Available online at www.sciencedirect.com
SCIENCE DIRECT
Image and Vision Computing 24 (2006) 680–692
image AND vision COMPUTING
www.elsevier.com/locate/imavis

# Integrating region and boundary information for spatially coherent object tracking

Desmond Chung [a], W. James MacLean [a,*], Sven Dickinson [b]

[a] Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada M5S 3G4
[b] Department of Computer Science University of Toronto, Toronto, Canada M5S 3G4

Received 23 July 2004; received in revised form 12 August 2005; accepted 9 September 2005

**Abstract**

The problem of segmenting image sequences based on 2D motion has been under study for many years now. Most early approaches were either region-based, doing some sort of robust motion estimation, or boundary-based, preferring instead to track the bounding contours of the moving image region. In this paper, we explore an approach based on a synergy between these two previous approaches. For example, while motion constraints are often in violation of their underlying assumptions at region boundaries, image edges are a rich source of information. The approach we propose uses feed-forward to use region-based information to propagate boundary estimates, feedback to use boundaries to improve motion estimation, and finally uses motion-based warping to compare image appearance between frames in order to provide additional information for the boundary estimation process.

We show results from an implementation in which a hierarchical, layered-motion estimation using parametric models is coupled with a distance-transform based active contour. The system is shown to provide stable and accurate segmentation in sequences with background motion, and multiple moving objects. Quantitative measures are proposed and reported for these sequences. Finally, a modification is detailed which allows the system to incorporate a Condensation algorithm tracker, but without requiring off-line learning in advance.


*Keywords:* Segmentation; Motion estimation; Boundary recovery; Parametric models; Motion layers

## 1. Introduction

The detection and measurement of object motion in image sequences is a central problem in computer vision and video processing. Accurate and reliable estimates of object motion and spatial extent are required for tasks such as video coding, object recognition, object avoidance during navigation, and accurate determination of the observer's motion in an environment where objects may have their own motion independent of that of the observer.

Attempts at motion estimation in image sequences have 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. When determining the motion of an image pixel, layered motion approaches generally utilize no information about the motion of neighbouring 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 model typically focus on object boundaries while ignoring the rich motion information encoded within the object boundaries.

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. The approach, which is described in detail in Section 3,

* Corresponding author. Tel.: +1 416 946 7285; fax: +1 416 946 8734.
*E-mail addresses:* desmond.chunglincheung@utoronto.ca (D. Chung), maclean@eecg.toronto.edu (W.J. MacLean), sven@cs.toronto.edu (S. Dickinson).

0262-8856/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.09.020

is general, and makes no assumptions about a static background, a static camera, or the number of moving objects. 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 directions for future research.

## 2. Related work

The notion of spatial coherence in visual motion analysis has been in the literature for quite some time. Yuille et al. ([1–3]) explore the notions of both spatial and temporal coherence, and present a mathematical theory which claims to provide explanations for a variety of motion perception phenomena, including the aperture problem and motion capture. The theory proposes the notion of a dense velocity field defined even where there is no local image evidence for motion, and which is estimated from image measurements.

Previous work on object segmentation and tracking can be divided into region based approaches ([4–10]) and boundary-based approaches ([11,12]; [13,14], [15]). Among the region-based approaches, some ([6–9]) can be classified as layered approaches, with the latter two using models to describe image regions. In [8], layered flow is computed using octagonal-shaped regions to limit the region of support 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. This method does not attempt to fit an accurate boundary to any region. In [9], an elliptical appearance model is learned and tracked. Both short-term (2-frame) and longer-term structure is represented and tracked. A model-less, layered approach is taken in [10], where the authors perform motion-based segmentation by computing motion parameters for fixed regions, then merging based on adjacency and similarity of motion parameters. Another model-less approach is found in [4,5], where a method for nonparametric flow estimation is given. A Markov random field model is used to provide a prior encoding the notion that neighbouring image points are likely to be related. A mean-field approximation is used to make the method computationally feasible, but the method gives no explicit estimate of the region boundaries.

Boundary-based approaches fall into two categories: probabilistic contour tracking ([11,12]) and active contour approaches ([16]; [13,14], [15]; [17,18]). A probabilistic formulation of curve tracking is presented in [11] that propagates a set of sampled states to approximate a posterior distribution on possible states given the observed data. The method requires an initial curve template and a learning phase to acquire a motion model, after which it can operate uninitialized on new image sequences.

It appears to require a stationary background as this is how the motion model is learned, and no results to the contrary are shown. This approach is furthered in [12] where the motion-learning phase is replaced with an explicit boundary motion model augmented with rough estimates of boundary motion derived from a layered flow model. The model does not directly address the issue of grouping motion edges to identify the boundary of a particular object. The approach in [12] can also be thought of as region-based in that an explicit flow model is provided for non-boundary regions.

The first attempt at integrating region information with an active contour boundary model is found in [16]. Here the authors use a constant, affine or homomorphic warp computed through a correlation approach to compute the displacement of the entire region, and to update the active contour between frames. The active contour is then allowed to settle on image edges. 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. All sequences presented by the authors have a single moving object against a static background or a single motion over the entire image.

Perhaps the closest work to that described in this paper is the geodesic active contour formulation, which automatically handles contour splitting and merging based on a single energy function. It is proposed in [13–15], which assumes a static background, so that an image differencing approach can be used to detect motion. 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. More recently, work by [17,18] employs level set contours for object boundary tracking, using colour and texture information to drive the active contour, in a manner similar to our proposed intensity constraints.

## 3. Integrating region and boundary constraints

An overview of the proposed system is shown in Fig. 1. The basic approach is to use a feed-forward, feed-back approach to combine region-based information, in our case motion constraints and intensity-consistency constraints, and boundary-based information, in our case the object boundary as determined by an active contour. Instead of relying solely on either type of information, we use the two types together to improve the results.

In our region-based module, gradient-based motion constraints are used to compute a parametric, layered flow model to estimate local image motion. Since motion constraints are often too sparse to perform proper boundary estimation due to lack of texture, we use the estimated motions to perform an intensity-based consistency check of each pixel against each motion layer. This is achieved by warping each pixel location according to the recovered motions, and comparing image intensities in the two frames accordingly. This yields a denser

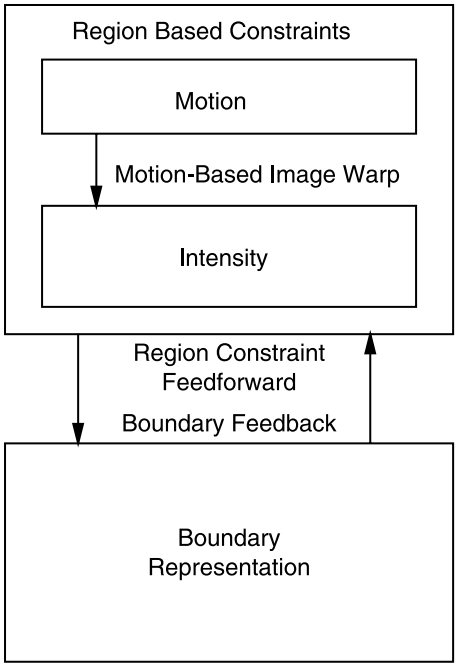



Fig. 1. This figure illustrates the basic structure of our approach. Region-based information is used to derive flow constraints, which are typically sparse. This information can be made denser by warping image pixels to find pixels with matching intensities in the two images. These constraints provide 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 only consider motion constraints within the contour during the motion estimation stage.

constraint representation that can be used to assist the active contour model. The boundary-based module consists of a distance-transform active contour. In addition to the usual edge-based energy terms in the contour's cost function, we introduce an additional, novel term based on the intensity consistency data from the region-based module. This places higher weight on image edges that carry relevant information about motion boundaries.

