Layered Motion Segmentation and Depth Ordering by Tracking Edges

Paul Smith, *Member*, *IEEE Computer Society*, Tom Drummond, *Member*, *IEEE Computer Society*, and Roberto Cipolla, *Member*, *IEEE*

Abstract—This paper presents a new Bayesian framework for motion segmentation—dividing a frame from an image sequence into layers representing different moving objects—by tracking edges between frames. Edges are found using the Canny edge detector, and the Expectation-Maximization algorithm is then used to fit motion models to these edges and also to calculate the probabilities of the edges obeying each motion model. The edges are also used to segment the image into regions of similar color. The most likely labeling for these regions is then calculated by using the edge probabilities, in association with a Markov Random Field-style prior. The identification of the relative depth ordering of the different motion layers is also determined, as an integral part of the process. An efficient implementation of this framework is presented for segmenting two motions (foreground and background) using two frames. It is then demonstrated how, by tracking the edges into further frames, the probabilities may be accumulated to provide an even more accurate and robust estimate, and segment an entire sequence. Further extensions are then presented to address the segmentation of more than two motions. Here, a hierarchical method of initializing the Expectation-Maximization algorithm is described, and it is demonstrated that the Minimum Description Length principle may be used to automatically select the best number of motion layers. The results from over 30 sequences (demonstrating both two and three motions) are presented and discussed.

Index Terms—Video analysis, motion, segmentation, depth cues.

1 Introduction

Motion is an important cue in vision, and the analysis of the motion between two images, or across a video sequence, is a prelude to many further areas in computer vision. Where there are different moving objects in the scene, or objects at different depths, motion discontinuities will occur and these provide information essential to the understanding of the scene. Motion segmentation (the division of a video frame into areas obeying different image motions) provides this valuable information.

With the current boom in digital media, motion segmentation finds itself a number of direct applications. Video compression becomes increasingly important as consumers demand higher quality for less bandwidth, and here motion segmentation can provide assistance. By detecting and separating the moving objects from the background, coding techniques can apply different coding strategies to the different elements of the scene. Typically, the background changes less quickly, or is less relevant than the foreground action, so can be coded at a lower bit rate. Mosaicing of the background [1], [2] provides another compact representation. The MPEG-4 standard [3] explicitly describes a sequence in terms of objects moving in front of a background image and, while initially designed for multimedia presentations, motion segmentation may be used to also encode real video in this manner.

Another relatively new field is that of video indexing [4], [5], where the aim is to automatically classify and retrieve video sequences based on their content. The segmentation of the moving objects enables these objects and the background to be analyzed independently. Classification, both on low-level image and motion characteristics of the scene components, and on higher-level semantic analysis can then take place.

1.1 Review of Previous Work

Many popular approaches to motion segmentation revolve around analyzing the per-pixel optic flow in the image. Optic flow techniques, such as the classic work by Horn and Schunk [6], use spatiotemporal derivatives of the pixel intensities to provide a motion vector at each pixel. Because of the aperture problem, this motion vector can only be determined in the direction of the local intensity gradient, and so in order to determine the complete field it is assumed that the motion is locally smooth.

Analyzing this optic flow field is one approach to motion segmentation. Adiv [7] clustered together pixels with similar flow vectors and then grouped these into segments obeying the same 3D motion; Murray and Buxton [8] followed a similar technique. However, the smoothing required by optic flow algorithms renders the flow fields highly unreliable both in areas of low gradient (into which results from other areas spread), and when there are multiple motions. The case of multiple motions is particularly troublesome, since the edges of moving objects create discontinuities in the flow field, and after smoothing the localization of these edges is difficult. It is unfortunate that these are the very edges that are required for a motion segmentation. One solution to this smoothing problem is to apply the smoothing in a piecewise fashion. Taking a small area, the flow can be analyzed to determine whether it best

- *The authors are with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK.*
E-mail: {pas1001, twd20, cipolla}@eng.cam.ac.uk.

Manuscript received 30 Oct. 2000; revised 24 Mar. 2002; accepted 13 Apr. 2003.
Recommended for acceptance by M. Irani.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 113077.

0162-8828/04/$20.00 © 2004 IEEE Published by the IEEE Computer Society

fits one smooth motion or a pair of motions, and these patches in the image can be marked and treated accordingly (e.g., [9], [10], [11]).

The most successful approaches to motion segmentation consider parameterizing the optic flow field, fitting a different model (typically 2D affine) to each moving object. Pixels can then be labeled as best fitting one model or another. This is referred to as a *layered representation* [10] of the motion field, since it models pixel motions as belonging to one of several layers. Each layer has its own, smooth, flow field, while discontinuities can occur between layers. Each layer represents a different object in the sequence, and so the assignment of pixels to layers also provides the motion segmentation.

There are two main approaches to determining the contents of the layers, of which the *dominant motion* approach (e.g., [1], [12], [13], [14], [15]) is the most straightforward. Here, a single motion is robustly fitted to all pixels, which are then tested to see whether they really fit that motion (according to some metric). The pixels which agree with the motion are labeled as being on that layer. At this stage, either this layer can be labeled as "background" (being the dominant motion), and the outlier pixels as belonging to foreground objects [14], [15], or the process can be repeated recursively on the remaining pixels to provide a full set of layers for further analysis [1], [12], [13].

