused on the problems of optical flow computation and motion layer segmentation, yet has paid relatively little attention to recovering accurate boundaries of moving objects. Layered motion approaches generally code no information about the motion of neighboring pixels and, as such, often yield support maps that are highly sparse. On the other hand, the object tracking community has typically focused on tracking the shape of a moving object, often assuming manual initialization of the tracking region, active contour, or model pose (in the case of model-based tracking). Trackers that do not assume an *a priori* object motion model typically focus on object boundaries while ignoring the rich motion information encoded in the region.

Each of these paradigms assumes a model of spatial coherence. The motion community seeks to label the pixels defining the *region* of the moving object, while the boundary-based tracking community seeks to label the pixels defining the *boundary* of the moving object. Each approach is not without its limitations. Motion constraints can be weak in areas of limited texture, while boundary constraints can be weak in areas of limited contrast. We attempt to bring together these two components in a novel manner to detect, track, and recover the shape of a moving object, effectively drawing on the strength of each component to overcome the weakness of the other.

In the following sections, we review related work, provide an overview of the approach, describe the components in detail, and demonstrate the approach on image sequences in which both the object and the background/camera are moving. We conclude with a discussion of the limitations of the approach, along with our directions for future research.

1.1. Previous Work

Previous work can be divided into region-based approaches [22, 21, 5, 20, 13, 9, 18, 10] and boundary-based approaches [11, 2, 17]. Among the region-based approaches, some [5, 20, 13, 18, 10] can be classified as layered approaches, with the latter two using models to describe image regions. Our own layered motion technique is very similar to that proposed in [18], where robust dominant motion estimation is applied to recursively recover motion estimates for independently moving layers in a sequence. Notably, this approach does not attempt to introduce any kind of spatial coherence constraints, and is thus not able to distinguish among objects in a scene that move with identical 2-D image motion. Moreover, general motion layer techniques of this type do not aim to recover accurate object boundaries or accommodate non-rigid object motion.

In [13], a method is described that adds a type of spatial coherence to a layered flow method by using octagonal-shaped regions to limit the region of support for motion constraints for a particular motion. In addition to the parameters for each motion, parameters for the size, shape and pose of each region are also computed, as well as a visibility ordering. While this method seeks to localize the effect of motion constraints, it does not attempt to fit an accurate boundary to any region. A model-less layered approach is taken in [9], where the authors perform spatial segmentation independently of motion, and derive a motion model for each region. Regions with like motions are labeled the same, and the union of all regions with a particular label gives the motion-based segmentation. Here, the challenge is updating the initial spatial segmentation between frames, and it is treated as a problem in graph manipulation. In this case, motion assumes a subservient role as it is used only to label regions. Another model-less approach is found in [22, 21], where a method for non-parametric flow estimation is given. A Markov random field model is used to provide a prior encoding of the notion that neighboring image points are likely to be related. A mean-field approximation can be used to make the method computationally feasible, but the method gives no explicit estimate of the region boundaries.

The use of normalized cuts for motion segmentation is introduced in [19], in which graph cutting techniques are used to recover a motion-related grouping of patches in the image sequence. The relationship between patches is defined on the basis of their motion similarity, as well as their spatial and temporal proximity in the image sequence. The method imposes a high computational overhead, and is thus restricted to very small image sizes in order to minimize the graph cutting complexity. As a result, it does not attempt to provide accurate shape recovery.

Boundary-based approaches fall into two categories: probabilistic contour tracking [11, 2] and active contour approaches [1, 17]. A probabilistic formulation of curve tracking known as *Condensation* is presented in [11] that propagates a set of sampled states to approximate a posterior distribution on the possible states given the observed data. The method requires a learned curve template and an additional learning phase to acquire a motion model, after which it can operate uninitialized on new image sequences. Our system can optionally incorporate elements of the *Condensation* tracker for boundary estimation while employing active contours and motion estimation to provide an on-line shape template and motion model to avoid off-line learning requirements.