In addition to using the intensity consistency constraints (ICC's) to assist the active contour in finding motion-relevant boundaries, the motion model for the layer associated with the active contour is used to warp the contour between frames, thus providing it with a more accurate starting location. This feed-forward improves the contour's performance by minimizing the iterations required to update the contour between frames, and avoiding nearby, extraneous image edges that might otherwise attract the active contour.

Finally, in order to improve motion estimates, only motion constraints within the contour are used to update motion estimates in subsequent frames. This feed-back allows the parametric flow model to focus on the motion characteristics of the object being tracked, without being distracted by similar motion elsewhere in the image. In the initial frames, the motion is estimated without this information, and the initial contour location is determined by connected-components analysis of motion constraints owned by each motion layer.

### 3.1. Region-based motion estimation

The first step in tracking objects is to estimate a layered motion model for the image sequence. Layered motion models have been explored extensively in the literature ([7,19,20]). In [7] the authors start with pre-computed optic flow, and cluster affine motion models in layers to account for the input flow field. In [19] multi-layer parametric optic flow is computed using a mixture of Gaussians augmented with a constant probability layer to accommodate outliers in the measurements. In formulating the model, the measurement conditional probabilities are assumed to be independent and identically distributed (i.i.d.). The paper describes fitting constant-flow models to each layer, although suggests the extensions appropriate for more complex models such as locally affine flow. In [20] a mixture approach is again used, although robust estimators are used instead of an outlier process in the mixture and a minimum-description-length (MDL) approach is used to determine the correct number of layers. Both assume i.i.d. image measurements. We follow the formulation of [19] in using a mixture model together with the EM algorithm to both integrate motion constraints and estimate motion parameters. While our approach uses a parametric model for flow, a dense, non-parametric method can (in principle) be substituted.

The brightness constancy constraint, $\nabla_{\vec{x}}^{\mathrm{T}} I\vec{u}(\vec{x}) + I_t = 0$, is well known, and is the starting point for the estimation of 2D image velocity $\vec{u}(\vec{x}) = [u_x(\vec{x})\, u_y(\vec{x})]^{\mathrm{T}}$. Each image location provides a constraint vector $\vec{c}(\vec{x}) = [I_x\, I_y\, I_t]^{\mathrm{T}}$ that satisfies (in the absence of noise) $\vec{c}(\vec{x})^{\mathrm{T}}\vec{u}_{\mathrm{h}} = 0$, where $\vec{u}_{\mathrm{h}} = [\vec{u}(\vec{x})^{\mathrm{T}} 1]^{\mathrm{T}}$ is a homogenous representation of $\vec{u}(\vec{x})$, and $\vec{u}(\vec{x})$ is assumed to be the motion experienced by the pixel at image location $\vec{x}$. The inner product of a constraint with a motion vector from another layer will be expected to yield a non-zero value, in general, thus providing a way to determine which constraints support which motion layers. The spatial image gradients are estimated by convolving with $G_x = (G(x,y;\sigma)/\partial x$ and $G_y = (G(x,y;\sigma)/\partial x$ where $G(x,y;\sigma)$ is a 2D Gaussian kernel with $\sigma = 1.5$ pixels. A constraint that has a zero value for $\nabla\vec{x}I$ corresponds to an image region with uniform intensity values, and provides no information about local motion. In practice, constraints with $||\nabla\vec{x}I||$ below a threshold of 2.5 intensity levels/pixel are ignored due to insufficient signal to noise ratio (SNR). Also, since in a coarse-to-fine flow estimation approach an upper limit can be placed the magnitude of $\vec{u}(\vec{x})$, it is possible to enforce the constraint $|I_t|/\|\nabla\vec{x}^{\mathrm{T}} I \nabla\vec{x} I\| < \|\vec{u}\|_{\max}$ in order to reject erroneous constraints. We use $\|\vec{u}\|_{\max}$ 2 pixels/frame.

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 parameters $\vec{\theta}$ is a likelihood function $p(\vec{c}(\vec{x})|\vec{\theta}, \vec{x})$ that indicates how well a constraint $\vec{c}(\vec{x})$ matches the motion. For example, for the constant motion model, we have $p_{\mathrm{constant}}(\vec{c}(\vec{x})|\vec{u}, \sigma) = G(\vec{c}(\vec{x})^{\mathrm{T}}\vec{u}_{\mathrm{h}}; 0, \sigma_{\mathrm{v}})$, where $G$ is a Gaussian density function. 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 $\pi_j$ are called *mixing parameters* and satisfy $0 \le \pi_j \le 1$ for all $j$ and $\Sigma_j \pi_j = 1$.

It is worth pointing out that the use of gradient-based approaches implicitly assumes, in practice, a certain amount of spatial coherence in the image velocity. This is due to the fact that the image spatial and temporal gradients have to be estimated using discrete approximations over a finite region. Constraints derived from image gradient measurements become anomalous if this region contains a motion boundary, and the traditional approach has been to use robust estimation techniques to remove these outliers, for example mixture models with outlier distributions ([19]) or robust estimators ([20]), or RANSAC ([21–23]) or something similar ([24]). We use an outlier layer with constant probability density in the range of 0.26–1.5 (based on value for $\sigma_v$ that decreases from 0.6 to 0.1 over the course of the EM iterations in a manner similar to simulated annealing) is used to model constraints not accounted for by other motion layers. An initial mixing proportion of 0.1 is assigned to the outlier layer, with the remaining mixing proportions being given an equal share of the remaining 0.9.

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

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

where

$$p(\vec{c}(\vec{x}_q)|\vec{\Theta}, \vec{x}_q) = \sum_{j=1}^{n} \pi_j p_j(\vec{c}(\vec{x}_q)|\vec{\theta}_j, \vec{x}_q)$$

and $\vec{\Theta} = [\vec{\theta}_1^{\mathrm{T}} \ldots \vec{\theta}_n^{\mathrm{T}} \pi_1 \ldots \pi_n]^{\mathrm{T}}$ is the collection of all model parameters. The EM algorithm ([25]) 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, so a good initial guess for parameters is helpful. For each EM step, we first compute the ownerships via Eq. (1), and then use these as weights in a least squares, analytic solution for the motion model parameters (either affine or constant) in the M-step. Standard singular value decomposition is used to compute the optimal parameter estimates.

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 [26–28], we perform a sequential analysis of dominant motions in the images. We start by computing a model for a single motion plus an outlier process designed to catch constraints not well-modelled 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. This continues until no further processes are added, or newly added processes become identical to existing ones.

To complete the discussion of the motion-layer estimation stage, it should be noted that our implementation is based on a multi-scale approach following that of [29]. The images are pre-processed into a pyramid representation, and motion estimation begins at the coarsest level of the pyramid, where presumably maximum displacement is on the order of one pixel or less, thus motivating the need for the $\|\vec{u}\|_{\max}$ parameter discussed above. Parameter estimation at each subsequent (finer) level computes residual flow after prewarping one of the images according to motion parameters from the coarser level—this is done separately for each motion layer. The residual flow parameters are then combined with those from the coarser level to get a final estimate for the current level, and the process moves on to the next level. At the coarsest levels there may not be enough constraints to reliably estimate full affine (or higher-level) parameter sets, and in such cases the model falls back to a translational model until finer resolution levels are reached.