The other main approach is to determine all of the motions simultaneously. This can either be done either by estimating a large number of motions, one for each small patch of the image, and then merging similar motions (typically by k-means clustering) [10], [11], [16], or by using the Expectation-Maximization (EM) algorithm [17] to simultaneously estimate motions and find the pixel labels [2], [18], [19]. The number of motions also has to be determined. This is usually done by either setting a smoothing factor and merging convergent models [19], or by considering the size of the model under a Minimum Description Length framework [18].

Given a set of motions, assigning pixels to layers requires determining which motion they best fit, if any. This can be done by comparing the pixel color or intensities under the proposed motions, but this presents several problems. Pixels in areas of smooth intensity are ambiguous as they can appear similar under several different motions and, as with the optic flow techniques, some form of smoothing is required to identify the best motion for these regions. Pixels in areas of high intensity gradient are also troublesome, as slight errors in the motion estimate can yield pixel of a very different color or intensity, even under the correct motion. Again, some smoothing is usually required. A common approach is to use a Markov Random field [20], which encourages pixels to be labeled the same as their neighbors [14], [15], [19], [21]. This works well at ensuring coherent regions, but can often also lead to the foreground objects "bleeding" over their edge by a pixel or two.

All of the techniques considered so far try to solve the motion segmentation problem using only motion information. This, however, ignores the wealth of additional information that is present in the image intensity structure. The image structure and the pixel motion can both be considered at the same time by assigning a combined score to each pixel and then finding the optimal segmentation based on all these properties, as in Shi and Malik's Normalized Cuts framework [22], but these approaches tend to be computationally expensive. A more efficient approach is that of *region merging*, where an image is first segmented solely according to the image structure, and then objects are identified by merging regions with the same motion. This implicitly resolves the problems identified earlier which required smoothing of the optic flow field, since the static segmentation process will group together neighboring pixels of similar intensity so that all the pixels in an area of smooth intensity, being grouped in the same region, will be labeled with the same motion. Regions will be delimited by areas of high gradient (edges) in the image and it is at these points that changes in the motion labeling may occur.

As with the per-pixel optic flow methods, the region-merging approach has several methods of simultaneously finding the motions and labeling the regions. Under the dominant-motion method (e.g., [4], [12], [23]), a single parametric motion is robustly fitted to all the pixels and then regions which agree with this motion are segmented as one layer and the process repeated on the rest. Alternatively, a different motion may be fitted to each region and then some clustering performed in parameter space to group regions with similar motions [24], [25], [26], [27], [28], [29]. The EM algorithm is also a good choice when faced with this type of estimation problem [19].

The final segmentation from all of these motion segmentation schemes is a labeling of pixels, each into one of several layers, together with the parameterized motion for each layer. What is not generally considered is the relative depth ordering of each of these layers, i.e., which is the background and which are foreground objects. If necessary, it is sometimes assumed that the largest region or the dominant motion is the background. Occlusion is commonly considered, but only in terms of a problem which upsets the pixel matching and so requires the use of robust methods. However, this occlusion may be used to identify the layer ordering as a postprocessing stage. Wang and Adelson [10], and Bergen and Meyer [29], identify the occasions when a group of pixels on the edge of a layer are outliers to the layer motion and use these to infer that the layer is being occluded by its neighbor. Tweed and Calway [30] use similar occlusion reasoning around the boundaries of regions as part of an integrated segmentation and ordering scheme.

Depth ordering has recently begun to be considered as an integral part of the segmentation process. Black and Fleet [31] have modeled occlusion boundaries directly by considering the optic flow in a small region and this also allows occluding edges to be detected and the relative ordering to be found. Gaucher and Medioni [32] also study the velocity field to detect motion boundaries and infer the occlusion relationships.

1.2 This Paper: Using Edges

This paper presents a novel and efficient framework for both motion segmentation and depth ordering using the motion of edges in the sequence. Previous researchers have found that that the motion of pixels in areas of smooth intensity is difficult to determine and that smoothing is required to resolve this problem, although this then

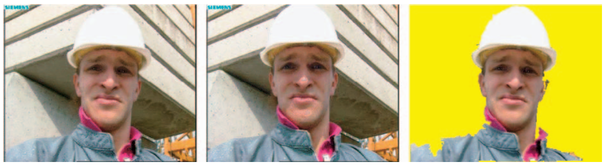


Fig. 1. "Foreman" example. Two frames from the "Foreman" sequence and the foreground layer of the desired segmentation. Two widely-separated frames are here shown only for clarity; this paper considers neighboring frames.

provides problems of its own. This paper ignores these areas initially, concentrating only on edges, and then follows a region-merging framework, labeling presegmented regions according to their motions. It is shown that the motion of these regions may be determined solely from the motion of their edges without needing to use the pixels in their smooth interior. A similar approach was used by Thompson [24], who also used only the motion of the edges of regions in estimating their motion. However, this is his only use of the edges, as a prelude to a standard region-merging approach. This paper shows that edges provide further information and, in fact, the clustering and labeling of the region edges provides all the information that can be known about the assignment of regions and also the ordering of the different layers.

This paper describes the theory linking the motions of edges and regions, and then develops a probabilistic framework which enables the most likely region labeling and layer ordering to be inferred from edge motions. This process may be performed over only two frames, but evidence can also be accumulated over a sequence to provide a more accurate and robust segmentation. The theoretical framework linking edges and regions is presented in Section 2. Section 3 develops a Bayesian formulation of this framework, and the basic implementation is presented in Section 4. This implementation is extended to use multiple frames in Section 5, and to segment multiple motions in Section 6. Results are given at the end of each of the implementation sections, while Section 7 draws some conclusions and outlines future directions for research.