Early attempts to integrate region information with an active contour boundary model include [1]. Here, the authors use a constant, affine or homomorphic warp computed through a correlation approach to compute the displacement of the entire region and then update the active contour between frames. Motion information is not used between frames to update the contour after the initial warp, and as such is not expected to discover evolving object structure except through image edges. A more holistic approach is proposed in [4], which uses level set techniques to minimize an objective function defined over the motion surfaces in a sequence, attempting to recover spatially coherent motion surfaces with the minimum area in the image. This approach does not propose any techniques to estimate the number of independently moving objects in the scene, and does not attempt to provide automatic initialization.

Perhaps the closest work to that described in this paper is the geodesic active contour formulation, proposed in [17], which assumes a static background, so that an image differencing approach can be used to detect motion. The geodesic active contour has elegant properties for splitting and merging different contours based on a single energy function. Difference images, local intensity statistics and intensity warping within the active contour region are all used to control the active contour. Whereas the geodesic active contour framework focuses on a more elegant active contour formulation while assuming a simpler motion model, we opt for a more elegant motion formulation while assuming a standard active contour model. As a result, while our active contour implementation is not currently topologically adaptive (although, in principal, we could also employ geodesic active contours), our approach does not assume a static background or a static camera.

2. Overview of the Approach

Our appoach to combining region (layered motion estimation) and boundary (active contour estimation) information is summarized in Figure 1. We begin by estimating the motion in successive frames using a parametric motion model, as described in Section 2.1. This process generates motion constraints by assuming brightness constancy, and classifies them according to a parametric motion model using the EM (Expectation Maximization) algorithm. Due to the aperture problem, these constraints are relatively sparse, and do not provide a good basis for object boundary recovery. We improve the density of segmentation constraints by

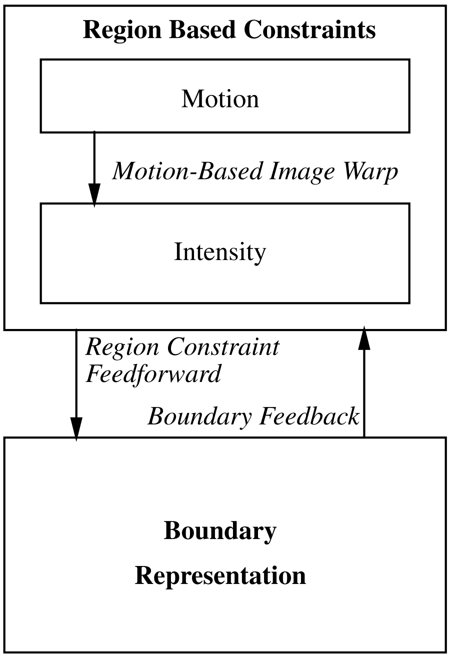


Figure 1. This figure illustrates the basic structure of our approach. Region-based information is used to derive flow constraints, which are typically sparse. This information can be made more dense by warping image pixels to find pixels with matching intensities in the two images. These constraints provide data for the active contour, and the contour is warped between frames according to the motion parameters. The contour reinforces spatial coherence by allowing us to consider only motion constraints within the contour.

warping the image according to the recovered motion, and comparing the intensities of corresponding pixels, as described in Section 2.2. Matching intensity values provides a sufficiently dense set of constraints to initialize an active contour whose shape is then governed by both motion and image gradient information, as described in Section 2.3. In subsequent frames, motion information is used to warp the recovered boundary (active contour) to the next frame in a *motion feedforward* step, described in Section 2.4.

Finally, the spatial coherence of the object boundary influences the next round of motion estimation by excluding motion constraints that fall outside the boundary. This *boundary feedback* step is described in Section 2.5.

