Motion Segmentation Incorporating Active Contours for Spatial Coherence

by

Desmond Ryan Chung Lin Cheung

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Sciences
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

Motion Segmentation Incorporating Active Contours for Spatial Coherence

Desmond Ryan Chung Lin Cheung

Master of Applied Sciences

Graduate Department of Electrical and Computer Engineering

University of Toronto

2004

This thesis describes a computer vision algorithm that detects and segments independently moving objects in a video sequence, recovering their shape over time. While traditional motion segmentation approaches employ learned or low-dimensional parametric models to represent object shape, we propose a hybrid framework that combines robust motion segmentation with active-contour-based boundary recovery techniques, to overcome each individual approach's limitations. Our framework proposes feeding forward motion segmentation results to initialize, constrain and propagate the active contour, while feeding back active-contour-based object boundary estimates to the motion segmentation process to provide spatial coherence. We develop a functional system based on this framework, introducing a novel motion-based intensity constraint, and an active contour formulation that incorporates motion segmentation results. Our results demonstrate the successful segmentation of sequences that include multiple moving objects and sequences with a moving background.

Acknowledgements

I begin by thanking Professor James MacLean, whose consistent instruction and support has motivated and guided this research. I also owe my sincere gratitude for the guidance and alacrity of Professor Sven Dickinson, inspiring my enthusiasm and assisting the development of the shape recovery aspects of this work.

Chakra Chennubhotla and Professor Allan Jepson provided invaluable discussion and suggestions, especially for the introduction of connected components techniques. I must also thank Hilda Faraji, whose blackboard discussions and friendship sustained my focus and spirits throughout this project.

Finally, I would like to thank my family: my parents and brothers, who embrace and enable every facet of my life, and Mitzie, who elevates it.

This work was made possible, in part, by the generous financial support of CITO, The University of Toronto, NSERC and MD Robotics.

Contents

Part I

Introduction

Chapter 1

Background

The process of distinguishing moving objects from their background and from each other in a video sequence is known as 'motion segmentation'. The ease with which motion allows humans to distinguish objects and their activities has motivated significant interest in motion segmentation computer vision community. This research aims to advance the exploitation of visual motion in addressing the larger challenge of computer-based visual interpretation.

This research develops a vision system that recovers the motion and shape properties of independently moving objects within a scene without manual initialization or prior knowledge of object shape or appearance. The system provides an object detection, segmentation and tracking framework that is able to deal with multiple independently moving objects and a moving background or moving camera. The results of this system identify the position and boundaries of any independently moving objects in the scene, continuously maintaining and updating this information to facilitate the application of more object-centric tasks, such as recognition.

The segmentation system aims to recover the shape and motion of multiple independently moving objects (*IMO*s) in a monocular video sequence. The system does not rely on the depth estimation-based segmentation techniques that might be available to a stereo or depth-based vision system. This system recovers scene segmentation for situations where such systems

are not available, such as for surveillance cameras already in place, conventional (monocular) video footage, and in low-power or low-bandwidth configurations where other systems are impractical. In addition, the segmentation system might be applied to footage captured by low-resolution imaging systems, or situations where contrast (and hence texture detail) may be poor, hindering the performance of alternative tracking systems. By combining motion segmentation and a class of boundary recovery techniques known as 'active contours', we design a system that might be suitable for these applications. We use the *MASC* acronym to refer to this system which performs *M*otion segmentation incorporating *A*ctive contours for *S*patial *C*oherence.

This introductory section presents a summary of the concepts behind imaging and the detection of visual motion in images, from the basic pinhole camera model to the theory of optical flow, and finally, the key obstacle to obtaining reliable motion segmentation of a video sequence: the aperture problem. These fundamental topics provide a background upon which the computer vision concepts developed and implemented in this research are based.

1.1 Image Formation: The Pinhole Camera

Image formation refers to the process by which a real-world scene is reproduced on a two-dimensional surface, as shown simplistically in Figure 1.1. A typical camera, for example, is aimed at a real-world scene and reproduces that scene on the light-sensitive negative, recording that scene on the negative so that it can be later reproduced in print form as a photograph. The camera's negative records the light reflected from each point within the target scene, so that each point in the scene has a corresponding 'image' in the negative. This description of image formation indicates the need for a one-to-one correspondence between scene points and image points, so that each image point reproduces only the light emitted by its corresponding scene point. If this were not the case, and an image point reproduced the light emitted by multiple scene points, the complete image would appear blurred, or 'out of focus'.

Perhaps the simplest technique of formation is the pinhole camera, illustrated in Figure 1.2.

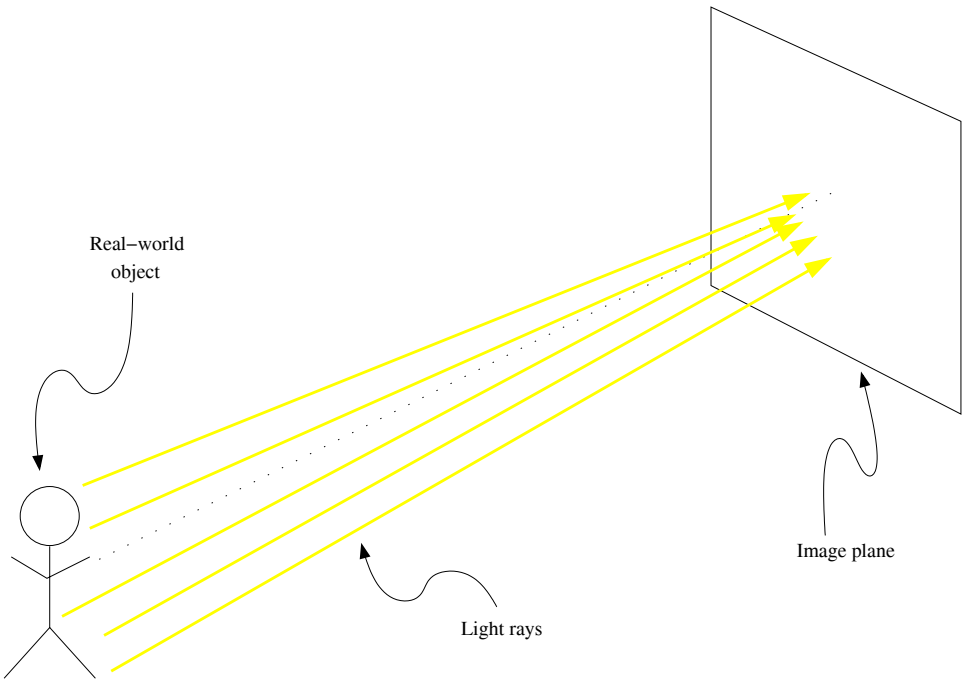


Figure 1.1: Imaging a real-world object on an image plane

The pinhole camera comprises an imaging surface, and a pinhole-sized aperture, the opening through which light is allowed to pass through to the imaging surface. The size of the aperture only permits the light coming from a single point in the real-world scene to reach its corresponding image point, forming a sharp, focused image. The drawback to using the pinhole camera is that the small aperture only allows a correspondingly small amount of light to reach the imaging surface in a given amount of time. Such a camera would require the negative surface to be exposed to the scene for a few seconds at the very least in order to correctly register the scene, and any camera motion or change in the scene during that period would degrade the quality of the image. Optical systems that incorporate lenses in order to allow larger apertures to be used, while maintaining the one-to-one scene to image point relationship, reduce this exposure time to fractions of a second, but the imaging principles remain.

The geometry of the pinhole camera illustrated in Figure 1.3 indicates the following image point to scene point relationship, taking the aperture as the coordinate center, also known as

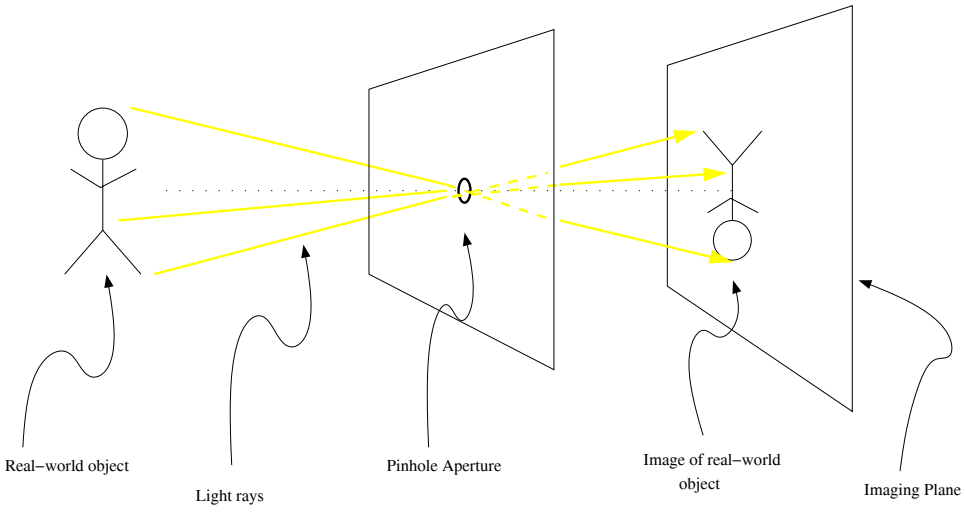


Figure 1.2: The pinhole camera model

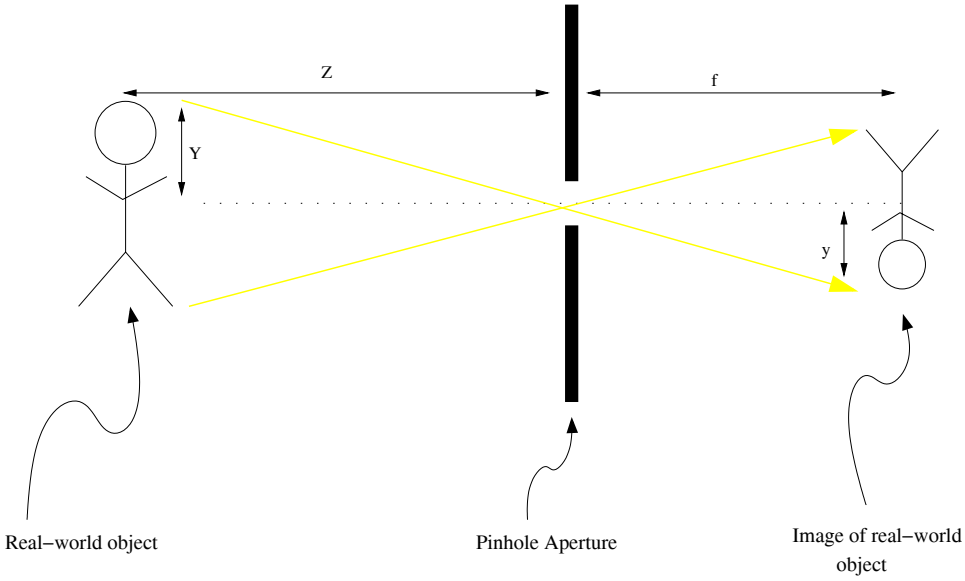


Figure 1.3: Pinhole camera imaging geometry

the optical center in imaging terms:

$$x = f\frac{X}{Z} \tag{1.1}$$

$$y = f\frac{Y}{Z} \tag{1.2}$$

Equations 1.1 and 1.2 formalize the relationship between the location of a point in the scene, and its image on the imaging surface. This simple pinhole camera model demonstrates the imaging process, and provides a clear relationship between scene and image points, providing a view to how motion in the real world generates related visual motion in a video sequence.

1.2 Motion Field and Optical Flow

The idea of estimating the motion of objects in a video sequence can now be interpreted in two senses, the first of which is the recovery of three dimensional motion of a real-world three dimensional object, projected onto the two dimensional imaging plane. Alternatively, we may estimate the two dimensional motion of the image of that object. The former would require an estimate of the three dimensional structure of the object, and we instead focus on two dimensional motion estimation. The exact nature of this projected motion is represented in the 'motion field' of a video sequence, which describes the motion of each image point, reflecting the (projected) motion of its corresponding scene point [50].

While the motion field is an ideal representation of visual motion, numerous examples may be presented to prove how the true motion field of a sequence is impossible to recover from the sequence alone. For example, a rotating sphere with a mirrored surface presents no evidence of its motion field. The intensity values on the surface of the sphere correspond to those of the stationary environment around the sphere. So even as the sphere rotates, any given image point on its surface appears to remain still in the video sequence. This introduces the correspondence problem in motion estimation: in order to estimate the motion of an object between two successive frames of a video sequence, we must know the precise position of the object in both frames. This information however, is the object segmentation of the sequence,

which is what we are trying to recover through motion estimation. This problem manifests itself as the Aperture Problem, described in greater detail in Section 1.3.

The notion of recovering the true motion field is thus considerably complicated by certain scene configurations. While the motion field is generated by the projection of moving scene points, in the opposite sense, the movement of image points allow for the approximate recovery of the motion field of the scene. That approximation is known as the optical flow field of a sequence. The optical flow field represents the movement in a video sequence based solely upon the evidence provided by the sequence. The optical flow field is most commonly estimated by making an assumption claiming that the intensity of a scene point's image remains constant while the scene point moves. The 'Brightness Constancy Constraint' [24] or BCC, therefore asserts that the appearance of a moving object in a video sequence does not change over short intervals, so that the optical flow field may be recovered by measuring the movement of constant intensity regions in the video sequence.

The BCC is formally expressed in Equation 1.3, by enforcing the claim that the brightness of any particular scene point, $I(x(t), y(t))$, in the image remains constant over small periods of time, where $(x(t), y(t))$ is a scene point coordinate whose position varies with the time parameter t.

$$\frac{dI(x,y,t)}{dt} = 0 \tag{1.3}$$

The chain rule for differentiation expresses this relation with respect to spatial and temporal (spatiotemporal) partial derivatives of the video sequence:

$$\frac{\partial I(x,y,t)}{\partial x}\frac{dx}{dt} + \frac{\partial I(x,y,t)}{\partial y}\frac{dy}{dt} + \frac{\partial I(x,y,t)}{\partial t} = 0 \tag{1.4}$$

Using the notational simplifications representing point velocity in Equation 1.5 and spatiotemporal derivatives in Equation 1.6:

$$v_x = \frac{dx}{dt} \qquad v_y = \frac{dy}{dt} \tag{1.5}$$

$$I_x = \frac{\partial I}{\partial x} \qquad I_y = \frac{\partial I}{\partial y} \qquad I_t = \frac{\partial I}{\partial t} \tag{1.6}$$

The BCC may be presented for a given image point as a sum of scalar products

$$I_x v_x + I_y v_y + I_t \;\; = \;\; 0 \tag{1.7}$$

or in vector form:

$$\vec{\nabla} I_{\vec{x}}^{\,T} \vec{v} + I_t = 0 \quad , \quad \vec{\nabla} I_{\vec{x}} = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \tag{1.8}$$

And also as:

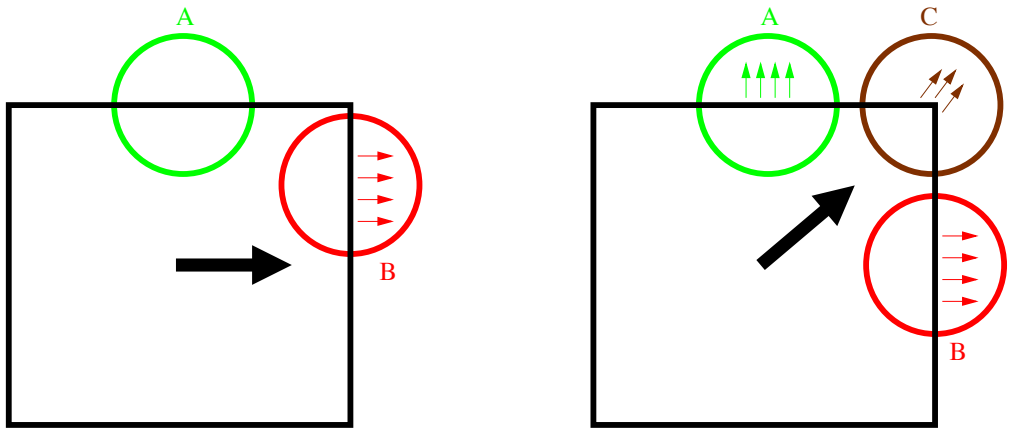
$$\vec{\nabla} I^T \begin{bmatrix} \vec{v} \\ 1 \end{bmatrix} = 0 \quad \text{where} \quad \vec{\nabla} I = \begin{bmatrix} I_x \\ I_y \\ I_t \end{bmatrix} \quad \text{and} \quad \vec{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{1.9}$$

As shown above, the BCC provides a means to estimating the motion of image points given the spatiotemporal gradients (I_x, I_y and I_t) of the video sequence. The primary obstacle to applying these equations is the Aperture Problem, described in the next section.

1.3 Aperture Problem

The 'Aperture Problem' of motion estimation [24] is a key problem posed by motion estimation, and provides strong motivation for the techniques developed in this research. The aperture problem relates the solution of the BCC expression in Equation 1.7 to the fact that at any given image point, only the component velocity perpendicular to the brightness gradient at that point can be estimated. This problem is illustrated in Figure 1.4(a), which illustrates how horizontal translation of the square cannot be identified at its horizontal edges in Region *A*, as the motion is imperceptible at these points. On the other hand, the motion is clearly identified around the vertical edges in Region *B*.

Figure 1.4(b) also shows that the component velocity at any single point does not necessarily describe the complete motion of an object. In this figure, the diagonal motion of the square produces distinct component velocities in regions *A*, *B* and *C*. The horizontal edge in region



(a) (b)

Figure 1.4: The aperture problem for a single object

A only provides evidence of the vertical component of the square's velocity, and similarly, the vertical edge in region *B* only provides evidence of the horizontal component of velocity. In contrast, the corner points in region *C* have brightness gradients perpendicular to the velocity of the square, and thus both horizontal and vertical components of the velocity are evident.

In order to overcome the aperture problem, it is necessary to estimate object motion using a large group of the component velocities present in a sequence. For example, by clustering the complete set of component velocities from the horizontal and vertical edges of the square in Figure 1.4(b), its diagonal translation is completely described. Component velocities are therefore referred to as motion constraints, as instead of providing a complete motion estimate, they simply provide a single constraint for that motion, while at least two constraints are required to estimate the translational velocity of a point.

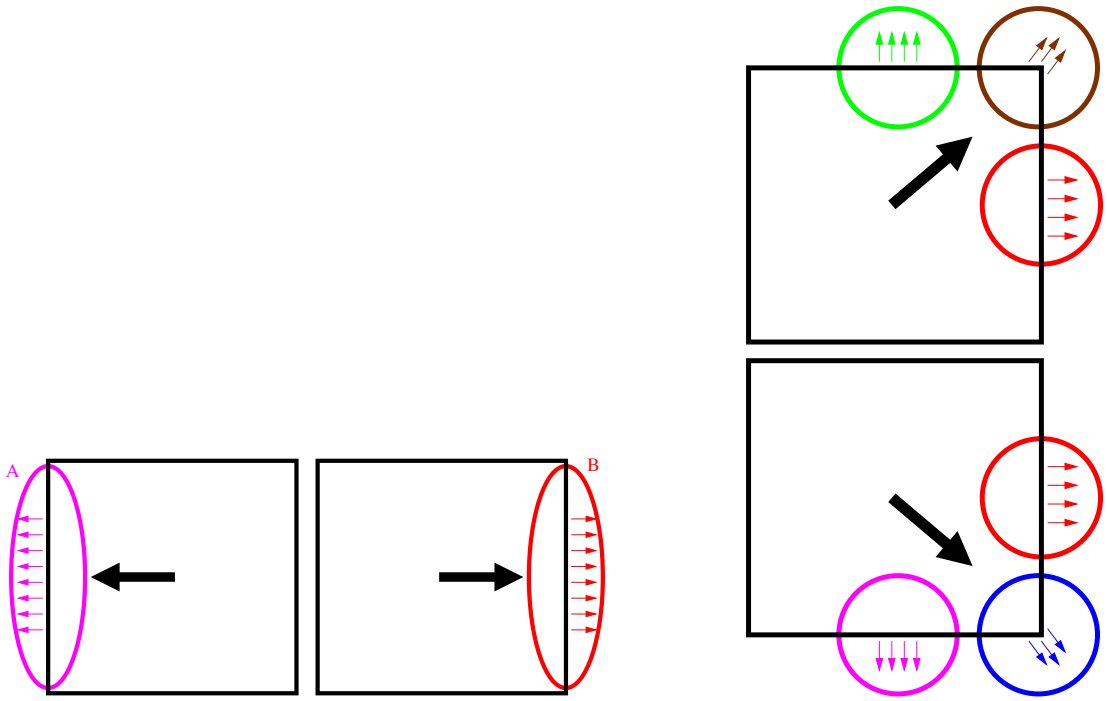
The key to correctly estimating motion then, is firstly recovering motion constraints accurately, and then intelligently grouping or clustering motion constraints from coherent objects. It is the latter of these two steps that addresses the aperture problem and presents obstacles that current computer vision research still does not address completely, the problem of how motion constraints are optimally clustered for motion estimation. In particular, the 'Generalized Aperture Problem' [29] exemplifies the clustering problem, by stating that while gathering motion

constraints from a larger region should improve the quality of motion estimates, doing so increases the possibility that motion constraints generated by other objects are included in our calculations, contaminating the motion estimate.

The classical optical flow estimation approach proposed by Horn and Schunck [24] makes the assumption that the optical flow field should vary smoothly, and simply integrates neighboring motion constraints to generate a smooth and continuous optical flow field. This approach assumes that neighboring points are part of the same object, making no provisions for discontinuities in the scene, such as the areas at object boundaries. By integrating motion constraints from multiple objects at object boundaries, the optical flow estimates in and around these areas correspond to neither object's correct optical flow field, resulting in a contaminated optical flow map.

An alternative approach introduced by Jepson and Black [29] seeks clusters of coherent motion constraints, as motion constraints generated by a single rigid moving object should not contradict each other. Both cases shown in Figure 1.4 show a single object, whose motion is easily estimated, as none of the constraints generated in either case are contradictory, so they are all *coherent*. Figure 1.5(a) shows an example where two objects move in different directions, generating two sets of constraints. The constraints generated by the left square contradict those generated by the right square, as they are in opposing directions. As a result, the two sets can be easily distinguished. However, this approach does not fully address the aperture problem as motion constraints generated by distinct moving objects may be coherent, as shown by Figure 1.5(b). This example shows how a portion of the constraints from multiple objects may be coherent, in this case, the horizontal constraints generated by both squares. As a result, the motion may be poorly estimated unless some additional criteria are imposed upon the constraint clustering process.

The two approaches suggest a possible solution to the aperture problem, using an 'intelligent window' to cluster motion constraints, as suggested by Figure 1.6. The ideal window would have the same shape as the boundary of the target object, so that the cluster would in-



(a) (b)

Figure 1.5: The aperture problem for multiple objects

clude all of the object's motion constraints, and exclude all other motion constraints present in the frame of a video sequence. Any other objects would then require their own particular integration windows with matching shape requirements, allowing the motion of all objects in the sequence to be recovered accurately.

Estimating an ideal window inherently requires that the shape of objects be known or recovered. As this research assumes no prior information about the objects in the scene is given, object shape must be recovered automatically by the system in order to recover the ideal window. The next section explores the clustering of coherent motion constraints using the approach suggested by Jepson and Black [29], followed by a description of a boundary (shape) recovery technique, two aspects that this research proposes to combine, resulting in a unified motion segmentation and shape recovery framework.

1.4 Motion Constraint Clustering

Motion constraint clustering is a critical part of the optical flow estimation process. It can be likened to deriving a 'layered' model of a video sequence [51], wherein every independently moving region in the sequence is represented as a separate layer. Objects may be composed of multiple moving regions, for example, a person may move one arm, keeping the rest of his body stable. We generalize the terminology here by using the term Independently Moving Object (*IMO*) to refer to any independently moving region whose motion can be approximated by a parametric motion model, discussed in Section 1.5. Multiple layers can then be superimposed upon each other in the correct order to reconstruct the original video sequence.

Our approach to recovering the *IMO*s in a video sequence is by using mixture model techniques. Figure 1.7 illustrates a simple one dimensional mixture model, a plot of two independent signals (the broken lines), which sum together to form a new 'mixture' (the solid line). In our context, mixture models apply specifically to compositions of probability density functions, so in our one dimensional example, each independent component is a density function

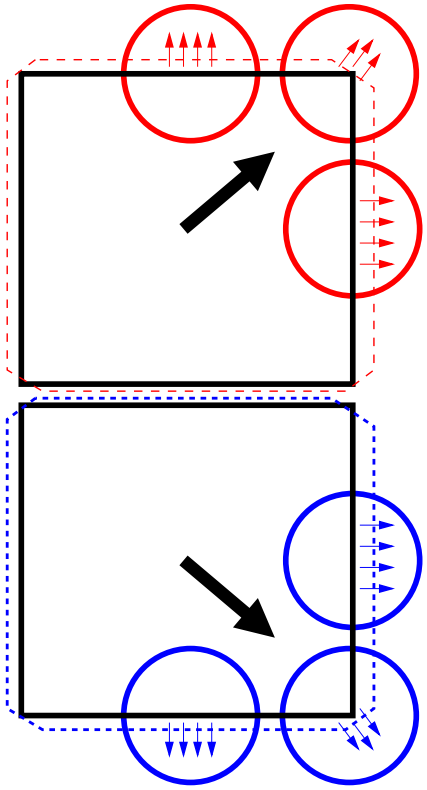


Figure 1.6: Intelligent motion constraint clustering groups motion constraints that are generated by distinct independently moving objects by using the clustering window, shown in the dotted lines, to select the appropriate motion constraints.

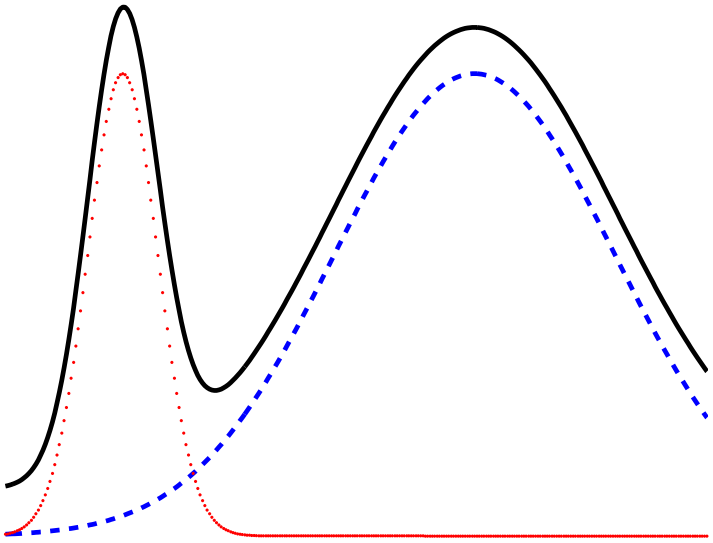


Figure 1.7: Mixture Model: The dotted line and dashed line represent the two independent signals that generate the mixture, observed as the solid line.

of x with component parameters $\vec{\theta}_j$. The mixture is also a density function that is composed of each component density weighted by a 'mixture proportion' to normalize the mixture density.

If the form of the original signals is known (for example if their form is known to be Gaussian, or close to Gaussian), then the original signals may be recovered from only the mixture signal using a technique known as the Expectation-Maximization (EM) algorithm [19]. The EM algorithm assumes that the mixture is made up of samples from a known number of independent Gaussian random variables. In addition, the algorithm assumes that a set of initial estimates of each independent signal's parameters is available. The algorithm iteratively refines the initial parameter estimates for the independent signals to optimize an objective function defined by the 'closeness of fit' of the estimated parameters to the actual observed mixture signal. As the EM algorithm only guarantees convergence to a local maximum in the objective function, it does not always recover the globally optimal parameters of the underlying independent signals. Given a reasonable set of initial guesses however, locally optimal estimates are typically very close to the globally optimum values.

As its name suggests, the EM algorithm is made up of two parts, the 'Expectation step'

and the 'Maximization step'. The expectation step compares current parameter estimates to the observed data set, measuring the 'fit' of each data point to the parameters of the signals that may have generated it. The maximization step uses these measurements to refine the parameter estimates using the data points that fit the former estimates best. For a simple mixture of Gaussians made up of N components, the initial parameter set estimates might comprise mean and standard deviation values, μ_n and σ_n, for each component, with each component having a mixture probability m_n. The mixture probability represents the prior probability of any sample coming from a given component.

The collection of parameters can also be represented in vector form for notational convenience:

$$\vec{m} = \begin{bmatrix} m_1 \\ \vdots \\ m_n \end{bmatrix} \quad \vec{\mu} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix} \quad \vec{\sigma} = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_n \end{bmatrix}$$

Now, given an initial estimate for the parameter set, the probability of an observation x_k can be calculated as

$$p(x_k|\vec{\mu},\vec{\sigma},\vec{m}) = \sum_{n=1}^{N} m_n p_n(x_k|\mu_n,\sigma_n) \,, \tag{1.10}$$

where for a mixture of Gaussians

$$p_n(x_k|\mu_n,\sigma_n) = \frac{1}{\sqrt{2\pi}\sigma_n} exp\left(-\frac{(x_k-\mu_n)^2}{2\sigma_n^2}\right) \tag{1.11}$$

Assuming the independence of samples, the probability of observation of the entire set of results can then be represented in log form:

$$\log P = \sum_{k=1}^{K} \log p(x_k|\vec{m},\vec{\mu},\vec{\sigma}) \tag{1.12}$$

This overall log data probability defines an objective function that gives the likelihood that the estimated parameter set $\{\vec{m},\vec{\mu},\vec{\sigma}\}$ generates the complete set of observations $\{x_k : k = 1,...,K\}$. By optimizing the objective function with respect to the component parameter set,

we are able to recover the locally optimal values of the component parameter set [39]. In order to do so however, the probability of a given data point, x_k, being generated by the n^{th} component must be related to determine the influence of each data point on each component optimization. Through Bayes' rule, we can calculate these 'ownership probabilities', which completes the E-step of the algorithm:

$$q_{nk} = \frac{m_n p_n(x_k|\mu_n, \sigma_n)}{\sum_{j=0}^{N} m_j p_j(x_k|\mu_j, \sigma_j)} \tag{1.13}$$

The maximization, or optimization, step comprises the recalculation of mixture proportions ensuring that the sum of ownerships for a point comes to one, while the sum of mixture proportions also comes to one. Furthermore, the parameter set $\{\vec{m}, \vec{\mu}, \vec{\sigma}\}$ must be recovered to satisfy the local optimum point of Equation 1.12. The partial derivative of the log data likelihood, $\log P$, with respect to the parameters $\{\mu_n, \sigma_n\}$, when set to zero defines the local extreme point of the log data likelihood function. Solving for the parameter set optimizes the objective function, completing the Maximization step:

$$\sum_{k=1}^{K} q_{nk} \frac{\partial}{\partial(\mu_n, \sigma_n)} \log p_n(x_k|\mu_n, \sigma_n) = 0 \tag{1.14}$$

By iterating between the two, the parameter set is continuously refined and the ownerships updated, so that they converge to locally optimal values.