2 Motion Segmentation Using Edges

Given frames from a sequence featuring moving objects, the task is to provide as an output a cut-out of the different objects, together with their relative depth ordering (see, for example, Fig. 1). The desired segmentation can be defined in terms of the pixels representing different objects or, alternatively, by the *edges* of the areas of the image representing the different objects. Edges are fundamental to the problem and it will be shown that the motion of the edges can be used to provide the solution.

Considering the pixels in Fig. 1, it can be noted that there are a number of areas of the image with very little variation in pixel color and intensity. No reliable motion information can be gained from these areas; it is the edges in the image which provide real motion information. (Texture can also give good motion information, but this provides a difficult matching problem.) Edges are very good features to consider for motion estimation: They can be found more reliably than corner features and their long extent means that a number of measurements may be taken along their length, leading to a more accurate estimation of their motion.

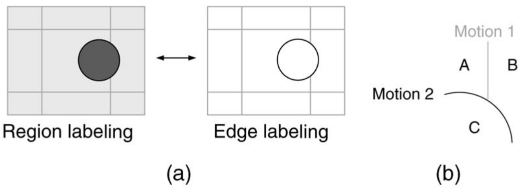


Fig. 2. Edges and Regions. (a) A region labeling and layer ordering (in this case black is on top) fully defines the edge labeling. The edge labeling can also give the region labeling. (b) T-junctions (where edges of different motion labelings meet) can be used to determine the layer ordering (see text).

Even when using edges, the task is also one of labeling regions since it is an enclosed area of the frame which must be labeled as a moving object. If it is assumed that the image is segmented into regions along edges, then there is a natural link between the regions and the edges.

2.1 The Theory of Edges and Regions

Edges in an image are generated as a result of the texture of objects, or their boundaries in the scene.[1] There are three fundamental assumptions made in this work, which are commonly made in layered-motion schemes, and will be valid in many sequences:

1. As an object moves all of the edges associated with that object move, with a motion which may be approximately described by some motion model.
2. The motions are layered, i.e., one motion takes place completely in front of another and the layers are strictly ordered. Typically, the layer farthest from the camera is referred to as the background with nearer foreground layers in front of this.
3. No one segmented image region belongs to two or more motion models and, hence, any occluding boundary is visible as an region edge in the image.

Given these assumptions, it is possible to state the relationship between the motions of regions and the motions of the edges that divide them. If the layer of each region is known, and the layer ordering is known, then the layer of each edge can be uniquely determined by the following rule:

- **Edge Labeling Rule.** The layer to which an edge belongs is that of the nearer of the two regions which it bounds.

The converse is not true. If only the edge labeling is known (and not the layer ordering), then this does not necessarily determine the region labeling or layer ordering. Indeed, even if the layer ordering is known, there may be multiple region labelings which are consistent with the edge labeling.

An example of a region and edge labeling is shown in Fig. 2a. On the left is shown a known region labeling, where

[1] Edges may also be generated as a result of material or surface properties (texture or reflectance). It is assumed that these do not occur but see the "Car" sequence in Section 4 for an example of the consequence of this assumption.

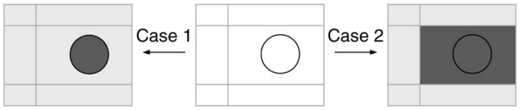


Fig. 3. Ambiguous edges and regions. If there is no interaction between the edges of the two objects, there are two possible interpretations of the central edge labeling. Either of the two motions could be foreground, resulting in slightly different region labeling solutions. In Case 1, the black circle is the foreground object. In Case 2, it is on the background (viewed through a rectangular window).

the dark circle is the foreground object. Since it is on top, all of its edges are visible and move with the foreground motion, labeled as black in the edge label image on the right. All of the edges of the gray background regions, except those that also bound the foreground region, move with the background motion and so are labeled as gray. The edge labeling is thus uniquely determined.

If, instead, the edge labeling is known (but not the layer ordering), it is still possible to make deductions about both the region labeling and the layer ordering. Regions which are bound by edges of different motions must be on a layer at least as far away as the furthest of its bounding edges (if it were nearer, its edges would occlude edges at that layer). However, for each edge, at least one of the regions that it divides must have the same layer as the edge. A region labeling can be produced from an edge labeling, but ambiguities may still be present—specifically, a single region in the middle of a foreground object may be a hole through to the background, although this is unlikely.

A complete segmentation also requires the layer ordering to be determined and, importantly, this can usually be determined from the edge labeling. Fig. 2b highlights a T-junction from the previous example, where edges with two different labeling meet. To determine which of the two motion layers is on top, both of the two possibilities are hypothesized and tested. Regions A and B are bounded by edges of two different motions, which can only occur when these regions are bounded by edges obeying their own motion and also an edge of the occluding object. These regions therefore must belong to the relative "background." The question is: Which of the two motions is the background motion? If it is hypothesized that the background motion is motion 2 (black), then these regions should be labeled as obeying motion 2, and the edge between them should also obey motion 2. However, it is already known that the edge between them obeys motion 1, so this cannot be the correct layer ordering. If motion 1 were background and motion 2 foreground, then the region labeling would be consistent with the edge labeling, indicating that this is the correct layer ordering.