2.1. Motion Estimation

Our approach begins with a multiscale layered motion estimation technique that is similar to the methods described in [10, 18]. In particular, we adopt a robust mixture model approach related to that described in [12], but in principle, any other layered motion approach that generates either a parametric motion or a dense, non-parametric motion can be used instead. The process is described here for application at a single scale to simplify its presentation. The *brightness constancy constraint* (BCC), $\nabla_{\vec{x}}^T I \vec{u} + I_t = 0$, is well-known, and is the starting point for the estimation of 2-D image velocity $\vec{u} = [u_x \ u_y]^T$. Each image location provides a constraint vector $\vec{c}(\vec{x}) = [I_x \ I_y \ I_t]^T$ that satisfies (in the absence of noise) $\vec{c}(\vec{x})^T \vec{u}_h = 0$, where $\vec{u}_h = [\vec{u}^T 1]^T$ is a homogeneous representation of $\vec{u}$, and $\vec{u}$ is assumed to be the motion experienced by the pixel at $\vec{x} = [x \ y]^T$. The inner product of a constraint with a motion vector from another layer will be expected to yield a non-zero value, in general.

Each motion layer has an associated parametric model that is either constant or affine, although any parametric model can be used. Associated with each parametric model and its associated parameters $\vec{\theta}$ is a likelihood function $p(\vec{c}(\vec{x})|\vec{\theta}, \vec{x})$ that indicates how well a constraint $\vec{c}$ matches the motion. For example, for the constant motion model, we have $p_{constant}(\vec{c}(\vec{x})|\vec{u}, \sigma) = G(\vec{c}(\vec{x})^T \vec{u}_h; 0, \sigma)$. Here, the notation $G(x; \mu, \sigma)$ represents a Gaussian density over x with mean μ and standard deviation σ. The likelihood of a particular constraint $\vec{c}(\vec{x})$ with respect to all motion layers is $p(\vec{c}(\vec{x})) = \sum_{j=1}^{n} \pi_j p_j(\vec{c}(\vec{x})|\vec{\theta}_j, \vec{x})$, where the π_j are called *mixing parameters* and satisfy $0 \le \pi_j \le 1$ for all j and $\sum_j \pi_j = 1$. An outlier layer is used to model constraints not accounted for by other motion layers.

The probability that a constraint comes from any particular layer j can be computed as

$$O(\vec{c}(\vec{x})|j) = \frac{\pi_j p_j(\vec{c}(\vec{x})|\vec{\theta}_j), \vec{x})}{\sum_{k=1}^{n} \pi_k p_k(\vec{c}(\vec{x})|\vec{\theta}_k), \vec{x}))} \quad (1)$$

and is called its *ownership* by that layer. Finally, the likelihood of the entire model with respect to a set of measured constraints $\{\vec{c}(\vec{x}_q)\}_{q=1}^{m}$ is given by

$$L(\vec{\Theta}) = \Pi_{q=1}^{m} p(\vec{c}(\vec{x}_q)) \ , \quad (2)$$

where $\vec{\Theta} = [\vec{\theta}_1^T \ldots \vec{\theta}_n^T \ \pi_1 \ldots \pi_n]^T$ is the collection of all model parameters. The EM algorithm [6] is an iterative technique for maximizing a model's likelihood with respect to the observed data. Since, in general, the likelihood is a non-linear function, the method may find a local minimum as opposed to the desired global minimum.

Each iteration of the EM algorithm requires that the number of models n be known, so it is necessary to determine this from the input sequence. Following [16], we start by computing a model for a single motion plus an outlier process designed to catch constraints not well modeled by the single motion. This can be considered a robust procedure for estimating the dominant image motion. By examining constraints owned by the outlier process, we can decide whether or not to add another motion to the model [18]. This continues until no further processes are added, or newly added processes become identical to existing ones.

2.2. Intensity Constraints

Motion constraints recovered and used in Section 2.1 are usually sparse, since areas with little or no texture will not produce motion constraints. While we can expect good constraints along the leading edge of the object, edges parallel to the motion may produce few or no constraints due to the BCC only detecting motion in the direction normal to image edges. It should be noted that this is not a problem specific to the BCC; any gradient-based motion constraint will suffer this fate. As such, motion constraints alone are insufficient for generating boundary estimates.