### 3.2. Intensity constraints

One reason that layer-based motion estimation has traditionally not been used to provide boundary estimates is the relative sparseness of the recovered motion constraints. This is a natural consequence of using the spatial gradient as part of the motion constraint: image regions with little or no texture do not provide useful constraints. Even if the image gradient is non-zero, the signal to noise ratio of each constraint must be considered, and a constraint is rejected if it fails to meet a minimum SNR (Section 3.1). Further, even if an image region has sufficient texture, if the spatial gradient vectors are perpendicular to the local motion then no useful constraints will be recovered, as $I_t$ will be zero in this case. The lack of motion constraints along edges parallel to the direction of motion is enough on its own to deter attempts at boundary recovery using only motion constraints. In practice, it is usually found that motion constraints alone are insufficient to drive boundary recovery.

In attempting to recover boundary constraints using motion constraints alone we are ignoring a rich and stable source of information, namely the intensity values themselves. Intuitively, we do not expect the appearance of an object to change radically from one frame to the next. While this notion is at the heart of gradient-based motion constraints, it can also be used directly. We propose a novel method for incorporating intensity information into the boundary recovery process. Given an estimate of the motion parameters for a particular object motion in the scene, it is possible to compare intensity values between frames to check for consistency with the proposed motion. For any pixel location in the first frame, $\vec{x}(t_1)$, we can compute its new location according to motion model $j$ by warping the pixel location according to the proposed motion, $\vec{x}_j(t_2) = \vec{x}(t_1) + \vec{u}(\vec{x}(t_1); \vec{\theta}_j)$.

The comparison of intensity values is done assuming a Gausian distribution for the intensity values. If $G(I;\mu,\sigma)$ is

a Gaussian density, then the likelihood of a given pixel location $\vec{x}$ matching the $j$-th motion model is given by

$$L_j(\vec{x}(t_1)) = G(I(\vec{x}(t_1), t_1) - I(\vec{x}_j(t_2), t_2); 0, \sigma_{\text{ccd}})$$

where $\sigma_{\text{ccd}}$ is the expected pixel-noise standard deviation for the camera in use. For modern CCD cameras, this value will typically be in the range 1–2. It should be noted that it might be worthwhile considering signal dependent noise models (such as that suggested by [30]) in which the value of $\sigma_{\text{ccd}}$ is dependent on the expected image intensity, giving what is typically referred to as *signal dependent noise*.

A low value of $L_j(\vec{x})$ suggests a low likelihood of motion $j$ at image location $\vec{x}$, whereas a high value indicates a higher likelihood. However, a high value for $L_j(\vec{x})$ alone is not sufficient, as the same image location may have high likelihood for other motion layers. This situation is expected in regions that have uniform intensity values. More than one motion layer may map $\vec{x}$ onto a location with the same intensity value, especially if the size of the uniform region is large compared to the motion magnitudes. It could be argued that intensity constraints are no better than motion constraints in this respect, but this is not the case. Intensity constraints may still be useful in uniform regions in helping to distinguish between motions of differing magnitudes when one or more of the motions is large enough to move into a region of different texture or intensity, as might be expected near an object boundary. Also, they show (in practice) better performance in regions of low texture where noise considerations preclude the use of motion constraints. In regions with strong texture, it is expected that one motion layer will have a stronger likelihood than the others for a particular image location. An example of intensity constraint likelihoods for two motion layers is given in Fig. 2.

To provide a more meaningful context for intensity constraints, we again invoke the notion of process ownership as used in a mixture model. Here we assign each pixel an ownership by each motion layer by combining the various intensity likelihood values as follows,

$$Q_j(\vec{x}) = \frac{\pi_j L_j(\vec{x})}{\sum_k \pi_k L_k(\vec{x})}.$$

In computing this ownership, the index k ranges from 0 to $N$ where $N$ is the number of motion layers in the model. A special meaning is attached to $L_0$. It is a constant value that assumes the role of an outlier process. This is important in order to account for any pixels that are not well modelled by any existing motion layer. Instead of forcing ownership by the motion layer that fits the data least poorly, the outlier process can assume ownership. In Section 3.3 we will use the intensity constraint ownerships to provide an additional force for the active contour, thus it is important to discount data that is not consistent with any motion layer.

While others have used the notion of warping image intensities and comparing them between frames, none have done so with the purpose of combining information across layers to provide additional support for a boundary hypothesis. Both [16], and [13] warp image intensities as part of their motion estimation, although neither considers relative likelihoods between different motion layers. In both cases, each pixel intensity is warped only according to a single motion model, not considering information contained in other motion layers. In [31] intensity mapping is considered as part of the minimization term used to estimate motion layer parameters. Jepson et al. [8] warp pixel intensities for multiple motion layers, but not for the purpose of developing boundary constraints.

Fig. 2. The middle image shows the intensity ownerships for the horizontal motion of a cup and hand from the frame shown at the top. Note that the reflection of the cup in the table surface is correctly identified. The lower image shows intensity ownerships for the (stationary) background. In each image, black is 0.0, white is 1.0, and values in between are given by levels of grey.

In this section we have talked about 'intensity constraints' as if they were referring to grey-scale values, however, this is overly restrictive. The same concept can be used for colour values (either with or without brightness information retained),

and it can be applied to other pixel features, for example phase response when a complex-valued filter has been used to pre-process the image, or even measures of local texture. Multiple cues can be used simulataneously.

### 3.3. Boundary estimation using active contours

The boundary-based information used in our model is derived using an active contour model. Active contours, or 'snakes' as they are sometimes called, have been used by vision researchers since their introduction by [32]. Our implementation uses distance transform (DT) contours ([33,34]), although we have also successfully employed gradient-vector flow (GVF) contours ([35]). The advantage of these formulations is that they descend into deep concavities easily, and converge quickly when initialized away from features of interest. Since we will initialize our contour using a convex hull based on intensity constraints (Section 3.3.1), these are useful characteristics. Each motion layer has an associated active contour, except for the outlier and background layers.

Our contour implementation uses the standard model of internal forces, used to promote even control point spacing (continuity) and smoothness, and external forces used to draw the contour to image features of interest. For active contour control points $\{\vec{x}_i\}_{i=1}^{s}$, where $\vec{x}_i = [x_{1i}\, x_{2i}]^{\mathrm{T}}$, we have

$$E_{\mathrm{contour}} = \sum_{i=1}^{s} [E_{\mathrm{int}}(\vec{x}_i) + E_{\mathrm{ext}}(\vec{x}_i)].$$

The internal energy term is given by

$$E_{\mathrm{int}}(i) = \alpha \vec{x}'_i + \beta \vec{x}''_i.$$

Here, $\alpha$ and $\beta$ represent the relative weighting of each term, and $\vec{x}'_i$ and $\vec{x}''_i$ are first- and second-order difference approximations, respectively. Image edges, motion constraints, and intensity constraints are represented in the external term,

$$E_{\mathrm{ext}}(i) = \gamma E_{\mathrm{DT}}(\vec{x}_i) + \delta E_{\mathrm{motion}}(\vec{x}_i) + \kappa E_{\mathrm{intensity}}(\vec{x}_i).$$

The values of $\alpha = 0.05$, $\beta = 0.01$, $\gamma = 1$ and $\delta = 1$ were used for all image sequences. Image edges are represented by their DT field ([33]), with

$$E_{\mathrm{DT}}(\vec{x}) = \begin{cases} -1, & \mathrm{d}(\vec{x}) < 1 \\ -1/\mathrm{d}(x), & \text{otherwise} \end{cases}$$

where $\mathrm{d}(x)$ is the distance (in pixels) of image location $\vec{x}$ to the nearest edge. Motion constraints are incorporated by using them to identify image edges owned by the current motion process, and subsequently providing an additional DT attractive force

$$E_{\mathrm{motion}}(\vec{x}) = \begin{cases} E_{\mathrm{DT}}, & Q_j(\vec{x}) > 0.5, \text{ where } j \text{ is the current motion} \\ 0, & \text{otherwise} \end{cases}$$

to those edges. Motion edges are therefore, used to modulate the external force exerted on the active contour by nearby image edges. When a motion edge is aligned with an image edge, the influence of that edge is boosted. Image edges normal to the object's motion direction that do not have motion constraint support have their influence attenuated, as presumably they belong to a different motion layer. Motion edges parallel to the object's motion, and even though they do not receive support from the motion constraints, still have support from the image edges, and in the absence of motion constraints give the active contour something to attach to.