Fig. 3 shows an ambiguous case. Here, there are no T-junctions and so the layer ordering cannot be determined. There are two possible interpretations, both consistent with the edge labeling. Cases such as these are ambiguous under any motion segmentation scheme and at least the system presented here is able to identify such ambiguities.

This section has shown that edges are not only a necessary element in an accurate motion segmentation, they are also sufficient. Edges can be detected in a frame, labeled with their motion, and then used to label the regions in between. In real images, it is not possible to determine an exact edge labeling and so instead the next section develops a probabilistic framework for performing this edge and region labeling.

3 Bayesian Formulation

There are a large number of parameters which must be solved to give a complete motion segmentation and for which the most likely values must be estimated. Given that the task is one of labeling, the regions of a static segmentation, finding their motion and determining the layer ordering, the complete model of the segmentation M consists of the elements $M = \{\Theta, F, R\}$, where

- Θ is the parameters of the motion models,
- F is the foreground-background ordering of the motion layers, and
- R is the motion label (layer) for each region.

The region edge labels are not an independent part of the model, but are completely defined by R and F from the Edge Labeling Rule of Section 2.

Given the image data D (and any other prior information assumed about the world), the task is to find the model M with the maximum probability given this data and priors:

$$\arg\max_{\boldsymbol{M}} \mathrm{P}(\boldsymbol{M}|\boldsymbol{D}) = \arg\max_{\boldsymbol{RF\Theta}} \mathrm{P}(\boldsymbol{RF\Theta}|\boldsymbol{D}). \tag{1}$$

This can be further decomposed, without any loss of generality, into a motion estimation component and region labeling:

$$\arg\max_{\boldsymbol{RF\Theta}} \mathrm{P}\left(\boldsymbol{RF\Theta}|\boldsymbol{D}\right) = \arg\max_{\boldsymbol{RF\Theta}} \mathrm{P}\left(\boldsymbol{\Theta}|\boldsymbol{D}\right) \mathrm{P}\left(\boldsymbol{RF}|\boldsymbol{\Theta D}\right). \tag{2}$$

At this stage, a simplification is made: It is assumed that the motion parameters Θ can be maximized independently of the others, i.e., the correct motions can be estimated without knowing the region labeling (just from the edges). This relies on the richness of edges available in a typical frame and the redundancy this provides. This motion estimate approaches the global maximum but, if desired, a global optimization may be performed once an initial set of motions and region labeling has been found; this is discussed in Section 6. Given this simplifying assumption, the expression to be maximized is:

$$\underbrace{\arg\max_{\boldsymbol{\Theta}} \mathrm{P}\left(\boldsymbol{\Theta}|\boldsymbol{D}\right)}_{\mathrm{a}} \underbrace{\arg\max_{\boldsymbol{RF}} \mathrm{P}\left(\boldsymbol{RF}|\boldsymbol{\Theta D}\right)}_{\mathrm{b}}, \tag{3}$$

where the value of Θ used in term b is that which maximizes term a. The two components of (3) can be evaluated in turn: first a, the motions, and then b, the region labeling and layer ordering.

3.1 Estimating the Motions Θ

The first term in (3) estimates the motions between frames (Θ encapsulates the parameters of all the motions). Thus far, this statistical framework has not specified how the most likely motion is estimated and neither are edges included. As explained in Section 2, edges are robust features to track, and they provide a natural link to the regions which are to be labeled. The labeling of edges must be introduced into the statistical model: They are expressed by the random variable e which gives, for each edge, the probability of it obeying each motion. This is a necessary variable, since in order to estimate the motion models from the edges it must be known which edges belong to which motion. However, simultaneously labeling the edges and fitting motions is a circular problem, which may be resolved by expressing the estimation of Θ and e in terms of the Expectation-Maximization algorithm [17], with e as the hidden variable. This is expressed by the following equation:

$$\arg\max_{\Theta_{n+1}} \sum_{\boldsymbol{e}} \log \mathrm{P}(\boldsymbol{e}\boldsymbol{D}|\boldsymbol{\Theta}_{n+1})\ \mathrm{P}\ (\boldsymbol{e}|\boldsymbol{\Theta}_n\boldsymbol{D}). \tag{4}$$

This iterates between two stages: The E-stage computes the expectation, which forms the bulk of this expression (the main computation work here is in calculating the edge probabilities $\mathrm{P}(\boldsymbol{e}|\boldsymbol{\Theta}_n\boldsymbol{D})$) and the M-stage then maximizes this expression, performing the maximization of (4) over $\boldsymbol{\Theta}_{n+1}$. Some suitable initialization is used and then the two stages are iterated to convergence, which has the effect of maximizing (3a). An implementation of this is outlined in Section 4.

3.2 Estimating the Labelings R and F.

Having obtained the most likely motions, the remaining parameters of the model $\boldsymbol{M}$ can be maximized. These are the region labeling $\boldsymbol{R}$ and the layer ordering $\boldsymbol{F}$, which provide the final segmentation. Once again, the edge labels are used as an intermediate step. Given the motions $\boldsymbol{\Theta}$, the edge label probabilities $\mathrm{P}(\boldsymbol{e}|\boldsymbol{\Theta}\boldsymbol{D})$ can be estimated, and from Section 2 the relationship between edges and regions is known. Term (3b) is augmented by the edge labeling $\boldsymbol{e}$, which must then be marginalized, giving