We have developed a new technique to improve the density of constraints for boundary estimation that directly embodies the notion of spatial coherence. Based on knowledge of the object's motion, recovered from a sparse set of motion constraints, we warp pixel intensities from one frame to the next and compare them. This makes intuitive sense, as we do not expect object appearance to change radically between successive frames. For each motion, we warp each pixel in the first image, $I_1(x, y)$, and compare it to its associated pixel in the next image, $I_2(x', y')$, where $[x' \; y']^T = [x \; y]^T + \vec{u}_q(\vec{x})$. The likelihood of the two pixels having the same intensity is computed as $L_q(x, y) = G(I(x', y', t_n) - I(x, y, t_{n-1}); 0, \sigma_{ccd})$, where G is a normal probability density, with σ_{ccd} related to expected noise levels in the CCD sensor. The subscript q indicates this is the likelihood for motion layer q. A high value indicates a high likelihood that the pixel intensity has been matched, whereas a low value suggests the pixel under consideration is not moving with the assumed motion.

Of course, it is possible to get a good match even for pixels that are not moving with the assumed motion. For example, pixels in a region with uniform intensity values will show a good match with any assumed motion. Because of this, we take the further step of combining these likelihood values for each pixel across all layers to get the probability that the pixel belongs to each layer. If $L_q(\vec{x})$ is the likelihood that the pixel at $\vec{x}$ matches the motion specified by motion layer q, then we compute this probability as

$$P_q(\vec{x}) = \frac{L_q(\vec{x})}{\sum_{i=1}^{n} L_i(\vec{x})} , \qquad (3)$$

where n is the number of motion layers. This equation shows a strong similarity to the computation of ownership values in the EM algorithm, and as such, we refer to the resulting probabilities as *intensity ownerships*.

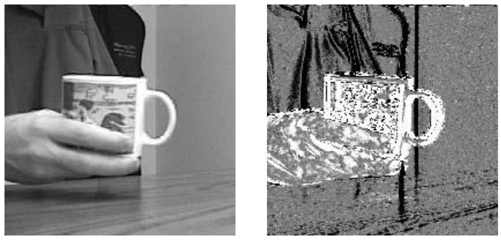


Figure 2. The right image shows the intensity ownerships for the horizontal motion of a cup and hand from the frame shown at the left. The gray levels in the right image represent intensity ownership estimates, black is 0.0, white is 1.0, and values in between are given by levels of gray. Values around 0.5 are considered "neutral".

This computation combines information across multiple motion layers, a fact which may facilitate extensions to our method to deal with objects with holes. It has the effect of assigning equal layer ownership for regions where little texture exists, while forcing regions with higher texture to show strong ownership by one of the motion layers. For example, in a sequence with two recovered motions, we expect the ownership of pixels in constant intensity regions to be 0.5. Regions with constant intensity values will show strong ownership if they are in close proximity to leading or trailing edges of an object in motion. In practice, regions with too-little texture to generate good motion constraints still have enough intensity variation to give denser clusters of high ownership values. An example of the recovered ownership maps for a two-layer sequence is shown in Figure 2. Finally, while we are applying this technique to intensity values, it is equally applicable to other features, such as pixel color, or phase values returned from a complex filtering of the image.

While both Bascle & Deriche [1] and Paragios *et al.* [17] warp image intensities, they only apply the warp to pixels already within the object boundaries and do not combine the information across motion layers to determine intensity ownership probabilities. By combining information across motion layers we are able to use information from other layers, including a possibly moving background, to increase our certainty about pixels in the object layer. This further assists in growing the object region if it deforms, and it allows us to deal with holes in objects. Jepson *et al.* [13] also warp intensity values to choose among possible motion models, but not for the purpose of determining intensity ownerships or accurate region boundaries.

2.3. Active Contour Implementation

Once motion and intensity constraints have been computed for a pair of image frames, we can initialize the active contour to track the object's boundary. Active contours are introduced in [15], and have been widely used in vision applications. For our experiments, we have adopted the distance transform-based (DT) active contour [3]. This approach was chosen to improve convergence when the active contour is not initialized close to the boundary, and to allow the active contour to quickly descend into deep concavities in the boundary. In those respects, the performance of the distance transform-based active contour closely matches that of the *Gradient Vector Flow* [23] active contour [7]. Since the active contour is initialized in the first frame using a convex hull of recovered intensity constraints (see below), effective concavity descent is essential.