Constraint clustering on the basis of motion constraint coherence takes the first step toward collecting the complete set of motion constraints generated by an object's motion. The simplest example of this is provided by two moving objects in a video sequence. Two objects undergoing distinct motion will each generate a set of motion constraints, and usually many constraints generated by one object will be incoherent with those from the other object. For example, the square of Figure 1.5(a) translating to the right side of the frame generates rightward motion constraints at its vertical edges, while the square translating to the left side of the frame generates leftward motion constraints at its vertical edges. The histogram of horizontal motion constraints generated by this scene would comprise two Gaussian forms (due to noise) with means representing the leftward motion and rightward motion of each square. The resulting

mixture model of horizontal motion constraints allows the two sets to be easily distinguished, as such a group of constraints could not be generated by a single rigid object.

The EM procedure is extended to recover the distinct underlying clusters of motion constraints in a sequence as suggested by Jepson and Black [29]. This formulation likens the video sequence to the observed mixture (solid black line) of Figure 1.7, and the *IMO*s within the sequence as the two underlying signals (dotted red and dashed blue lines) that compose the mixture. The details of the application of mixture models for motion estimation are covered in Chapter 4.

1.5 Motion Models

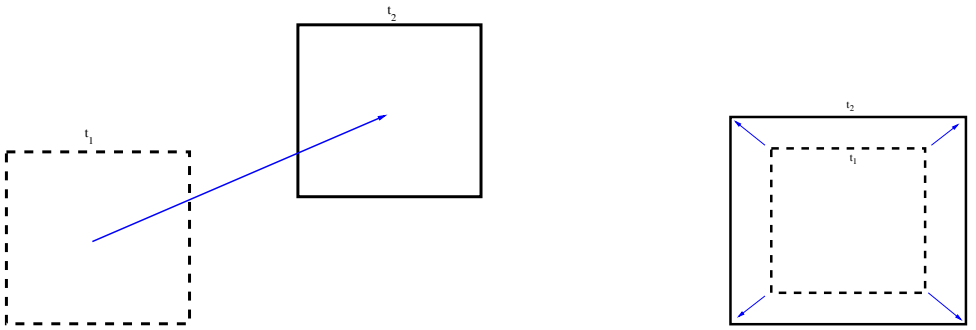
Motion constraint clustering is done with the assumption that a real object's motion, projected onto the imaging plane, may be represented by a plausible 'motion model'. The simplest motion model is the translational or constant motion model, in which the only plausible motion an object in a sequence can undergo is that of simple translation. Figure 1.8(a) demonstrates the simple translation of a square parallel to the image plane.

Equations 1.15 and 1.16 describe the change in coordinates of a pixel between consecutive frames of a video sequence, as it moves from $\vec{x}$ to $\vec{x}'$, undergoing translation $\vec{v}$. Note that the vector $\vec{v}$ represents a discrete displacement between consecutive frames, as opposed to a velocity at a time-instant.

$$\vec{x}' = \vec{x} + \vec{v} \tag{1.15}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{1.16}$$

Real world objects rarely restrict their range of movements to translations parallel to a plane however, and the scaling motion shown in Figure 1.8(b), demonstrates the result of the target object moving toward the camera (or vice versa). A constant motion model is clearly unable



(a) The real world square motion is a translation parallel to the image plane, represented in the video sequence as a simple translation.

(b) As the real world square moves directly toward the camera, its image scales in size, a change that cannot be represented by a simple translation.

Figure 1.8: Object Motion

to describe the motion of the square in this case. Instead, alternative higher order parametric motion models must be used to represent the motion of a group of pixels that belong to an object.

An affine motion model uses four additional parameters in addition to the two translational parameters used by the constant model. The additional parameters make up the matrix $\mathbf{A}$ in Equation 1.17, a 2×2 matrix that can be used to describe location dependent movement of pixels.

$$\vec{x}' = \mathbf{A}\vec{x} + \vec{b} \tag{1.17}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{1.18}$$

In addition to simple translation, the affine motion model is also able to represent the effects of scaling, shearing and rotation in the plane. Assuming the depth of objects is relatively small with respect to the distance of objects to the camera, the affine motion model provides

a suitable parametric model for representing and clustering object motions. In the example shown in Figure 1.8, the scaling of the square about the center of the frame can be represented by the affine parameter set

$$\mathbf{A} = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{1.19}$$

where $a > 1$, to represent the effect of the square becoming larger.

Higher order parametric motion models such as the projective motion model are able to overcome the orthographic projection assumption, at the expense of instability in their recovery and the need for a larger number of motion constraints to estimate them. In general, any function $\vec{v}(\vec{x}) = f(\vec{x}, \vec{\theta})$ can be used as long as the motion model parameter set, $\vec{\theta}$, can be estimated.

1.6 Motion Segmentation Principles

Motion segmentation classifies regions in a frame as belonging to distinct objects by using the information generated by the motion estimation process. The regions associated with a set of coherent motion constraints are used to estimate the position and boundaries of the *IMO* responsible for generating those constraints. We perform the clustering of coherent motion constraints under the additional assumption that objects are spatially coherent, that is, all the parts of a single *IMO* are usually encapsulated within a single connected region. It is implicit that parts of other *IMO*s are usually not present in that connected region, but we do not assume this to always be the case. There are common exceptions to this assumption, notably cases of transparency and occlusion. In general however, the assumption of spatial coherence allows us to intelligently cluster motion constraints that typically correspond to a single coherent object.

This type of region-based segmentation provides a powerful means of recovering an initial estimate of the motion and shape of any *IMO* in the scene, however the aperture problem reveals itself again, through the problem of classification of ambiguous edges, as shown in

Figure 1.5(b). In this example the two objects both generate identical motion constraints around their adjacent vertical edges, making it difficult to resolve that the two vertical edges are part of distinct objects. The incomplete segmentation of a sequence motivates the combination of motion and shape estimation techniques, since the recovery of accurate object motion estimates also ultimately requires accurate object boundary estimates, as suggested in Section1.3.

1.7 Boundary Recovery

Boundary recovery is introduced as a means of gradually capturing the shape of the object, using active contour techniques to generate a closed contour that lies on the object boundary. An active contour is made up of an ordered set of discrete points, over which we define an energy functional. The energy functional has two primary components: internal energy, whose value is a function of the position of contour points with respect to each other, and external energy, whose value is a function of the position of contour points with respect to image features. As a result, the shape of the active contour can be related in quantitative terms to characteristics such as smooth curvature through the internal energy term. Similarly, the adherence of an active contour to edges in the image can be related by the external energy term. A correctly defined energy functional term thus quantifies desirable qualities of the active contour. We can use this to adjust the position of an initial contour, as shown in Figure 1.9(a), into a contour with smooth curvature, that is also aligned with image edges, resulting in a curve as shown in Figure 1.9(b).

If applied correctly, the boundary recovery process can be used to recover an accurate boundary of a target *IMO*, which in turn, can be used to delineate the ideal constraint clustering boundary for motion estimation and segmentation, principles that are explored in detail in this thesis. We begin with a description of thesis organization in the next section.

(a) Initial Contour (b) Final Contour

Figure 1.9: Active contour execution adjusts the position of the initial contour to that of the final contour. Each discrete contour point is shown as a small circle.

1.8 Thesis Organization

The remainder of this thesis describes our design and implementation of a motion segmentation system, presenting experimental results and the conclusions we arrived at after its development.

We begin with a review of recent research in the area of motion segmentation and object tracking in Chapter 2, that includes a summary of the original contribution of this thesis. The overview in Chapter 3 provides a high-level description of our segmentation system, describing how the various components interact to segment and track *IMO*s in a video sequence. The overview is followed by a detailed description of the design and implementation of each component. The motion estimation component is described in Chapter 4, and then the motion-based intensity constraint classification algorithm is discussed in Chapter 5. In Chapter 6, we detail the connected components analysis algorithm used to spatially segment objects in the sequence, and the active contour techniques used to recover object boundaries are explored in Chapter 7. The system design discussion is concluded in Chapter 8 with a review of the system as a whole. Here, we pay particular attention to the interaction between components in generating the final result: a segmented sequence in which *IMO* boundaries are efficiently

recovered.

Chapter 9 presents a series of experimental results generated by our segmentation system, and an accompanying analysis of each case. Finally, conclusions and future work directions are discussed in Chapter 10.

Chapter 2

Literature Review

Recovering the boundary, position and motion of the independently moving objects in a video sequence is effectively expressed through the notion of layers [51]. In describing a video sequence as a composition of layers, each layer is made to represent the shape and appearance of an *IMO*. By superimposing the complete set of layers in their correct 'visibility' order [30], the video sequence may be completely reconstructed by adjusting the position of each layer according to its independent motion in the sequence. The objective of this research can then be likened to the deconstruction of a video sequence into a set of *IMO* layers [52], with particular emphasis made upon recovering accurate object boundaries and accurate motion estimates for each *IMO* layer. Recovering a layer-like description of a video sequence requires that several key problems be addressed: segmenting the *IMO*s in the scene and subsequently tracking each *IMO* for as long as it is visible.

For the purposes of this research, our object segmentation relies upon motion to distinguish objects, as our sole interest is in exploring visual motion cues. As our focus is on *IMO*s only, techniques that segment sequences using appearance model databases or static intensity segmentation are not discussed here. Our object segmentation discussion focuses on motion-based segmentation, noting that a powerful motion segmentation system might be composed of dynamic and static segmentation approaches. Our research supports this idea by introducing

static boundary recovery techniques to supplement pure motion segmentation.

This literature review surveys the the range of related research, beginning with a discussion of the key research in motion estimation, which form the basis of motion segmentation techniques. The review continues with a discussion of object segmentation and tracking techniques, concluding with a discussion of the boundary recovery research relevant to object segmentation and tracking.

2.1 Motion Estimation

There are three main classes of motion estimation techniques: gradient-based approaches, correlation approaches and feature tracking approaches. We explore each of these areas with emphasis on the gradient-based approaches, which are of particular relevance to this research since they provide a reliable means to achieve motion-based object segmentation and tracking.

2.1.1 Gradient-based Motion Estimation

Gradient-based approaches, used in this research, rely on spatiotemporal gradient estimates to form motion constraints at each point in the sequence (see Section 1.2). Clustering coherent motion constraints attempts to address the aperture problem [29] by assuming that the motion constraints collected into a coherent cluster are generated by a single coherent object. As a result, this class of techniques ties the concept of motion estimation very closely to that of motion segmentation, as some segmentation of the frame must be recovered in order to generate a valid motion estimate.

The concept of using spatiotemporal intensity gradient constraints (BCC) relates closely to the research of Horn and Schunck [24] on optical flow. A dense set of optical flow estimates is calculated by regularization, integrating a single motion constraint with its neighbors through diffusion. As this process overcomes the aperture problem by blindly assuming that all the neighbors of a given pixel belong to the same object, the process exhibits unstable behavior

around object boundaries. Various techniques have been proposed to extend this approach to handle object discontinuities, such as Nagel's proposal of 'oriented smoothness' [40]. This approach employs second derivatives of the image to estimate edges in the image, to ensure that smoothness constraints are not applied across these edges.

Parametric motion models [37, 8] have been introduced to represent the valid motion of a single object. As parametric motion models are 'fitted' to the motion constraints of a single object in a video sequence, their use has prompted the need to develop clustering algorithms that are able to effectively collect sets of motion constraints in a coherent manner. Robust estimation techniques for estimating parametric optical flow [7, 8] take the approach of recovering a single dominant motion among a set of constraints, deterministically weighting each constraint based on its consistency with an evolving parameter set. In doing so, consistent constraints maintain and reinforce the parameter set, while inconsistent constraints are removed from consideration. The use of mixture models for motion constraint clustering [29], extends the robust statistical approach to apply a dynamic weighting to each motion constraint while iteratively refining the motion parameter set. This is the technique adapted for the *MASC* system, for initial scene segmentation and object motion estimation. Ju *et al.* [32] take a midway approach, using robust statistical techniques to calculate the affine motion parameters of (possibly) multiple objects in regular, fixed-size regions of the target sequence. Spatial smoothing is then applied to ensure optical flow field transitions between the regions to yield a continuous flow field.

One of the primary disadvantages of gradient-based techniques in estimating optical flow relates to the temporal aliasing effect that occurs due to large object displacements [23] (detailed in Section 4.1.2). As a result, gradient-based techniques must rely upon multiscale techniques [23, 6, 8], which estimate object displacements at multiple scales, starting at the smallest scale where image displacements are significantly smaller, and only refining displacement estimates at the increasingly larger scales. The multiscale approach is adopted for this research to ensure its applicability in typical video sequences.

Alternative means of generating motion constraints have been proposed, notably the use of phase-based motion constraints [20], which suggest improved performance over BCC-based motion constraints, due to their relative insensitivity to changes in illumination and shading [21, 5, 1]. These properties ensure that the spatiotemporal gradient values reflect actual changes in object position as opposed to changes in illumination. Along similar lines, Mann *et al.* [38] propose the substitution of pixel intensity values with photoquantigraphic values calculated on the basis of camera light response. This approach aims to eliminate the effects of differences in exposure between consecutive frames in a video sequence due to the common automatic exposure control of typical video cameras. Although phase and photoquantimetric-based motion constraints are not employed in the *MASC* system, the flexibility of the *MASC* system encourages the incorporation of such techniques where the application calls for them. Phase-based techniques in particular are very computationally intensive to accommodate, thus their invariance properties must be weighed against that cost.

2.1.2 Correlation-based Motion Estimation

Correlation-based motion estimation techniques are the other primary means of estimating optical flow in video sequences. Correlation techniques find the optimal displacement of a fixed sized region between two consecutive frames by comparing the intensities within target region in the first frame to the intensities of comparably sized neighboring regions in the second frame [33, 27]. The displacement that matches the initial region with its counterpart in the second frame is taken as the optical flow for the region. Various refinements have been suggested to improve the performance of correlation-based techniques [5], for example using bilinear interpolation in region matching to estimate sub-pixel displacements, and using affine region transformations instead of simple translations. Nevertheless, the approach suffers from the need to make a choice of window size, which cannot be performed automatically, nor can fixed size regions generate optimal results over a varied scene [27].

2.1.3 Feature-based Motion Estimation

A less related motion estimation technique is based on feature tracking, which tracks the movement of generic features such as corners and other high contrast points, to measure the movement within a video sequence. The registration of features within a window, proposed by Lucas and Kanade [37], presents this technique as a means of estimating stereo disparity. The adaptive feature tracking method presented by Shi and Tomasi [45] describes a tracking algorithm that selects optimal tracking features and continuously measures the 'tracking quality' of each feature over time. The sparseness of typical tracking features usually makes feature tracking techniques unsuitable for motion segmentation, while the need to track generic features within limited size windows prevents this approach from maintaining good tracking over large feature displacements.

2.2 Object Segmentation and Tracking

The aperture problem suggests that motion estimation using gradient-based techniques is inherently related to motion-based segmentation. Assuming the correct parametric motion model is used, the recovery of coherent clusters of motion constraints corresponds to the segmentation of the image. This occurs because motion constraints typically occur in spatially coherent clusters, being generated by a spatially coherent object.

While many motion estimation techniques exploit the consistency of motion constraints [29, 7, 8], such approaches do not necessarily produce effective segmentation, as they do not assume the spatial coherence of objects. This results in topographically disparate regions being classified as part of a single coherent object. This is often an incorrect assumption, although it may be valid occasionally, for example in the case of transparency. Most segmentation techniques explicitly account for the spatial coherence of objects using a variety of approaches to restricting the topographical flexibility of a suspected object.

In this section, we profile techniques used to segment and track independently moving ob-

jects in video sequences. From background differencing techniques and approaches incorporating spatial segmentation priors to curve-based techniques, we attempt to clarify the position of the *MASC* system in the spectrum of published research. Object tracking, while often tackled as an isolated domain, typically relies on explicit initialization on the target object to initiate tracking [9, 28, 49]. Tracking techniques can be seen as a class of techniques which overlap substantially with segmentation techniques, however in many respects the tracking problem may be considered a subset of the segmentation problem. Approaches that incorporate initial segmentation as well as tracking [30, 3, 25] are clearly more general than pure tracking techniques, and applicable to unsupervised systems, comparable to the *MASC* system.

2.2.1 Background Differencing Techniques

Intuitively, the simplest segmentation and tracking approaches are based on background differencing- assuming the background is stationary and constant, and a reference frame of the background is available. Any changing areas in such a scene can then be detecting and identified as moving objects, applying additional segmentation techniques to identify the multiple moving objects in the scene if necessary. Stauffer and Grimson apply background differencing principles using adaptive background mixture models to tune the approach to a real-time tracking system [47]. This approach however, is clearly limited to a stationary camera and requires a stable reference background.

2.2.2 Appearance Model-based Techniques

Appearance model-based techniques aim to learn the appearance of the objects of interest over a training sequence or during the course of the target sequence itself. This appearance model is then used to seek and identify regions in the video sequence that correspond to the target, segmenting these regions and tracking them over time by measuring the incremental region displacements.

The two dimensional 'sprites' proposed by Jojic and Frey [31] are dynamic appearance models of coherent moving regions in the video sequence, learned stochastically over the entire duration of the target video sequence. This method is shown to deal effectively with occlusion however it is only demonstrated when applied to the sequence as whole, and thus cannot be applied on-line.

El-Maraghi's approach introduces the use of dynamic appearance models in an on-line framework [28], actively learning the appearance model of a manually selected region, and using the appearance model to robustly track the region. The three-part appearance model comprises stable, transient and outlier components that allow the tracker to function correctly under noisy conditions and with occlusion. This tracking approach does not attempt to address the detection and segmentation issues, but demonstrates an alternative tracking approach that might be easily integrated with the *MASC* framework's tracking part.

The 'active blobs' proposed by Sclaroff [43] use an intelligent interaction of shape and appearance, but requires manual initialization. Each active blob is made up of a color texture map overlaid upon a three-dimensional shape mesh. Tracking is performed by warping the shape mesh (and its texture map) to register it against similar areas in succeeding frames. The three-dimensional nature of the mesh allows for a complex set of deformations. In addition, the matching process is implemented in commodity graphics hardware to conveniently exploit the high performance texture mapping and texture mapping available to provide high performance.

The multiple object segmentation and tracking techniques proposed by Irani *et al.* [25] employ robust motion segmentation techniques for initial automatic segmentation. The system then develops a windowed dynamic intensity appearance model, which is created by using the recovered motion parameters to register *IMO*s in the sequence, so that the object of interest appears stable over a span of frames. The span of registered frames are used to identify the temporally stable regions, which should correspond to the *IMO* of interest. A 'motion measure' function assigns the likelihood of motion at each point, so that dense region classifications may be made. This work however, does not impose any specific spatial coherence constraints,

limiting the bounds and connection of objects, or preferring compact objects against widely distributed objects. Similarly, the registration and comparison model does not necessarily segment object boundaries correctly when they are set against uniform intensity backgrounds. In the case of partial transparency, this is not necessarily the case. For example, the motion constraints generated by a car might be coherent around its sheet metal, however the transparent glass areas would generate motion constraints consistent with either the interior of the car, or the objects observed through the glass.

2.2.3 Techniques Incorporating Spatial Segmentation Priors

Tao and Sawhney [48] employ an ellipsoidal Gaussian spatial segmentation prior to enforce spatial coherence in a multiple object tracker. A dynamic appearance model is implemented by comparing intensity values within a layer's object region through multiple frames in motion, using a Gaussian likelihood function for matching. These two constraints and a parametric motion model are incorporated into a single objective function, optimized using the EM algorithm to segment the sequence into discrete layers. This approach uses a state machine to manage the initialization and removal of objects in the image sequence through time. The approach effectively tracks multiple independently moving objects in a scene, but is forced to rely upon its simple spatial model of ellipsoidal objects, detracting from its versatility.

Jepson *et al.* employ an octagonal spatial segmentation prior [30] whose edge likelihood varies as a half-Gaussian profile, allowing for a 'cushion' region in which the true object boundary is proposed to lie. The 'polybones' model is forced to accommodate objects with its octagonal shape prior, and so is not able to represent complex objects coherently, instead using multiple polybones to represent such objects. A background layer is used to handle background or camera motion, and new objects may be initialized at any time.

Segmentation employing spatial and intensity constraints are explored by Weiss [53] by incorporating spatial proximity and intensity difference within the constraint clustering objective function itself. The EM algorithm is then used to optimize the objective function, yielding

segmentation that propagates through uniform areas up to region boundaries. But this approach does not aim to recover accurate boundaries, and makes no attempt to explicitly use edge information.

The use of Markov random field models to enforce spatial constraints [22] provides an intelligent approach to linking areas, however these techniques place extreme demands on computational resources. An alternate technique that uses static segmentation and a modified MRF implementation to assign spatial priors is also suggested by Weiss [54], however the approach still does not attempt to recover accurate boundaries. In addition, the modified MRF implementation does not necessarily improve performance over typical MRF's, which require significant computational resources to perform effectively.

2.2.4 Area Integration-based Techniques

Wang and Adelson [51, 52] segment sequences into layers by the segmentation of spatiotemporal gradient constraints. Their approach recovers regional optical flow estimates from fixed size (5×5) neighborhoods, which impose spatial coherence constraints. Affine motion model parameters are then fitted to 20×20 neighborhoods of motion constraint pixels, and finally, *k-means* clustering techniques are then used to cluster the region motion estimates, collecting groups of regions. It should be noted that the groups are not spatially coherent by nature, to cope with transparency and occlusion.

Smith *et al.* [46] conceptually extend the optical flow recovery technique proposed by Ju *et al.* [32], by pre-segmenting the initial frame into intensity coherent regions using edge segmentation, before estimating the motion at the edges. The EM algorithm and a Minimum Description Length requirement are used to accommodate and initialize multiple object proposals in the scene. The edge ownerships are used to generate region ownership estimates by propagation in a maximum likelihood framework, thus completing the region segmentation process. However, this technique leaves a significant proportion of poorly labeled regions, especially regions with uniform intensity who share a nearly equal proportion of edges with

distinct objects.

The multi-layer segmentation process proposed by Darrell and Pentland [17] enforces spatial coherence using a line-process approach within contiguous regions of motion support to regularize optical flow estimation. This technique creates a number of layers, which are then further grouped using robust statistical processes to overcome transparency, occlusion and noise. Morphological constraints are further applied to the segmentation process to favor objects that are compact and coherent, however these spatial constraints are relatively weak and are not aimed at accurate shape recovery.

2.2.5 Curve and Boundary-based Techniques

While segmentation by its nature requires some element of shape recovery, the inclusion of accurate object shape as one of the *MASC* system objectives extends the notion of spatial coherence beyond a generic object model that models object shapes as simple geometric shapes or two-dimensional probability distributions [30, 48].

Active contours [34] provide a means of shape recovery, but by their formulation, require 'good' initialization to function correctly in their shape recovery and tracking roles. The use of motion to assist in their initialization and evolution through the duration of a sequence improves their stability, execution time and reliability.

Some tracking systems have been based purely upon boundary recovery techniques, for example Leymarie and Levine [36], applying a multiscale active contour implementation to track and adapt to the non-rigid deformations of cells in biological sequences. This approach does not explicitly attempt to directly recover the motion of the target objects. Terzopoulos and Szeliski [49] take this purely active contour-based approach by applying Kalman filtering techniques to smooth the evolution of the active contour in time, in an attempt to accommodate object motion within the snake framework itself.

The 'Active Shape Model' of Cootes et al. [13, 4] extends boundary recovery-based approaches using learning and subspace techniques, parameterizing the various boundary forms

of a fixed database of objects, and exhaustively searching for such objects in sequences. The parameterizations are able to accommodate a common range of deformations in the target objects, permitting the development of human shape trackers [4], that are able to detect, segment and track humans and provide an indication of their physical configuration.

The technique is further pursued by Isard and Blake [26], who introduce conditional density progapation techniques to estimate contour motion and deformation parameters during tracking. This approach is restricted to detecting and tracking template-based contours, and requires a training stage to learn a stochastic model of contour deformation and parameterized motion. This approach goes beyond the typical Kalman filtering model which uses a unimodal Gaussian distribution to propagate the contour, by using factored sampling techniques to approximate multimodal distributions for propagation. As a result, this approach is unable to deal with unknown objects, but could be incorporated within the *MASC* system which would provide a steady, or slowly changing object boundary for on-line learning.

2.3 Hybrid Techniques

By combining the aspects of segmentation, tracking and accurate shape recovery, hybrid techniques attempt to address segmentation and tracking within a single framework. Thus, while typical motion segmentation research attempts to resolve optical flow purely through the use of spatiotemporal constraints [29, 7], here we draw upon the assumption of spatial coherence to provide an added set of constraints.

The motion segmentation research of Cremers [14, 15] is based on a cost functional whose optimization requires the simultaneous solution of *IMO* motion parameters and the approximate *IMO* boundary location. The level set technique employed in this work allows the boundary representation to automatically merge and split to accommodate the topographical changes in the target *IMO*, and the cost function also attempts to minimize this boundary length to encourage compactness and spatial coherence. However, this approach is limited to consid-

ering only motion constraints for its boundary estimation, and is also limited to single scale gradient-based motion estimation, restricting the range of object displacement between consecutive frames.

Bascle *et al.* [2] combine region tracking and segmentation with active contour techniques, using region-based motion estimation to propagate an active contour between frames. Assuming the active contour is correctly configured in the initial frame, it is able to deform in the succeeding frame to accommodate non-rigid deformations in the target object. The affine parameters are Kalman filtered to temporally smooth the motion estimates, however the technique does not employ robust estimation techniques for motion estimation between frames, and also does not integrate any region-based segmentation information in the active contour's operation. A similar approach by Bascle and Deriche [3] substitute the gradient-based motion estimation techniques of [2] by correlation-based motion estimation, permitting larger displacements between frames.

Paragios and Deriche develop the theory of Geodesic Active Regions [41], an approach that is perhaps closest to our own approach of segmenting and tracking a set of multiple independently moving objects in a video sequence. The Geodesic Active Region technique uses geodesic active contours to refine motion segmentation results, by using the active contours to isolate each motion coherent region recovered by the segmentation. Once initialized, the isolated regions can then be analyzed individually to recover their motion. The segmentation relies on the assumption of a static background in the video sequence, from which a dense region segmentation map can be obtained. The dense region segmentation is used to assist the contour's evolution in each new frame, after which the contour is allowed to further deform according to typical geodesic active contour criterion. The static background assumption imposes the same limitations upon this technique as other techniques that make this assumption (Section 2.2.1), limiting its applicability. In addition, the lack of robust motion estimation may prevent the system from operating effectively where image noise and limited *IMO* transparency is present. Limited transparency may be observed when tracking a car with a window, for example. The

segmentation and tracking system described in this research adopts a similar hybrid approach by actively combining motion segmentation and tracking techniques with active contour-based boundary recovery techniques, and the next section describes our contribution to this area of research.

2.4 Original Contribution

The segmentation framework presented in this thesis proposes the integration of region and boundary-based techniques to address the problem of object segmentation and tracking, in the belief that the effective combination of these techniques can result in a better solution than one based on only one of those philosophies. Our framework aims to integrate seemingly disparate components to enable strong collaboration, yet still remain flexible enough to allow for the substitution of individual components, based on application-specific requirements without compromising the overall integration of the system. The resulting system proposes a novel solution to the problems of automatic initialization, multiple object tracking and accurate shape recovery within a single framework. It also demonstrates a vision system that can provide higher level vision components with useful motion and shape data for object recognition, visual surveillance and video compression.

Part II

Methodology

Chapter 3

The *MASC* Segmentation System

The principle of our segmentation system is to combine region and boundary information, allowing techniques based on both approaches to overcome any weaknesses present in an approach that only relies on one of them. We employ motion estimation techniques to provide region information and active contour techniques to recover boundary information, so we use the *MASC* acronym to refer to this system which performs *M*otion segmentation using *A*ctive contours for *S*patial *C*oherence.

The segmentation procedure of resolving the motion and shape of *IMO*s within a video sequence, comprises the following parts:

- Detect and coarsely segment *IMO*s in the sequence using motion estimation methods;
- Use the coarse segmentation to initialize and refine the boundary estimates of each *IMO* to recover an 'accurate' *IMO* shape representation;
- Continuously track each *IMO* to maintain a stable segmentation over time, incorporating boundary estimates within the motion estimation process to improve performance.

Detection and coarse segmentation of *IMO*s in the scene is performed by region-based motion segmentation methods, which allow boundary-based methods to be instantiated. Information from both types of methods are then used to continuously track the *IMO*s throughout

the sequence, and eventually recover an accurate *IMO* shape representation. Special considerations are also made for the scene background which may be stationary or moving due to camera motion or movement of the background itself. The ultimate goal of providing a completely automatic segmentation of the image is achieved by initializing an independent tracking process for each *IMO* present in the scene.

This section presents an overview of the region and boundary-based components, and the way in which information is shared between them. The remainder of this chapter goes on to develop each of the components in full detail, presenting their background, implementation and analysis. In addition, the interaction between components is described throughout the thesis, but formally discussed in Chapter 8.

3.1 Methodology

The *MASC* system aims to provide a general framework for tightly integrating region-based motion techniques with boundary-based active contour techniques. The result of its development is a complete motion segmentation and tracking system that aims to recover *IMO* boundaries accurately. Figure 3.1 illustrates the configuration of the various components within the system, and a summary of the diagram is provided below.

The motion segmentation component recovers the overall segmentation of a scene and the approximate locations of each *IMO* by performing motion estimation upon the entire frame. The motion segmentation generates a coarse estimate of each *IMO*'s location and boundary, as well as an estimate of the motion of each *IMO*. A spatial segmentation process is applied to segment two or more *IMO*s share similar motions, if there is sufficient spatial separation between those *IMO*s. The resulting boundary and location estimates are typically sparse however, as motion segmentation is performed on spatiotemporal gradient constraints (motion constraints), which are only present in areas with significant edge structure.

In order to increase the effectiveness of the motion segmentation process, a motion-based

intensity constraint classification operation is performed to increase the density of the region segmentation map, generating a new motion segmentation map. The intensity of pixels between between two successive frames is compared using the detected *IMO* motion between the frames to warp the *IMO* from its position in the second frame to its position in the first. By comparing the original frame to the warped second frame, the regions whose intensity remains constant in the two frames provide a good indication of which regions are part of the target *IMO*.

Spatial segmentation based on the Connected Components Analysis (CCA) algorithm is then applied to the motion segmentation maps, in order to identify a single *IMO* of interest, as there may be more than a single *IMO* with a given motion pattern. The spatial segmentation aims to isolate the single largest spatially coherent region identified by the motion segmentation map to provide a coarse *IMO* boundary estimate. The remaining areas are removed from further consideration within the current process, but are available for inclusion in separate tracking processes as distinct objects.

The segmentation result is used to initialize the boundary-based active contour around the target *IMO*. The active contour position evolves from its motion-based initial position to account for the edges in the image, while incorporating motion segmentation information in its adjustments. In turn, the boundary estimates recovered by the active contour are used by the motion segmentation algorithm to localize its behavior, confining the area over which *IMO* motion estimates are derived.

The *IMO* motion and boundary is continuously refined and used throughout the sequence to maintain steady tracking, while a global motion estimation process continues to operate in non-*IMO* regions, in the search for new *IMO*s that might appear in the scene. The global process maintains awareness of the rest of the scene so that when combined with the individual *IMO* tracking processes, a complete detection, segmentation and tracking procedure is established and maintained.