$$\arg\max_{\boldsymbol{RF}} \mathrm{P}(\boldsymbol{RF}|\boldsymbol{\Theta}\boldsymbol{D}) = \arg\max_{\boldsymbol{RF}} \sum_{\boldsymbol{e}} \mathrm{P}\ (\boldsymbol{RF}|\boldsymbol{e}\boldsymbol{\Theta}\boldsymbol{D})\ \mathrm{P}\ (\boldsymbol{e}|\boldsymbol{\Theta}\boldsymbol{D}) \tag{5}$$

$$= \arg\max_{\boldsymbol{RF}} \sum_{\boldsymbol{e}} \mathrm{P}\ (\boldsymbol{RF}|\boldsymbol{e})\ \mathrm{P}\ (\boldsymbol{e}|\boldsymbol{\Theta}\boldsymbol{D}), \tag{6}$$

where the first expression in (5) can be simplified since $\boldsymbol{e}$ encapsulates all of the information from Θ and $\boldsymbol{D}$ that is relevant to determining the final segmentation $\boldsymbol{R}$ and $\boldsymbol{F}$, as shown in Section 2.

The second term, the edge probabilities, can extracted directly from the motion estimation stage—it is used in the EM algorithm. The first term is more difficult to estimate, and it is easier to recast this using Bayes' Rule, giving

$$\mathrm{P}(\boldsymbol{RF}|\boldsymbol{e}) = \frac{\mathrm{P}\ (\boldsymbol{e}|\boldsymbol{RF})\ \mathrm{P}\ (\boldsymbol{RF})}{\mathrm{P}\ (\boldsymbol{e})}. \tag{7}$$

The maximization is over $\boldsymbol{R}$ and $\boldsymbol{F}$, so $\mathrm{P}(\boldsymbol{e})$ is constant. The prior probabilities of $\boldsymbol{R}$ and $\boldsymbol{F}$ are independent, since whether a particular layer is called "motion 1" or "motion 2" does not change its labeling. Any foreground motion is equally likely, so $\mathrm{P}(F)$ is constant, but the last term, $\mathrm{P}(R)$, is not constant. This term is used to encode likely labeling configurations since some configurations of region labels are more likely than others.[2] This leaves the following expression to be evaluated:

$$\arg\max_{\boldsymbol{RF}} \sum_{\boldsymbol{e}} \mathrm{P}\ (\boldsymbol{e}|\boldsymbol{RF})\ \mathrm{P}\ (R)\ \mathrm{P}\ (\boldsymbol{e}|\Theta D). \tag{8}$$

The $\mathrm{P}\ (\boldsymbol{e}|\boldsymbol{RF})$ term is very useful. The edge labeling $\boldsymbol{e}$ is only an intermediate variable, and is entirely defined by the region labeling $\boldsymbol{R}$ and the foreground motion $\boldsymbol{F}$. This probability, therefore, takes on a binary value—it is 1 if that edge labeling is implied and 0 if it is not. The sum in (8) can thus be removed and the $\boldsymbol{e}$ in the final term replaced by the function $\boldsymbol{e}(\boldsymbol{R}, \boldsymbol{F})$, which provides the correct edge labels for given values of $\boldsymbol{R}$ and $\boldsymbol{F}$.

$$\arg\max_{\boldsymbol{RF}} \underbrace{\mathrm{P}(\boldsymbol{e}(R, F)|\boldsymbol{\Theta}\boldsymbol{D})}_{a} \underbrace{\mathrm{P}(R)}_{b}. \tag{9}$$

The variable $\boldsymbol{F}$ takes only a discrete set of values (for example, in the case of two layers, only two: either one motion is foreground, or the other). Equation (9) can therefore be maximized in two stages: $\boldsymbol{F}$ can be fixed at one value and the expression maximized over $\boldsymbol{R}$ and the process then repeated with other values of $\boldsymbol{F}$ and the global maximum taken.[3] The maximization over $\boldsymbol{R}$ can be performed by hypothesizing a complete region labeling and then testing the evidence (9a)—determining the implied edge labels and then calculating the probability of the edge labeling given the motions —and the prior (9b), calculating the likelihood of that particular labeling configuration. An exhaustive search is impractical and, in the implementation presented in Section 4, region labelings are hypothesized using simulated annealing. Maximizing this expression is identical to maximizing (3b) and is the last stage of the motion segmentation algorithm: The most likely $\boldsymbol{R}$ and $\boldsymbol{F}$ represent the most likely region labeling and layer ordering.

4 Implementation for Two Motions, Two Frames

The Bayesian framework presented in Section 3 leads to an efficient implementation. This section describes how a video frame may be divided into two layers (foreground and background) using the information from one more frame. This is a common case and also the simplest motion segmentation situation. Many of the details in this two motion, two frame case apply to more general cases, which are mostly simple extensions. Sections 5 and 6 cover the multiple-frame and multiple-motion cases, respectively.

The system progresses in two stages, as demonstrated in Fig. 4. The first is to detect edges, find motions and label the edges according to their probability of obeying each motion. These edge labels are sufficient to label the rest of the image. In the second stage the frame is divided into regions of similar color using these edges and the motion labeling for these regions which best agrees with the edge labeling is then determined.

4.1 Estimating the Motions Θ and Edge Labels e

As explained in Section 2, edges are fundamental to the segmentation problem, and also provide the only robust source of motion information. The motion segmentation approach proposed in this paper begins with finding edge

[2] For example, individual holes in a foreground object are unlikely. This prior enables the ambiguous regions mentioned in Section 2 to be given their most likely labeling.