Following [15], our active contour has an objective function with both internal and external terms. For active contour control points $\{\vec{x}_i\}_{i=1}^{s}$, where $\vec{x}_i = [x_i \; y_i]^T$, we have

$$E_{contour} = \sum_{i=1}^{s} \left[E_{int}(\vec{x}_i) + E_{ext}(\vec{x}_i) \right] \; . \tag{4}$$

The internal energy term maintains even spacing of the control points and smooth curvature over the length of the contour [15], while image edges, motion constraints, and intensity constraints are represented in the external term,

$$E_{ext}(i) = \gamma E_{DT}(\vec{x}_i) + \delta E_{mo}(\vec{x}_i) + \kappa E_{inten}(\vec{x}_i) \; . \tag{5}$$

In our formulation, image edges are represented by their DT field [3]. Motion constraints are incorporated by using them to identify corresponding image edges, and subsequently providing an additional DT-based attractive force E_{mo} to those edges. When a motion edge is aligned with an image edge, the influence of that edge is boosted through this term. Image edges normal to the object's motion direction, which do not have motion constraint support, have their influence attenuated, as presumably they belong to a different motion layer. The DT fields corresponding to the edge-based terms are generated as described in [23], with weighting terms γ and δ.

Information from intensity constraints is incorporated by adding a "balloon force" [3] on the active contour points based on the intensity constraints at that point. The force acting on contour point $\vec{x}_i$ is

$$\vec{f}_{inten}(\vec{x}_i) = \kappa M(\vec{x}_i) \vec{n}(\vec{x}_i) \tag{6}$$

where $\vec{n}(\vec{x}_i)$ represents the vector normal to the active contour at $\vec{x}_i$ and

$$M(\vec{x}_i) = 2P_q(\vec{x}_i) - 1 \; . \tag{7}$$

Recalling that $\sum_i P_i(\vec{x}) = 1$, we see that $M(\vec{x}_i)$ will be positive when the object being tracked has an ownership > 0.5, negative when the ownership is < 0.5, and zero when the ownership is 0.5. Having a zero-force for $P_q(\vec{x}_i) = 0.5$ is somewhat geared toward a two-layer model. For a model with more layers it may be appropriate to have this force positive only when $P_q(\vec{x}_i)$ exceeds the individual ownership by all other layers.

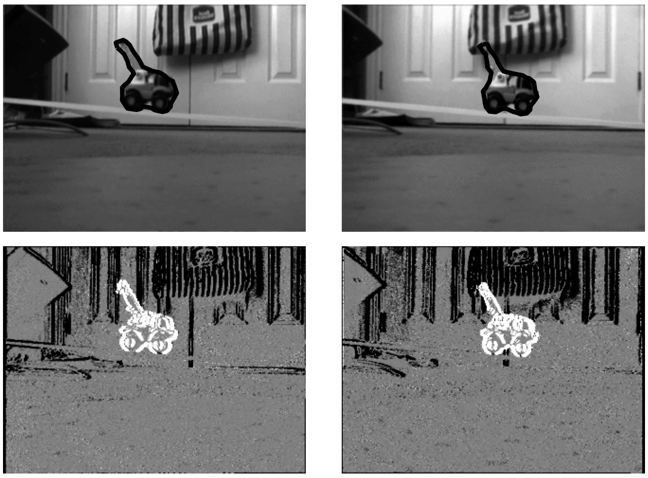
The sign of of $M(\vec{x}_i)$ indicates the direction of the intensity constraint force (expansion or contraction), and its magnitude indicates the strength of the force, scaled by weighting factor κ. So assuming $\vec{n}(\vec{x}_i)$ is directed toward the exterior of the closed contour, a positive $M(\vec{x}_i)$ value encourages expansion, a negative value contraction, and a zero value neither. Note that in regions where this force is zero, the active contour moves to the nearest motion constraint supported edge. The active contour's optimization follows [15, 23], which describes an iterative solution to the active contour energy minimization problem, incorporating the two internal force terms described above, and external force maps.

The active contour's initialization in the first frame is based on a convex hull contour placed around the target object's estimated bounding pixels. This is accomplished by thresholding the intensity constraints, and performing a local clustering by applying connected components analysis. The clustering is necessary because we do not want to allow outlier constraints to adversely affect the convex hull.