Through time, the active contour boundary estimates assist motion estimation by isolating

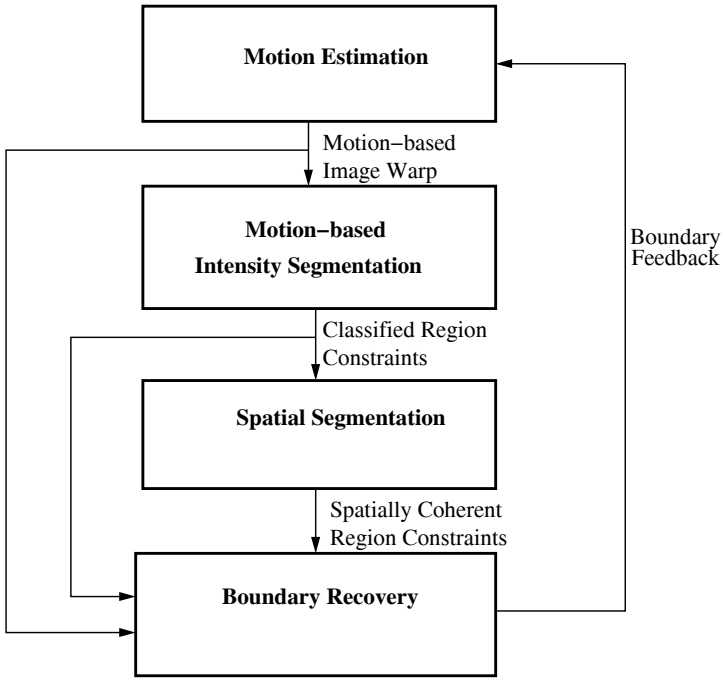


Figure 3.1: *MASC* Segmentation System Block Diagram

all relevant *IMO* motion constraints within its boundary, removing noise from the *IMO* motion estimation process. In turn, the improved motion estimates assist the motion segmentation process by providing richer region-based information to the active contour, improving boundary estimates. The boundary estimates gradually improve over time, to generate an accurate representation of *IMO* shape.

The interaction of components through time results in a system that consciously balances global criteria affecting the scene as a whole, and local criteria affecting specific areas in the frame. The interaction also integrates the dynamic (motion-based) and static (edge-based) properties of image sequences in an effective manner. The primary goals of detection, segmentation, stable tracking and shape recovery are thus met.

Chapter 4

Motion Estimation

Motion estimation is performed by the *MASC* system to recover the coarse segmentation of a sequence and generate estimates of each *IMO*'s parametric motion. Continuous analysis of the video sequence permits the estimation process to detect any new *IMO*s that appear in the scene, so that a new tracking process may be initialized for new *IMO*s. Once a new *IMO* has been detected and an initial segmentation for it obtained, an independent *IMO* tracking process is initiated for the new *IMO*. As a result, initial segmentation is only required when any given *IMO* first appears in the scene.

Initial detection and segmentation of *IMO*s in a scene is achieved by performing a global analysis of spatiotemporal gradient constraints in the scene, which must be clustered consistently with respect to parameterized motion models. Each motion model represents the consistent motion of a single *IMO*, so that the clustering process isolates constraints that move with a single motion. As the analysis is global, multiple *IMO*s may be present in the scene, requiring the application of robust clustering techniques to differentiate between the sets of constraints. Successfully applying the motion estimation process for each *IMO* produces a segmentation that assigns ownership of spatiotemporal gradient constraints to distinct motion processes, providing a crude segmentation of the scene into a set of motion-consistent regions, which may not be spatially coherent. However, these results permit the execution of boundary recovery

techniques which refine the position and shape estimates of each *IMO* using techniques that do incorporate spatial coherence properties. This information is fed back to the motion estimation component, assisting the motion segmentation to provide more accurate estimates of *IMO* motion.

Once an *IMO* tracking process has been initiated (after initial segmentation), motion estimation is continually applied to the *IMO* region to update its motion parameter estimates and global motion-based segmentation estimates. In this case however, the motion estimation is only applied within the isolated *IMO* region, but in the same robust manner as for the global estimation, to account for changes in *IMO* shape, transparency and occlusion. This chapter explores the process behind motion estimation using gradient-based techniques. We begin by describing single scale motion estimation, and extend the implementation to multiple scales in Section 4.2. We conclude with a discussion of how the estimation process is applied to segment and estimate the motion of multiple *IMO*s in a video sequence.

4.1 Single Scale Motion Estimation

The principles of optical flow and the techniques for estimating it from a video sequence are described in Section 1.2, This section describes the particular algorithm implemented in the *MASC* system to robustly estimate *IMO* motion parameters and perform motion-based region segmentation.

The estimation of coherent motion comprises the accurate recovery of motion constraints for a given pair of frames, followed by the clustering of motion constraints according to parametric motion models to identify and segment the coherent motion layers. Each coherent motion layer logically represents one or more *IMO*s undergoing a single plausible motion in the sequence. Because of this, motion-based classification of constraint regions results in a layered segmentation of the sequence at that point in time.

Figure 4.1 illustrates the motion estimation process at a single scale, for the recovery of

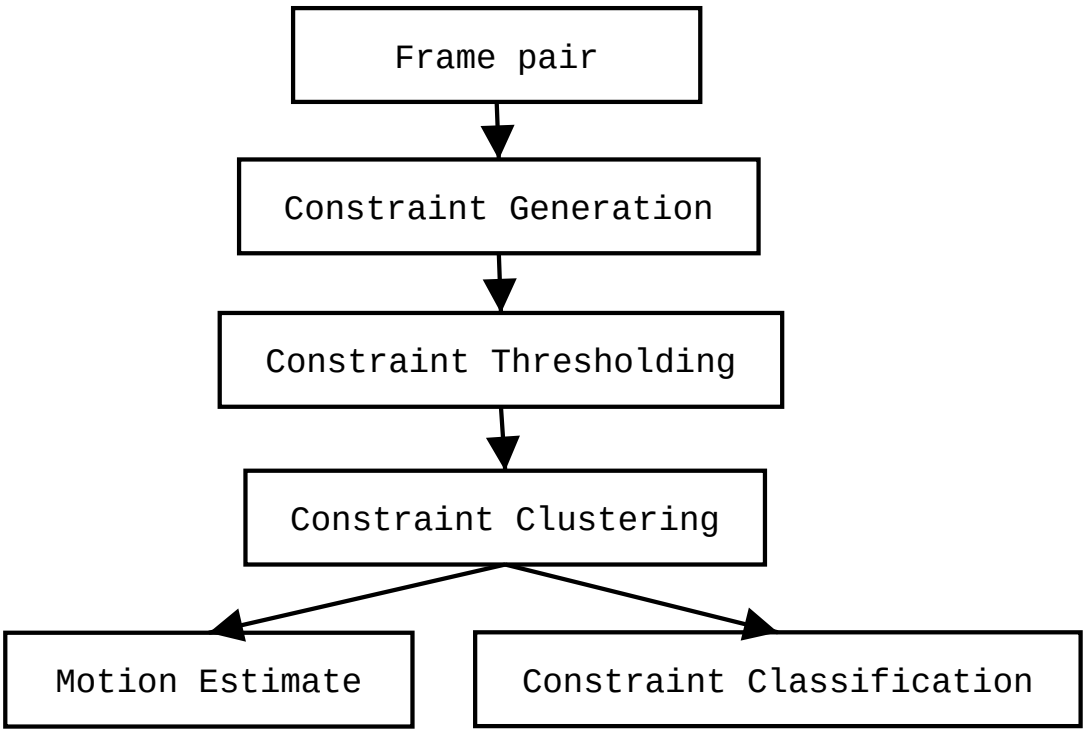


Figure 4.1: Motion Estimation Process

a single motion process. The motion estimation process is in fact applied at multiple scales, on a number of subsampled versions of the sequence as well as at the original size. This allows the estimation procedure to maintain effectiveness and accuracy even for relatively large *IMO* displacements, as this is not possible at a single scale. Our discussion of spatiotemporal constraints in Section 4.1.1 describes the unsuitability of single scale motion estimation for large displacements due to the difficulty in estimating temporal derivatives, and the extension to multiscale motion estimation is discussed in detail in Section 4.2. Each component shown in Figure 4.1 is described below, detailing the motion estimation process at a single scale.

4.1.1 Spatiotemporal Constraints

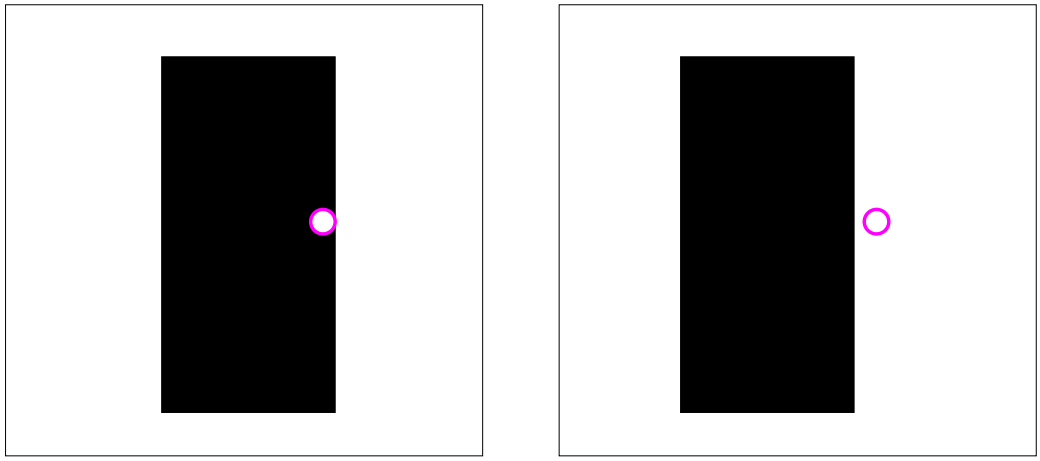
The *MASC* system relies on brightness constancy [24] to estimate motion, however it may be easily adapted to use alternative motion constraints based on other principles such as phase constancy [21], which provides better robustness to variable lighting conditions at the expense

of significant computational overhead.

The spatiotemporal gradient (motion) constraints introduced in Section 1.2 are made up of three components: the x and y dimensional spatial gradients of a single frame, I_x and I_y, and the temporal gradient of that frame, I_t. The spatial gradient components are estimated by calculating the numerical derivative of the intensity image along the x and y directions. To estimate the spatial derivatives we use a Gaussian smoothing filter to interpolate the image signal, after which a discrete three-point difference kernel may be applied to approximate the derivative [50]. In fact, as both the difference and smoothing filters are separable, they can be combined into a pair of 1-D filters that can be applied separately in the x and y directions.

Temporal gradient estimation uses two successive frames to estimate I_t, a process known as two-frame flow estimation, although more frames can be used for the approximation. The temporal gradient is approximated as the temporal difference between the two successive frames, which are Gaussian smoothed prior to calculation to remove discontinuities in intensity changes between the frames, so that the difference calculation more closely approximates the true temporal derivative. The Gaussian smoothing filter is configured to have a standard deviation of $\sigma = 1.5$ pixels, so that displacements of intensity discontinuities (such as a strong edge) are smoothed assuming that *IMO* displacements will be approximately within this range. The σ value is indicative of the distance from a point over which the smoothing is effective, as it determines the effective width of the filter. As a result, temporal derivative values are thus valid for displacements less than approximately 2 pixels, an adequate displacement range within a single scale.

The maximum displacement size that can be estimated can be justified intuitively by analyzing the behavior of temporal gradient estimates for a small region about a moving intensity discontinuity or edge. Even for a very small displacment, the temporal difference of a pixel on a moving edge will be the same as it would be for a larger displacement, as illustrated in Figure 4.2. This effect prevents the effective estimation of the true motion of the black block, as the temporal gradient estimate is constant irrespective of actual block displacement. If a Gaus-



(a) Frame 1 (b) Frame 2

Figure 4.2: The temporal difference for a pixel in the region within the light circle is constant for small or large displacements of the black rectangle as it moves leftward, preventing effective motion estimation.

sian smoothing filter is used to smooth the sharp edges of the block, the change in intensity becomes continuous and smooth even at the edges, but the smoothing is only effective if the displacement of the block is less than the effective width of the filter, roughly indicated by σ. Note that while increasing the σ value increases the valid range of the displacements for which derivatives may be estimated, it also decreases the accuracy of those estimates. As a result, it is impossible to estimate larger displacements with reasonable accuracy at a single scale. We ovecome this problem through the introduction of multiscale estimation in Section 4.2.

Finally, the three-vector, $\nabla I = \begin{bmatrix} I_x & I_y & I_t \end{bmatrix}^T$, is used to represent each spatiotemporal gradient constraint, which is present at every point in the image. Not every spatiotemporal gradient value is a valid motion constraint however, as will be shown in the next section.

4.1.2 Constraints Thresholding

A large portion of spatiotemporal gradient constraints are not valid for motion calculation. There are two primary classes of invalid constraints, constraints that are unreasonably small,

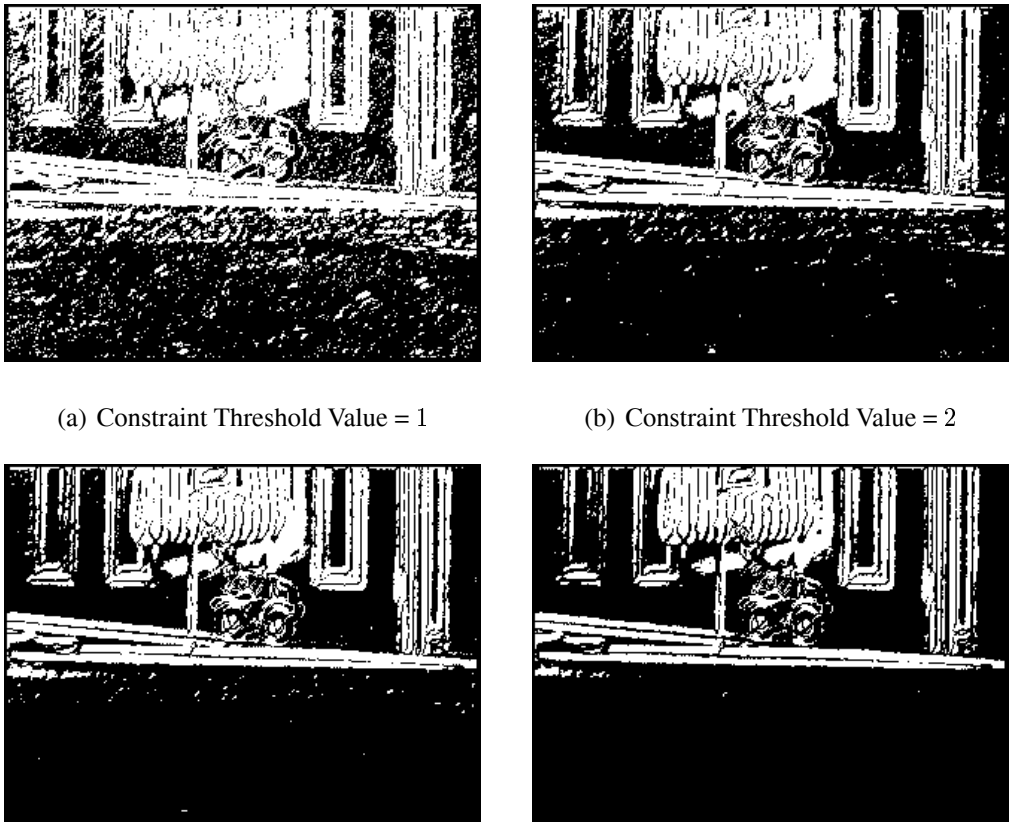
and constraints that indicate motion beyond our range of estimation. Minimum thresholding eliminates constraints with very low spatial gradient magnitude, $|\nabla I_{\vec{x}}| = ||[I_x, I_y]^T|$, due to the unreliability of these constraints. When the spatial gradient at a point in the frame is very small, such as in regions of uniform texture, noise has a large influence on the constraint value. As a result, any calculations using these noisy values will be correspondingly noisy. These constraints can be seen as having an extremely poor signal-to-noise ratio (SNR), where the minimal BCC evidence they provide is corrupted by sensor and quantization error. The minimum $|\nabla I_{\vec{x}}|$ threshold is configured to remove noisy gradient estimates, filtering out spatiotemporal gradient constraints derived from uniform intensity regions.

This thresholding step is intuitive with respect to the aperture problem, which indicates that regions of uniform brightness provide no evidence of motion, while a small amount of noise added to such a region produce constraints with small magnitude that have no relation to the true brightness changes in the scene. As such, they cannot provide valid motion constraints for motion estimation process, and instead simply introduce a set of noisy constraints. This process also suggests an intuitive process through which the minimum threshold value can be interpreted, by visual inspection of the valid constraint map. Valid constraints should only exist at or around regions where texture or edges are present, as these are the only regions where motion can be locally distinguished.

The threshold value for the minimum constraint magnitude is heavily dependent upon the spatiotemporal constraint generation process, specifically the means of smoothing and the difference approximation technique chosen. The *MASC* implementation uses a 3-point central difference approximation, and a Gaussian smoothing kernel with a standard deviation of 1.5 pixels. In this configuration, testing indicates the optimum minimum threshold value to be in the range of 3 units, so that we impose the requirement that all valid constraints have a spatial derivative magnitude, $||[I_x, I_y]^T|| > 3$. For the tow truck test sequence shown in Figure 4.3, the valid constraint maps over a range of minimum threshold values are shown in Figure 4.4, providing intuitive justification for the choice of threshold.

(a) Frame 45 (b) Frame 46

Figure 4.3: Tow Truck Sample Sequence: Toy tow truck rolls down the ramp toward the right of the scene against a stationary background.



(a) Constraint Threshold Value = 1 (b) Constraint Threshold Value = 2

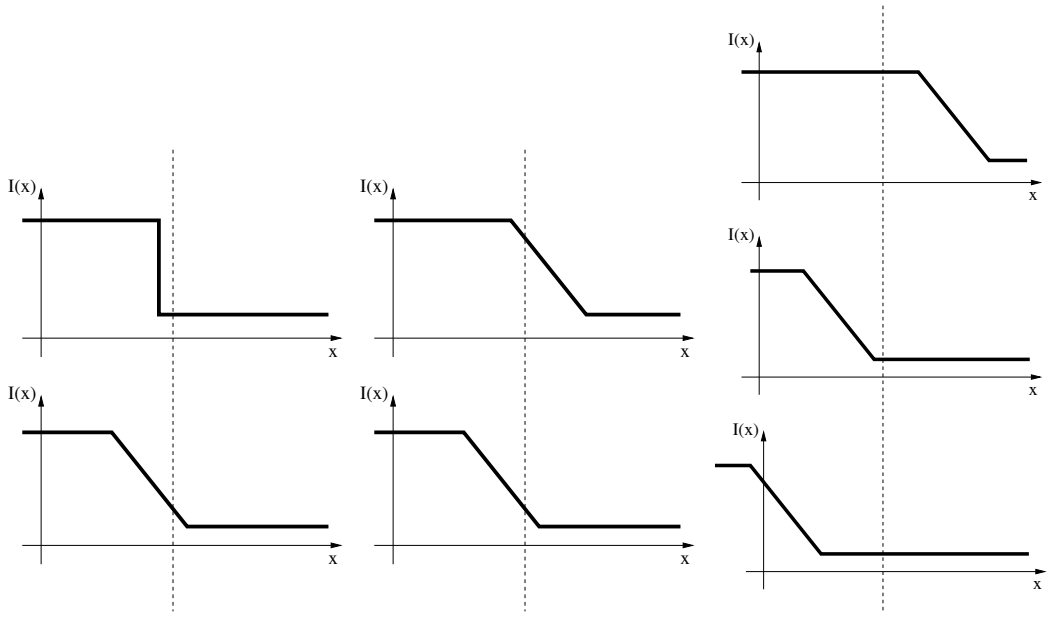
(c) Constraint Threshold Value = 3 (d) Constraint Threshold Value = 4

Figure 4.4: Tow Truck Sequence Thresholded Constraint Map

While constraints with small spatial gradient magnitude are dominated by noise, constraints with extremely large magnitude are generated by large displacements or noise. Many large displacements destabilize the temporal gradient estimates as the size of such displacements prevent the temporal differencing procedure from generating a useful approximation of the temporal derivative at a point, due to temporal aliasing. The condition is demonstrated in Figure 4.5, where we show the change of intensity due to motion of a step edge that might occur due to a translating rectangle, such as that shown in Figure 4.2. The step edge at the top of Figure 4.5(a) undergoes smoothing, reducing the gradient of the step to that shown in the lower part of Figure 4.5(a). The correct temporal derivative estimate is obtained through temporal differencing for the small displacement of Figure 4.5(b). However, for larger displacements beyond the size of the smoothing filter's effective width, the temporal difference obtained has no relation whatsoever to the true temporal derivative at the point when it was first sampled., as shown in Figure 4.5(c).

Large displacements therefore have the same effect as overly large temporal differences between image samples, and the example of Figure 4.5 demonstrates that temporal gradient estimates can be wildly inaccurate for large displacements and at low sample rates. In fact, the two conditions are equivalent: increasing the frame capture rate reduces the size of displacements between consecutive frames, however this places significantly higher demands on the available system bandwidth and processing power, so this approach is typically impractical. Meanwhile, the aliasing effect prevents single scale motion estimates from being accurate for large displacement magnitudes [23]. A multiscale motion estimation method (described in Section 4.2) uses several sub-sampled versions (scales) of the sequence to capture larger displacements. These large displacement estimates are used to recursively refine the motion estimates at larger scales, so that only a limited range of displacement need to be measured at any given scale.

As the development of constraints follows the assumption that only small displacements are valid at any given scale, the upper thresholding process described below eliminates a large



(a) The original step edge (top), and the smoothed edge (bottom)

(b) The original edge position (top) is displaced by a small amount (bottom)

(c) The original edge position (top) is displaced by a large amount (middle), generating a specific temporal difference value. However, if the displacement is even larger (bottom), the difference in displacement size is not reflected by the temporal difference at all, due to temporal aliasing.

Figure 4.5: The temporal aliasing phenomenon is demonstrated for a smoothed step edge.

class of the constraints that could only be generated by large displacements outside of the range of interest. These constraints may be generated legitimately, such as by a relatively large area in the frame with a very smoothly changing textured intensity undergoing a large displacement. As a result of the large smoothly changing area, the temporal aliasing phenomenon would not occur, and the motion constraints generated by this region would provide reliable evidence of the true motion. However, we defer large displacement handling to the multiscale formulation, so that we may assume that at any single scale, the *IMO* displacements will be within a reasonable range for estimation. In this case, a range of two pixels is indicated by the description of spatiotemporal constraint computation. We therefore remove the motion constraints that could only correspond to large displacements, although many motion constraints that may be generated by large displacements may still remain, as each constraint only provides one component of a point's velocity.

We identify constraints that indicate displacements outside of the range of interest by analyzing the minimum pixel displacement that could result from a given motion constraint. A pixel's displacement, $\vec{v}$, generated by the motion constraint ∇I, satisfies the BCC if

$$\nabla I_{\vec{x}} \cdot \vec{v} + I_t = 0 \tag{4.1}$$

$$\nabla I_{\vec{x}} \cdot \vec{v} = -I_t \tag{4.2}$$

$$\left|\nabla I_{\vec{x}} \cdot \vec{v}\right| = \left|I_t\right| . \tag{4.3}$$

Now using the cosine formula:

$$\left|\nabla I_{\vec{x}} \cdot \vec{v}\right| = \left|\nabla I_{\vec{x}}||\vec{v}||cos\alpha\right| , \tag{4.4}$$

where α is the angle between the two vectors $\nabla I_{\vec{x}}$ and $\vec{v}$. $\nabla I_{\vec{x}}$. So we obtain the relation

$$|\vec{v}| = \frac{|I_t|}{|\nabla I_{\vec{x}}||cos\alpha|} \tag{4.5}$$

This relation allows a maximum gradient magnitude threshold to be derived and applied to the set of spatiotemporal constraints:

$$\frac{|I_t|}{|\nabla I_{\vec{x}}|} \leq \text{(Maximum Displacement)} \tag{4.6}$$

The maximum displacement threshold is set to two pixels of displacement for the *MASC* system, allowing the robust clustering techniques to handle any noisy constraints that still exist within this range. The multiscale technique introduced in Section 4.2 allows the motion estimation to continue to perform correctly even for large displacements.

4.1.3 Mixture Models for Optical Flow

Once a valid set of motion constraints has been calculated for the two-frame pair, the constraints must be clustered to recover the coherent motion patterns. The constraint generation process allows us to assume that at any given image scale, displacements are 'reasonable', that is within the range we can reliably estimate given the limitations of gradient-based flow estimation. This section introduces the use of mixture models to recover optical flow as suggested by Jepson and Black [29]. Mixture models and the EM algorithm were introduced in Section 1.4, and this section discusses their application in clustering motion constraints to recover parametric optical flow. This approach assumes that the constraint set represents a 'mixture' of motions generated by various layers within the scene, with random noise also present. Jepson and Black [29] address the aperture problem by clustering motion constraints that are strictly consistent with each other, so that all the motion constraints within a single motion layer support a common velocity estimate up to a given tolerance. The same constraint classification criterion is adopted in this work, and is described below.

The Brightness Constancy Constraint (BCC) of Equation 1.4, can be expressed as

$$\vec{\nabla} I_k \cdot \vec{v}_n \;\; = \;\; 0 \; , \tag{4.7}$$

where $\vec{\nabla} I_k$ is the spatiotemporal constraint vector corresponding to pixel k, and $\vec{v}_n$ is a vector representing the velocity of the *IMO* layer with index n:

$$\vec{v}_n \;\; = \;\; \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix}$$

The third component of $\vec{v}_n$ is used to represent the temporal displacement of the *IMO* corresponding to the spatial displacements, in this case one, as Δt is measured in frames. The motion parameter estimation process described below estimates the motion parameters as a three-vector, but only in two degrees of freedom. A similar case is observed for the higher order affine motion model, where a seven-vector is solved for in six degrees of freedom. In either case, the actual motion parameters may be obtained by simply normalizing the solution with respect to the temporal displacement (last element in the vector), to make it equal to one.

The consistency of a (spatiotemporal) motion constraint with a velocity $\vec{v}_n$, is measured by its deviation from the BCC, Equation 4.7. We employ a likelihood function defined by the angular error of the spatiotemporal gradient constraint $\vec{\nabla} I_k$ with respect to velocity $\vec{v}_n$, approximating this angular error with its sine, assuming small angular error and noting that under such conditions, they are apporixmately equivalent [29]:

$$d_{nk} = \frac{\vec{\nabla} I_k \cdot \vec{v}_n}{|\vec{\nabla} I_k||\vec{v}_n|} \tag{4.8}$$

We use this deviation estimate, d_{nk}, to define a Gaussian likelihood function:

$$P(\vec{\nabla} I_k|\vec{v}_n) \quad = \quad \frac{1}{\sqrt{2\pi}\sigma_v} \exp\left(-\frac{d_{nk}^2}{2\sigma_v^2}\right) \tag{4.9}$$

While the d_{nk} error function has finite range and the Gaussian likelihood function has infinite range, it should be noted that the above likelihood function is only used to model reasonably accurate constraints. An outlier distribution is introduced in Section 4.1.5 to model constraints with larger error. In doing so, σ_v is an estimate of the error in estimating the spatiotemporal gradients, sensor noise, and must also account for the modeling error associated with using a parametric motion model to represent the projection of three dimensional motion. For example, the use of a translational motion model cannot represent an object's rotation in the plane, and thus any part of the motion constraints resulting from a rotational motion must be accounted for as modelling error. A reasonable value for σ_v is in the range $[0.1, 0.5]$ [29, 28], however a simulated annealing process is introduced in Section 4.1.7 that systematically varies σ_v to assist the optimization of the EM algorithm.

In the more general case, the motion parameters of an *IMO* are represented by the vector, $\vec{\theta}_n$, instead of the translational motion vector, v_n, to accomodate higher-order parametric motion models. Taking this into account, the likelihood function can be used in the Expectation Maximization (EM) algorithm described in Section 1.4, so with the mixture probabilities of the *IMO*s in vector $\vec{m}$ and given the *IMO* motion parameter set in the mixture model, $(\vec{\theta}_1, \ldots, \vec{\theta}_N)$, the probability of observing any constraint, $\vec{\nabla} I_k$, is

$$p(\vec{\nabla} I_k | \vec{m}, \vec{\theta}_1, \ldots, \vec{\theta}_N) \quad = \quad \sum_{n=0}^{N} m_n p_n(\vec{\nabla} I_k | \vec{m}, \vec{\theta}_n) \; . \tag{4.10}$$

The log likelihood of the entire constraint set can then be calculated as

$$\log L(\vec{m}, \vec{\theta}_1, \ldots, \vec{\theta}_N) \quad = \quad \sum_{k=1}^{K} \log p(\vec{\nabla} I_k | \vec{x}_k, \vec{m}, \vec{\theta}_1, \ldots, \vec{\theta}_N) \; , \tag{4.11}$$

allowing us to define the ownership probability for each constraint, $\vec{\nabla} I_k$, with respect to the motion parameters of the nth motion as

$$q_{nk} \quad = \quad \frac{m_n p_n(\vec{\nabla} I_k | \vec{x}_k, \vec{\theta}_n)}{\sum_{j=0}^{N} m_j p_j(\vec{\nabla} I_k | \vec{x}_k, \vec{\theta}_j)} \; . \tag{4.12}$$

Ownership estimation concludes the E-step of the EM algorithm, by calculating the probability of a motion constraint being generated by a specific *IMO*. Calculating ownership values for each constraint under each *IMO* motion parameter set quantifies the relationship between every constraint with every *IMO*, determining the influence each motion constraint will have on the *IMO* motion parameter refinement in the M-step of the algorithm. Conversely, each ownership value identifies the probability with which the motion constraint belongs to an *IMO*, so that the image may now be coarsely segmented by assigning each motion constraint to the *IMO* with maximum ownership for that constraint. The M-step of the algorithm involves optimizing $L(\vec{m}, \vec{\theta}_1, \ldots, \vec{\theta}_N)$ with respect to the motion parameter set, $(\vec{\theta}_1, \ldots, \vec{\theta}_N)$, and is described in the next section.