[3] This stage is combinatorial in the number of layers. This presents difficulties for sequences with many layers, but there are many real sequences with a small number of motions (for example, 34 sequences are considered in this work, all with two or three layers).

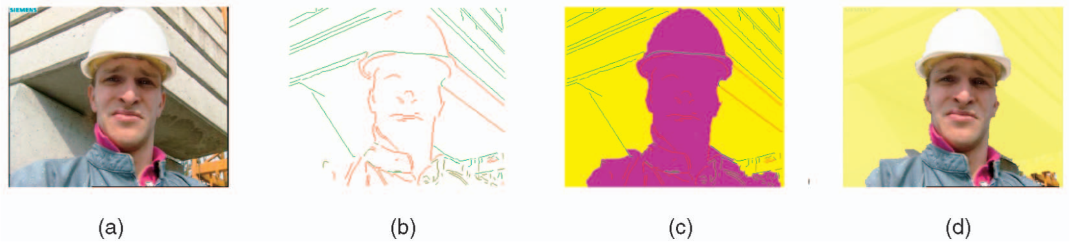


Fig. 4. Foreman segmentation from two frames. (a) Frame 1. (b) Edges labeled by their motion (the color blends from red (motion 1) to green (motion 2) according to the probability of each motion). (c) Maximum a posteriori region labeling. (d) Final foreground segmentation.

chains in the frame, in this case using the Canny edge detector [33] followed by the grouping of edgels into chains (e.g., Fig. 4b). The motions of these edge chains must then be found so that they can be assigned to clusters belonging to each of the different moving objects. The object and background image motions are here modeled by 2D affine transformations, which have been found by many to be a good approximation to the small interframe motions [10], [14].

Multiple-motion estimation is a circular problem. If it were known which edges belonged to which motion, these could be used to directly estimate the motions. However, edge motion labeling requires making a comparison between different known motions. In order to resolve this, Expectation-Maximization (EM) is used [17], implementing the formulation (4) as described below.

4.1.1 Maximization: Estimating the Motions

If the edge label probabilities $\mathrm{P}(e\Theta_n|D)$ are known, (4) can be maximized, and here the expression $\log \mathrm{P}\ (eD|\Theta_{n+1})$ is estimated and maximized using techniques derived from group-constrained snake technology [34]. For each edge, sample points are assigned at regular intervals along the edge (see Fig. 5a). The motion of these sample points are considered to be representative of the edge motion (there are about 1,400 sample points in a typical frame). The sample points from the first frame are mapped into the next (either in the same location or, in further iterations, according to the current motion estimate), and a search is made for the true edge location. Because of the aperture problem, the motion of edges can only be determined in a direction normal to the edge, but this is useful as it restricts the search for a matching edge pixel to a fast one-dimensional search along the edge normal.

To find a match, color image gradients are estimated in both the original image and the proposed new location using a 5×5 convolution kernel in the red, green, and blue components of the image. The match score is taken to be the sum of squared differences over the three colors, in both the x and y directions. The search is made over the pixels normal to the sample point location in the new image, to a maximum distance of 20 pixels.[4] The image distance d^k, between the original location and its best match in the next image, is measured (see Fig. 5b). If the score is below a threshold, "no match" is returned instead.

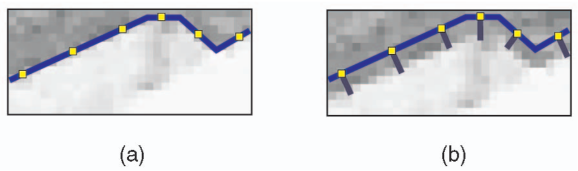


Fig. 5. Edge tracking example. (a) Edge in initial frame, with sample points. (b) In the next frame, where the image edge has moved, a search is made from each sample point normal to the edge to find the new location. The best-fit motion is the one that minimizes the squared distance error between the sample points and the image edge.

At each sample point the expected image motion due to a 2D affine motion $\boldsymbol{\theta}$ can be calculated. A convenient formulation uses the Lie algebra of image transformations [34]. According to this, transformations in the General Affine group GA(2) may be decomposed into a linear sum of the following generator matrices:

$$\boldsymbol{G}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{G}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{G}_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
$$\boldsymbol{G}_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{G}_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{G}_6 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{10}$$

These act on homogeneous image coordinates $(x \quad y \quad 1)^T$, and are responsible for the following six motion fields in the image:

$$\boldsymbol{L}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \boldsymbol{L}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \boldsymbol{L}_3 = \begin{pmatrix} -y \\ x \end{pmatrix}$$
$$\boldsymbol{L}_4 = \begin{pmatrix} x \\ y \end{pmatrix} \quad \boldsymbol{L}_5 = \begin{pmatrix} x \\ -y \end{pmatrix} \quad \boldsymbol{L}_6 = \begin{pmatrix} y \\ x \end{pmatrix}. \tag{11}$$

The task is to estimate the amount, α_i, of each of these deformation modes.

Since measurements can only be taken normal to the edge, α_i may be estimated by minimizing the geometric distance between the measurements d^k and the projection of the fields onto the unit edge normal $\hat{\boldsymbol{n}}^k$, over all of the sample points on that edge, or set of edges

$$\sum_k \left(d^k - \textstyle\sum_j \alpha_j ({\boldsymbol{L}_{\boldsymbol{j}}}^k \cdot \hat{\boldsymbol{n}}^k) \right)^2, \tag{12}$$

which is the negative log probability of $\log \mathrm{P}(\boldsymbol{e}\boldsymbol{D}|\Theta_{n+1})$, from (4), given an independent Gaussian statistical model. This expression may be minimized by using the singular value decomposition to give a least squares fit. In practice,

[4] Testing has revealed that the typical maximum image motion is of the order of 10 pixels, so this is a conservative choice. An adaptive search interval, or a multiresolution approach, would be appropriate in more extreme cases.