To incorporate the information provided by the intensity constraint ownerships, we incorporate an additional force into the active contour cost function. This force, which takes the form of a balloon force (directed in or out along the contour normal at a control point, ([34])), is based on comparing the intensity constraint ownership of the motion layer to which the active contour is associated, with the largest ownership by any other layer. The force acting on contour point $\vec{x}_i$ is

$$\vec{f}_{\mathrm{intensity}}(\vec{x}_i) = \kappa M(\vec{x}_i)\vec{n}(\vec{x}_i) \propto \nabla E_{\mathrm{intensity}}$$

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

$$M(\vec{x}_i) = Q_j(\vec{x}) - \max_{\vec{j}, j \neq 0} Q_{\vec{j}}(\vec{x}).$$

We see that $\vec{f}_{\mathrm{intensity}}(\vec{x}_i)$ will be positive when the motion layer associated with the contour has greatest ownership of the intensity consistency data, and negative if not (a net force of zero occurs when there is no unique strongest owner). Here the maximum of the other ownerships is computed excluding the outlier process. The reasoning for formulating the force in this way is as follows. If any motion layer has a strong ownership for this image location, it will dominate this force. If it is the layer associated with the active contour, then the force will be positive, otherwise it will be negative. In the event that all motion layers have roughly equal ownership of the location (suggesting a uniform region), then the resulting force will be small and have little effect on the contour. The sign 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 $\kappa$. So assuming $\vec{n}(\vec{x}_i)$ is directed outward, 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 or the nearest image edge. The value of $\kappa$ was set to a value in the range of 0.9–1.3 for regions identified as not belonging to the current motion (with a negative value of $M(\vec{x}_i)$ providing a contraction force), 1.3 for regions identified as belonging to the current motion (thus providing expansion), and a value in the range $-0.01$ to $-0.02$ for neutral regions to provide slow contraction in regions of uncertain ownership. The active contour's optimization follows ([32,35]), which describe an iterative solution to the active contour energy minimization problem, incorporating the two internal force terms described above, and external force maps. Every five iterations a point management routine is invoked to add or remove points so that the distance between control points always remains in the range of 4–8 pixels. When the distance between two control points

exceeds 8 pixels, a new control point is added midway, and when two points move closer than 4 pixels one of the control points is removed.

*3.3.1. Active contour initialization*

The active contour's initialization in the first frame is automatic, and is based on a convex hull contour placed around the target object's estimated bounding pixels. This is accomplished by thresholding the intensity constraints from a motion-only analysis in the first pair of frames, and then using a median filter to remove spurious constraints, and finally performing connected-components analysis (CCA) clustering ([36]) on the intensity-consistency constraint data. The CCA clustering is necessary because we do not want to allow outlier constraints to adversely affect the convex hull, nor do we want to initialize a contour around multiple objects moving with the same motion. An example is shown in Fig. 3. The CCA analysis starts by defining an affinity matrix

$$W_{ij} = \begin{cases} \exp\left\{\dfrac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}\right\}, & |\vec{x}_i - \vec{x}_j| < d_{\max} \\ 0, & |\vec{x}_i - \vec{x}_j| \geq d_{\max} \end{cases}$$

for all pairs of points in a motion layer $\vec{x}_i$ and $\vec{x}_j$[1]. A threshold $\tau$ is used for comparing affinity values and determining whether to mark the corresponding labels as equivalent or generate new labels. The working values of $d_{\max}=5$, $\sigma=3$ and $\tau=0.8$ were determined through a series of experiments on representative image sequences.

*3.4. Motion feed-forward*

In addition to $E_{\text{motion}}$, the region-based motion estimation of Section 3.1 is combined with the active-contour-based boundary estimation of Section 3.2 through a feed-forward mechanism. After the motion layer parameters have been estimated for any frame pair at $t$ and $t+1$, the associated active contour is updated based on edge information for the current frame, and the motion information. In order to correctly propagate the active contour between frames, the motion model for the contour is used to estimate motion for each of the contour's control points. These points are then updated according to

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{u}(\vec{x}_i(t); \vec{\theta}_j).$$

In this way, the active contour is propagated between frames in a manner that is consistent with the motion estimates. The advantage of this is that when it comes time to update the active contour in the next frame, its control points will already be close to their final positions, and very few iterations of the contour optimization will be needed. Also, if the relative motion between frames is large, this method of propagation will help the active contour bypass spurious image structure,

[1] Here the subscripts to not refer to control points of an active contour, but rather provide a way of indexing over all points in a motion layer.

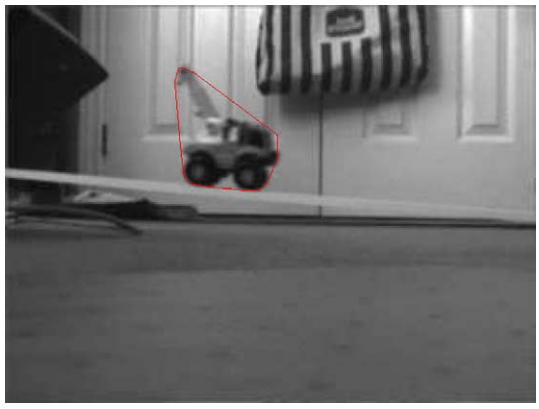


Fig. 3. This figure shows the convex hull of intensity constraints used to initialize an active contour around the tow truck. The red line is the active contour's starting point in the first frame.

such as edges internal to the object or part of another image region, which otherwise might cause the active contour optimization to settle into an incorrect local minimum.

It is worth noting that the shape of the contour may change as an object is tracked over time. This can happen as a result of the object deforming, or becoming partially occluded. In this case, the change in the contour shape will be primarily driven by image edges ($E_{\text{DT}}$), intensity consistency ($E_{\text{intensity}}$) and motion constraints ($E_{\text{motion}}$). Normal contour optimization is sufficient to track such changes.

*3.5. Boundary feed-back*

The second manner in which the region- and boundary-based information are integrated is through feed-back from the active-contour boundary to the motion estimation stage. In a traditional layered-flow model, motion parameter updates are based solely on the EM ownerships for individual motion constraints. However, once an object with a bounding contour has been identified, it is desirable to set these ownerships to zero outside the contour in order that the motion estimates for the object being tracked are not affected by similar, but possibly unrelated, motions outside the region. Other regions with similar motions can be tracked separately by spawning new motion layers for them with similar motion parameters, but with separate ownerships. In this sense, the active contour provides a computationally simple yet effective way in which to enforce the notion of spatial coherence in the motion estimates.

## 4. Results

In the following results, the assumed value for CCD sensor noise is $\sigma_{\text{ccd}}=1.0$, and the $\pi_j$'s are assumed equal in computing the intensity constraint ownerships. Computation time is roughly 1–2 s per frame using Matlab™ on a 2.6 GHz dual Xeon processor workstation with 4 GB of physical ram (although only one processor actively performs the computations). Before presenting the results, we begin with a brief description of the sequences considered.