4.1.4 Motion Parameter Optimization

Under a parametric motion model, the velocity of a point on an *IMO* can be expressed as a function of image location, $\vec{x}$, as $\vec{v}(\vec{x}) = \begin{bmatrix} v_x(\vec{x}) & v_y(\vec{x}) & 1 \end{bmatrix}^T$. To simplify computation for higher order parametric models, it is convenient to represent this velocity as a product of a matrix $\mathbf{M}(\vec{x})$, and a vector $\vec{\theta}$:

$$\vec{v} \;=\; \mathbf{M}(\vec{x})\vec{\theta} \tag{4.13}$$

This form assigns the the parameters of the motion model to the $\vec{\theta}$ vector, separating them from the point coordinate data that is required to calculate point velocity for higher order motion models. However, if the *IMO* is moving under the assumption of a constant motion model, *all* points on that *IMO* will have image velocity $\vec{v}$. In this case, point velocity is not dependent on point location, so the matrix $\mathbf{M}$ is simply the identity matrix, and $\vec{\theta}$ represents point velocities:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \vec{\theta} = \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} \tag{4.14}$$

In contrast, the affine motion model represents image point velocity as a function of point location:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} \;=\; \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{4.15}$$

The image velocity components v_x and v_y can therefore be expressed as

$$v_x \;=\; A_{11}x + A_{12}y + b_1 \tag{4.16}$$

$$v_y \;=\; A_{21}x + A_{22}y + b_2\ . \tag{4.17}$$

In this case, the vector $\vec{\theta}$ contains the affine flow parameters,

$$\vec{\theta} \;=\; \begin{bmatrix} A_{11} & A_{12} & A_{21} & A_{22} & b_1 & b_2 & 1 \end{bmatrix}^T\ , \tag{4.18}$$

and the matrix $\mathbf{M}$ is now a function of point location, $\vec{x}$:

$$\mathbf{M}(\vec{x}) = \begin{bmatrix} x & y & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & x & y & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.19}$$

Higher order parametric models, such as the projective motion model, may also be represented in a similar manner. The motion parameter optimization process is applicable to any parametric model, so that they may also be incorporated within the *MASC* system. Each *IMO*'s motion parameters may be estimated by optimizing a least squares objective function defined over the spatiotemporal gradient constraint set, $\vec{\nabla} I_k$, for $k = 1, ...K$. In the simplest case, we assume all the constraints are generated by the *IMO* of interest, that is, there are no other *IMO*s in the scene. Reiterating the brightness constancy assumption:

$$I_x v_x + I_y v_y + I_t = 0 \ , \tag{4.20}$$

which may now be expressed in the terms defined above:

$$\vec{\nabla} I^T \vec{v} = 0 \tag{4.21}$$

$$\vec{\nabla} I^T \mathbf{M} \vec{\theta} = 0 \tag{4.22}$$

Where

$$\vec{\nabla} I = \begin{bmatrix} I_x & I_y & I_t \end{bmatrix}^T \tag{4.23}$$

To find the motion parameters, $\vec{\theta}$, that best satisfy the constraint set in the least squares sense, we minimize the cost function, Ψ with respect to $\vec{\theta}$:

$$\Psi(\vec{\theta}) = \sum_{k=1}^{K} (\vec{\nabla} I_k^T \mathbf{M}_k \vec{\theta})^2 \tag{4.24}$$

$$= \sum_{k=1}^{K} (\vec{\theta}^T \mathbf{M}_k^T \vec{\nabla} I_k \vec{\nabla} I_k^T \mathbf{M}_k \vec{\theta}) \tag{4.25}$$

$$= \sum_{k=1}^{K} (\vec{\theta}^T \mathbf{D}_k \vec{\theta}) \tag{4.26}$$

$$= \vec{\theta}^T (\sum_{k=1}^{K} \mathbf{D}_k) \vec{\theta} \tag{4.27}$$

The value of $\vec{\theta}$ that yields the minimum Ψ is the least squares solution of the motion estimation problem. We can solve for $\vec{\theta}$ by finding the eigenvector of matrix $\mathbf{D} = \sum_{k=1}^{K} \mathbf{D}_k$, corresponding to $\mathbf{D}$'s smallest eigenvalue.

The least squares motion estimation process described above assumes that the complete spatiotemporal gradient constraint set, $\vec{\nabla} I_k$, is generated by a single moving *IMO* whose motion is well modeled using the assumed parametric motion model. These two requirements are a result of the sensitivity of the least squares optimization process to noise, as the optimization performs very poorly when the residual noise in the constraint set is non-Gaussian [8]. Within the mixture model framework however, the ownership probabilities of the Expectation step of the EM algorithm provide a quantitative indication of the coherence of constraints with respect to a given motion parameter set.

Assuming each scene point with index k has ownership probability q_{nk} with respect to motion n, the $\mathbf{D}$ matrix of Equation 4.27 relating to *IMO* n can now be expressed as the following weighted sum,

$$\mathbf{D} \;=\; \frac{\sum_{k=1}^{K} q_{nk} M_k^T \vec{\nabla} I_k \vec{\nabla} I_k^T M_k}{\sum_{k=1}^{K} q_{nk}} \;. \tag{4.28}$$

This can be likened to calculating a weighted mean value (for a matrix in this case). Calculating the eigenvector of D that corresponds to its minimum eigenvalue produces an estimate for the motion parameter vector, $\vec{\theta}$, that is optimized with respect to the ownership probabilities. This optimization completes the M-step of the EM algorithm used to robustly estimate *IMO* motion parameters.

By iterating between the E-step and M-step, the ownership and motion parameter estimates are continuously refined and improved, within the limits of the EM algorithm. The process can be terminated when the data's log likelihood, shown in Equation 4.11, ceases to increase noticeably. It should be noted however, that the quality of the results from this algorithm depend on the motion parameter set used in the first iteration, due to the inability of the EM algorithm to guarantee recovery of the global maximum of the log data likelihood function.

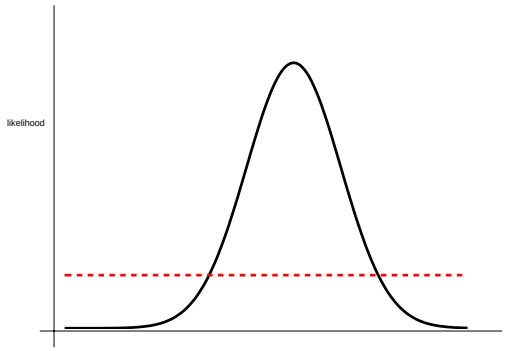


Figure 4.6: The outlier distribution: The Gaussian inlier distribution (solid line) claims ownership of constraints within ρ standard deviations of its mean. Beyond this range, the uniform outlier distribution (dashed line) claims ownership to any errant constraints.

Motion parameter initialization for the *MASC* system is closely related to the multiscale motion estimation process, discussed in Section 4.2.

4.1.5 Outlier Components

The EM process described above effectively clusters constraints generated by distinct *IMO*s, however, in any real data set a small but significant proportion of constraints do not correspond to any specific *IMO*, as their values are corrupted by noise and violations of the BCC. An additional layer called the outlier layer is used to assume ownership of such stray constraints, by claiming ownership of any constraints that are not 'strongly' claimed by an *IMO*.

The outlier layer is configured as a distinct *IMO* layer, with index $(n = 0)$, for which we approximate a sum of many Gaussian distributions to model the outlier layer in the mixture by using a simple uniform distribution for outlier likelihood [29]. The outlier likelihood, p_0, is assigned by considering the situation in which the probability of a constraint being an outlier is one half. Taking the scenario where one *IMO* is present, the ownership of a constraint can be expressed as

$$q_{1k} = \frac{m_1 p_1(\vec{\nabla} I_k | \vec{x}_k, \vec{\theta}_1)}{\sum_{j=0}^{1} m_j p_j(\vec{\nabla} I_k | \vec{x}_k, \vec{\theta}_j)}, \tag{4.29}$$

wherein the mixture proportions are defined to sum to 1, so that given an inlier mixture proportion, m_1, the corresponding outlier proportion is: $m_0 = 1 - m_1$.

$$\frac{1}{2} = \frac{m_1' p_1(\vec{\nabla} I_k | \vec{x}_k, \vec{\theta}_1)}{(1 - m_1')p_0 + m_1' p_1(\vec{\nabla} I_k | \vec{x}_k, \vec{\theta}_1)} \tag{4.30}$$

$$p_0 = \frac{m_1'}{(1 - m_1')} p_1(\vec{\nabla} I_k | \vec{x}_k, \vec{\theta}_1) \tag{4.31}$$

Now assuming this outlier ownership probability is assigned to constraints that occur ρ standard deviations from the mean, we obtain an expression for the outlier likelihood in terms of the expected inlier mixture, m_1', standard deviation, σ_v and a 'confidence' constant, ρ:

$$p_0 = \frac{m_1'}{(1 - m_1')\sqrt{2\pi}\sigma_v} \exp\left(-\frac{\rho^2}{2}\right) \tag{4.32}$$

The constant ρ indicates the distance from the mean (in units of the standard deviation, σ_v) where there remains high confidence that constraints are generated by the *IMO* of interest, but have been mildly corrupted by noise. Jepson and Black [29] employ values of $m_1' = 0.9$ and $\rho = 2.5$ to estimate this likelihood, and our experimental results confirm the validity of these values.

4.1.6 Motion Model Selection

The selection of which motion model to use intuitively depends on the content of the video sequence being analyzed. For the *MASC* system, we assume that the affine motion model is an effective parameterization by the large class of *IMO* motion that can be represented by this model. It is also reasonable to assume that typically the distance of *IMO*s from the camera is significantly greater than the comparative depth of *IMO*s.

In practical terms, the typical frame size of test sequences used with the *MASC* system, 320×240 pixels, makes estimating motion parameters for higher order models very difficult. Constant (translational) motion parameter estimates cannot be cannot be reliably calculated for image patches smaller than 30×30 pixels, and for affine motion models, the image patch size should be larger still, approximately 50×50 [8]. Note that these dimensions refer to the

approximate *IMO* sizes, not simply the image sizes, as the motion constraints of interest are only generated by the *IMO*. The *MASC* motion estimation algorithm employs the affine motion model wherever the frame size exceeds 50×50 pixels, and where more than 200 spatiotemporal gradient constraints are available for an *IMO* region. As an alternative, the constant motion model is employed when the frame size exceeds 30×30 pixels, and where more than 50 spatiotemporal gradient constraints are available. The estimation process dynamically selects the appropriate model depending on the available constraint set and frame pair.

Functionally, the affine and constant motion models may easily be replaced by higher order parametric models such as the projective motion model, at the cost of stability and complexity. As higher order motion models require a proportionally larger number of constraints to generate stable motion estimates, the conditions that determine transitions between the various possible motion models in use must be defined. However, making such an implementation practical would require using larger image sizes, greatly increasing the computational demands of the *MASC* system.

It should be noted that the motion segmentation unit does not take spatial distribution into account, classifying only on the basis of the spatiotemporal gradient constraints at any given point. As a result, noise in disparate regions of the image may cause constraints to be included in an otherwise compact cluster, and two *IMO*s with the same motion properties will be classified as one. Median filtering is applied to remove isolated noise, and in practice this is quite effective. *IMO* segmentation and high noise regions must be dealt with using spatial segmentation techniques described in the following Chapters.

4.1.7 Motion Variance Selection

With all the above components in place, robust motion estimation at a single scale may be performed by generating a set of initial motion parameters, and then by performing successive iterations of the EM procedure described above. A key consideration, already discussed, is the choice of σ_v, which we have already pointed out should reasonably be in the range $[0.1, 0.5]$.

Setting σ_v to a larger value permits the algorithm to incorporate a larger set of data points into its optimization improving its ability to recover from a poor initial motion parameter setting. However, this also forces the algorithm to include a large range of imprecise data points, causing a correspondingly larger error in the final estimated motion parameters. Conversely, using a smaller σ_v should improve the quality of the parameter estimates, but only if the initial parameter estimates are already very close to true maximum, as there is correspondingly very little tolerance to noise.

The EM algorithm's iterative nature suggests an alternative means of managing σ_v selection, through a simulated annealing approach [28]. This approach suggests setting σ_v at a large value during initialization, and then gradually decreasing it after each subsequent iteration, down to a minimum σ_v value. The *MASC* algorithm uses an initial value of $\sigma_v^i = 0.5$, and a final value of $\sigma_v^f = 0.2$. This process ensures large noise and initial value tolerance at the beginning of EM optimization, while the final smaller σ_v value refines the final motion parameter estimates about the correct clean data points.

4.2 Multiscale Motion Estimation

Multiscale methods are required by any gradient-based motion estimation technique in order to reliably recover large displacements in a video sequence [23, 6, 8]. This can be attributed to the unreliability and instability of temporal gradient estimates for large displacements (see Section 4.1.1). Large displacements can be estimated using gradient-based methods by sub-sampling the video sequence, and recovering the large displacements at a smaller image scale. To do so, we construct an image pyramid of the frame pair under consideration, where each level of the pyramid represents a given scale of an image frame, beginning with the original (largest) scale, with level index l_1, and ending with the smallest subsampled scale, index l_{max}. An image pyramid is shown in Figure 4.7.

The pyramid is created beginning with the original image at level l_1, applying a Gaussian

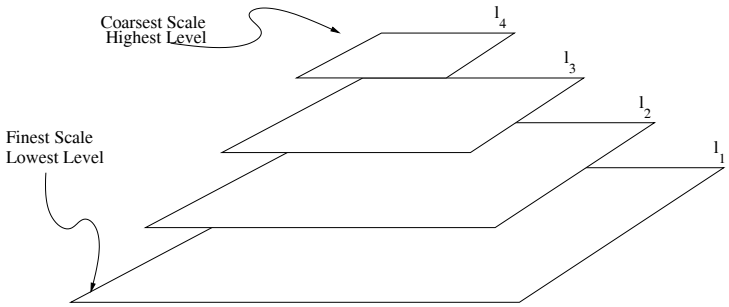


Figure 4.7: The image pyramid is made up of subsampled versions of the original image. The original image forms the lowest level, or scale, of the pyramid, and the smallest subsampled version is referred to as the highest level.

smoothing filter and then subsampling by a factor of 2, so that at level l_2 the image has width and height dimensions half that of level l_1. This procedure is recursively applied to create the higher levels of the pyramid, with the highest level, l_{max}, made up of an image at least 30×30 pixels, to allow for stable single scale motion estimation (see Section 4.1.6). To perform the multiscale motion estimation, we first create image pyramids for two consecutive frames, I_1 and I_2, from the sequence. We can now perform single scale motion estimation at each scale, beginning at highest level of the pyramid, l_{max} to estimate the largest displacements first.

The single scale motion estimation procedure allows us to recover a single valid motion at a given level, attributing any other motion patterns to the outlier class. Assuming the motion parameters, $\vec{\theta}$, are estimated for a coherent *IMO* at level l_q, the motion vectors are projected onto the next lower level, l_{q-1}, and I_1 at level l_{q-1} is warped according to the projected motion of the *IMO*, producing $I_1^w(\vec{x}) = I_1(\vec{x} - \hat{\vec{v}}(\vec{x}))$. As a result of this pre-warping, the displacement of the *IMO* of interest is significantly reduced between frames I_1^w and I_2 at level l_{q-1}, in order to reduce the *IMO* displacement magnitude at level l_{q-1} to a size below two pixels. Assuming the motion estimate at level l_q is accurate to less than one pixel displacement, the residual displacement at level l_{q-1} is within reasonable estimation scale, less than two pixels displacement. This procedure is recursively applied between each overlapping pair of image

scales, from level l_{max} to level l_1, continuously refining the motion estimate. The resulting motion estimate is accurate at the lowest, original image scale, but able to capture displacements as large as $2^{l_{max}}$ pixels.

The recovered motion parameters at level l_q must be rescaled before they can be used to warp I_1 at level l_{q-1}, as a single pixel displacement at level l_q corresponds to two pixels displacement at level l_{q-1}, We must identify the corresponding motion parameter scaling for the affine motion model parameters employed by the *MASC* system. At level l_q, a point $\vec{x}_1^q$ is displaced under affine transformation to

$$\vec{x}_2^q = \mathbf{A}\vec{x}_1^q + \vec{b}\,, \tag{4.33}$$

where $\mathbf{A}$ is a 2×2 matrix, and $\vec{b}$ is a two-dimensional translation vector. Assuming a coordinate origin at the centre of the frame, the corresponding points at level l_{q-1} are $\vec{x}_1^{q-1}$ and $\vec{x}_2^{q-1}$, whose coordinates can be related to the upper level point coordinates by a scaling factor α (set to two for the *MASC* system), so that $\vec{x}_1^q = \alpha\vec{x}_1^{q-1}$, and $\vec{x}_2^q = \alpha\vec{x}_2^{q-1}$. We can express the equivalent affine transformation at level l_{q-1} by observing that

$$\begin{aligned} \vec{x}_2^q &= \alpha\vec{x}_2^{q-1} && (4.34)\\ &= \mathbf{A}(\alpha\vec{x}_1^q) + \alpha\vec{b} && (4.35)\\ &= \mathbf{A}\vec{x}_1^{q-1} + \alpha\vec{b}\,. && (4.36) \end{aligned}$$

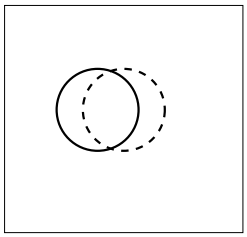
So an affine transformation can be projected from level l_q to level l_{q-1} by preserving the matrix $\mathbf{A}$ unchanged, and scaling the translation vector b by the appropriate inter-level scale factor α.

The pre-warping process is illustrated by Figure 4.8, in which we attempt to recover the displacement of the circle using a multiscale approach with two levels, with the actual displacement of the circle at the lowest level l_1 shown in Figure 4.8(b). The circle's position in frame 1 is shown by the solid circle, and its position in frame 2 is shown by the dashed circle. Assuming a circle displacement of magnitude $u_1 = 3$ at level l_1, we subsample the frame pair to generate a half-size frame pair at level l_2, shown in Figure 4.8(a). At this level, the circle

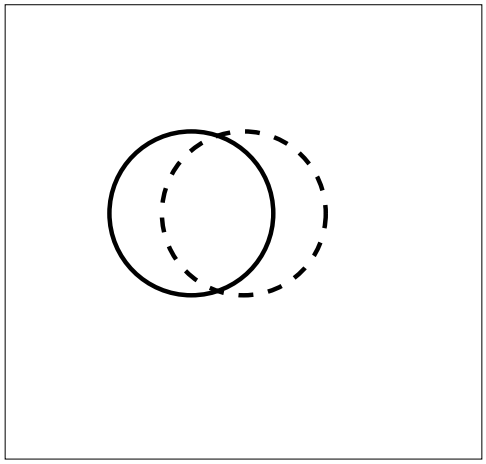
displacement is halved in magnitude to $u_2 = 1.5$, facilitating displacement estimation, as the maximum displacement size we are able to estimate a single scale is 2 pixels. The recovered displacement at level l_2 is used to warp the circle's initial position at level l_1, so that the circle's displacement at this level becomes considerably smaller, shown in Figure 4.8(c). As the displacement estimate at level l_2 is typically accurate to well within one pixel, the residual displacement at level l_1 shown in Figure 4.8(c) is well within the range of estimation. Summing the projected warp displacement and residual displacement at level l_1 yields the true displacement of the circle at that level. This pre-warping and residual motion estimation and summation process is performed recursively for q pyramid levels, so that the system is able to accommodate large displacements, provided the top level of the image pyramid contains a sufficient number of motion constraints for motion estimation.

Notably, a given I_1^w image pyramid corresponds to the hierarchically detected motion of only a single *IMO*, as the lower levels of the image pyramid are warped by the detected motion parameters of the target *IMO*. As a result, a unique I_1^w image pyramid is required for each distinct *IMO* under analysis.

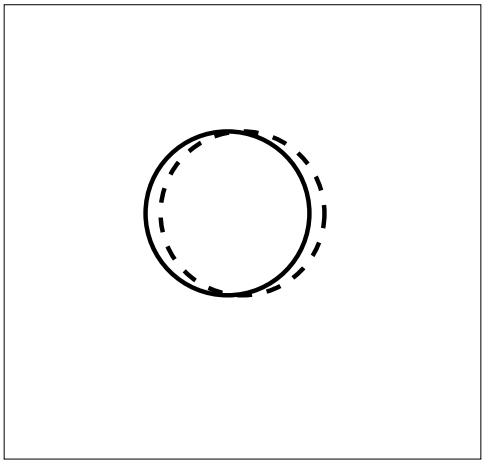
The hierarchical motion parameter estimation process greatly affects the single scale motion parameter initialization process, as for all but the highest level of the Gaussian pyramid, *IMO* motion can be assumed to be close to zero, as the primary component of motion should already be estimated at a higher scale, and removed at the current scale through warping. As a result, for all but the highest scale of the pyramid, initial single scale motion parameters are set to random values around the zero displacement range. At the highest level, the range of recovered motion parameters is still restricted by the thresholding of motion constraints to limit the estimation of pixel displacements to 2 pixels at a given scale. For the test sequences used (all consisting of no more than three *IMO*s at any time), it is sufficient to initialize the motion parameters of a tracking process to the (unweighted) least squares constant motion model optimum for the entire constraint set.



(a) Original circle positions at level l_2



(b) Original circle positions at level l_1



(c) Pre-warped frame 1 circle position (solid line) and frame 2 circle position (dashed line) at level l_1

Figure 4.8: Pre-warping of the image pyramid is illustrated by the displacement of a circle at multiple scales, with the circle position in frame 1 illustrated by the solid circle, and its position in frame 2 illustrated by the dashed circle.

4.3 Multiple *IMO* Segmentation

The motion estimation procedures discussed above recover the motion of a single *IMO* in the video sequence. Where multiple *IMO*s are present in the sequence, a technique known as dominant *IMO* recovery [8, 25] is applied to recover motion estimates for all *IMO*s in the scene. Taking the case of single scale motion estimation, mixture model motion estimation may be applied assuming only a single *IMO*'s presence in the scene. As a result, the robust estimation procedure recovers the motion parameters of the dominant single *IMO* present in the scene, assigning the ownership of constraints generated by other *IMO*s to the outlier layer of the mixture.

The constraints assigned to the target *IMO* may then be removed from further consideration, and the single *IMO* motion estimation process then re-applied to the outlier-owned constraints, to recover another coherent set of *IMO* motion constraints. The recursive application of this process recovers a set of motion parameters for all *IMO*s in the scene, and the process terminates when the population of motion constraints under consideration becomes negligible, or if no coherent motion can be recovered.

Chapter 5

Motion-Based Intensity Constraint Classification

The motion segmentation procedure described in the previous chapter generates a coarse region segmentation through its classification of motion constraints, which are present wherever valid spatiotemporal gradient values exist. Binary motion constraint maps, such as those shown in Figure 4.4 illustrate how sparse the results are, when relying solely upon motion-based constraint classification. Despite providing a reasonable indication of where *IMO*s may be, the requirements of the *MASC* system encourage the development of an improved segmentation technique that can provide denser region segmentation maps for each *IMO*.

The motion-based intensity constraints introduced here exploit the BCC explicitly, and are essentially an extension to the standard motion constraint classification techniques discussed in the previous chapter. By improving the density of region segmentation estimates, the initialization of active contours about *IMO*s is improved, and the region-based active contour forces described in Section 7.4 have greater impact on *MASC* segmentation performance.

5.1 Motion-based Intensity Constraints

While the motion estimation process provides a coarse and sparse region segmentation map of the *IMO*'s in the scene, it also provides the estimated motion parameters of each *IMO*. We further exploit the BCC assumption that an *IMO*'s pixel intensities remain constant over short periods of time, so that the independent motion of an *IMO* can be used as a means to identify the pixels in the image that belong to the *IMO*.

For two consecutive frames, I_1 and I_2, the estimated motion for *IMO* n between the two frames is used to warp each pixel in the first frame, forming a warped frame, I_{1n}. The *IMO*'s position in I_{1n} is expected to be identical to it's actual position in I_2, notwithstanding any non-rigid deformations of the *IMO*. By the BCC, the intensity difference image $I_n^d = |I_2 - I_{1n}|$ should ideally be zero at any points on the target *IMO*, as these regions are assumed to maintain constant intensity. On the other hand, some points that are not part of the *IMO* can be expected to have a large value in I_d^n, as the warp would not correspond to their motion. The large class of exceptions to this hypothesis are dealt with below.

A Gaussian metric is used to compare the warped intensity differences so that for any given pixel,

$$L^n(x,y) = \mathcal{G}(I_n^d(x,y); 0, \sigma_{ccd}) \ , \tag{5.1}$$

with the Gaussian likelihood function $\mathcal{G}$ defined as

$$\mathcal{G}(I_n^d(x,y); 0, \sigma_{ccd}) \ = \ \frac{1}{\sqrt{2\pi}\sigma_{ccd}} exp\left(-\frac{(I_n^d(x,y))^2}{2\sigma_{ccd}^2}\right) \ . \tag{5.2}$$

Equation 5.2 compares the intensity difference of a single point's distinct images in two consecutive frames. The σ_{ccd} factor thus aims to model the sensor noise that might be present in the video recording device, and is conservatively set to a value of $\sigma_{ccd} = 5$. The likelihood function, $\mathcal{G}$, maps points in I_n^d with zero or near-zero values to have relatively high values, as they can be said to have a high likelihood of belonging to the target *IMO*, n. Points with comparatively high I_n^d values are mapped to low likelihood values with similar justification.

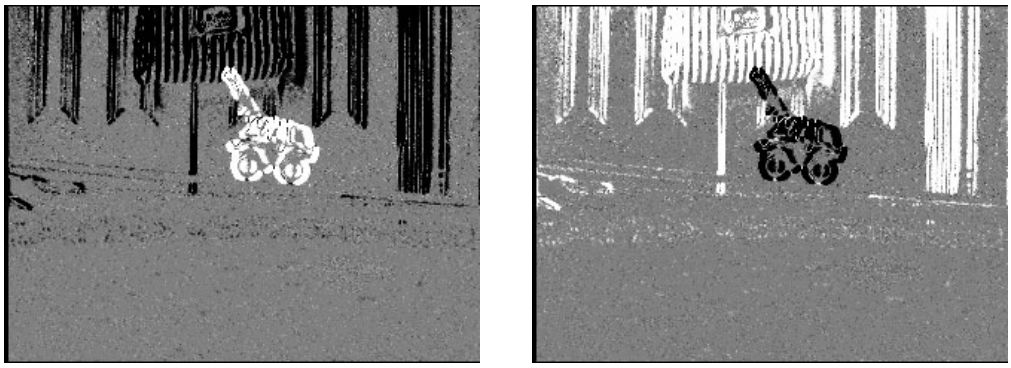
It is trivial to envision a common scenario where points with near-zero values in I_n^d may not correspond to *IMO* pixels. For example points in regions of uniform intensity would have near-zero I_n^d values for a large range of motion warps, as the intensity of the nearby neighbors of such points all share near-identical intensity as the point itself. To account for such occurrences, we formulate the probability that a point is part of an *IMO* by combining the likelihood values for that point over all *IMO*s in the scene:

$$P_n(x,y) \quad = \quad \frac{L_n(x,y)}{\sum_{j=1}^{N} L_j(x,y)} \tag{5.3}$$

This expression combines a set of likelihoods for a single data point to generate a single probability of 'ownership' in a very similar sense to that of the EM algorithm, described in Section 1.4. We therefore also adopt the term 'ownership' to refer to these values, which provide a strong indication of which *IMO* motions (and hence which *IMO*s) best account for the warped intensity difference images, $\{I_n^d : n = 1, \ldots, N\}$.

The complete set of ownerships therefore consist of pixels that have a high probability of belonging to a given *IMO*, and other pixels that are ambiguous in that respect. In the simplest case where a single *IMO* moves against a stationary background, ambiguous pixels are present mainly in uniform regions in the frames, and each *IMO* (the background is a unique *IMO*, and the foreground *IMO* generate a total of two *IMO*s in the scene) would claim equal ownership over these pixels. This case is shown in Figure 5.1 using our reformulated motion segmentation maps that use motion-based intensity constraints instead of pure motion constraints. The bright areas represent areas of high ownership probability, while the dark areas represent areas of low ownership probability. This can be seen clearly in the figure, where textured areas that belong to the target *IMO* are shown in white, while textured areas that clearly do not belong to the target *IMO*, are given as black. Logically, ambiguous areas are shown in both segmentation maps in gray, as the ownership is roughly equal between both *IMO*s in the scene.

The key advantage of this approach is in comparison to the purely motion-based segmentation, which only provides region classification where valid spatiotemporal gradients are present. This process is able to exploit far subtler texture and edge data to generate a denser



(a) *IMO* Layer Ownership Map (b) Background Layer Ownership Map

Figure 5.1: Motion-based intensity constraints are used to generate improved motion segmentation maps for Tow Truck 1 Sequence Frame 50

classification map. As a result, the motion-based intensity constraints produce an improved segmentation map, especially in sequences where lightly textured areas are present. In absolutely uniform regions, there is no advantage in using the motion-based intensity constraints, however, performance is no worse. The improved performance of motion-based intensity constraints is plainly illustrated in the tow truck segmentation maps shown in Figure 5.2, where the pure motion segmentation map is compared to the motion-based intensity constraint classification. The binary ownership maps are generated by thresholding, so that the (white) pixels belonging to the tow truck are those for which its ownership values are highest among all other *IMO* layers in the scene. As shown, the sparsity of motion constraints results in a relatively poor segmentation, when compared to the dense segmentation provided by the motion-based intensity constraints classification. Our reformulated motion-based intensity constraint classification maps are hereafter referred to as motion segmentation maps.

5.2 Multiple *IMO*s with Similar Motion Parameters

A special case arises when two *IMO*s with identical motion parameters are present in a sequence, as they would be identified as only a single *IMO* by the motion segmentation process.

(a) Motion constraint segmentation Tow Truck map

(b) Motion-based intensity constraint classification Tow Truck map

Figure 5.2: Comparison of motion constraint segmentation and motion-based intensity constraint classifciation Maps for Tow Truck 1 Sequence, Frame 33

For active contour initialization, CCA spatial segmentation routines (described in Section 6) are applied to the segmentation maps to ensure that the active contour is only initialized about a single *IMO*. This approach is effective as long as the two *IMO*'s are sufficiently detached in the video sequence, as it is otherwise impossible to segment the *IMO*s from each other on the basis of their motion. At later stages in the sequence, this ambiguity between *IMO*s with similar motion no longer poses a significant problem, as the active contours about each distinct *IMO* remain distinct so long as a reasonable distance between *IMO*s is maintained.

5.3 *MASC* Integration of Motion-based Intensity Constraints

The results generated by the motion-based intensity constraint classification are used in two distinct parts of the *MASC* system. The first use is during the initialization of a new *IMO* in the scene, where the segmentation map for a single *IMO* is used to initialize the active contour of the *IMO*. In this case, the segmentation map must be reduced to a binary map around which an initial active contour is generated. To do so, a threshold is applied to the segmentation map, so that only those regions in the frame with high confidence of *IMO* presence are isolated.

The CCA-based spatial segmentation routine is then applied to this binary map to isolate the single primary compact region to generate a conservative estimate of the *IMO*'s location. This final compact, binary segmentation map is then passed on to the active contour initialization routines.

The second use of motion segmentation map results is in implementing the active contour normal forces (see Section 7.4), an application for which spatial segmentation is not necessary, as the initial active contour applies an implicit spatial segmentation by only expanding to encapsulate well connected *IMO* regions.

Chapter 6

Connected Components Analysis

The motion-based intensity constraint classification process segments the frame into regions of coherent motion, however this does not account for the fact that multiple *IMO*'s may be present within a single region of coherent motion. This common situation must be addressed to prevent the *MASC* system from classifying such a group of distinct *IMO*s as a single *IMO*. The *C*onnected *C*omponents *A*nalysis (CCA) procedure performs spatial segmentation of a motion coherent region to resolve any spatially disjoint regions if any such regions exist. Spatially separate *IMO*s moving with very similar motion are thus segmented and dealt with as distinct entities. The following discussion demonstrates the application of CCA to a representative set of test data, illustrating the application and effectiveness of the CCA procedure to the type of data sets present in the segmentation framework.

The CCA procedure comprises two primary steps:

1. Determine the segmentation between spatially coherent areas in the motion coherent region ;

2. Generate and assign distinct labels to each of these distinct regions.

Furthermore, given a parameterized minimum *IMO* size setting, any segmented *IMO*s that are smaller than the minimum size parameter may be eliminated, ensuring the exclusion of

extraneous noise. This minimum size parameter relates directly to the minimum number of motion constraints required to perform effective motion segmentation (Section 4.2), imposing the limitation that very small objects cannot be reliably segmented using motion. The CCA procedure results in labelling of every element in a motion coherent region. The challenge of doing so in a logical, spatially coherent manner is addressed, to recover one or more plausible *IMO* areas within the motion coherent region.