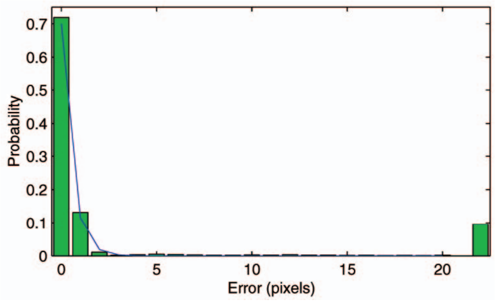


Fig. 6. Distribution of sample point measurement errors d^k. The probability of “no match” is shown on the far right. A Laplacian distribution is overlaid, showing a reasonable match.

reweighted least squares [35] is used to provide robustness to outliers, using the weight function

$$w(x) = \frac{1}{1+|x|}, \qquad (13)$$

(for a full description, see [36]). This corresponds to using a Laplacian (i.e., non-Gaussian) model for the errors, and is chosen because it gives a good fit to the observed distribution (see Fig. 6). Having found the α_i, an image motion $\boldsymbol{\theta}$ is then given by the same linear sum of the generators:

$$\boldsymbol{\theta} = \boldsymbol{I} + \alpha_i \boldsymbol{G}_i. \qquad (14)$$

To implement the M-stage of the EM algorithm (4), (12) is also weighted by $\mathrm{P}(\boldsymbol{e}|\Theta_n D)$ and then minimized to obtain the parameters of each motion $\boldsymbol{\theta}_{n+1}$ in turn. These are then combined to give Θ_{n+1}.

4.1.2 Expectation: Calculating Edge Probabilities

The discrete probability distribution $\mathrm{P}(\boldsymbol{e}|\Theta D)$ gives the probability of an edge fitting a particular motion from the set of motions $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$, and the E-stage of (4) involves estimating this. This can be done by considering the sample points used for motion estimation (for example, in Fig. 5, the first edge location with zero residual errors is far more likely than the second one). It may be assumed that the residual errors from sample points are representative of the whole edge, and that these errors are independent.[5] The likelihood that the edge fits a given motion is thus the product of the likelihood of a correct match at each sample point along the edge. Given Θ, the sample points are matched under each motion and each edge likelihood calculated. Normalizing these gives the probability of each motion.

The distribution of sample point measurement errors d^k has been extracted from sample sequences where the motion is known. The sample points are matched in their correct location and their errors measured, giving the distribution shown in Fig. 6. This histogram is used as the model when calculating the likelihood of a correct match for a sample point given a residual error or the fact that no match was found.

4.1.3 Initialization

The EM iteration needs to be started with some suitable initialization. Various heuristic techniques have been tried, for example initializing with the mean motion and the zero motion, but it is found that (in the two motion case, at least) the optimization well is sufficiently large for a random initialization to be used to begin the EM. An initial match for all the sample points is found in frame 2 by searching for 20 pixels normal to the edge. The edges are then randomly divided into two groups, and the sample points from the two groups are used to estimate two initial motions and the EM begins at the E-stage. The advantage of a random initialization is that it provides, to a high probability, two motions which are plausible across the whole frame, giving all edges a chance to contribute an opinion on both motions.

When using multiple frames (see Section 5), the initialization is an estimate based on the previous motion and the motion velocity in the previous frame. In the case where the number of motions is more than two, or is unknown, a more sophisticated initialization technique is used, as outlined in Section 6.

4.1.4 Convergence

The progress of the algorithm is monitored by considering the total likelihood of the most likely edge labeling, i.e.,

$$\prod_{\text{edges}} \max_{j} \mathrm{P}\ (\text{Edge is motion } j|\Theta \boldsymbol{D}), \qquad (15)$$

where these probabilities are taken from $\mathrm{P}(\boldsymbol{e}|\Theta D)$. This likelihood increases as the algorithm progresses (although not strictly monotonically) and then levels out. It is common for some edges to be ambiguous and for these to oscillate somewhat between the two motions, even after convergence. It is sufficient to declare convergence when the likelihood has not increased for 10 iterations, which usually occurs after 20-30 iterations. For a typical image of 352×240 pixels, this takes about three seconds on a 300MHz Pentium II.

4.2 Finding Edges and Regions

Having obtained the set of edges, and labeled these according to their motions, it is now time to build on these to label the rest of the pixels. First, a segmentation of the frame is needed, dividing the image into regions of the same color. The implementation presented here uses a scheme developed by Sinclair [38] (also used in [30]) but other edge-based schemes, such the morphological segmentation used in [29] or variants of the watershed algorithm [39], are also suitable.

Under Sinclair’s scheme, seed points for region growing initialized the locations furthest from the edges (taking the peaks of a distance transform of the edge image). Regions are then grown, gated by pixel color, until they meet, but with the image edges acting as hard barriers. Fig. 7 shows two example segmentations.