### 4.1. Sequences

The results presented are from four sequences, two from the authors' lab and two that have been used by multiple researchers. Each sequence is described below:

#### 4.1.1. Scoop sequence

In this sequence, a toy construction vehicle rolls down a ramp from left to right. The vehicle has a complicated shape, and the background is stationary.

#### 4.1.2. Tow truck sequence

In this sequence, a toy tow-truck rolls down the same ramp, but the background is no longer stationary. The camera is handheld, with dominant motions in both the left and right horizontal directions. The tow truck also has a complex shape due to the towing boom and hook arrangement. A second version of this sequence, the same except for having a stationary background, is used for comparison purposes.

#### 4.1.3. Frey–Jojic sequence

In this sequence, first used in [37], two subjects move horizontally against a stationary (but noisy) background. They move towards each other, and in the full sequence one passes in front of the other, fully occluding the other.

#### 4.1.4. Flower garden sequence

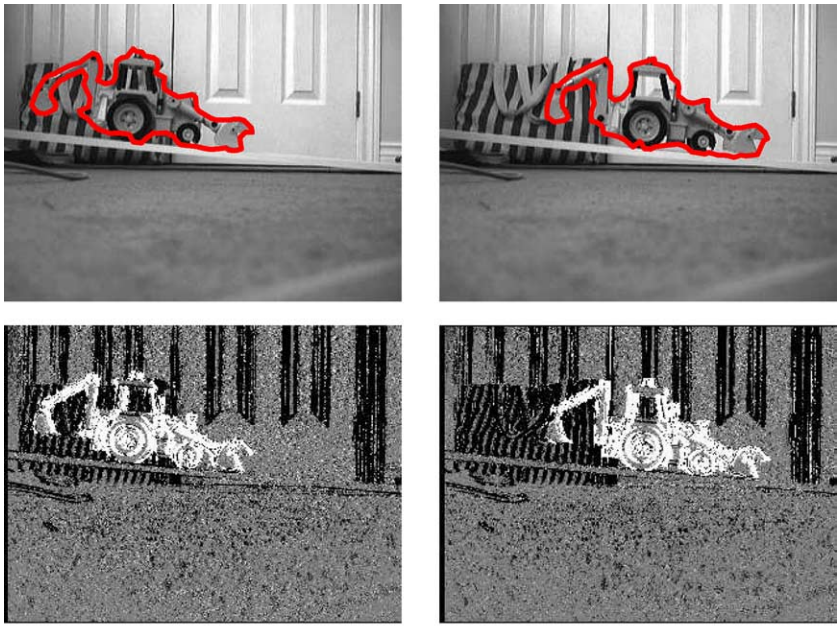
In this standard test sequence, an observer translates roughly fronto-parallel to a scene with a tree in the foreground, a flower garden, which slopes away from the observer behind the tree, and a house in the background. While there are no independently moving objects here, the amount of parallax between the tree and the garden and house is substantial. The motion and the geometry of the flower garden, mean that a simple translational model is insufficient. Strictly speaking, a full projective model is required to account for the foreshortening in the flower garden region, but our analysis is based on fitting an affine model.

### 4.2. Performance

Results from each of the four sequences are shown in Figs. 4–7. In the top row of each figure are shown two frames from the sequence with bounding contours shown for the independently moving objects (or, in the case of the flower garden sequence, the tree). In the bottom row of each figure are shown the intensity constraint maps for the moving objects, with each map corresponding to the frame displayed above it.

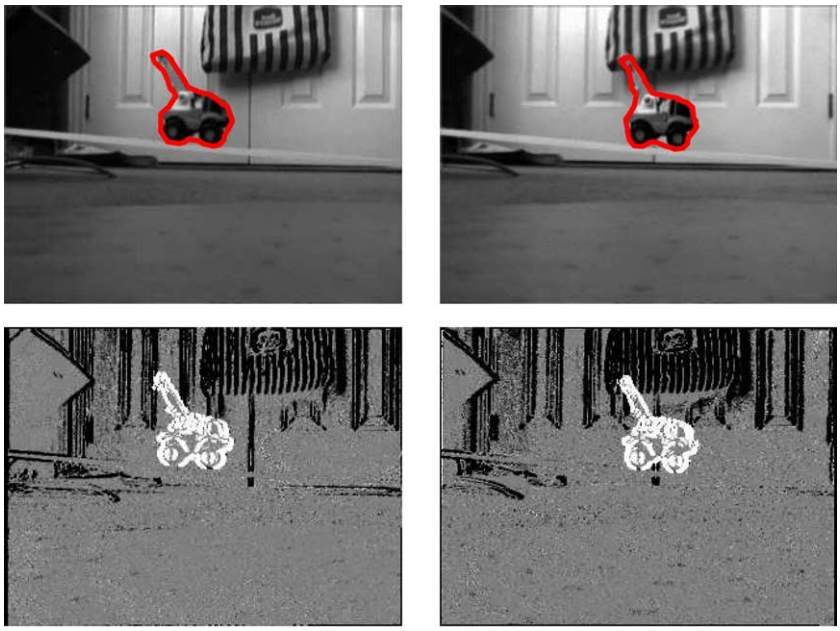
In each case the foreground objects are seen to be correctly identified, and their boundaries recovered accurately. A number of specific cases are worth commenting on further. In general, shape is well recovered in most instances. In particular, the complex shapes of the towtruck and even the construction vehicle (the latter with deep concavities) are seen to be well recovered. It should be noted that small and very narrow concavities are somewhat problematic (e.g. the space between the tree branches in the flower garden sequence). This is partially due to the difficulties of propagating active contours into deep concavities, and partially due to the finite spatial extent of the gradient filter kernels $G_x$ and $G_y$ used to generate the motion constraints.

Using sequential process generation to determine the number of motion layers from the outlier constraint population works well for the first three sequences presented. In the case of the flower garden, it just gives two motions, although it could be argued that three might be more appropriate if the house motion



Frame 31 Frame 50

Fig. 4.



Frame 31 Frame 50

Fig. 5.

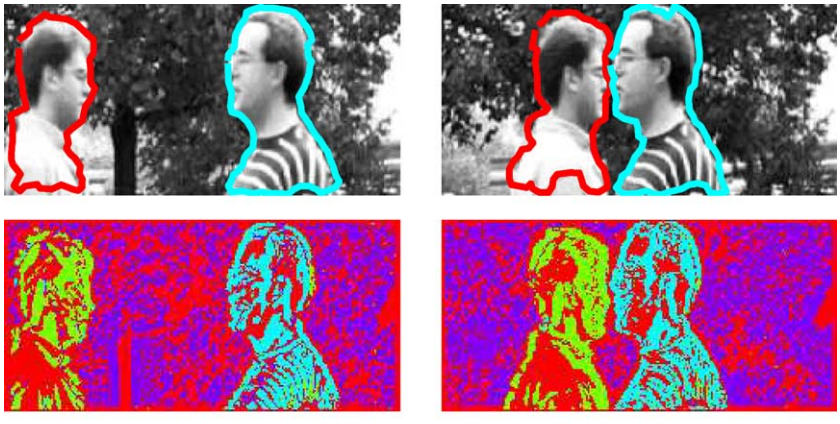
is considered separate from the garden motion. This is partly an issue of the value of $\sigma_v$ used in the process ownership function.