6.1 Connected Components Labeling

Connected components analysis techniques are generally used on binary images to assign a unique label to every connected area in the image [44]. The binary image example in Figure 6.1(a) shows a simple binary image with two connected regions that we use to demonstrate the labeling algorithm. The 'Classical' connected components algorithm takes two passes over the binary image while assigning the values of a label map, with the same dimensions as the binary image. We begin at the top-left corner of the image, scanning across each row. We assign an ascending numerical label to any unlabeled pixel, and propagate pixel labels to any connected neighbors right, below, or below-right of the pixel under examination , Figure 6.1(b). If any labeled pixel is found to have connected neighbors with a different label, the equivalence of the two label values is noted in the Equivalence Table. After the first pass is completed, the Equivalence Table is examined and modified to ensure that all equivalent labels reference the minimum label value in their set, Figure 6.1(c). The second pass of the algorithm comprises a similar examination of the label map, substituting labels with their minimum equivalence label from the Equivalence Table. The resulting label map assigns to unique labels for each connected component to each pixel (Figure 6.1(d)).

For its application in the *MASC* system, we extend the notion of connectedness from an immediate connected neighbor to a neighbor within some radial distance. This is equivalent to the application of typical connected components labeling to a dilated binary image, using a

1	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0
1	1	1	0	1	1	0	0
1	1	1	1	1	0	0	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	1	1	0
1	1	1	0	0	0	0	0

(a) Original Binary Image

1	1	1	0	0	0	0	0
1	1	1	0	2	2	0	0
1	1	1	0	2	2	0	0
1	1	1	1	1	0	0	0
1	1	1	0	0	0	3	0
1	1	1	0	0	0	3	0
1	1	1	0	0	4	3	0
1	1	1	0	0	0	0	0

(b) First Pass of CCA

Label	Equivalence Label
1	1
2	1
3	3
4	3

(c) Equivalence Table

1	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0
1	1	1	0	1	1	0	0
1	1	1	1	1	0	0	0
1	1	1	0	0	0	3	0
1	1	1	0	0	0	3	0
1	1	1	0	0	3	3	0
1	1	1	0	0	0	0	0

(d) CCA Final Result

Figure 6.1: CCA example: When applied to the original binary image (*a*), the first pass of CCA scans down each column, assigning labels to connected components (*b*). The first pass cannot immediately identify all connected components correctly, but notes their connectedness in an equivalence table (*c*). A second pass through the pre-processed data replaces all labels with their equivalence labels.

disk structuring element. We describe this extension by introducing affinity matrices, and then by describing the application of the classical connected components algorithm to operate upon affinity matrices.

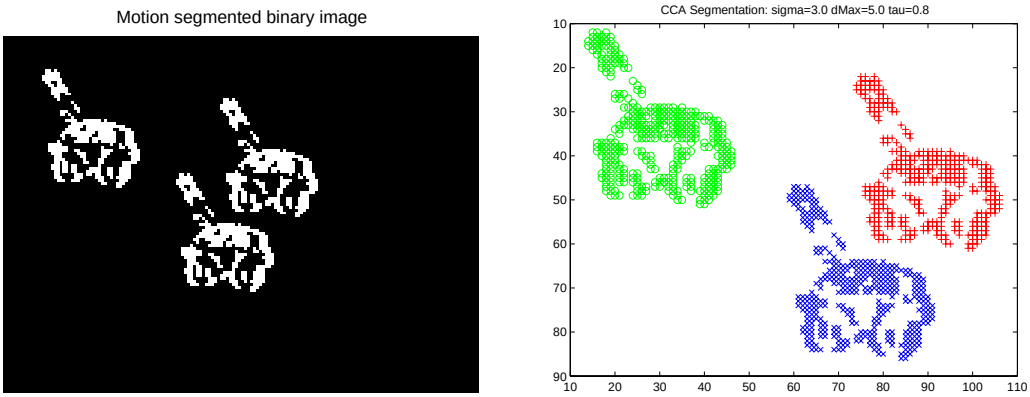
6.2 Affinity Matrix Properties

The basis of our CCA segmentation is an inter-point affinity matrix, whose values are based upon the distances between each point in the motion segmented regions. An example is shown in the left binary image of Figure 6.2, where three identical trucks have been segmented as a single motion coherent layer (as they share the same motion), and we wish to identify each truck as a separate entity. The test sequence is challenging and representative of the problems addressed by the segmentation system, in particular, note the proximity of the two trucks in the right of the test image, and also the detachment of the top section of each truck's towing mast. The segmentation therefore aims to distinguish and uniquely label the three trucks present in the test image.

An affinity matrix, W, assigns the affinity between any two motion coherent points i and j (white pixels in the binary image of Figure 6.2) :

$$W_{ij} = \begin{cases} \exp \frac{(\vec{i}-\vec{j})^2}{2\sigma^2} & , |\vec{i}-\vec{j}| < d_{max} \\ 0 & , |\vec{i}-\vec{j}| > d_{max} \end{cases} \tag{6.1}$$

The affinity matrix thus assigns inter-point affinities in the range of $(0, 1]$ for points that are closer than d_{max}, and affinities of 0 for points that are farther than d_{max} from each other. Setting a threshold, τ, 'cuts' in the image may be made wherever the connecting affinity between two areas is below the threshold value, forming multiple distinct regions that may then be labeled uniquely, as shown in Figure 6.2(b). The following section describes the extension to classical CCA for its application to affinity matrices, and it is followed by the process of selection of reasonable values for the variables used in the *MASC* CCA segmentation.



(a) Binary input image.

(b) CCA segmentation, with each distinctly segmented *IMO* assigned a unique symbol.

Figure 6.2: CCA Segmentation Example.

6.3 Connected Components Labeling for Affinity Matrices

The affinity matrix structure proposed in the previous section assigns non-zero affinities for points separated by less than d_{max} pixels, and uses a Gaussian metric with standard deviation σ to assign affinities. Given such a matrix and an affinity threshold, τ, we can now apply CCA to determine the connected components with intra-component affinities greater than τ, in the target binary image.

The modified CCA algorithm is very closely related to the classical CCA algorithm, comprising two passes across the columns of the affinity matrix, as each column corresponds to one pixel in the binary image. Similarly, the pixel labels are maintained in a column vector (the *label vector*) with the same number of elements as one row or column of the affinity matrix. We begin with the leftmost column of the affinity matrix and assign a new label to that pixel in the label vector. We then examine subsequent columns in the affinity matrix in order, comparing the target pixel's affinities for previously examined pixels against the τ threshold. This involves examining the target column from the top entry, up to but not including the entry on the matrix diagonal (corresponding to the pixel itself).

- If the τ threshold is exceeded by any affinity value in this set and the target pixel is

unlabeled, we can propagate the label from the connected pixel to the target pixel.

- If the τ threshold is exceeded by an affinity value in this set and the target pixel has already been labeled, we note the equivalence of the two labels in an Equivalence Table.

- If no affinity in this set exceeds the τ threshold, a new label is generated for the target pixel.

Once all columns in the affinity matrix have been examined, the Equivalence Table is modified again to ensure that all equivalent labels reference the minimum label value in their set. Like the Classical algorithm, the second pass requires the examination of the label vector, substituting labels with their minimum equivalence label from the Equivalence Table. The resulting label vector assigns unique labels for each connected component to each pixel in the affinity matrix, which can be reshaped to produce the segmented binary image.

6.4 Variable Selection

The selection of σ, d_{max} and τ values rely upon experiments which aim to generate reliable segmentations of a wide range of target cases, and the principle of their selection is presented here. The maximum distance parameter, d_{max}, determines the inter-point distance beyond which affinity is set to zero, and thus the distance at which *IMO*s are 'obviously' distinct. The main reason for using this variable however, is to improve the sparsity of the affinity matrix, W, speeding up computation. Our selection therefore seeks to find the minimum value of d_{max} for which the target image is not over-segmented, as shown in the top right image of Figure 6.3. Increasing d_{max} ensures that the τ value is responsible for determining the segmentation, as the d_{max} value simply aims to eliminate redundant affinity values. As shown in Figure 6.3, the segmentation remains stable and reasonable when d_{max} is set to values above 5 units, and this value is intuitively reasonable. Note that although the results are only shown for fixed values of τ and σ, it should be noted that an appropriate selection for d_{max} should be reasonable for

segmentation using any τ and σ values, as long as the chosen d_{max} value is sufficiently high.

The σ value is used to determine the strength of inter-point relationships, and therefore directly affects the choice of threshold, τ. The two must thus be chosen together, using the histogram of affinities as a guide. We aim to choose σ such that two tightly clustered sets of affinities, one near zero and another near one, are apparent in the histogram. In doing so, the value of τ can be set near the bottom of the cluster near one, generating a strong segmentation. Figure 6.4 indicates the effect of varying the τ parameter with σ fixed, while Figure 6.5 demonstrates the interaction between the two variables in generating equivalent segmentation. While it is intuitively desirable to select the minimum σ value that generates reasonable segmentation, it is similarly desirable to select the maximum possible τ value.

As a compromise between the criteria discussed above, the working system uses the following CCA segmentation variable values: $d_{max} = 5$, $\sigma = 3$ and $\tau = 0.8$. Segmentation results for various test scenarios are presented in the next section to illustrate the performance of the CCA segmentation procedure.

6.5 CCA Results Analysis

In this section, the performance of CCA segmentation is examined, summarized in Figure 6.6, where the main problem of under-segmentation is illustrated. The CCA segmentation makes cuts on the basis of point-to-point affinity, and thus as the moving (lower) truck makes any connection to either of the stationary trucks, the connected trucks are classified as a single *IMO*. This situation, shown in the top and bottom rows of Figure 6.6 is an inherent property of the CCA algorithm, which does not take into account the intra-cluster affinities when making cuts. This failure however, is quite an acceptable property of the segmentation algorithm, and quite in-line with our assumption of spatial coherence, in which spatially coherent regions moving with identical motion are likely to belong to a single *IMO*. This assumption is upheld by the segmentation results demonstrated here, and the correct segmentation shown in the center row

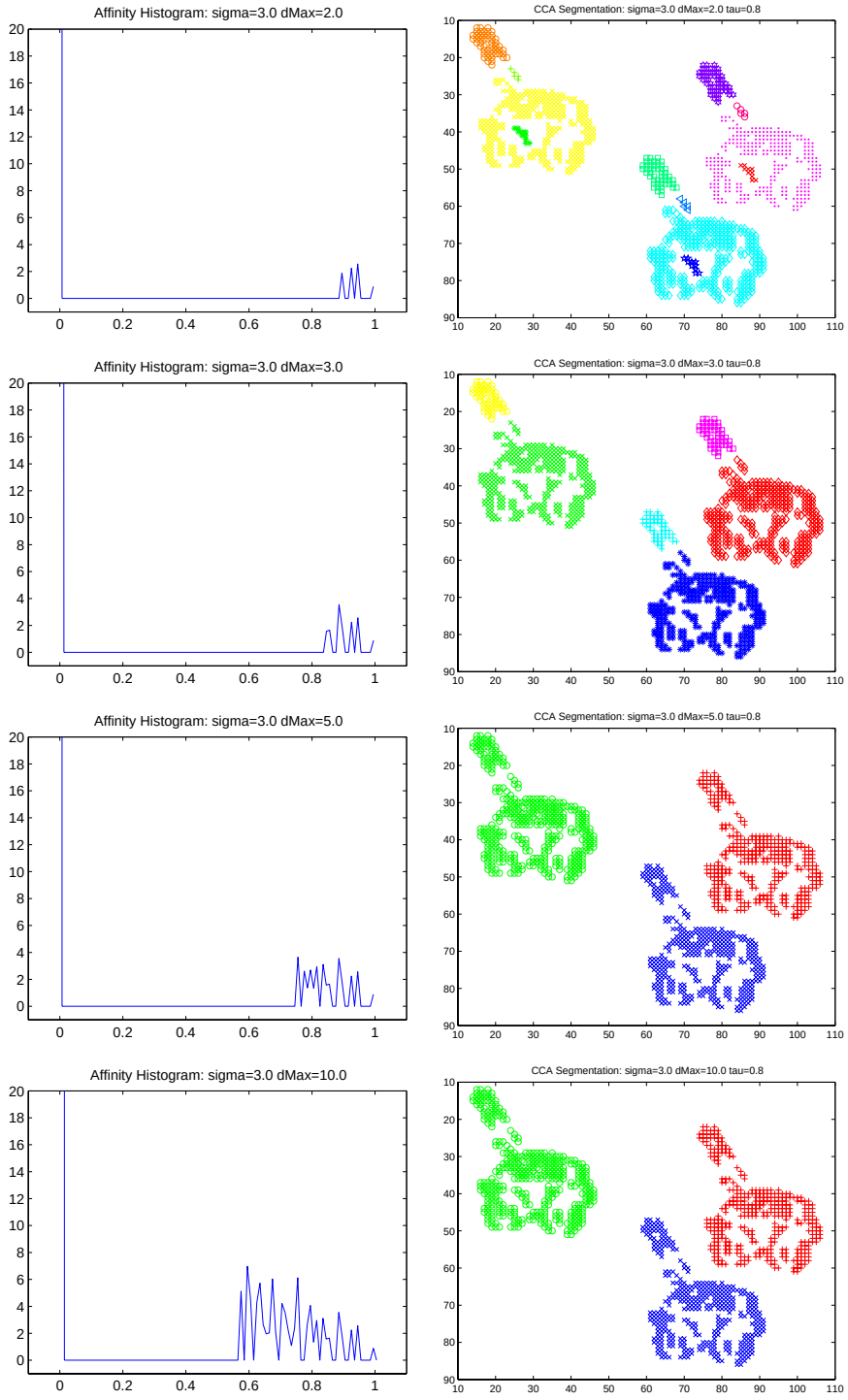


Figure 6.3: d_{max} Variable Selection Results. The histograms in the left column illustrate the distribution of affinities, which spread as the d_{max} threshold is raised. The corresponding CCA segmentation is shown in the right column.

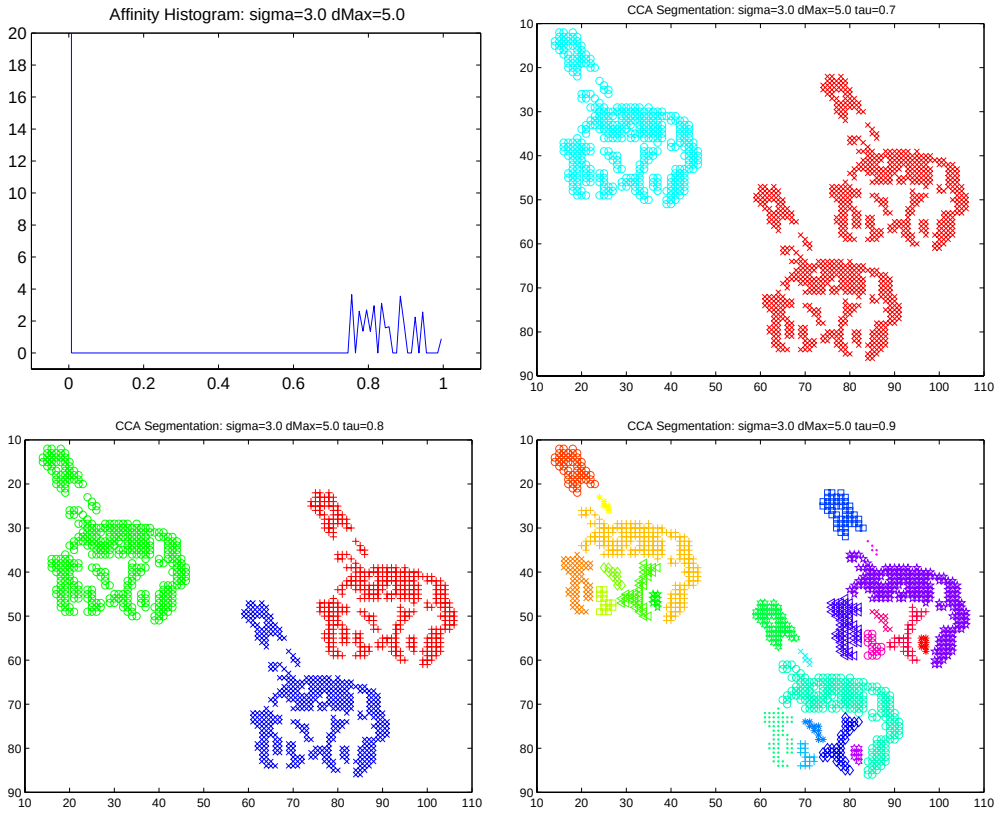


Figure 6.4: τ Variable Selection Results. Variation of the τ segmentation threshold illustrates the transition from under-segmentation to over-segmentation in the right column, with the correct segmentation shown bottom left.

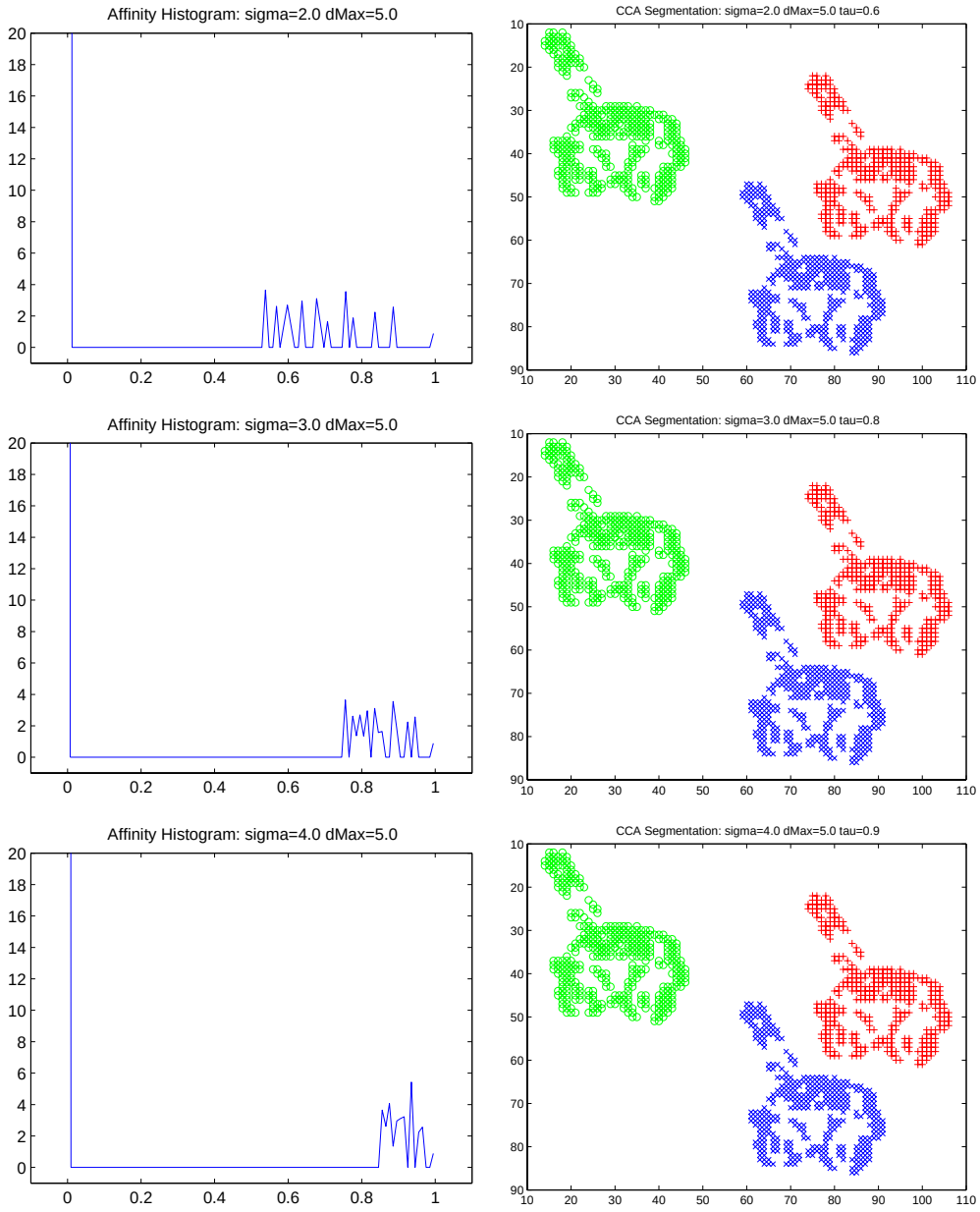


Figure 6.5: σ Variable Selection Results. The σ and τ variables work in tandem to generate the desired segmentation threshold, see text for selection criteria.

of Figure 6.6, which also illustrates that the segmentation degrades smoothly.

Although noise filtering is applied to motion segmented binary regions before spatial segmentation is performed, Figure 6.7 is presented to demonstrate CCA segmentation performance under noisy conditions, in particular, the non-filtered version of the motion segmented binary map. The results shown in the right column of the image demonstrate similar performance to that in the non-noisy case, with noisy binary pixels classified separately. A final failure case is demonstrated in Figure 6.9, where a motion segmented satellite binary image is shown on the left, and its CCA segmented properties shown. The disconnectedness in the binary image does not allow the correct segmentation to be recovered, however the main central part of the satellite is correctly recovered due to the connectedness of its components.

The CCA segmentation routines effect spatial coherence constraints on motion segmentation data by uniquely labeling spatially distinct regions in a motion segmented binary image. The above results indicate that this need is met, and that the variables chosen for those purposes allow good performance and graceful degradation even under noisy conditions. The coarse *IMO* region isolated by the CCA segmentation is then used to initialize the active contour that binds boundary-based spatial coherence constraints to the motion estimation component of the *MASC* system, as described in the next Chapter.

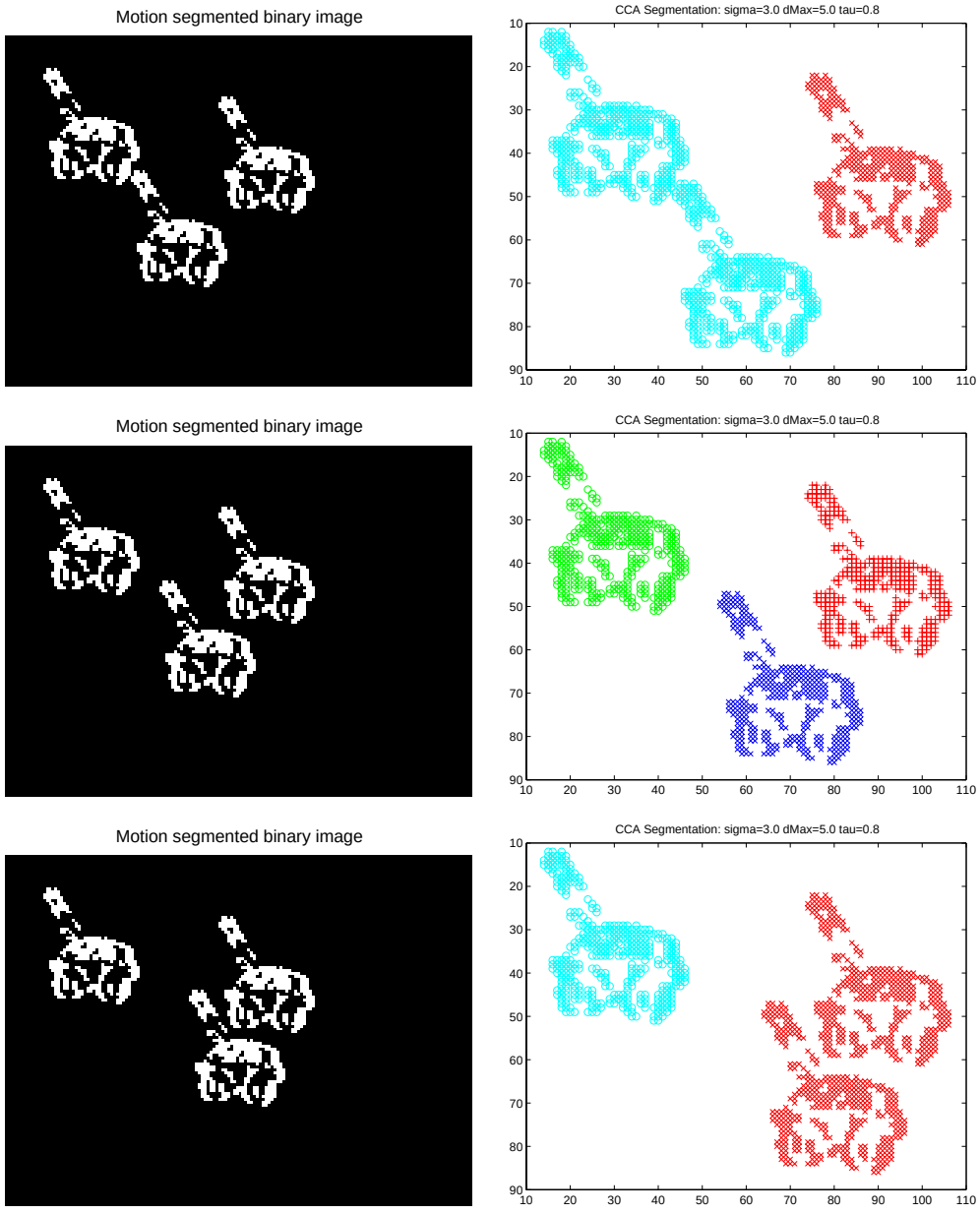


Figure 6.6: 3truck CCA Segmentation. CCA segmentation of the three trucks when the bottom truck translates horizontally to the right, illustrating the behavior of the CCA segmentation, and the way in which *IMO*s are merged and segmented.

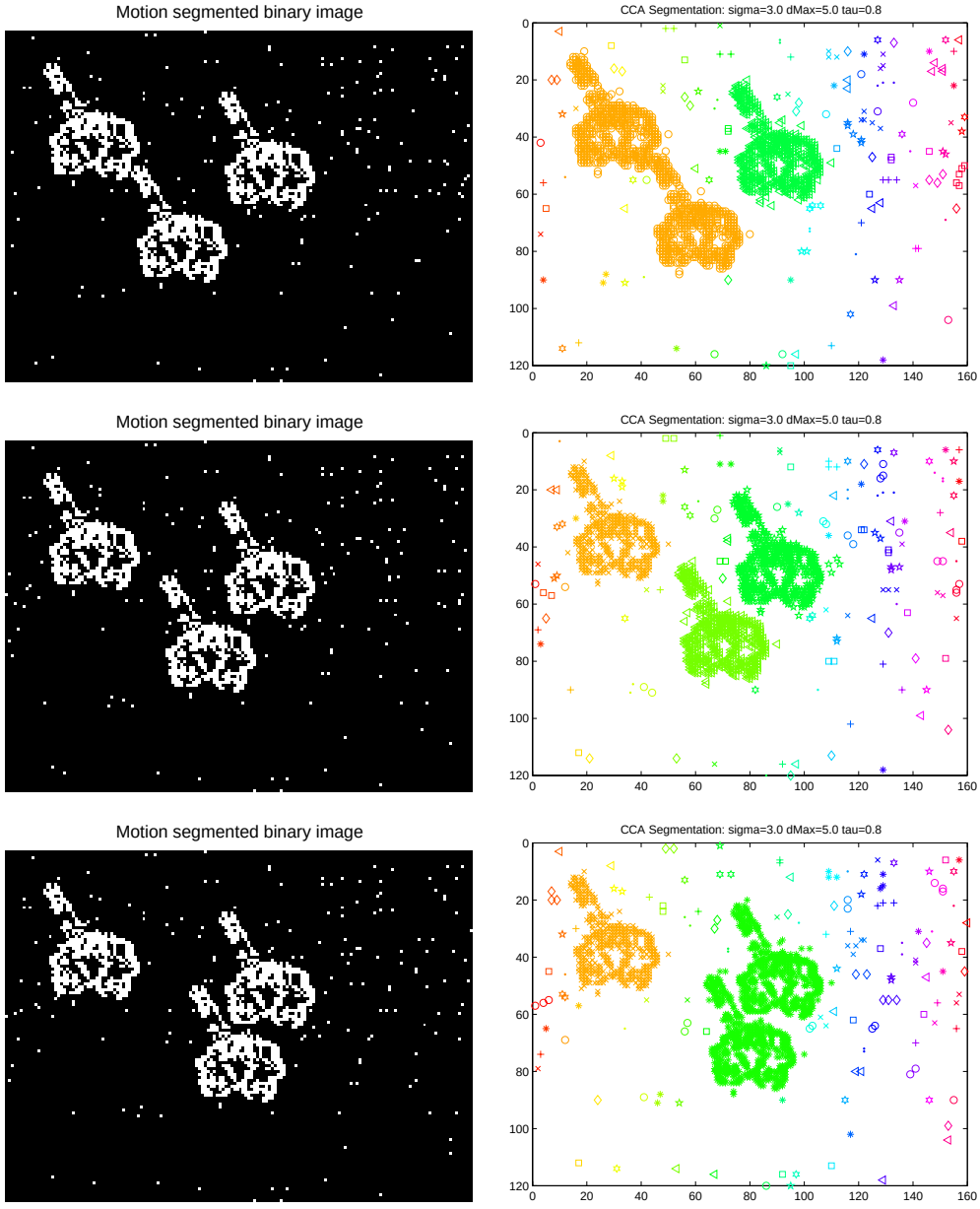


Figure 6.7: 3truckNoisy CCA Segmentation. Demonstration of the behavior of the CCA procedure under noisy conditions, in which the three trucks are correctly segmented and noise pixels are classified separately. Note that in the center row, the trucks are correctly segmented.

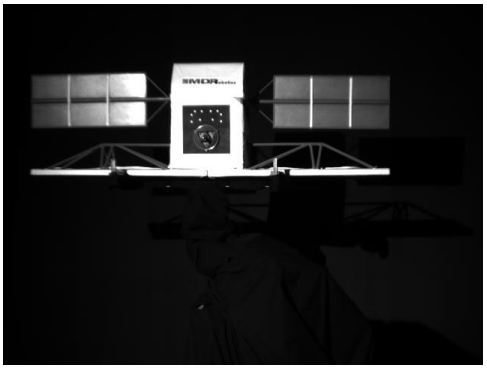


Figure 6.8: Satellite test image (Image courtesy of MD Robotics): A model satellite is suspended by a hidden robotic arm.

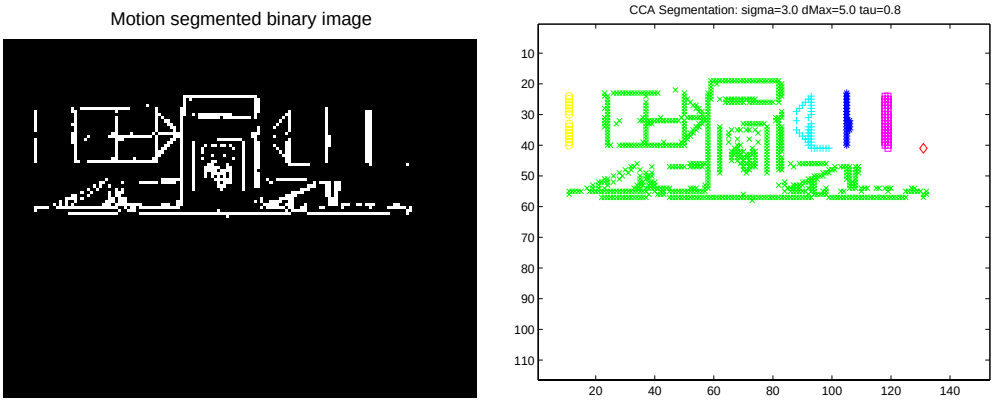


Figure 6.9: Satellite test image CCA Segmentation. The CCA segmentation is unable to coherently label the entire satellite, as the valid spatiotemporal constraints generated by the satellite have large holes within the satellite area.

Chapter 7

Active Contours

Active contour techniques are the basis of boundary recovery and boundary tracking within the *MASC* system. Their aim is to recover the shape of the target *IMO* over time using image edge information and through tight integration with region information from the motion segmentation component.