In addition, an alternative *Condensation*-based curve tracker [11] can be adopted in place of the active contour module after initialization. We use the shape template provided by the active contour to initialize this alternative boundary estimation technique, which is appropriate for objects undergoing any parametric deformation, although our current implementation employs an affine motion model.

2.4. Motion Feedforward

Information from the motion estimation step is used to move the active contour between subsequent frames. Since the active contour lies on the object boundary, it should move with the same velocity as the object, such that each boundary control point, $\vec{x}_i$, moves with velocity $\vec{u}(\vec{x}_i)$, given by the object's motion parameters and the control point's location. As each new frame is acquired and the interframe motion is computed, the active contour is moved from its final position in the previous frame to its expected position in the new frame, by moving each of its control points according to the estimated flow at their locations. During the initial frames of the sequence, the active contour is then updated again by the active contour optimization techniques, iteratively minimizing its energy function



Frame 31 Frame 50

Figure 3. **This figure shows two frames from a sequence with a toy tow truck moving down a ramp. The camera is hand-held and undergoing considerable and erratic motion. The active contour position is shown by the bold black line around the toy truck. The intensity constraints for the tow truck's motion layer are shown in the bottom row, with the tow truck assigned high ownership shown by the white areas, and the background in black.**

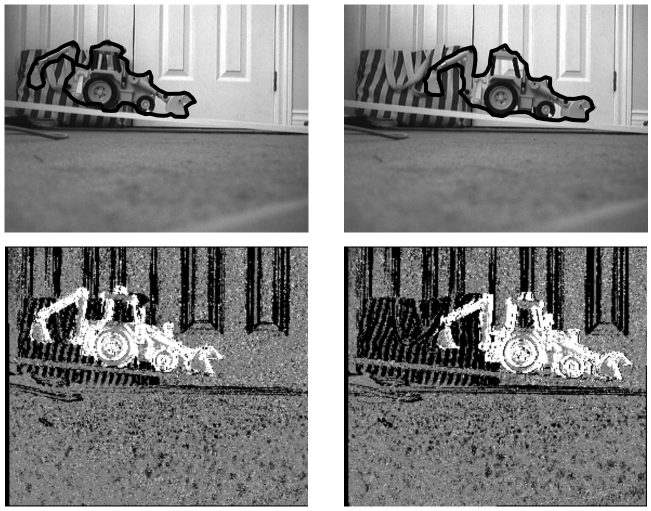
with respect to the intensity constraints for the current frame and local image edges. It is possible that new intensity constraints will appear outside the active contour boundary. Whenever the active contour finds itself in the interior of a region of intensity constraints, it will expand in order to "collect" the new constraints outside the boundary.

2.5. Boundary Feedback

In the first frame, when the active contour has not yet been initialized, all motion constraints are used in the motion computation. However, once the active contour has been initialized, we use it to improve the motion estimation step by using only motion constraints found inside the active contour. In this way, the inside/outside spatial coherence provided by the boundary is used to improve the spatial coherence of the region (motion) model. As a result, we limit the effect of constraints that are not derived from the object of interest.

3. Results

Results from our technique are demonstrated in Figure 3. This sequence shows a toy tow truck rolling down a ramp.

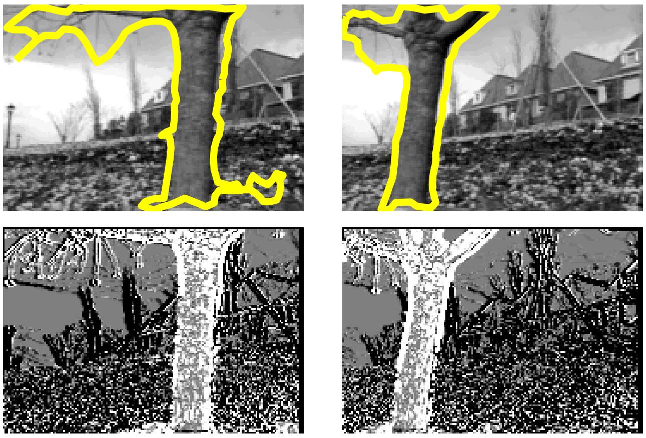


Frame 31 Frame 50

Figure 4. **The bulldozer truck illustrates the segmentation of an object that possesses deep concavities and also transparency, through the operator window. The motion-directed active contour descends into the deep concavities created by the backhoe, while the background area that is visible through the operator window is correctly segmented in the intensity constraint maps shown in the lower row.**