An apparent problem in both the Scoop and towtruck sequences is that the area between wheels of the tow truck, or between the front wheels and the bucket of the construction vehicle, are segmented as part of the vehicle and not the background. This is a consequence of the method employing no top-down contextual knowledge. Because these regions are textureless, there is no evidence to support them belonging preferentially to either the foreground or background. In this case, the presence of a strong edge along the ramp becomes the driving force for the contour. In the event that background texture becomes available in these regions at a later time, as is the case in the towtruck sequence, the boundary is seen to move inwards due to negative value of $M(\vec{x})$ in these regions.

In the Scoop sequence the background is visible through the operator's cab. It can be seen that this is correctly labelled in the intensity constraints. This will allow for future enhancements to the method to recover 'hole boundaries'.

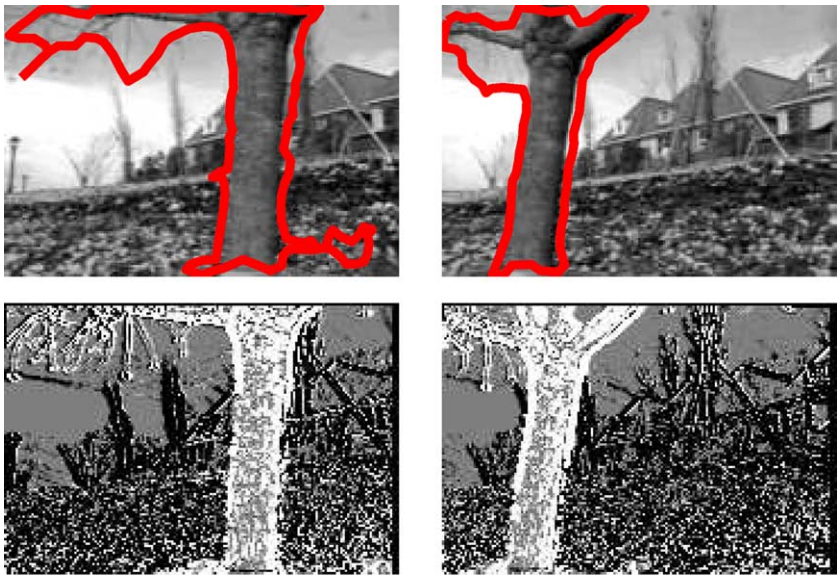
In the Frey–Jojic sequence, it is seen that the lower edge of the left subject's shirt is difficult to recover due to lack of texture and its being along the image boundary-at present the system does not pay special attention to image dimensions. Later in the sequence, when the right subject occludes the left one, the contour of the left subject contracts as should be expected. There is a slight amount of overlap in the contours—at present there are no forces acting between the contours to prevent such occurrences, but this is not difficult to add.

Finally, in the flower garden sequence the tree is seen to be well segmented along its trunk, although in frame 32 some of



Frame 2 Frame 15

Fig. 6. The false-colour coding for the intensity ownership constraints in the bottom two images is: purple-motion 0 (background), green-motion 1, cyan-motion 2 and red-indeterminate.



Frame 32 Frame 58

Fig. 7.

the flower garden is incorrectly segmented with it. This is understandable since it has a very similar motion (it lies in roughly the same plane as the tree trunk) and, as mentioned above, there is no contextual information employed. Also, some of the tree's branches are incorrectly included even though the recovered motion parameters do not properly account for their motion. This indicates a slight inaccuracy in our recovered affine parameters. This has a subsequent effect on the intensity constraint maps, where the branches are identified as belonging to the tree model (but they do not lie in the same plane as the tree trunk), thus attempting to drive the active contour around them. It is difficult for the contour to descend into the concavities between these branches, leading to an inaccurate shape recovery.

The results shown here demonstrate the importance of combining boundary and motion estimation in the segmentation problem. Our segmentation is able to incorporate static edge information via the active contour. This is particularly important with respect to bounding edges, as motion constraints generated along these edges are often in violation of the brightness constancy assumption used to generate them. Further, the propagation of the contour between frames based on estimated motion improves the likelihood of tracking the correct edge from one frame to the next. Some amount of smooth deformation is also tolerated, as the active contour can adjust for small amounts of mis-registration when the curve is propagated.

### 4.3. Motion warp comparisons

In order to quantitatively assess the quality of recovered motion parameters, Fig. 8 shows the results of comparing the tracked object images by warping with the motion parameters computed for the tracked object, and creating a histogram of the resulting absolute intensity differences. Cubic interpolation was used in computing the image warps prior to comparison.

### 4.4. Ground-truth comparisons

In order to better assess the performance of the system with respect to the boundary contour, a comparison is made in one sequence against a manually segmented contour, and the results are shown in Fig. 9. The average absolute distance of the recovered contour from that of the ground-truth are 1.5 pixels. The main regions in which discrepancies occur are seen to have little texture in the background, leading to an ambiguity of ownership. For example, the region between the wheels, as described earlier, can easily be understood as part of the truck if no prior knowledge of the truck's shape is assumed. The grey areas average 13.77% the size of the interior of the ground-truth contour for a stationary-background sequence involving the tow truck (ranging from 11.03 to 17.44%), and an average of 14.92% for the moving-background sequence involving the same tow truck (ranging from 11.98 to 18.00%).

In Fig. 10, the mean distance of the control points on the active contour with respect to the ground truth contour is shown. It shows that the control points remain close to the ground-truth contour, with a slight bias towards being outside- this can be attributed to (1) the inclusion of textureless background regions with the estimated object boundary (as discussed above) and (2) the finite size of the kernels used to estimate image spatial gradients, leading to a blurring of edge information which subsequently acts on the active contour via the data-based energy terms.

### 4.5. Incorporation of condensation tracker

In order to show how our proposed region-boundary paradigm can be related to other tracking techniques,

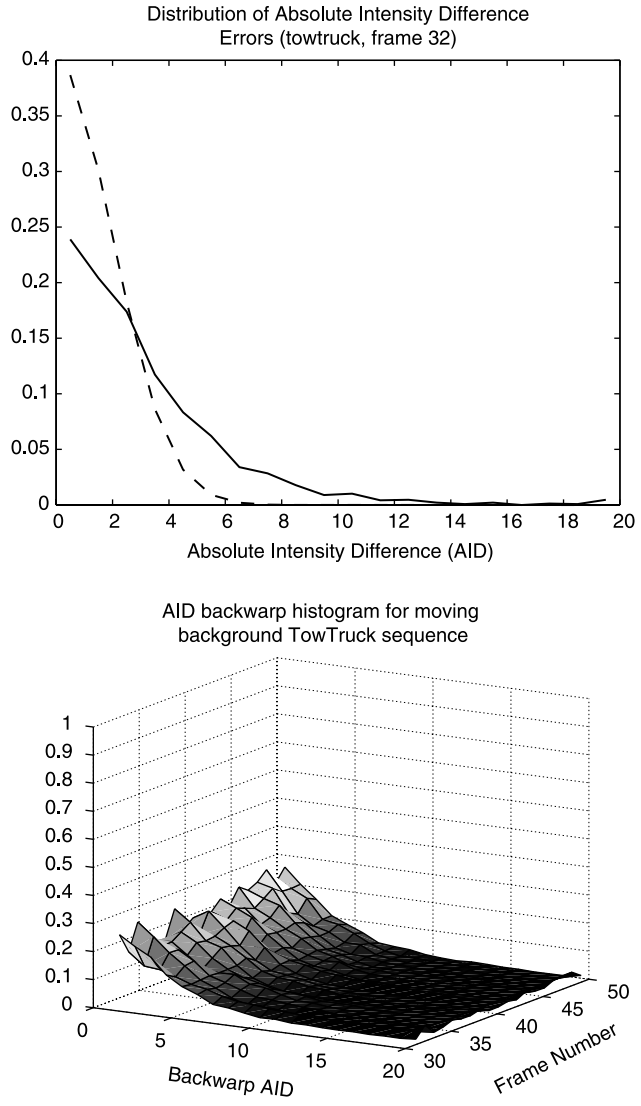



Fig. 8. One measure of the quality of recovered motion parameter is to compare the intensity value of pixels in an image with the intensity values of a second image warped according to the motion parameters that relate the two images. The comparison is performed for pixels within the tracked object contour. Top: the normalized histogram of absolute intensity differences is given by the solid line. All intensity differences greater than 20 have been lumped into the final bin-it can be seen that very few pixels fall into this category, suggesting most strong edges have been well matched. For comparison, a Gaussian density ($\sigma = 2$) for absolute differences is also shown (dotted line), modelling expected intensity differences due to CCD sensor noise alone. Bottom: this graph shows the result of performing this operation over an entire tracked sequence. The difference distribution can be seen to be consistent over the tracked frames.