Our active contour follows the traditional 'snake' formulation [34], but incorporates alternative criteria derived from region-based motion segmentation information. Additional modifications assist the active contour to reliably detect concavities in the target *IMO*. Including region-based constraints requires modification to the active contour formulation, to allow global information from the entire scene to affect the behavior of each active contour control point. The active contour also uses motion segmentation results for its initialization, so that the active contour begins execution near the target *IMO* edges.

Accurate boundary estimates simplify the motion segmentation process over time by providing an increasingly accurate and temporally stable *IMO* boundary. The strong spatial coherence constraint provided by a good boundary estimate shifts the motion segmentation calculation from a global process to a very local, *IMO* centered process. The motion estimation process no longer needs to segment constraints generated by other *IMO*s while estimating the target *IMO*'s motion, as it is now supplied with an estimate of the ideal constraint clustering

window, proposed in Section 1.3. Instead, the target *IMO*'s motion is recovered while removing a relatively small outlier population from consideration. This reduced complexity of the clustering process improves motion estimation performance, which in turn improves segmentation.

The specific active contour technique employed is not critical to the tracking performance of the system as a whole. Alternative active contour implementations that employ level set formulations [10] provide increased topological sensitivity at the expense of some complexity, and these alternative techniques are equally applicable within the proposed segmentation framework. The discussion below evaluates the performance of a subset of active contour techniques that effectively address our primary concerns of boundary recovery: accurate shape recovery and the incorporation of region information.

7.1 Traditional Active Contours

Active contours are used to recover a boundary estimate of *IMO*s in static images, so for the purposes of this research the active contours are assumed to be closed, as they must provide a closed contour about the *IMO* of interest. An active contour consists of an ordered set of control points in a closed loop, and is assumed to be initialized close to the boundary of the target *IMO*. The initialization of the active contour is an important area of discussion, as the traditional contour's optimization process is a purely local function of its individual control point locations, preventing any global information from influencing the behavior of the active contour. This is addressed in the *MASC* system through the addition of normal forces to the traditional active contour formulation, described below.

The traditional active contour formulation [34] ensures that the set of control points formed by the contour meet the following criteria:

- The control points are evenly spaced;
- The control points form a contour with smooth curvature; and

- The contour adheres to image edges.

The active contour's position is optimized by moving each of its individual control points to minimize an energy functional defined on the entire contour. The terms of the energy functional are given in Equation 7.1, representing internal and image-based (external) forces that ensure the contour meets the traditional criteria of continuity (the even spacing of control points), smoothness of curvature and adherence to image edges. The equation expresses the energy of the contour as a function of the index s of control points along the contour, so that minimizing $E(s)$ minimizes the energy functional at each discrete control point over the entire contour.

$$E(s) = \sum_s \left(\underbrace{\alpha E_{continuity}(s) + \beta E_{curvature}(s)}_{\text{internal forces}} + \underbrace{\gamma E_{image}(s)}_{\text{image-based force}} \right) \tag{7.1}$$

The α, β and γ coefficients in the energy functional adjust the relative weighting of each term, so that the three criteria can be balanced according to the priorities of the system. For example, if a smooth contour is more important that strict adherence to image edges, β can be set proportionally higher than γ.

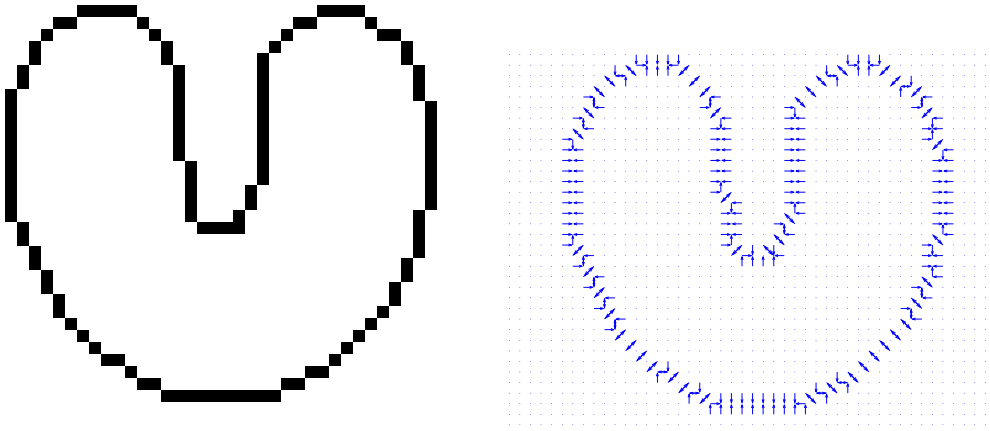
Given an initial control point set, $\vec{p}(s), s = 1...n$, the points can be configured to form a closed contour by assigning the last point the same coordinates as the first point, and making appropriate adjustments in the energy calculations. The continuity term in the energy functional, $E_{continuity}$, can be expressed as

$$E_{continuity}(s) = (\overline{d} - ||\vec{p}(s) - \vec{p}(s-1)||)^2 , \tag{7.2}$$

where $\vec{p}(s)$ represents the vector coordinate of the control point of interest, and $\vec{p}(s-1)$ that of the control point before it. $\overline{d}$ represents the average distance between all the control points that make up the contour, so that the continuity term $E_{continuity}$ maintains even spacing between all points along the entire contour.

The curvature of the contour is approximated using the second derivative of the contour, which we express discretely as

$$E_{curvature}(s) = ||\vec{p}(s-1) - 2\vec{p}(s) + \vec{p}(s+1)||^2 . \tag{7.3}$$



(a) Sample Edge Image (b) Traditional Snake External Image Forces

Figure 7.1: Active Contour Edge Forces

Equation 7.3 approximates the curvature of the contour, so that minimizing this expression with respect to the position of points attempts to keep the contour points as collinear as possible, ensuring smoothness throughout its length.

The traditional image-based term is typically a function of the spatial gradient of the image, $\nabla I_{\vec{x}}$:

$$E_{image}(s) \quad = \quad -||\nabla I_{\vec{x}}(\vec{p}(s))|| \tag{7.4}$$

The E_{image} energy term is lowest at points of high spatial gradient, typically wherever image edges exist, so that the contour attempts to bind itself to edges in the image. The image forces are illustrated in Figure 7.1, where the increasing gradient in areas close to edges attracts control points toward the edges, but only if control points are initially close enough to be affected by these short range forces (which are limited to affecting control points two or three pixels radial distance from the edge).

The active contour is optimized by adjusting the location of each contour control point within a local neighborhood to wherever the overall contour energy is lowest, allowing the contour to meet the three required criteria. The traditional optimization process solves a set of Euler equations in an iterative fashion as proposed in the original active contour research

[34] and summarized in Section 7.7, however the greedy optimization technique provides an intuitive approach that often demonstrates comparable performance [55]. The greedy minimization process operates in a loop upon the contour control points, iteratively moving each control point within a specified neighborhood (typically a square 3×3 or 5×5 area) to wherever in the neighborhood the contour energy functional is minimized. The process is repeated over the length of the contour until the proportion of control points that move in a single iteration drop below a selected threshold or until a maximum number of iterations over the length of the contour have been performed.

Numerous techniques have been developed to improve the performance of the original active contour formulation [34]. Aspects such as computational efficiency [55, 35], shape coherence [56, 12] and flexibility [10] have all been developed and improved. Although based on the traditional formulation, our active contour approach incorporates improvements that address key shortcomings of the traditional approach with respect to the segmentation and tracking goals of this research. The next section discusses the specific shortcomings of traditional active contours that are relevant to their application within the *MASC* framework.

7.2 Traditional Active Contour Shortcomings

The key shortcomings of traditional active contours for this application are the need for good initialization, the inability to descend into *IMO* boundary concavities and a purely local optimization process that does not account for region-based information.

Automatic initialization of the active contour close to the target *IMO*'s boundary must be generated to eliminate the need for a human operator. While the traditional active contour makes no provisions for automatic initialization, the layered motion segmentation and CCA-based spatial segmentation provides a rough estimate of the target *IMO* location and boundary that is used by the active contour initialization process discussed in Section 7.3. The use of automatic initialization procedures also brings about the possibility for non-ideal initialization.

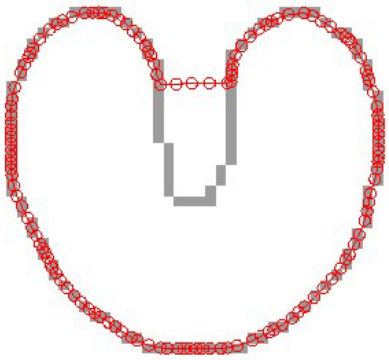


Figure 7.2: Traditional active contour image-based forces are not able to ensure the contour's descent into large concavities due to their limited range.

Poor initialization drastically affects the ability of a traditional active contour to recover a good *IMO* boundary. The *MASC* system's operation upon a sequence of images (as opposed to a single image) provides scope for recovery from poor initialization over time, if the active contour energy functional (Equation 7.1) can be modified to account for region-based *IMO* classification. Normal forces are introduced to the active contour's execution in Section 7.4, addressing this problem.

Beyond initialization, deep concavities in *IMO*s are typically not recovered effectively by traditional active contours, due to the relatively short range of traditional image-based forces, as shown in Figure 7.1. The horizontal image forces generated by the sides of the concavity do not cause the contour to descend into the concavity, and the depth of the concavity prevents forces from the bottom of the concavity from ensuring the descent of the contour. This results in the situation shown in Figure 7.2, where the final contour clearly does not not descend into the concavity. This shortcoming prevents the recovery of reliable *IMO* boundaries for many classes of shapes, and longer range image-based forces are introduced in Section 7.5 to address this issue.

7.3 Active Contour Initialization

The problem of initialization is a key obstacle to any active contour implementation, as traditional active contours are not able to focus attention upon any area beyond their local scope of a few pixels radius about any given control point. This shortcoming can be seen intuitively by examining the active contour energy function of Equation 7.1, where each control point is influenced only by internal forces, and local image gradient. As such, a control point is only attracted to the edges closest to its current location, making the assumption that the entire contour is initialized close to the correct set of target *IMO* edges.

Traditional active contour approaches defer the initialization process to a human-based interface, however the automatic nature of the *MASC* segmentation system makes it crucial that the system not require any such interaction. Motion segmentation information provides a very useful indication of the approximate target *IMO* boundary location, and thus the *MASC* system integrates this data in its active contour initialization process. The example given below demonstrates the way in which motion segmentation information is used to generate an initial guess of the closed contour around the target *IMO*.

For the tow truck sample sequence shown in Figure 7.3, the combined motion estimation and motion-based intensity constraint classification component (referred to as the motion segmentation component in this section) identify the target *IMO* as the white area shown in the motion segmentation map in Figure 7.5(a). It is useful to initialize the active contour around the motion segmented target *IMO* area, that is, the white area in Figure 7.5(a), corresponding to the tow truck. It should be noted however, that active contour techniques are specifically attracted to edges in the target image, specifically those edges about the initial control point locations. As such, there is a strong motivation to initialize control points at the edges that the motion segmentation process indicates are the exterior edges of the target *IMO*.

The edge map of the frame, shown in Figure 7.4, indicates the complete set of image edges generated using a Canny edge detector. Masking the edge map with the high confidence

(white) areas of the motion segmentation map isolates the target *IMO* edges. A convex hull [1] is generated around the isolated *IMO* edges so that all such edges are enveloped within the initial closed contour, Figure 7.6(a). Active contour techniques are then applied to contract the convex hull about the *IMO* edges so that it tightly wraps the motion segmented target *IMO* as shown in Figure 7.6(b). This process is applied using the isolated *IMO* edge set as the sole source of image-edge forces to the initialization active contour process, ignoring any other edges present in the original image. In doing so, the active contour is initialized closely about the target *IMO*.

The motion segmentation process does not necessarily isolate the full set of edges belonging to the target *IMO*, due to the aperture problem preventing the distinction of moving regions where the edges of the moving region are parallel to the direction of motion. However, the resulting initial contour generated usually provides a reasonable automatic initialization. The further requirement that the active contour automatically recover from poor initialization through the incorporation of motion segmentation data goes even further lengths to ensure that the initialization described here is sufficient for our purposes. The next section describes the procedures taken to ensure that even poorly initialized active contours are able to recover a good *IMO* boundary over time within the *MASC* system.

7.4 Region-based Normal Forces

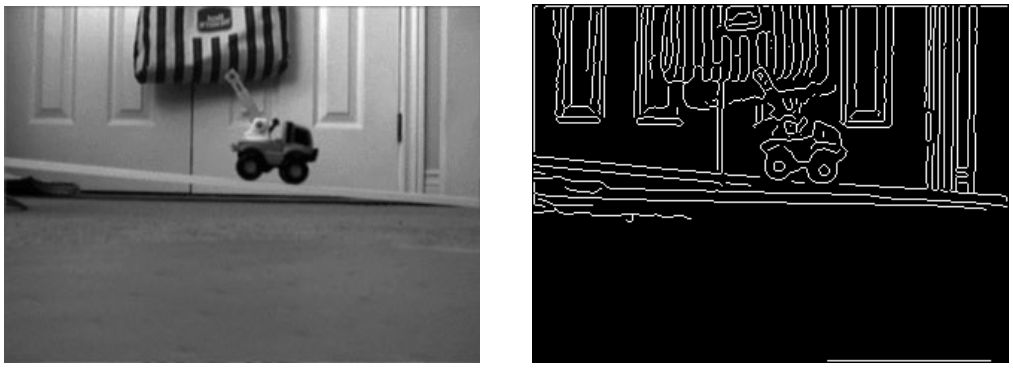
We address both the need to overcome poor initialization and concavity descent by the integration of motion-segmentation data with the traditional active contour formulation. This is achieved by specifying additional criteria to the contour's original properties in operation, so that the active contour:

- Adheres to image edges outside the estimated target *IMO* interior area;
- Does not adhere to image edges outside the estimated *IMO* border.

[1] Convex hull generation techniques [18] are widely available, for example MATLAB provides the built-in *convhull()* function.

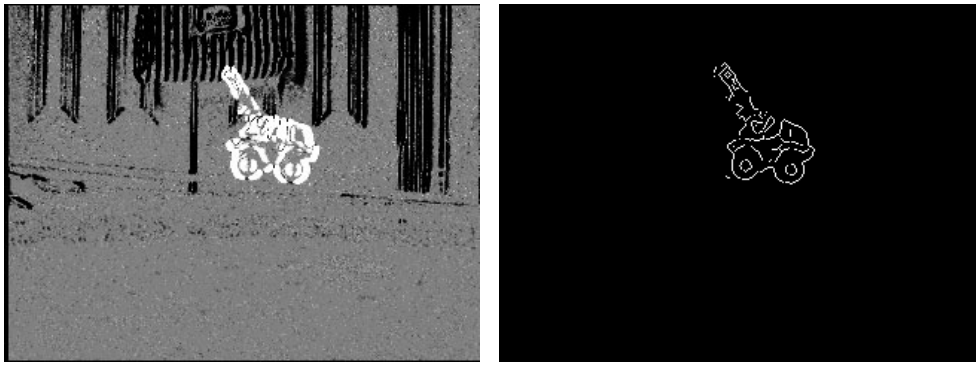
(a) Frame 30 (b) Frame 50

Figure 7.3: Tow Truck Sample Sequence: Toy tow truck rolls down the ramp toward the right of the scene against a stationary background.



(a) Tow Truck Sequence 1 Frame 50 (b) Edge Map

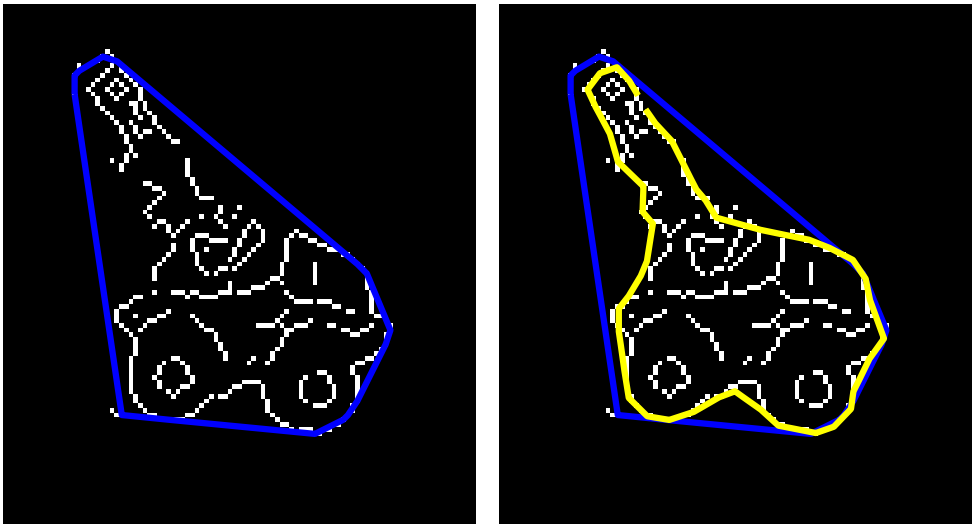
Figure 7.4: Tow Truck Sample Sequence Edge Map



(a) Motion segmentation map: White areas correspond to the target *IMO* areas, black areas to background areas, and gray areas are ambiguous areas.

(b) Motion segmented edges of the target *IMO*.

Figure 7.5: Tow Truck Motion Segmentation.



(a) Convex hull shown as dark blue line.

(b) Contracted convex hull shown as bright yellow line.

Figure 7.6: Tow Truck Convex Hull Initialization

These additional criteria are met by implementing 'normal forces' at each contour control point so that it may expand out of the target *IMO*'s interior (white areas in Figure 7.5(a)), and contract away from the non-*IMO* area (black areas in Figure 7.5(a)). The active contour thus gains awareness of the motion segmentation region classification results during its execution, and is pushed toward the boundary of the *IMO* area, where the *IMO* bounding edges should lie.

As a result, while traditional active contours are optimized under the myopic influence of purely local image-based forces and internal constraints, region-based normal forces introduce global information from the motion segmentation process into the active contour's execution. These external forces supplement the purely local image-based force in the traditional active contour so that the active contour is not only attracted to edges in the image, but those edges in the image that likely belong to the target *IMO*'s boundary. Similarly, the motion segmentation indicates areas where edges have a high probability of *not* belonging to the target *IMO*, and the normal forces cause the active contour to contract away from these areas. Our revised active contour formulation is thus able to overcome poor initialization and also maintain awareness of the global segmentation of the frame.

The segmentation framework requires additional terms and modifications to the traditional energy functional of Equation 7.1 in order to meet the additional contour criteria of adhering to edges that are expected to be part of the *IMO* boundary. The resulting energy functional shown in Equation 7.5 is optimized in the same manner as the traditional active contour, but simply incorporates new or modified terms in its formulation:

$$E(s) = \int_s (\underbrace{\alpha E_{continuity}(s) + \beta E_{curvature}(s)}_{\text{internal forces}} + \underbrace{\gamma E_{image}(s) + \delta E_{normal}(s)}_{\text{image-based forces}}) ds \quad (7.5)$$

Our test results employ weighting constant values of $\alpha = 0.05$, $\beta = 0.01$, $\gamma = 1$ and $\delta = 1$. These values balance the effect of internal constraints against the need to closely adhere to image boundaries. The δ term that determines the weighting of normal forces is left to unity, as the normal forces incorporate separate weighting terms (described below) that reflect their true weighting.

The normal forces are implemented in the active contour formulation in a similar manner to image-based forces, by exerting a directional force at any given control point. This force is simply normal to the direction of the contour at the control point of interest, so that it exerts an expansion or contraction force when activated. For a given control point: $\vec{p}(s) = (x(s), y(s))^T$, the vector normal to the active contour at that point, $\vec{n}(s)$ is approximated as

$$\vec{n}(s) = \begin{bmatrix} -(y(s) - y(s-1)) \\ x(s) - x(s-1) \end{bmatrix} . \tag{7.6}$$

This vector is then normalized, $\hat{n} = \frac{\vec{n}}{|\vec{n}|}$, so that the force exerted by the normal force at any control point is only dependent on the weighting coefficient δ, and the motion segmentation map value at that point. As the active contour is initialized with control points ordered in counter-clockwise order about the estimated boundary of the *IMO*, $\hat{n}$ is directed outside the *IMO* area delineated by the closed contour.

The motion segmentation map shown in Figure 7.7 has the corresponding active contour result super-imposed upon it, to demonstrate the function of normal forces. High probability *IMO* areas are shown in white, ambiguous areas in gray and low probability *IMO* areas in black. Normal forces in the white high probability *IMO* areas are configured for expansion, pushing the active contour outside areas considered to be part of the *IMO*, so that the active contour remains at the exterior of the target *IMO*. This is done by multiplying the normal force in white areas by a positive weighting factor, so that they retain their original direction. Black, low probability, *IMO* areas are configured to exert a contraction force so that control points in these areas contract away from non-*IMO* areas and back to the area near the target *IMO* boundary. To do so, the normal force in black areas is multiplied by a negative weighting factor, reversing the direction of $\hat{n}$ so that it points inward.

Finally, normal forces in gray ambiguous areas are eliminated, so that in such areas only internal and image-based forces influence the behavior of the active contour. This is done by setting the normal force to zero in these areas. By doing so, the active contour behaves like a typical snake (with long-range image-based forces) in this area that is determined to

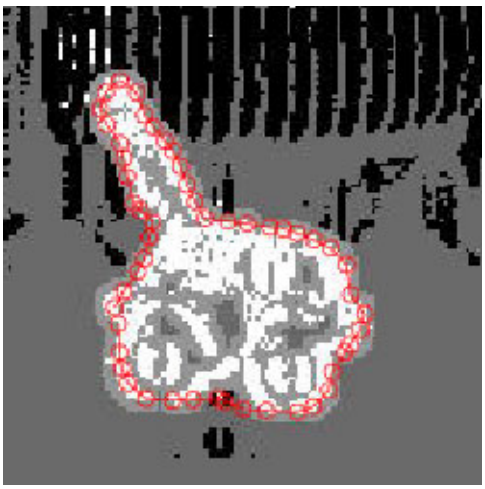


Figure 7.7: Normal forces are implemented in the white and black areas of the motion segmentation map, causing expansion in (white) *IMO* interior areas, and contraction from (black) non-*IMO* interior areas.

be neither inside the *IMO*'s interior, nor inside the area determined to be part of other *IMO*s. This ensures that the motion segmentation results are used to guide the active contour to the target *IMO* border, within the limits of the motion segmentation's accuracy. The normal force term can therefore take on any of three values depending on the location of $\vec{p}(s)$ in the motion segmentation map generated by the motion-based intensity constraint classification process. As such, the motion-based intensity constraint classification ownership function for the target *IMO*, $P_n(\vec{p}(s))$, described in Section 5.1, can be used to assign the normal force value at a point:

$$Force_{normal}(s) = \begin{cases} \upsilon\hat{n}(s) & , \quad P_n(\vec{p}(s)) > 0.5 \\ -\omega\hat{n}(s) & , \quad P_n(\vec{p}(s)) < 0.5 \\ 0 & , \quad P_n(\vec{p}(s)) = 0 \end{cases} \tag{7.7}$$

The weighting terms υ and ω determine the relative strength of the contraction and expansion force, and it should be noted that these terms may absorb closely the value of the δ weighting coefficient in Equation 7.5. The combination of the three terms determine the ultimate balance of the normal forces relative to the other forces that affect the active contour's behavior. Under

test conditions, the normal force weighting factors are set to the values $\upsilon = 0.8$ and $\omega = 0.6$, so that normal forces approach but do not exceed the influence of image-based forces. This setting quantifies our confidence in the image-based forces, over the comparatively coarse nature of the region-based normal force.

The motion segmentation map, shown in Figure 7.5(a), still shows large ambiguous areas in the image where no strong motion classification exists, the gray areas in the figure. Such areas occur in the uniform areas of the image where no valid motion or motion-based intensity constraint classification exits, so normal forces cannot be applied in these areas. This condition is mitigated by the introduction of long-range image edge-based forces that are applied in this areas, discussed in the next section, which improve the ability of the active contour to adhere to the correct boundaries within concavities and within motion-segmented target *IMO* areas.

7.5 Long-Range Image Edge-Based Forces

The traditional image edge-based forces shown in Figure 7.1 have relatively short range, and this property affects the need for a good initialization. Increasing the range of these edge-based forces assists the active contour in finding and adhering to the *IMO* edges, but only if the *IMO* edges are identified. The problem can be clearly visualized by imagining the external image forces that would be exerted by the complete edge map for the tow truck sequence shown in Figure 7.4 where extended range forces from all *IMO*s in the scene would affect the active contour. Ideally, edge-based forces should affect the active contour when they originate from the *IMO* of interest's edges only, such as those edges shown in Figure 7.5(b). In this case, the active contour easily acquires the correct boundary of the target *IMO*.

To assist the active contour's adherence to the correct edges, the motion segmentation map is used to determine which edges have a high probability of belonging to the target *IMO*, which edges are ambiguous, and which edges have a low probability of belonging to the *IMO*. Edges with a low probability of belonging to the target *IMO*(edges in the black areas of a

segmentation map) are removed from consideration when generating the image-based force for a given *IMO*, so that they do not exert any influence on the active contour. The nature of the aperture problem prevents the classification of many ambiguous edges, even though those edges may be part of the target *IMO*. As a result, both the high probability edges of the *IMO* and edges in ambiguous regions are used to generate the long-range image-based forces, confining the active contour to adhere to edges in the target *IMO*'s bounding areas only. The steady tracking and improvements of the motion estimation process over time improve the motion segmentation results, improving the active contour's performance. Meanwhile the ambiguous edges remain attractive to the active contour allowing it to adhere to these edges wherever they are closer than high probability *IMO* edges.

The motion segmentation does result not correctly segment edges that are either ambiguous due to the aperture problem, or edges that belong to *IMO*s that have the exact same motion as the target *IMO*, but the issue of the aperture problem is assisted by the motion-based intensity constraint classification process discussed in Chapter 5, while cases where two *IMO*s share similar motion parameters is addressed through their spatial segmentation, using the CCA method presented in Chapter 6. Extending the range of image-based forces assists the active contour to overcome poor initialization, and to descend into concavities in the *IMO*'s boundary. The distance transform techniques used to extend the edge-based forces are introduced in the next section.

The distance snake formulation effectively replaces the traditional external image forces in the active contour energy functional given in Equation 7.1. They are calculated by generating an image-edge map of the region of interest (the motion segmented *IMO* being tracked), and then by applying a Euclidean distance transform to the edge image [12], generating an image-sized 2-D map whose values correspond to the distance from a given point to the nearest edge. A potential map, $P(\vec{x})$ can be defined on the distance transform map, $d(\vec{x})$, as

$$P(\vec{x}) = \begin{cases} -1 & , \quad d(\vec{x}) < 1 \\ \frac{-1}{d(\vec{x})} & , \quad otherwise \end{cases} . \tag{7.8}$$

The potential is defined as the negative inverse of distance to the nearest image edge of interest, and the image-based force can then be defined (and optionally normalized) as shown below:

$$D(\vec{x}) = -\nabla P(\vec{x}) \tag{7.9}$$

$$D(\vec{x}) = \frac{-\nabla P(\vec{x})}{|\nabla P(\vec{x})|} \tag{7.10}$$

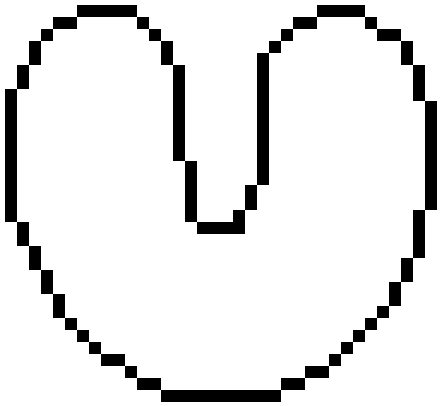
The negative of the spatial gradient of the potential map orients the image-based force map vectors toward the closest edge of interest. In order to improve the speed of convergence, the normalized force map of Equation 7.10 may be used, by ensuring that image forces are of unit magnitude even at long distances from image edges. In execution, the new distance-based image forces are substituted in place of the traditional image-based force. The new forces work cleanly alongside other external forces and the traditional internal forces of the active contour, and the new formulation can be optimized in an identical manner.

The distance transform edge map is generated by applying a Canny[2] edge segmentation [50] to the target image, revealing the significant edges within it. The motion segmentation map is then used to identify low-probability *IMO* areas in the image, and edges within these areas are removed from the edge map. It is this filtered edge map that is used to generate the distance transform-based image force of Equation 7.5. The vector forces at each point are illustrated in Figure 7.8(c).

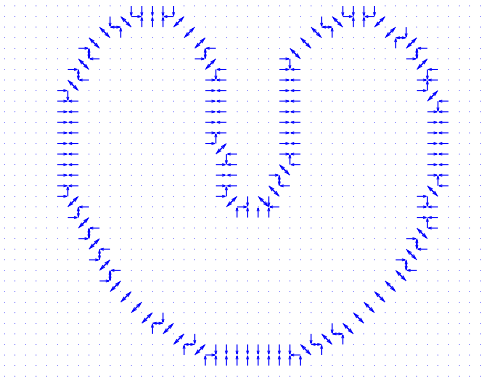
7.6 Active Contour Propagation Between Frames

An additional region-based force is used to improve active contour performance: the recovered motion parameters of the target *IMO* are used to shift the active contour between subsequent frames of the sequence after initialization. This follows the idea that the boundary of the *IMO* should move with the same motion parameters as the interior of the *IMO*, notwithstanding any non-rigid deformations.

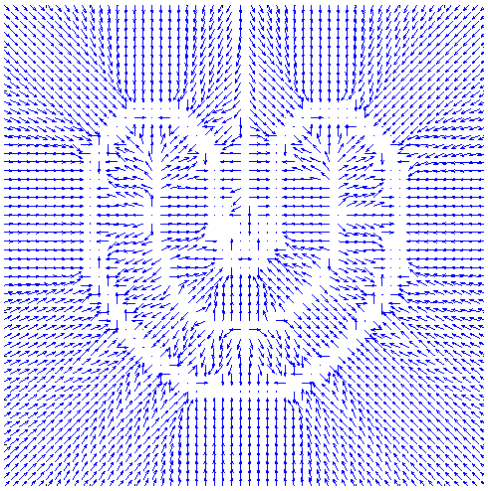
[2]This common edge detection technique is widely documented in computer vision literature, and is available in the MATLAB Image Processing Toolbox as part of the *edge()* function.



(a) Sample Edge Image



(b) Traditional Snake External Image Forces



(c) Distance Transform External Image Forces

Figure 7.8: Active Contour Edge Forces

The inter-frame contour propagation step shifts the position of the active contour from its final position in the previous frame, to its motion predicted position in the current frame. This simplifies the problems faced by the active contour optimization stage, so that it does not need to account for the rigid parametric motion of the *IMO*. The active contour execution simply addresses non-rigid *IMO* deformations and corrects the boundary estimate for propagation errors due to the assumption of a parametric motion model.

The refinement of the active contour over multiple frames is a noteworthy aspect of the active contour application in a video sequence: the active contour optimization about a rigid *IMO* will be continually refined at every time step, so the net result is equivalent to running the active contour optimization for a very large number of iterations over time. This allows the active contour to eventually recover an accurate *IMO* boundary even for complex *IMO*s, without necessitating the complete optimization of the active contour (through a very large number of iterations) at any single frame.