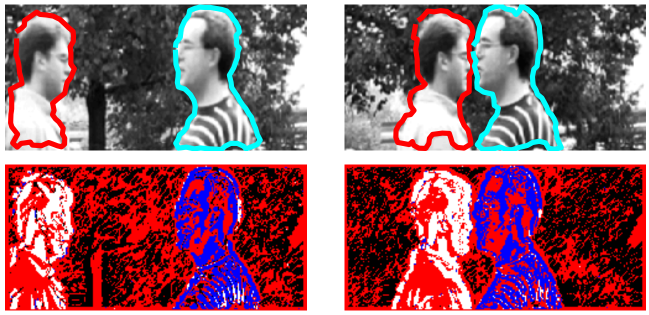
The camera is hand-held, and the background region undergoes motions that are roughly horizontal, and in both leftward and rightward directions. Our motion estimation component automatically fits two affine motion layers to the first two frames to start the algorithm running. The background motion is detected first due to the fact that it generates many more motion constraints than the truck. The motion layer of the truck object is recovered by estimating motion parameters based on constraints marked as outliers and inserting a new layer into the model.

The active contour is able to effectively describe the shape of the tow truck through the sequence, demonstrating the ability to consistently segment an independently moving object against a moving background. One apparent problem is that the active contour does not fully enter the concavity between the truck's wheels initially. However, this is not incorrect, as the lack of texture seen in this region during the first few frames makes it as likely to belong to the truck as to the background. In later frames, any background texture that appears in this region is identified as belonging to the distinct background motion layer, and the active contour is able to move inward due to the negative value of $M(\vec{x}_i)$ in this region.



Frame 2 Frame 28

Figure 5. This figure shows two frames from the flower garden sequence with the active contour indicated as a bold black line. The lower images show the respective intensity constraint maps for the foreground tree object, indicated by the white areas. The motion-detected background areas are shown as the black regions.



Frame 2 Frame 15

Figure 6. The walking sequence illustrates a scene in which two men walk toward each other. Motion segmentation-based intensity constraint maps (lower row) provide an effective coarse segmentation of the image, assisting the active contours to recover each man's profile.

A second image sequence of a more complicated bulldozer shape is shown in Figure 4, with front and rear scoops that form deep concavities in its shape. Our active contour is able to descend completely into the large concavities, notably recovering the shape of the rear scoop. Also of interest is the ownership of background structure through the window of the operator's compartment, which is correctly identified by the region-based segmentation technique. This feature will facilitate future work in tracking interior holes in objects while they move. One notable flaw in the active contour's shape can be seen in the area between the front wheel and the front scoop, where the small concavity is not recovered, primarily due to the small size of this concavity, which prevents the effective motion segmentation of this small area, as can be seen in the intensity constraint maps.

Figure 5 illustrates the application of our segmentation system to the flower garden sequence. The motion estimation process effectively segments the foreground tree object from the slower moving background area, and the active contour successfully recovers the tree's shape. It can also be seen that the active contour has difficulty descending into the small concavities created by the thin branches at the top left of the tree trunk. Although the active contour appears to be aligned to the edges created by the clouds, the contour is in fact forced to remain in this position by the normal forces that push the contour outwards to accommodate the thin branches of the tree. These branches are clearly indicated in the intensity constraint maps. The relatively small width of the concavities prevent the the active contour from descending into them, and this is visible in both frames of the flower garden sequence.

A walking sequence [14, 8] is shown in Figure 6, in which two men approach each other against a stationary but noisy background. Both independently moving objects are correctly segmented from the background and from each other by the motion segmentation process, while the resulting active contours correctly recover their moving outlines.