a prototype version of the system was developed based on the Condensation algorithm by [38]. The basic idea is to use the motion update our technique provides to the active contour instead of the deterministic component of the update step in the Condensation algorithm, thus removing the need for a learning stage to train the dynamical model.

Our Condensation implementation follows its description in [38], using the six parameters of affine motion as the system state variables. The state variables thus represent the affine motion of the object boundary between the previous

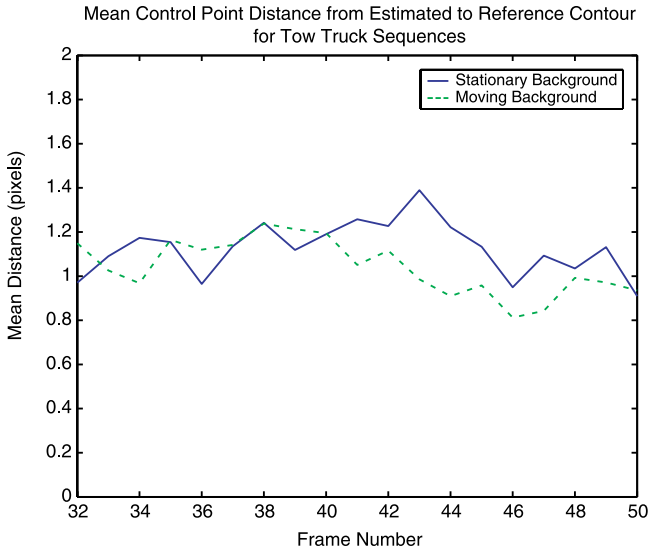



Fig. 9. The mean distance of the active contour control points from the ground-truth contour is shown for two tow-truck sequences, one with a moving background and one without. The somewhat better performance on the moving background sequence is likely due to larger relative velocities, leading to stronger forces acting on the active contour. The slight bias can be attributed to the finite receptive field of the gradient estimation algorithms.

frame and the current frame, for which an estimate is already provided from our flow-based motion estimation component. As a result, the region information used to derive motion estimates is directly incorporated in the contour-based tracker at this stage, replacing the typical sample propagation model for system state variable sample $\vec{r}^{(n)}$,

$$\vec{r}_t^{(n)} = A\vec{r}'^{(n)}_{t-1} + B\vec{w}_t^{(n)},$$

with

$$\vec{r}_t^{(n)} = \vec{\Theta}_t^{(n)} + B\vec{w}_t^{(n)}$$

where $\vec{w}$ is a vector of normal random variates and $B$ is a diagonal matrix of standard deviation values that control the stochastic diffusion of state samples. Here $\vec{\Theta}_t^{(n)}$ is the current motion layer parameter estimate. For the affine state vector, the leading four diagonal entries of B are identically assigned to account for small affine deformations, in the neighbourhood of 0.1, while the remaining entries corresponding to the translation parameters are considerably higher, in the range of two or more pixels. The observation measurement and weighting follows the approach described in [38] for 2D curve matching, by seeking edge features that lie on lines normal to the boundary curve. In our case, the normal lines are not explicitly set at uniform intervals of the curve, but at each active contour control point, the set of which are roughly evenly spaced along its length. The observation density is therefore, computed as

$$P\left(\vec{z}|\{\vec{x}_i\}_{i=1}^s\right) \propto \exp\left\{-\sum_{i=1}^{s}\frac{1}{2s\sigma_z^2}f\left(|\vec{z}-\vec{x}_i|,\mu\right)\right\}$$

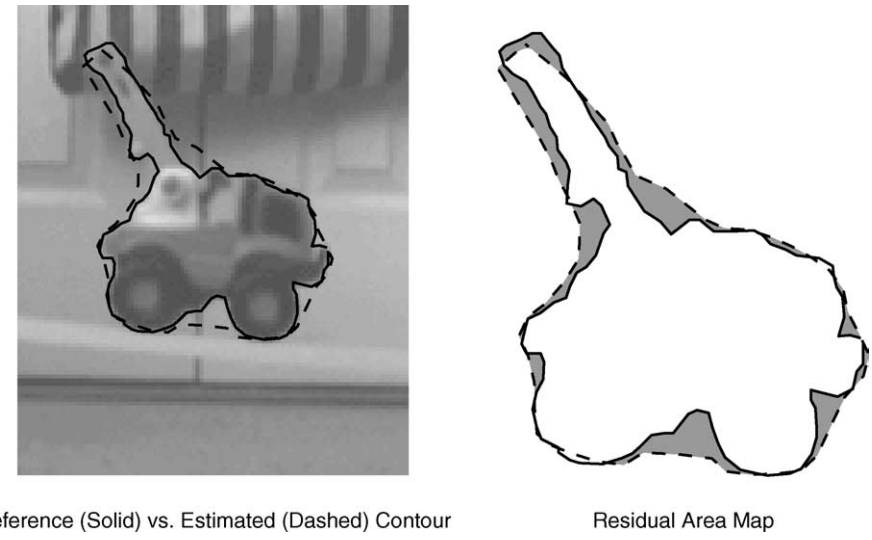



Fig. 10. The images here show a comparison of the automatically recovered boundary (dashed line) with a hand-segmented boundary (solid line). The left image shows both contours overlaid on the image, while the right shows the area between the two contours in grey, with the contours themselves in black. Note that the largest region of discrepancy is between the wheels where there is no discernable texture, and contextual information-available to the person performing the manual segmentation but not the algorithm-is required to correctly segment in this area.

where $f(\nu,\mu)=\min(\nu^2,\mu^2)$, and $\vec{z}$ is the location of the Canny-derived edge point closest to control point $\vec{x}_i$ along the normal to the curve at $\vec{x}_i$. The maximum feature distance parameter, $\mu \approx 7$ pixels, is used to constrain the feature search along the normal lines to a reasonable number of standard deviations. The equivalent 1D observation density along each normal has standard deviation $\sigma_z$, usually set to a value less than one to ensure close curve-to-edge alignment. With these changes in place, the standard Condensation algorithm ([38]) is executed to recover the object boundary, replacing the functionality of the active contour component. The resulting performance is found to be comparable to those results already given in this section.