Non-rigid *IMO* deformations may be caused by changes in the *IMO*'s shape or topology, for example due to articulation. Assuming the movement due to articulation is small relative to the overall motion of the *IMO*, the active contour optimization can easily account for it. On the other hand, very large motions due to articulation might cause the articulated parts of the *IMO* as a distinct *IMO*, causing the *MASC* system to initialize a new active contour process around that part. The other important class of non-rigid *IMO* deformations are a result of the error in frame-to-frame contour propagation due to the assumption of a parametric motion model. This occurs when the *IMO*'s true motion can only be accurately parameterized by a higher order motion model than that which is being used at the time, for example, where a constant motion model is used to represent an *IMO*'s rotation in the image plane. As the constant motion model is only able to represent translation in the plane, the propagation of the contour between consecutive frames does not properly align the contour with the *IMO* in the new frame. The active contour aims to accommodate such errors by moving individual contour points to coherent edges in their vicinity, which have a strong possibility of being the correct

IMO edges under the assumption of smooth motion. The effectiveness of inter-frame contour propagation are demonstrated in the results presented in Section 9.1, that show the predicted contour and the final contour for various test frames.

7.7 Active Contour Optimization

This section describes the active contour optimization algorithm proposed by Kass *et al.* [34], used with modifications in the *MASC* system. The solution process is described here for completeness, without details of the derivation.

The active contour energy functional is a discrete sum of internal and external energy terms, summed over the length of the contour:

$$E_{snake} = \sum_{s=1}^{n} E_{int}(s) + E_{ext}(s) \tag{7.11}$$

where we define the internal energy term as

$$E_{int}(s) = \frac{\alpha|\vec{p}(s) - \vec{p}(s-1)|^2}{2h^w} + \frac{\beta|\vec{p}(s-1) - 2\vec{p}(s) + \vec{p}(s+1)|^2}{2h^4} \; . \tag{7.12}$$

For notational simplicity, we let $f_x(s) = \frac{\partial E_{ext}}{\partial x(s)}$, and $f_y(s) = \frac{\partial E_{ext}}{\partial y(s)}$. The discrete expression of the Euler equations that minimize the energy function equation is

$$\begin{aligned} &\alpha[\vec{p}(s) - \vec{p}(s-1)] - \alpha[\vec{p}(s+1) - \vec{p}(s)] \\ &+2\beta[\vec{p}(s-2) - 2\vec{p}(s-1) + \vec{p}(s)] \\ &-\beta[\vec{p}(s-1) - 2\vec{p}(s) + \vec{p}(s-1)] \\ &+\beta[\vec{p}(s) - 2\vec{p}(s+1) + \vec{p}(s+2)] \\ &+[f_x(s), f_y(s)] = 0 \; . \end{aligned} \tag{7.13}$$

We may now express Equation 7.13 in matrix form, where $\vec{p}(s) = [x(s), y(s)]^T$:

$$\mathbf{A}\vec{x} + \vec{f_x}(\vec{x}, \vec{y}) = 0 \tag{7.14}$$

$$\mathbf{A}\vec{y} + \vec{f_y}(\vec{x}, \vec{y}) = 0 \tag{7.15}$$

We can solve the Euler equations of 7.14 and 7.15 by defining an Euler method with step size ζ:

$$\mathbf{A}\vec{x}_t + \vec{f}_x(\vec{x}_{t-1}, \vec{y}_{t-1}) = -\zeta(\vec{x}_t - \vec{x}_{t-1}) \quad (7.16)$$

$$\mathbf{A}\vec{y}_t + \vec{f}_y(\vec{x}_{t-1}, \vec{y}_{t-1}) = -\zeta(\vec{y}_t - \vec{y}_{t-1}) \quad (7.17)$$

The time derivatives on the right hand side of Equations 7.16 and 7.17 go to zero as we approach equilibrium through iteration. We can solve the incremental equations by matrix inversion:

$$\vec{x}_t = (\mathbf{A} + \zeta\mathbf{I})^{-1}[\vec{x}_{t-1} - \vec{f}_x(\vec{x}_{t-1}, \vec{y}_{t-1})] \quad (7.18)$$

$$\vec{y}_t = (\mathbf{A} + \zeta\mathbf{I})^{-1}[\vec{y}_{t-1} - \vec{f}_y(\vec{x}_{t-1}, \vec{y}_{t-1})] \quad (7.19)$$

Taking g_x and g_y to represent the distance transform image-based force maps and taking N_x and N_y to represent the normal force maps, we can express the incremental solution equations as

$$\vec{x}_t = (\mathbf{A} + \zeta\mathbf{I})^{-1}[\vec{x}_{t-1} - \vec{g}_x(\vec{x}_{t-1}, \vec{y}_{t-1}) - \vec{N}_x(\vec{x}_{t-1}, \vec{y}_{t-1})] \quad (7.20)$$

$$\vec{y}_t = (\mathbf{A} + \zeta\mathbf{I})^{-1}[\vec{y}_{t-1} - \vec{g}_y(\vec{x}_{t-1}, \vec{y}_{t-1}) - \vec{N}_y(\vec{x}_{t-1}, \vec{y}_{t-1})] \quad (7.21)$$

We incorporate additional force weighting parameters in their respective force maps, setting the relative weighting of the external forces with respect to each other. The next chapter summarizes the interaction between *MASC* components, and presents details regarding the implementation and performance of the *MASC* system.

Chapter 8

MASC Segmentation System Summary

The preceding chapters provide a detailed description of the *MASC* segmentation system's components, this chapter provides an overall summary of the *MASC* system, emphasizing the interaction between the various components. The block diagram of the *MASC* system is repeated in Figure 8.1 to reiterate the relationships between components.

8.1 Motion Segmentation Feedforward During Initialization

Hierarchical motion estimation detects and segments *IMO*s within the scene, providing motion parameter estimates for the background and *IMO*s. This information in turn, is used to generate motion-based image warps for each layer, so that the motion-based intensity constraint classification process can be applied, generating region segmentation data. CCA-based spatial segmentation is performed to distinguish similarly moving (but spatially distinct) *IMO*s in the scene, generating a binary segmentation map for each *IMO*. The binary segmentation map is then used to initialize an active contour about each *IMO* which performs boundary recovery for the *IMO*. The active contour directly incorporates motion-based intensity constraint classification data in its execution by adopting normal forces that direct the contour to expand or contract based on that data. *IMO* motion estimates are further exploited after initialization for contour propagation, described below.

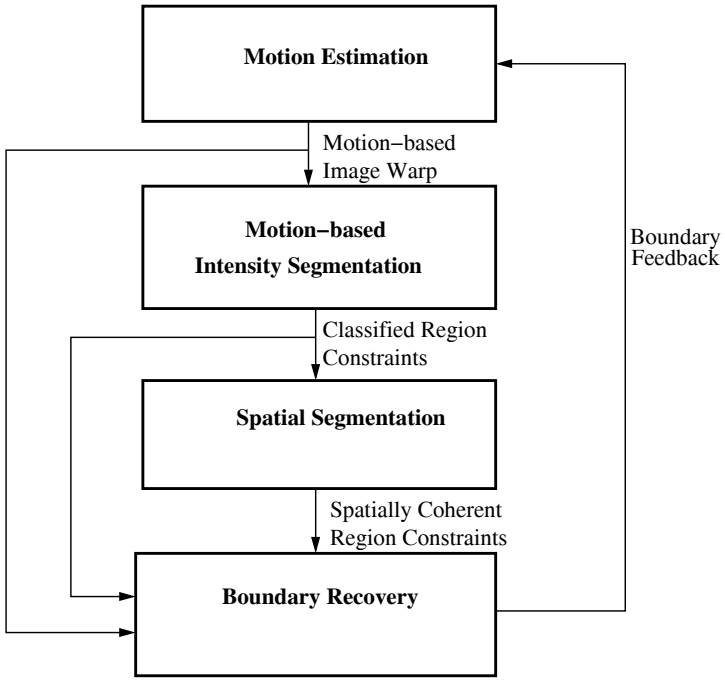


Figure 8.1: *MASC* Segmentation System Block Diagram

The initial motion segmentation process initializes an active contour about each *IMO* in the scene, providing boundary-based spatial coherence constraints for each *IMO* and the background layer. The background layer makes up the entire area in the scene that is not bounded by an *IMO* active contour. These spatial coherence constraints can now be used by the system as part of the active contour feedback process.

8.2 Active Contour Feedback

Once an active contour has been initialized, it is optimized to align itself with nearby *IMO* edges and within the *IMO* boundary area indicated by the motion-based intensity constraint classification. This boundary can now be used by the motion estimation process for the next consecutive frame as an estimate of the *IMO*'s boundary, so that only motion constraints from within this region are used to estimate the motion of the *IMO*. This motion estimation process continues to be performed using robust, hierarchical techniques within this confined region,

allowing the *MASC* system to accomodate *IMO* artefacts (such as transparency) and large displacements.

This approach contrasts with the initialization process whereby the motion constraints are segmented on the basis of their motion coherence, irrespective of their distribution throughout the scene. The CCA-based spatial segmentation component attempts to ensure spatial compactness, however, this does not simplify or change the motion estimation process. The active contour provides a boundary within which most motion constraints will be generated by the target *IMO*, improving the accuracy and reducing the complexity of the motion estimation.

8.3 Motion Segmentation Feedforward After Initialization

After the active contour initialization has been performed for an *IMO*, motion estimation and accordingly, the motion-based intensity constraint classification are based on the motion constraints found within the bounded *IMO* area. The motion estimate recovered at this point can be used to propagate the active contour from the previous frame into the next frame, so that the active contour's optimization does not need to compensate for the rigid *IMO* motion. This greatly improves the performance of the active contour, as it is effectively performing continuous refinement on the *IMO*, accumulating a large number of refinement iterations over the duration of the sequence. The typical region-based external forces continue to direct the active contour to expand around the *IMO* or contract away from non-*IMO* regions, continuing this aspect of the original feedforward interaction.

The outline of *MASC* component interaction illustrates the way in which each component of the *MASC* system is able to incorporate results from other components within fundamental aspects of it own operation. This allows the *MASC* system's components to collaborate and overcome the weakness of their individual limitations. The next section provides a description of the platform used to implement the *MASC* segmentation system, and some indications of the performance of the system.

8.4 *MASC* System Implementation

The *MASC* system described in this thesis was developed and implemented in the MATLAB 6.0 environment of The MathWorks Inc., running on RedHat Linux 7.2 for x86. The primary hardware platform used was based on dual Intel Pentium Xeon CPUs in SMP configuration, operating at 2.6Ghz. As the MATLAB 6.0 environment is not able to exploit SMP facilities at this time, performance should be identical for a single CPU configuration.

The typical performance of the algorithm on the primary test platform is roughly seven to ten seconds of processing time per frame, for a sequence containing a single *IMO* and background layer. As each additional *IMO* requires independent motion segmentation and active contour processes, additional computing loads are imposed on the system, reducing performance accordingly. The next chapter presents results generated by the *MASC* system when applied to a variety of video sequences.

Part III

Results and Conclusion

Chapter 9

Experimental Results

In this chapter, we present a set of experimental results that demonstrate the segmentation and tracking performance of the *MASC* system. The target video sequences include sequences in which a single *IMO* moves against a stationary background, a sequence with a moving background due to camera motion and also a multiple motion sequence in which multiple *IMO*s move against a stationary background.

Most video sequences used to test the *MASC* segmentation system were captured at thirty frames per second at a resolution of 640×480 pixels, and subsampled to 320×240 pixels for *MASC* analysis. A SONY VFW-DL500 digital video camera was used for capture, providing non-interlaced, uncompressed frame captures in 8-bit grayscale. MD Robotics provided the satellite test sequence (Section 9.4), at a resolution of 640×480, subsampled to 320×240 for analysis. The cup test sequence (Section 9.5) was acquired by Professor James MacLean at the University of Toronto using a SONY DCR-TRV510 NTSC Digital8 hand-held digital video camera with compression, at a resolution of 720×480 pixels, subsampled to 360×240 for analysis. The Jojic-Frey test sequence (Section 9.6) is used with permission from Professor Brendan Frey of the University of Toronto, and the sequence has a resolution of 244×118 pixels.

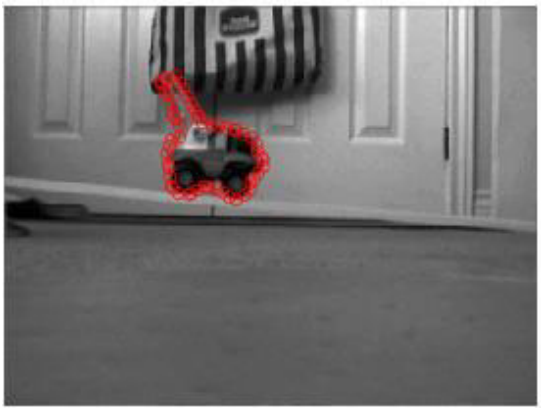
9.1 Tow Truck Sequence 1

The tow truck video sequence introduced earlier is a view of a toy tow truck rolling down a ramp from the left side of the scene to the right, shown in Figure 9.1. The camera and background are stationary, and thus the tow truck is the only moving object in view. In this sequence, the stationary background is represented as a unique *IMO*, and the toy truck is also a unique *IMO*. Results are discussed referring to the toy truck as the active *IMO*. The segmentation process begins at Frame 31 of the sequence, and Figure 9.1(a) illustrates the results of the initial motion-based intensity constraint classification. The white areas of the segmentation map indicate areas corresponding to the active *IMO*, while the black areas indicate areas corresponding to the sequence background. The remaining gray areas of the image are ambiguous areas for which motion segmentation provides no useful distinction due to the aperture problem, as these areas typically arise due to lack of texture or edges perpendicular to any motion.

The tow truck is clearly identified in the sequence by the motion-based intensity constraint classification process, but speckled regions in the background are still assigned the white classification due to noise. These noisy classifications are removed from consideration during the initialization of the active contour by the CCA segmentation module, discussed in Section 9.3. The isolated object layer is then used to initialize the active contour as described in Section 7.3. The resulting active contour, after initialization and optimization is shown in Figure 9.1(b). Once the object contour has been initialized, it can be propagated through successive frames using the recovered motion estimates from within the contour, with the results of motion segmentation and the active contour's execution shown in Figures 9.1(c) to 9.1(f).

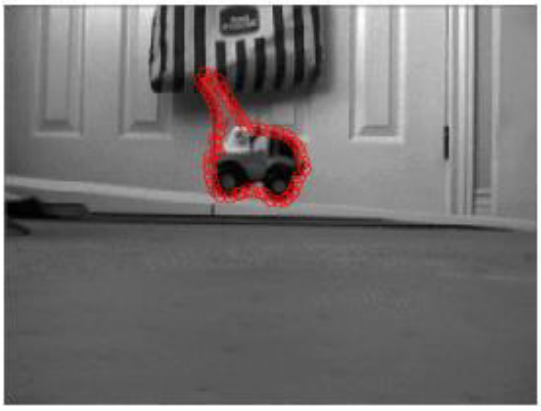
Figure 9.2 demonstrates the effectiveness of contour propagation during sequence analysis, whereby after initialization, the active contour's position from the previous frame is shifted to its predicted position in the current frame based on the motion estimate of the object. By doing so, the initial contour position in the current frame should be very nearly correct, notwithstanding any non-rigid object deformations. The near complete overlap of the predicted and final contour positions shown in the figure fully support the effectiveness of the propagation.

(a) Frame 31: Layers

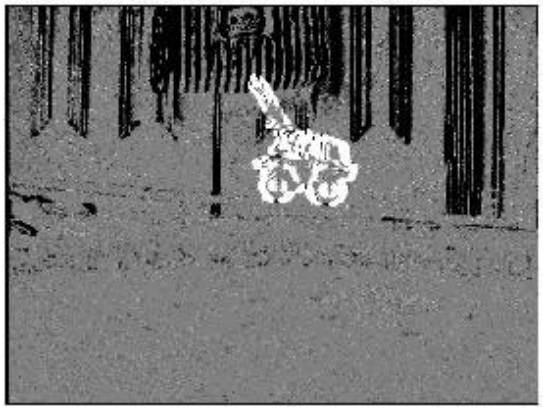


(b) Frame 31: Contour Initialization

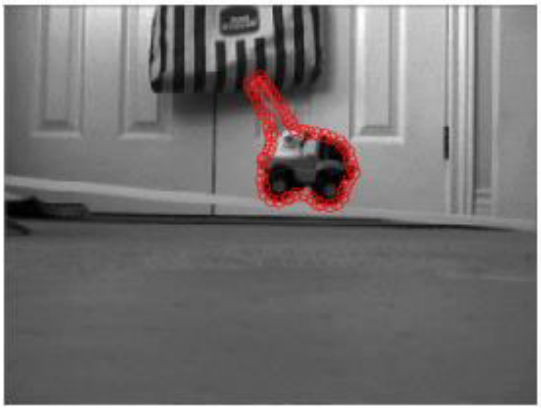
(c) Frame 40: Layers



(d) Frame 40: Contour

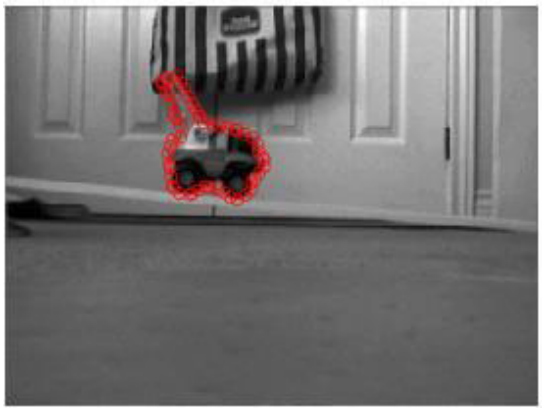


(e) Frame 50: Layers

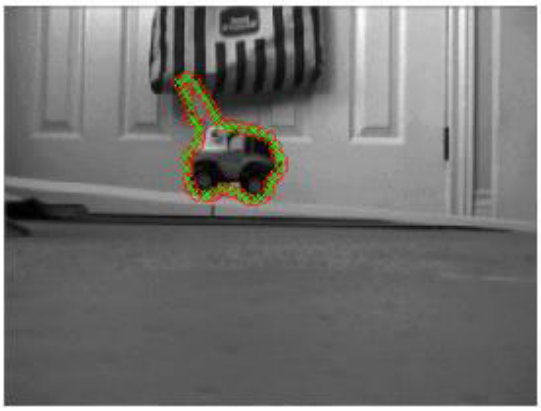


(f) Frame 50: Contour

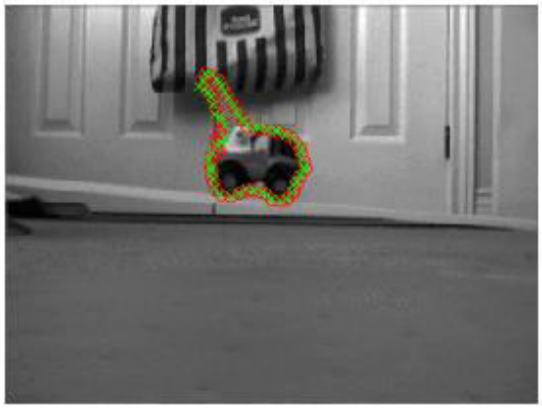
Figure 9.1: Tow Truck Sample Sequence 1: Toy tow truck rolls down the ramp toward the right of the scene against a stationary background. The motion segmentation maps are shown in the left column, while the active contour result is shown in the right column.



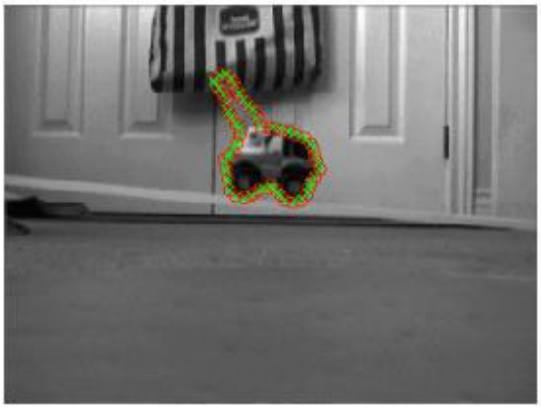
(a) Frame 31: Contour Initialization



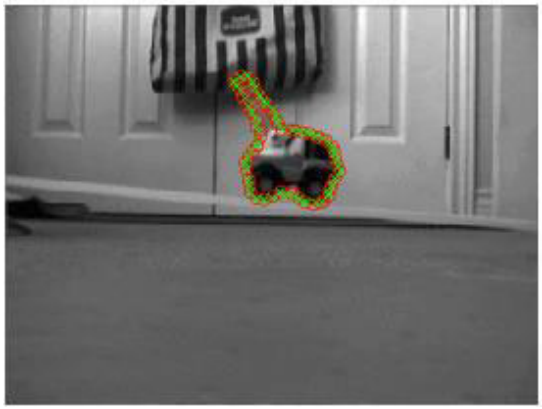
(b) Frame 35: Contour



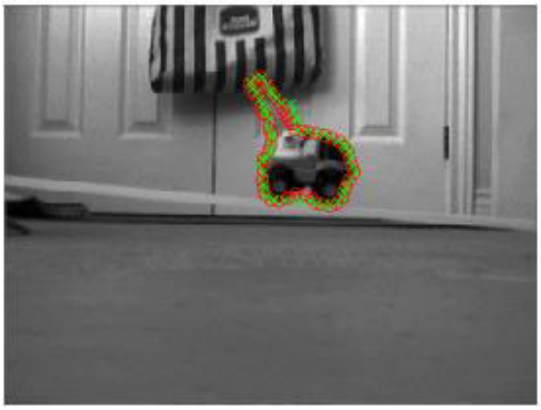
(c) Frame 40: Contour



(d) Frame 43: Contour



(e) Frame 47: Contour



(f) Frame 50: Contour

Figure 9.2: Tow Truck Sample Sequence 1 Illustrating Inter-Frame Active Contour Propagation: The green contour with '$\times$' point markers represents the propagated contour, while the red contour with '$\circ$' point markers represents the the final contour.

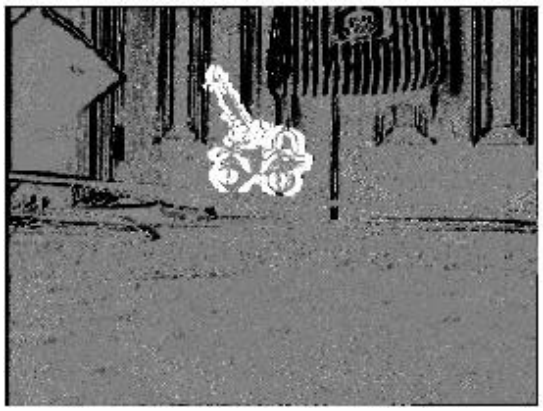
The detail recovery performed by the active contour appears to be quite effective in capturing the tow bar of the truck, and also the depression in the base of the truck between the wheels. The active contour is not successful in capturing the very small concavity that exists between the driver's cabin and the towing rig, however the very small entry to the concavity makes it very difficult for the active contour to descend into this small but deep area. Indeed, the small dimensions of the concavity make it difficult even for the region segmentation component to identify. For more tractable artefacts, the scoop sequence presented in Section 9.3 demonstrates the *MASC* system's ability to handle deep concavities and transparent *IMO* areas.

9.2 Tow Truck Sequence 2

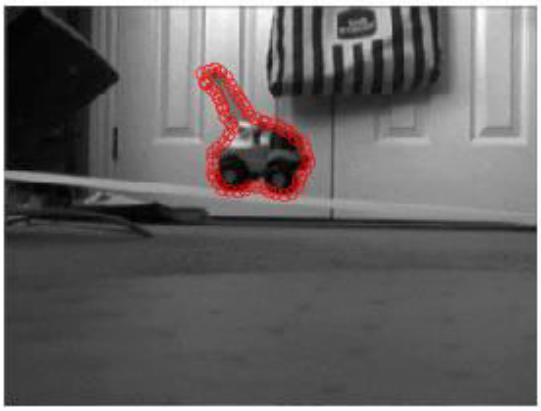
The second tow truck sequence sees the same toy truck rolling down the ramp from the left to the right side of the scene, but now the camera pans left, right then left again, causing the background to move independently of the tow truck. This result demonstrates the treatment of the background as a layer with independent motion whose boundary is simply the entire frame. The robust motion estimation process allows *IMO*s to be present in this area without affecting the background motion estimation process. In addition, as each *IMO* in the scene is detected and delimited by an active contour, these regions are removed from the consideration of background motion estimation. The results for this sequence are shown in Figure 9.3, and illustrate the ability of the *MASC* system to consistently segment an *IMO* against a moving background, with the results being nearly identical to the results for the stationary background case shown in Figure 9.2.

9.3 Scoop Sequence

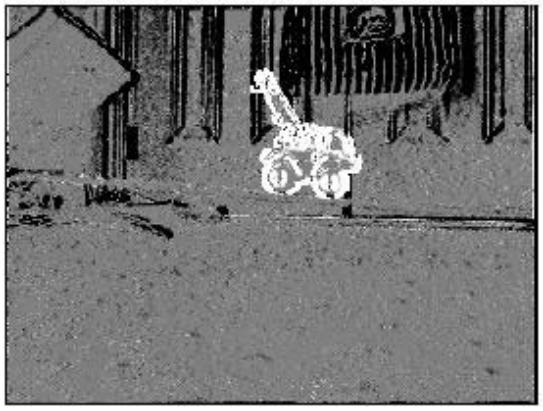
The scoop sequence shown in Figure 9.4 substitutes the tow truck in the first two test sequences for a more complicated bulldozer toy. The bulldozer has front and rear scoops that form signif-



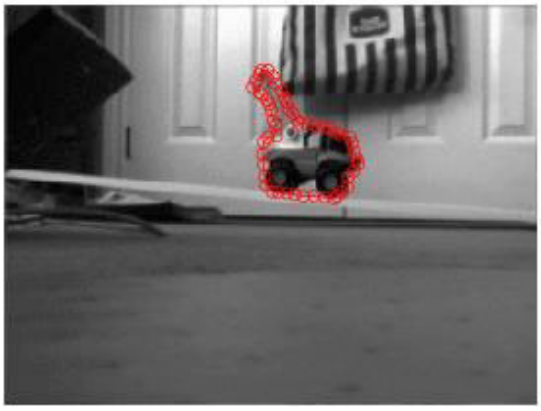
(a) Frame 31: Layers



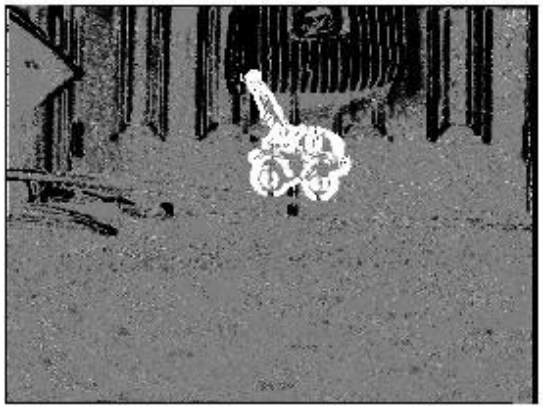
(b) Frame 31: Contour Initialization



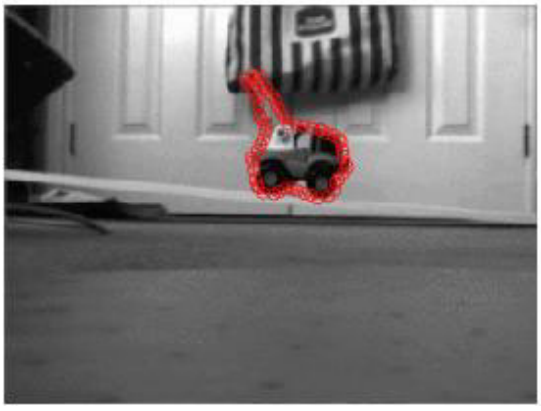
(c) Frame 40: Layers



(d) Frame 40: Contour



(e) Frame 50: Layers



(f) Frame 50: Contour

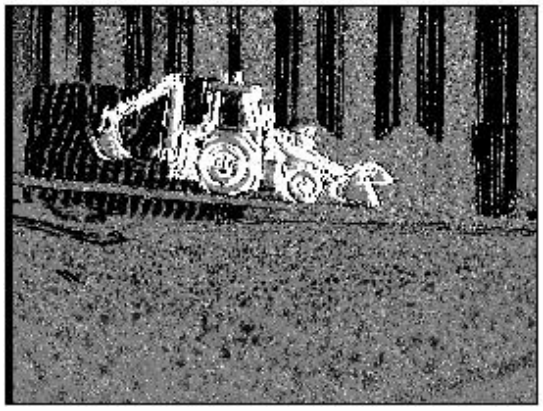
Figure 9.3: Tow Truck Sample Sequence 2: Toy tow truck rolls down the ramp toward the right of the scene against a moving background. The camera pans gently to the left and right of the scene, keeping the toy truck in view at all times. Motion segmentation maps (left column) and active contour results (right column) are nearly identical to those of the stationary background test case.

icant concavities in the shape of the object, posing a serious challenge to conventional active contour techniques that attempt to recover such boundaries. The *MASC* active contour implementation is shown to overcome this problem, descending into the concavities and accurately recovering the shape of the rear scoop structure.

A flaw in the active contour boundary recovery can be seen at the base of the front scoop, where the motion segmentation process is unable to disambiguate the near horizontal edge of the front scoop. This problem is a result of the relevant edge being parallel to the direction of motion, and the very even texture about both sides of that edge. As the region segmentation map illustrates, the edge and its surrounding area are ambiguous with respect to the motion-based intensity constraint classification process, preventing the contour from being initialized correctly. Furthermore, as neither area changes in appearance for the duration of *MASC* analysis, this condition cannot be corrected. The small concavity at the left side of the front scoop is also incorrectly classified by the region segmentation process due to its size, similar to the situation caused by tow truck's small concavity behind the operator cabin, described in Section 9.1.

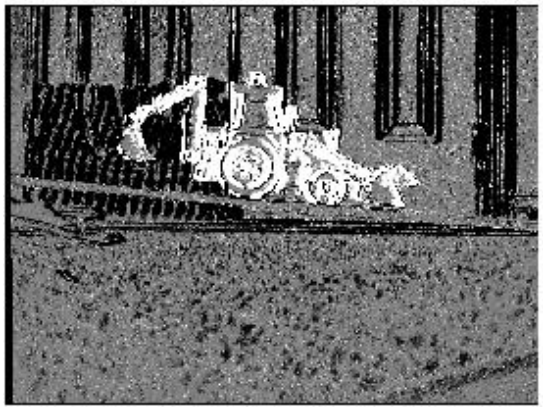
An additional challenge presented by this sequence is the transparency phenomenon produced by the empty driver's window of the bulldozer, through which background texture is clearly visible in Figures 9.4(b) and 9.4(f). The robust motion estimation process is able to accurately segment the complete bulldozer region, with the background area visible through the transparent driver window being correctly classified in the region classification maps of Figures 9.4(a) and 9.4(e).

An additional pair of results illustrate the removal of speckled intensity consistency noise, removed by the CCA segmentation techniques discussed in Chapter 6, and shown in Figure 9.5.