The results shown by these image sequences demonstrate two other important strengths of this segmentation technique: first, the use of the active contour to generate the final segmentation result incorporates local edge information into the boundary estimate that would only be included implicitly in a purely motion-based technique, in the form of the spatiotemporal gradient constraints that by their definition require smoothing of the region. Secondly, the interframe propagation of the active contour imposes an implicit short-term smoothness constraint on the final shape estimate, but allows for smooth deformation within the well defined constraints of the active contour.

4. Summary and Conclusions

In this paper, we show how motion layer estimation and boundary tracking offer separate but complementary forms of spatial coherence that can be combined to yield improved object boundary detection. The main advantages of our approach are its low complexity, its ability to accurately recover the closed boundaries of moving objects, and its generality in terms of allowing for multiple moving objects, moving backgrounds, or a moving camera. The approach is novel in its use of intensity constraints to provide driving forces for the active contour, as well in its feedforward and feedback relationships between the motion estimation and boundary estimation steps. The derivation of intensity constraints is not specific to intensity, and other local image properties such as color and phase can be used to improve the density of constraints used by the active contour.

In future work, we plan to incorporate topologically adaptive active contours to accommodate holes in the boundary recovery, incorporate on-line shape learning for the *Condensation* algorithm to accommodate non-rigid parameterized deformation beyond the affine range, and explore other image features such as color and phase to improve our region-based segmentation results.

Acknowledgments

The authors would like to thank Allan Jepson and Chakra Chennubhotla for their discussion and comments. The authors gratefully acknowledge the support of MD Robotics, CITO and NSERC.

References

[1] B. Bascle and R. Deriche. Region tracking through image sequences. In *Proceedings of the 5th International Conference on Computer Vision*, pages 302–307, Boston, MA, June 1995. IEEE Computer Society Press.

[2] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion boundaries. *International Journal of Computer Vision*, 38(3):231–245, 2000.

[3] L. Cohen and I. Cohen. Finite element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), November 1993.

[4] D. Cremers and S. Soatto. Variational space-time motion segmentation. In *Intl. Conf. on Computer Vision*, volume 2, pages 886–89, Nice, 2003.

[5] T. Darell and A. Pentland. Robust estimation of a multi-layered motion representation. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 173–177, Princeton, New Jersey, October 1991.

[6] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[7] A. Dumitras and A. Venetsanopoulos. A comparative study of snake models with application to object shape description in bi-level and grey-level images. In *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Baltimore, USA, June 2001.

[8] B. Frey, N. Jojic, and A. Kannan. Learning appearance and transparency manifolds of occluded objects in layers. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, June 2003.

[9] M. Gelgon and P. Bouthemy. A region-level motion-based graph representation and labeling for tracking a spatial image partition. *Pattern Recognition*, 33:725–740, 2000.

[10] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1), February 1994.

[11] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[12] A. D. Jepson and M. J. Black. Mixture models for optical flow computation. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York*, pages 760–761, Los Alamitos, California, June 1993. IEEE Computer Society Press.

[13] A. D. Jepson, D. J. Fleet, and M. J. Black. A layered motion representation with occlusion and compact spatial support. In *European Conf. on Computer Vision, ECCV 2002*, volume 1, pages 692–706. Springer-Verlag, 2002.

[14] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, December 2001.

[15] D. T. M. Kass, A. Witkin. Snakes: Active contour models. *International Journal of Computer Vision*, 1(2):321–331, 1987.

[16] W. J. MacLean, A. D. Jepson, and R. C. Frecker. Recovery of egomotion and segmentation of independent object motion using the EM algorithm. In *Proceedings of the 1994 British Machine Vision Conference*, pages 175–274, York, England, September 1994.

[17] N. Paragios and R. Deriche. Geodesic active contours and level set methods for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.

[18] H. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–830, 1996.

[19] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, January 1998.

[20] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York*, pages 361–366, Los Alamitos, California, June 1993. IEEE Computer Society Press.

[21] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 520–527, 1997.

[22] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 321–326, 1996.

[23] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow,. *IEEE Transactions on Image Processing*, pages 359–369, March 1998.