## 5. Conclusions

The task of motion segmentation has long been approached from two separate sides, one in which motion layers are estimated without regard to the bounding contour of the moving object, and the other in which the bounding contour is tracked without regard to motion information inside the boundaries. In this paper, we have sought to reconcile these two approaches by making an explicit attempt to exploit spatial coherence in the motion of the image pixels. These two sources of information are complementary; for example, boundary tracking can deal with pixels that often violate constancy constraints used in motion layer estimation. Their combination is intuitively satisfying, and yields both improved motion estimation and boundary recovery. Our proposed framework allows for a moving background or observer, and multiple moving objects. We have introduced a novel mechanism that we call 'intensity constraints' to provide driving forces for the active contour. Finally, the use of feed-forward and feed-back mechanisms to link the motion layer and boundary estimation steps leads to a synergistic segmentation and tracking algorithm. It should be noted that the approach is general, in that it is not tied to a specific set of motion constraints (for example, phase- or colour-based constraints can also be used, or even non-gradient based methods for motion estimation).

We are continuing to develop the proposed method, and work is currently progressing in the following areas. First, topologically adaptive active contours are being investigated that will allow for recovery of non-simply connected object motions. We are also studying 'stiffening constraints' at contour control points to identify those parts of the bounding contour that are stable, and those that are articulating. In the case of articulating objects that are composed of rigid parts, this approach will allow for decomposition of objects based on these parts, as well as facilitate the tracking of objects under occlusion. It is also desirable to investigate the incorporation of non-parametric flow estimation for two reasons. The first is to give the method a range of estimation capabilities beyond constant, affine or other parametric motions. The second reason is to allow tracking of freely deformable objects. The challenge in incorporating non-parametric flow estimation lies in assigning ownership probabilities to motion constraints used in the layered motion model.

## Acknowledgements

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

## References

[1] A.L. Yuille, N.M. Grzywacz, The motion coherence theory, Proceedings of the Second International Conference on Computer Vision, Tampa, FL, 1988, pp. 344–353.

[2] N.M. Grzywacz, J.A. Smith, A.L. Yuille, The motion coherence theory, Proceedings of the IEEE Workshop on Visual Motion, Irvine, CA, 1989, pp. 148–155.
[3] A.L. Yuille, N.M. Grzywacz, A mathematical analysis of the motion coherence theory, International Journal of Computer Vision 3 (1989) 155–175.
[4] Y. Weiss, E.H. Adelson, A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1996, pp. 321–326.
[5] Y. Weiss, Smoothness in layers: Motion segmentation using nonparametric mixture estimation. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 520–527.
[6] T. Darell, A. Pentland, Robust estimation of a multi-layered motion representation, Proceedings of the IEEE Workshop on Visual Motion, Princeton, New Jersey, 1991, pp. 173–177.
[7] J.Y.A. Wang, E.H. Adelson, Layered representation for motion analysis. Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, IEEE Computer Society Press, Los Alamitos, California, 1993, pp. 361–366.
[8] A.D. Jepson, D.J. Fleet, M.J. Black, A layered motion representation with occlusion and compact spatial support European Conference on Computer Vision. ECCV, Springer, pp. 692–706 .
[9] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, Robust On-Line Appearance Models For Visual Tracking. IEEE Conference on Computer Vision and Pattern Recognition, Kauai, 2001, pp. 415–422 .
[10] M. Gelgon, P. Bouthemy, A region-level motion-based graph representation and labeling for tracking a spatial image partition, Pattern Recognition 33 (2000) 725–740.
[11] M. Isard, A. Blake, ContourTracking By Stochastic Propagation Of Conditional Density in: ECCV96 vol. II Springer. pp. 343–356.
[12] M.J. Black, D.J. Fleet, Probabilistic detection and tracking of motion boundaries, International Journal of Computer Vision 38 (3) (2000) 231–245.
[13] N. Paragios, R. Deriche, Geodesic active regions for motion estimation and tracking, Proceedings of the seventh International Conference on Computer Vision, IEEE Computer Society Press, Los Alamitos, CA, 1999, pp. 688–694.
[14] N. Paragios, R. Deriche, Unifying Boundary And Region-Based Information For Geodesic Active Tracking IEEE Computer Vision and Pattern Recognition, 2, IEEE Computer Society Press, 1999, pp. 300–305.
[15] N. Paragios, 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) (2000) 266–280.
[16] B. Bascle, R. Deriche, June Region tracking through image sequences Proceedings of the Fifth International Conference on Computer Vision. IEEE Computer Society Press, Boston, MA, 1995. pp. 302–307.
[17] A. Yilmaz, X. Li, M. Shah, November Contour-based object tracking with occlusion handling in video acquired using mobile cameras, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1531–1536.
[18] A. Yilmaz, X. Li, M. Shah, January Object contour tracking using level sets, Proceedings of the Asian Conference on Computer Vision, Korea, 2004.
[19] A.D. Jepson, M.J. Black, Mixture models for optical flow computation. Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, IEEE Computer Society Press, Los Alamitos, California, 1993, pp. 760–761.
[20] S. Ayer, H.S. Sawhney, Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding, Proceedings of the Fifth International Conference on Computer Vision, IEEE Computer Society, IEEE Computer Society Press, 1995, pp. 777–784.
[21] M.A. Fischler, R.C. Bolles, Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography, Communications ACM 24 (6) (1981) 381–395.
[22] Torr P.H.S., 1994 Outlier detection and motion segmentation. PhD thesis, University of Oxford.
[23] P.H.S. Torr, D.W. Murray, Stochastic motion clustering, in: J.-O. Eklundh (Ed.), Lecture Notes in Computer Science, Computer Vision-ECCV'94 801, Springer, Berlin, 1994, pp. 328–337.
[24] H.S. Sawhney, Y. Guo, R. Kumar, Independent motion detection in 3d scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (10) (2002) 1191–1199.
[25] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society B 39 (1977) 1–38.
[26] M. Irani, B. Rousso, S. Peleg, Detecting and tracking multiple moving objects using temporal integration ECCV92, Springer, 1992, pp. 282–287.
[27] Black M.J., Anandan P., 1993 The robust estimation of multiple motions: Affine and piecewise-smooth flow fields. Technical Report, Xerox
[28] W.J. MacLean, A.D. Jepson, R.C. Frecker, Recovery of egomotion and segmentation of independent object motion using the EM algorithm, Proceedings of the 1994 British Machine Vision Conference, British Machine Vision Association, York, England, 1994, pp. 175–274.
[29] J. Bergen, P. Anandan, K. Hanna, R. Hingorani, Hierarchical model-based motion estimation, Proceedings of the European Conference on Computer Vision, Santa Margherita, Italy, 1992, pp. 237–252.
[30] A.K. Jain, Fundamentals of Digital Image Processing, Prentice–Hall, 1988.
[31] R. Zhang, N. Paragios, D. Metaxas, Implicit representations towards recovery and reconstruction of motion layers, IEEE Workshop on Geometric, Variational and Level Set Methods in Computer Vision, Nice, 2003.
[32] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, International Journal of Computer Vision 1 (2) (1987) 321–331.
[33] L. Cohen, I. Cohen, November 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) (1993).
[34] L.D. Cohen, On active contour models and balloons Computer Vision, Graphics, and Image Processing. 53(2) (1991) 211–8.
[35] C. Xu, J.L. Prince, March Snakes, shapes, and gradient vector flow IEEE Transactions on Image Processing, 1998 359–96.
[36] L.G. Shapiro, G.C. Stockman, Computer Vision, Prentice–Hall, New Jersey, 2001.
[37] N. Jojic, B.J. Frey, Learning flexible sprites in video layers, Kuaii, Hawaii IEEE Conference on Computer Vision & Pattern Recognition, 1, 2001, pp. 199–206.
[38] M. Isard, A. Blake, Condensation — conditional density propagation for visual tracking, International Journal of Computer Vision. 29(1) (1998) 5–28.