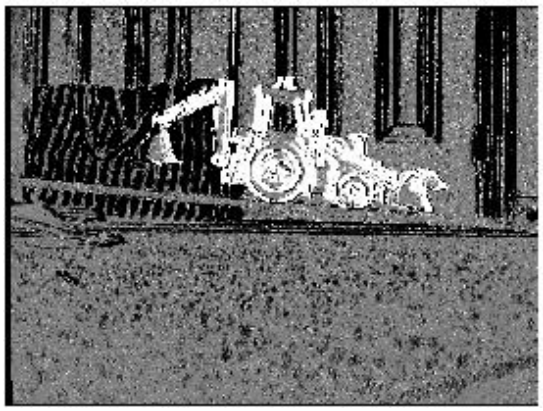
(a) Frame 41: Layers

(b) Frame 41: Contour Initialization



(c) Frame 45: Layers

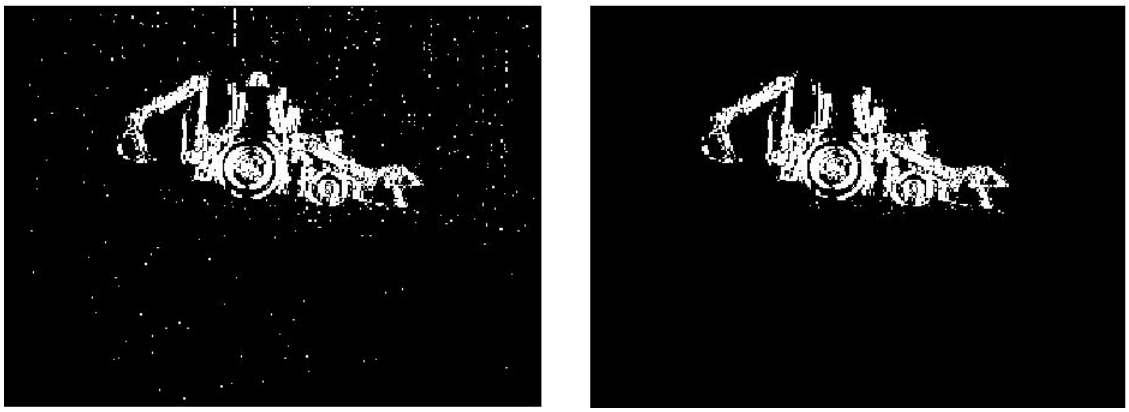
(d) Frame 45: Contour



(e) Frame 50: Layers

(f) Frame 50: Contour

Figure 9.4: Scoop Sample Sequence: Toy bulldozer truck rolls down the ramp toward the right of the scene against a stationary background.



(a) Frame 45: Scoop layer (b) Frame 45: Scoop layer after CCA noise removal

Figure 9.5: Scoop Sequence CCA Noise Removal: The scoop sequence motion segmentation maps scoop layer is typically noisy. Noise is removed using CCA techniques.

9.4 Satellite Sequence

For the satellite test video sequence[1], a satellite model moves across the scene from left to right, mounted upon an 'invisible' robotic arm, simulating a typical scene in space. The lack of any visible background texture in this scene requires a change in the *MASC* system settings, disabling the background motion recovery and segmentation process. The entire background area is ambiguous, so the *MASC* system is unable to classify any part of it to a useful extent.

The satellite itself is adequately resolved, and poses an interesting challenge due to the extreme lighting conditions of the scene and the extreme concavities formed by the solar panels. Region classification maps indicate that the satellite is correctly segmented, with the exception of the extreme upper corners of the solar panels, which lack sufficient definition and contrast. As a result, these parts of the image are classified as ambiguous regions, and are not correctly recovered by the *MASC* system. Besides the solar panel corners, the active contour is generally able to recover the correct boundary and descend into the deep concavities of the solar panels despite the lack of background texture in the image, using the long range distance transform

[1]Courtesy of MD Robotics

active contour external force to seek and recover the correct object edges.

9.5 Cup Sequence

The cup sequence[2] in Figure 9.7 shows a handheld cup entering the scene from the left, moving through to the right of the scene. Analysis begins at Frame 41 in Figures 9.7(a) and 9.7(b), where the cup is only partially visible. As more and more of the cup becomes visible, the bounding active contour is able to expand about the newly visible areas to accommodate the complete cup. This is made possible by the textured pattern on the cup providing strong motion evidence the motion-based intensity constraint classification module. The dense region classification in the cup area, shown in the region segmentation maps of Figure 9.7, allows the *MASC* system to apply expansive normal forces to the active contour in these regions, causing it to expand around the new cup area. Texture around the hand areas is not strong enough to cause a similar effect here, preventing the active contour from correctly capturing the complete moving structure of the hand and cup.

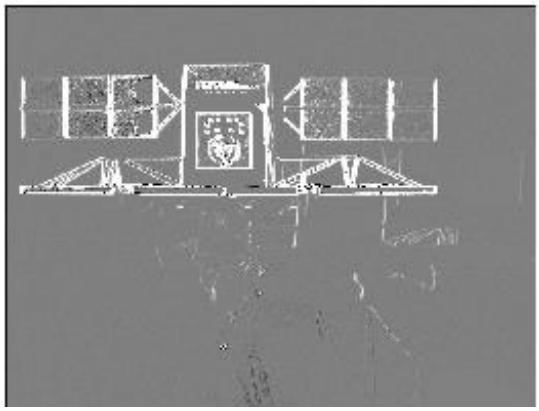
9.6 Jojic-Frey Sequence

The Jojic-Frey[3] test sequence is presented in research on adaptive appearance mixture models [31], and is used here in Figure 9.8 to demonstrate the *MASC* system's analysis of multiple *IMO* sequences.

The two persons in the sequence walk across the scene in opposite directions, meeting at the mid-point of the image, where one of them occludes the other. The *MASC* system correctly segments both *IMO*s as well as the stationary background, as shown by the region classification maps in the left column of Figure 9.8. Similarly, the active contours shown in the right column

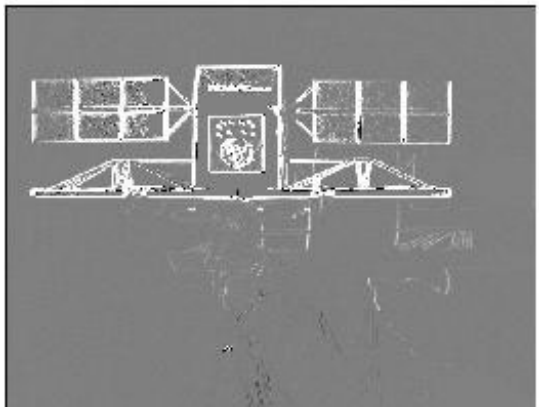
[2]Courtesy of Professor James MacLean, Department of Electrical and Computer Engineering at the University of Toronto

[3]Courtesy of Professor Brendan Frey, Department of Electrical and Computer Engineering at the University of Toronto



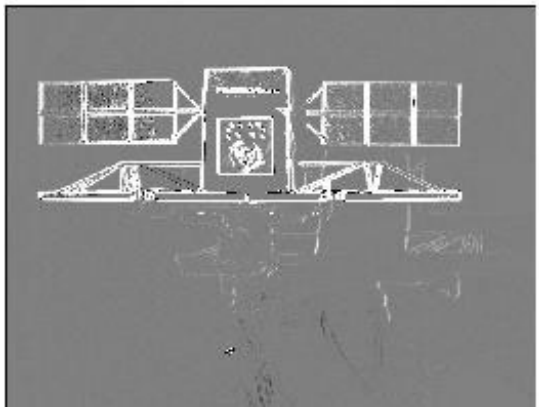
(a) Frame 96: Layers

(b) Frame 96: Contour Initialization



(c) Frame 104: Layers

(d) Frame 104: Contour



(e) Frame 110: Layers

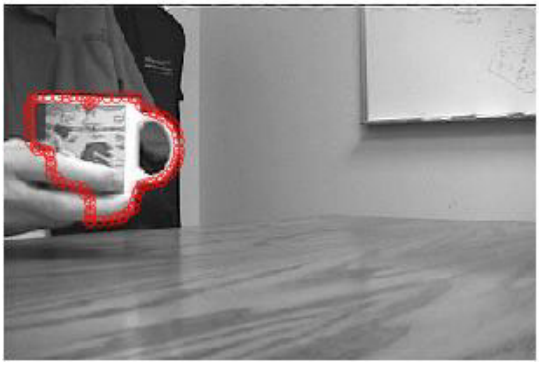
(f) Frame 110: Contour

Figure 9.6: Satellite Sample Sequence: Satellite model moves into view from left to right, against little visible background, except for shadows.

(a) Frame 41: Layers

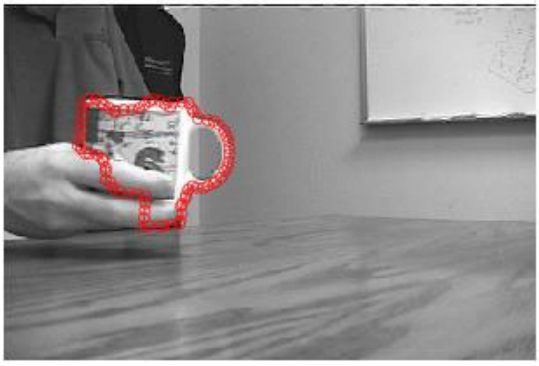
(b) Frame 41: Contour Initialization

(c) Frame 50: Layers



(d) Frame 50: Contour

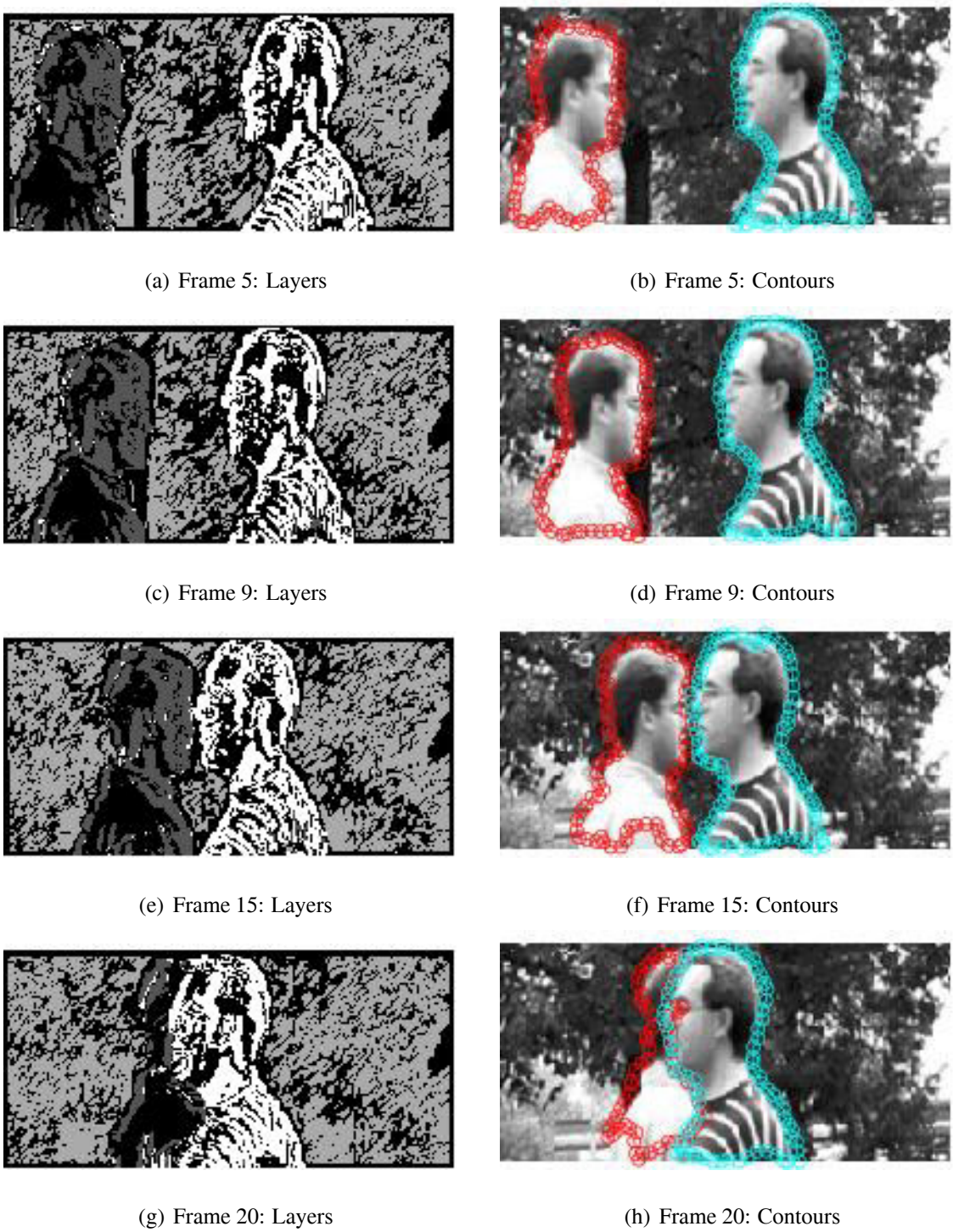
(e) Frame 60: Layers



(f) Frame 60: Contour

Figure 9.7: Cup Sample Sequence: Handheld cup moves into view from left to right, against a stationary background, with visible reflections and shadows on the table surface.

accurately recover the moving outlines of both persons in the scene. Figures 9.8(f) and 9.8(h) demonstrate the effectiveness of the active contour external normal forces in managing the shape of the active contour with respect to region classification information. It is clear that the contour of the occluded person reliably contracts away from the area of the occluding person, while maintaining the correct boundary segmentation of the scene.



(a) Frame 5: Layers

(b) Frame 5: Contours

(c) Frame 9: Layers

(d) Frame 9: Contours

(e) Frame 15: Layers

(f) Frame 15: Contours

(g) Frame 20: Layers

(h) Frame 20: Contours

Figure 9.8: Jojic-Frey Sample Sequence: Two persons walk horizontally across the scene in opposite directions against a stationary background.

Chapter 10

Conclusion

The objectives of this research were to develop a segmentation system to detect, segment and track *IMO*s in a video sequence. The key assumptions imposed on our solution were that no prior *IMO* information would be available and that the background or camera might move. The results presented in Chapter 9 indicate that the *MASC* system we have developed is able to address these situations while generating a good estimation of the *IMO* shape through the active contour. We now describe the conclusions we have drawn upon completion of development of the *MASC* system, providing an analysis of the successes and failures of the system. Finally, we provide suggestions for future work that might advance the utility of the *MASC* system and hybrid region-boundary segmentation techniques in general, and a brief concluding discussion on the broader implications of the research.

10.1 Contributions of the *MASC* System

The most tangible achievement of the *MASC* system is its demonstrated ability to segment multiple *IMO*s in a video sequence, and its ability to deal with moving backgrounds. This is achieved within the requirements of achieving a reasonable estimate of the *IMO* shape, and of dealing with shapes for which no prior information is available. The *MASC* system is also able to demonstrate a system that effectively combines region segmentation techniques with

boundary segmentation techniques, integrating well established motion segmentation and active contour techniques within a common framework. The close integration has several key aspects that are discussed in detail here.

Motion segmentation performs the initial detection and segmentation of *IMO*s in a sequence, generating motion segmentation maps and parametric motion estimates for the *IMO*s. The active contour initialization routines utilize the motion segmentation maps to form initial contours around the target *IMO*s, overcoming a key obstacle in active contour implementations. This is the first feedforward component of the interaction between the two components, that is further pursued in the active contour's execution. The *MASC* system's active contour formulation directly incorporates motion segmentation information through normal forces that apply expansion or contraction forces to the active contour based on region classification. The normal forces assist in addressing another key problem in traditional active contour implementations, that of ensuring that the active contour binds to the correct edges in the image, as opposed to the closest edge to the initial contour location. While region segmentation is the basis of the initial contour itself, significant changes in the shape of the *IMO* during the sequence cannot be detected by a traditional active contour formulation. Examples of such changes are the the penetrating cup of Section 9.5, where an increasingly larger portion of the cup is visible in the scene, and the *IMO* occlusion example in Section 9.6 that shows one *IMO* rapidly occluding another. The normal forces improve the responsiveness of the active contour to such changes, allowing the active contour to incorporate a far broader class of global information into its traditionally local optimization process.

The feedforward process is further strengthened through the use of the parametric motion estimates to propagate the active contour after initialization between subsequent frames. This propagation approach ensures that the active contour's initial position at any given frame is at the position estimated by its deformation in the previous frame, and the estimated rigid motion of the *IMO*. As a result, the active contour's execution in the current frame does not need to account for the rigid motion of the *IMO*. The active contour's execution serves to accommodate

any non-rigid deformations of the *IMO*, and to accommodate for any error in motion estimation due to the use of a parametric motion model. The use of motion estimates in this manner thus simplifies the task of the active contour, so that fewer iterations are required to optimize the active contour, reducing the computational demands of the active contour component and allowing the active contour to provide good boundary estimates even when the number of optimization iterations per frame are limited.

The utility of the feedforward process is balanced by the feedback of the active contour results to the motion segmentation component, which uses the *IMO* boundary estimates to isolate the motion constraints generated by each *IMO*. As a result, motion estimation for each *IMO* can be performed given a good estimate of the 'ideal clustering window' proposed in Section 1.3. Most motion constraints within this window are generated by the target *IMO*, while a small proportion of constraints may be generated by noise, occluding *IMO*s and transparency phenomena. The application of robust EM-based methods to estimate motion removes incoherent constraints of that nature from consideration, so that the resulting motion estimate results from the clustering of coherent constraints from the target *IMO*, providing near ideal conditions for estimation. Through this feedback, the active contour component imposes strong spatial coherence constraints upon the motion segmentation process after active contour initialization, removing a significant part of the burden of segmentation. An implicit element of temporal coherence is also introduced by the active contour, as its propagation between frames implies that we expect the *IMO* (and specifically, the *IMO* shape) to persist temporally in the sequence.

In addition, the feedforward and feedback mechanisms of the *MASC* system allow global (region-based) constraints such as the global segmentation of the whole scene to be integrated within the execution of the myopic local contour process. Similarly, the region-based operation of the motion segmentation component is able to incorporate the very local segmentation results of the active contour into its own operation. The overall *MASC* system uses each component to overcome the other's weaknesses, resulting in a better system than what might be achieved by focusing on only one approach.

From a purely algorithmic aspect, the *MASC* framework introduces two novel approaches to segmentation. The first is the motion-based intensity constraint classification technique, which introduces a novel extension to pure motion segmentation techniques. It uses motion estimates and the BCC to improve the density of motion segmentation estimates, and introduce a probabilistic framework for doing so. The second novel aspect is the use of active contour normal forces described earlier.

The more general conclusion of the *MASC* system emphasizes that this research presents a general framework for integrating motion segmentation and boundary recovery techniques. This research presents an implementation of that framework (the *MASC* system), employing a hierarchical BCC-based motion segmentation component, and a modified active contour boundary recovery component. However, as indicated in the suggestions for future work in Section 10.3, either component might be substituted for more appropriate application-specific modules if required. The feedforward and feedback channels need not change significantly to accommodate the substitution.

10.2 Shortcomings of the *MASC* System

The *MASC* system exhibits a number of shortcomings that we have not yet addressed, as many are beyond the scope of this research. Proposals to rectify many of these shortcomings are presented in the suggestions of future work. The performance of the current *MASC* system implementation presents a key hurdle to any real world application, as it currently takes seven to ten seconds to process a single frame in the *MASC* sequence. It should be noted that the current MATLAB 6 implementation is far from performance optimized, as the MATLAB language itself is interpreted.

Regarding generic video segmentation, backgrounds with significant depth variations are not uncommon, and the *MASC* system currently does not attempt to deal with such scenarios. A confined room for instance, might have five visible walls (planes), each of which would

require an independent projective motion process to model, if the camera was moving in such a setting. The current *MASC* system makes little provision for this condition, however a separate problem suggests a possible solution. The presence of smaller *IMO*s in front of larger *IMO*s is an additional problem currently not addressed: transparency allows *IMO* processes to exist 'on top' of other *IMO*s in the field of view and requires inference of the visibility of each *IMO* in the scene. Introducing visibility data for each *IMO* might also allow for multiple background planes to be used to represent complicated background structures.

The use of the BCC to estimate motion is another drawback of the *MASC* system in applications where illumination of the sequence may vary significantly. Despite this drawback, alternatives would only introduce further complications to the *MASC* system while relying on similar feature constancy assumptions (such as phase [20] or photoquantigraphic constancy [38]). The benefits of invariance to changing illumination must be weighed against the additional system complexity and computational overhead if these alternative approaches are to be adopted.

10.3 Future Research Suggestions

Our suggestions for future research directions comprise suggestions that apply to individual components of the *MASC* system, and suggested directions to improve the *MASC* system as a whole, as well as the fundamental framework it is built upon. We present our suggestions for each of these in order, below.

Discrete Component Research Suggestions

The simplest suggestions for future research are direct improvements to the current components, for example, the introduction of higher order motion models, such as the projective motion model, for motion segmentation. Higher order motion models represent a wider range of *IMO* motion at the cost of increased computational complexity and lower computational sta-

bility. Given larger frame dimensions, a larger number of constraints might be used to improve computational stability, making this a viable option for a faster, better optimized computational platform.

The introduction of phase constancy- and photoquantigraphic constancy-based motion recovery algorithms to improve robustness of the segmentation under variable lighting conditions is a similar direct improvement of the motion segmentation component. As pointed out in Section 10.2, such modifications would increase the complexity and computational demands of the system, however such costs may be justified. For example, a space-based application of the segmentation system would require strong immunity against the rapidly changing brightness of exterior conditions of a satellite's orbit. In additon, the automatic gain control present in video cameras changes the brightness of pixels on the basis of overall scene brightness, violating the brightness constancy constraint in many conditions, and motivating the use of alternative constraint sets.

While the active contour component of the *MASC* system explicitly imposes strong spatial coherence constraints on the motion segmentation process, the notion of temporal coherence is only implicit through its frame to frame propagation of the active contour. Explicit temporal coherence constraints might be introduced to the *MASC* system through the on-line learning of shape models, similar to the 'active shape model' (ASM) concept [13]. Although the ASM approach requires training on the target *IMO*s, the notion of continuous on-line learning and weakly enforcing the learned shape models during a sequence provide a natural means to incorporate temporal coherence principles into the *MASC* system.

Kalman filtering techniques provide an immediate means of integrating temporal coherence into the active contour boundary recovery. Furthermore, the *CONDENSATION* contour tracker [26] introduces non-Gaussian density propagation techniques that improve boundary recovery performance in cluttered environments. In addition, by using the *MASC* system's estimated *IMO* motion parameters to replace the trained dynamical model of original *CONDENSATION* tracker, the resulting revised *MASC* implementation provide strong temporal coherence as well

as spatial coherence, and this suggestion has been implemented recently [11]. Future work aims to incorporate weak shape priors to this model learned on-line using the active contour shape recovery.

Finally, the introduction of active contours with inherent topological independence [10, 42] would allow our boundary representation to dynamically split and merge to accommodate motion coherent *IMO*s, so that a separate spatial segmentation component may no longer be necessary.

MASC System Research Suggestions

One of the primary limitations of the current *MASC* system implementation is the lack of multiple object handling routines, dynamically managing the 'birth' and 'death' of *IMO*s in the scene over time. One key *IMO* management aspect that should be incorporated in such a module is the merging and splitting of *IMO*s, for example in a scene where an *IMO*'s parts stop articulating while the *IMO* continues to undergo rigid motion. The previously independently moving parts of the *IMO* should then be merged with the rest of the main *IMO*, and then they should be split off once again if they begin to move independently later. Along similar lines, an improved *IMO* management system might also incorporate a probabilistic framework to continuously infer the transparency and visibility of *IMO*s, along the lines of Jepson *et al.* [30]. Such a framework might attempt to explicitly address transparency and visibility, improving the performance of the system when *IMO*s in the scene are occluded.

In more practical terms, the current performance limitations of the *MASC* system, suggest that an implementation in a compiled programming language (as opposed to the current interpreted MATLAB scripts) such as C or C++ could provide immediate performance benefits. However, as the current *MASC* implementation requires seven to ten seconds per frame for analysis, even a speedup of two orders of magnitude would only approach performing real-time segmentation at the current capture rate of thirty frames per second. Improvements on the order of nearly three orders of magnitude would be required to operate the *MASC* system

at typical video frame rates of around thirty frames per second. This level of performance might be achieved through implementation in re-programmable hardware systems, in a similar manner to that demonstrated by Darabiha *et al.* [16] shown for stereo depth measurement.

Beyond incremental improvements, the true utility of the *MASC* system will be demonstrated by its integration with higher-level components that use the segmentation, tracking and shape information that the *MASC* system recovers. Recognition components can use the shape and localization information to simplify their tasks, and add robustness to their execution. Positive recognition matches can be fed back to the *MASC* system to improve segmentation performance, providing even stronger spatial priors for both the motion and shape recovery components. The results from such a combination of the *MASC* segmentation and recognition could provide the shape, location, motion, identity and perhaps even the pose of the target *IMO*.

10.4 Discussion

The segmentation framework proposed by this thesis embodies the idea that motion estimation and shape recovery require a solution that addresses both problems in unison. While spatial regularization techniques have been a part of even the earliest motion estimation techniques, the *MASC* system introduces a shape constraint for motion estimation that does not require prior information about the *IMO*s it segments, and does not restrict shape representation to a low-dimensional parametric shape descriptor. The active contour used by the *MASC* system permits the representation of arbitrary shape, working under the assumption that *IMO*s generally possess spatial coherence and compactness. This research emphasizes the importance of accurate shape recovery for motion estimation, and presents a functional implementation in the *MASC* system, taking another step toward endowing computer vision systems with effective visual awareness of the world.

Bibliography

[1] J.L. Barron, D.J. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal Of Computer Vision*, 12(1), 1994.

[2] B. Bascle, P. Bouthemy, R. Deriche, and F. Meyer. Tracking complex primitives in an image sequence. In *Proceedings of the IAPR International Conference On Pattern Recognition*, pages 426–431, Jerusalem, Israel, October 1994.

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

[4] A. Baumberg and D.C. Hogg. Learning flexible models from image sequences. In *Proceedings of the European Conference on Computer Vision*, Stockholm, 1994. Springer-Verlag.

[5] S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.

[6] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, pages 237–252, Santa Margherita, Italy, May 1992.

[7] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 231–236, Berlin, Germany, May 1993.

[8] M.J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1), January 1996.

[9] A. Blake and M. Isard. *Active Contours*. Springer-Verlag London Limited, 1998.

[10] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the IEEE International Conference on Computer Vision*, Cambridge, Massachusetts, June 1995.

[11] D. Chung Lin Cheung, W.J. MacLean, and S. Dickinson. Integrating region and boundary information for improved spatial coherence in object tracking. Submitted for publication review to ECCV 2004.

[12] 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.

[13] T.F. Cootes, C.J. Taylor, and D.H. Cooper et al. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[14] D. Cremers. A variational framework for image segmentation combining motion estimation and shape regularization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 53–58, Madison, June 2003.

[15] D. Cremers and S. Soatto. Variational space-time motion segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, Nice, 2003.

[16] A. Darabiha, J. Rose, and W.J. MacLean. Video-rate stereo depth measurement on programmable hardware. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, June 2003.

[17] T. Darrell and A. Pentland. Robust estimation of a multilayered motion representation. In *Proc. IEEE Workshop on Visual Motion*, Princeton, New Jersey, October 1991.

[18] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag London Limited, 2000.

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

[20] D.J. Fleet and A.D. Jepson. Computation of component image velocity from local phase information. *International Journal Of Computer Vision*, 5(1), August 1990.

[21] D.J. Fleet and A.D. Jepson. Stability of phase information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12), 1993.

[22] M. Gelgon and P. Bouthemy. A region-level graph labeling approach to motion-based segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 1997.

[23] F.C. Glazer. Hierarchical gradient-based motion detection. In *DARPA Proceedings of Image Understanding Workshop*, pages 733–748, Los Angeles, California, February 1987.

[24] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 16, 1981.

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

[26] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proceedings of the European Conference on Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[27] M.R.M Jenkins, A.D. Jepson, and J.K. Tsotos. Techniques for disparity measurement. *Computer Vision Graphics and Image Understanding: Image Understanding*, 53(1), 1991.

[28] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust on-line appearance models for visual tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, Kauai, December 2001.

[29] A.D. Jepson and M. Black. Mixture models for optical flow. Technical Report RBCV-TR-93-44, Department Of Computer Science, Univ. of Toronto, 1993. Res. in Biol. and Comp. Vision.

[30] A.D. Jepson, D.J. Fleet, and M.J. Black. A layered motion representation with occlusion and compact spatial support. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 692–706. Springer-Verlag London Limited, 2002.

[31] 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.

[32] S.X. Ju, M.J. Black, and A.D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, October 1996.

[33] D.S. Kalivas and A.A. Sawchuk. A region matching motion estimation algorithm. *Computer Vision Graphics and Image Understanding: Image Understanding*, 54(2), 1991.

[34] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the IEEE International Conference on Computer Vision*, London, 1987.

[35] B. Leroy, I.L. Herlin, and L.D. Cohen. Multi-resolution algorithms for active contour models. In *Proceedings of the 12th International Conference on Analysis and Optimization of Systems Images, Wavelets and PDE'S*, 1996.

[36] F. Leymarie and M.D. Levine. Tracking deformable objects in the plane using an active contour model. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 15, June 1993.

[37] B.D. Lucas and T. Kanade. An iterative image-registration technique with an application to stereo vision. In *Proceedings of International Joint Conferences of Aritificial Intelligence*, Vancouver, British Columbia, 1981.

[38] Steve Mann, Corey Manders, and James Fung. Painting with looks: photographic images from video using quantimetric processing. In *Proceedings of the 10th ACM International Conference on Multimedia*, Juan les Pins, France, December 2002.

[39] G.J. McLachlan and K.E. Basford. *Mixture models: Inference and applications to clustering*. Marcel Dekker, Inc., New York, 1988.

[40] H-H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33(3), November 1987.

[41] N. Paragios and R. Deriche. Geodesic active regions for motion estimation and tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 688–694, Corfu, Greece, 1999.

[42] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3), March 2000.

[43] S. Sclaroff and J. Isidoro. Active blobs. In *Proceedings of the IEEE International Conference on Computer Vision*, January 1998.

[44] L.G. Shapiro and G.C. Stockman. *Computer Vision*. Prentice-Hall Inc., New Jersey, 2001.

[45] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, June 1994.

[46] P. Smith, T. Drummond, and R. Cipolla. Segmentation of multiple motions by edge tracking between two frames. In *Proceedings of the British Machine Vision Conference*, 2000.

[47] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 1999.

[48] H. Tao, H.S. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 134–142, 2000.

[49] D. Terzopoulos and R. Szeliski. *Tracking With Kalman Snakes*, chapter 1, pages 3–20. Springer-Verlag London Limited, 1998.

[50] E. Trucco and A. Verri. *Introductory Techniques For 3-D Computer Vision*. Prentice-Hall Inc., New Jersey, 1998.

[51] J. Y. A Wang and E. H. Adelson. Representing moving images with layers. *The IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.

[52] J. Y. A. Wang and E. H. Adelson. Spatio-temporal segmentation of video data. In *Proceedings of SPIE on Image and Video Processing II*, volume 2182, pages 120–131, February 1994.

[53] Y. Weiss and E.H. Adelson. Perceptually organized em: A framework for motion segmentation that combines information about form and motion. Technical Report 315, Massachussetts Institute of Technology Media Laboratory Perceptual Computing Section, 1995.

[54] 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 the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 321–326, 1996.

[55] D.J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision Graphics and Image Understanding: Image Understanding*, 55(1), January 1992.

[56] Chenyang Xu and Jerry L. Prince. Snakes, shapes and gradient vector flow. *IEEE Transactions on Image Processing*, March 1998.