4.3 Labeling Regions R and Motions and F

Having obtained the regions, term (3b) (the region labeling and layer ordering) can be maximized given the motions Θ. According to (9), this can be performed by hypothesizing possible region and foreground motion labelings and

[5] This is not, in fact, the case but making this assumption gives a much simpler solution, while still yielding plausible statistics. See [36] for a discussion of the validity of this assumption and [37] for an alternative approach.

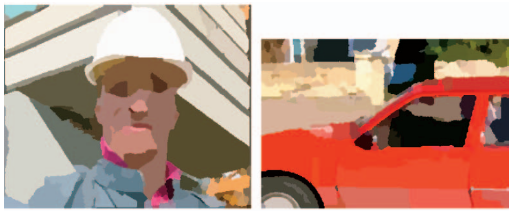


Fig. 7. Edge-based static segmentations of frames from the "Foreman" and "Car" sequences.

calculating their probabilities (9a), combining with a configuration prior (9b), and selecting the most probable.

4.3.1 Region Probabilities from Edge Data

The first term, (9a), calculates the probability of a region labeling and layer ordering given the data, $\mathrm{P}(\boldsymbol{e}(\boldsymbol{R},\boldsymbol{F})|\boldsymbol{\Theta D})$. First, the edge labels $\boldsymbol{e}(\boldsymbol{R},\boldsymbol{F})$ are computed using the Edge Labeling Rule from Section 2. This implies a labeling m^k for each sample point k in the frame: they take the same label as the edge to which they belong. Assuming independence of the sample points, the desired probability is then given by

$$\mathrm{P}(\boldsymbol{e}(\boldsymbol{R},\boldsymbol{F})|\boldsymbol{\Theta D} = \prod_k \mathrm{P}(m^k|\boldsymbol{\Theta D}). \tag{16}$$

The probability of a particular motion labeling for each sample point, $\mathrm{P}(m^k|\boldsymbol{\Theta D})$, was calculated earlier in the E-stage of EM. The likelihood of the data is that from Fig. 6, and is normalized according to Bayes' rule (with equal priors) to give the motion probability.

4.3.2 Region Prior

Term (9b) encodes the a priori region labeling, reflecting the fact that some arrangements of region labels are more likely than others. This is implemented using an approach similar to a Markov Random Field (MRF) [20], where the probability of a region's labeling depends on its immediate neighbors. Given a region labeling $\boldsymbol{R}$, a function $f_r(\boldsymbol{R})$ can be defined which is the proportion of the boundary which region r shares with neighbors of the same label. A long boundary with regions of the same label is more likely than very little of the boundary bordering similar regions. A probability density function for f_r has been computed from hand-segmented examples and can be approximated by

$$\mathrm{P}(f_r) = \frac{0.932}{1+\exp\left(9-18f\right)} + 0.034 \qquad 0 < f_r < 1. \tag{17}$$

$\mathrm{P}(1)$ is set to 0.9992 and $\mathrm{P}(0)$ to 0.0008 to enforce the fact that isolated regions or holes are particularly unlikely. The prior probability of a region labeling $\boldsymbol{R}$ is then given by

$$\mathrm{P}(\boldsymbol{R}) = \prod_{\text{regions r}} \frac{\mathrm{P}(f_r(\boldsymbol{R}))}{\sum_{l=1}^{\text{layers}} \mathrm{P}(f_r(\boldsymbol{R}\{r=l\}))}, \tag{18}$$

where $f_r(\boldsymbol{R}\{r=l\})$ indicates the fractional boundary length which would be seen if the label of region r were substituted with a different label l.

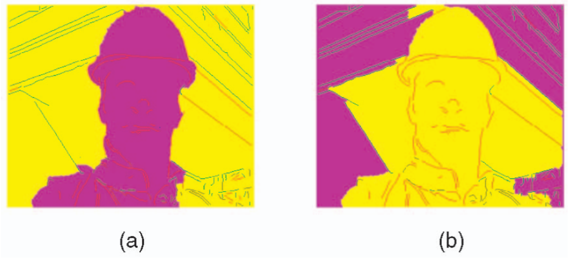
4.3.3 Solution by Simulated Annealing

In order to maximize over all possible region labelings, simulated annealing [40] is used. This begins with an initial guess at the region labeling and then repeatedly tries flipping individual region labels one by one to see how the change affects the overall probability. (This is a simple process since a single region label change only causes local changes and so (18) does not need completely reevaluating.) The annealing process is initialized with a guess based on the edge probabilities and a reasonable initialization is to label the regions according to the majority of its edge labelings. The region labels are taken in turn, considering the probability of the region being labeled motion 1 or 2 given its edge probabilities and the current motion label of its neighbors. At the beginning of the annealing process, the region is then reassigned a label by a Monte Carlo approach, i.e., randomly according to the two probabilities. As the iterations progress, these probabilities are forced to saturate so that gradually the assignment will tend toward the most likely label, regardless of the actual probabilities. The saturation function, determined empirically, is

$$p' = p^{1+(n-1)^{0.07}}, \tag{19}$$

where n is the iteration number. This function is applied to each of the label probabilities for a region before normalization. The annealing process continues for 40 iterations, which is found to be sufficient for a good solution to be reached. Each pass of the data tries flipping each region, but the search order is shuffled each time to avoid systematic errors.

In order for the edge labeling to be generated from the region labeling $\boldsymbol{R}$, the layer ordering $\boldsymbol{F}$ must also be known, but this is yet to be found. This parameter is independent of $\boldsymbol{R}$ and so a fixed value of $\boldsymbol{F}$ can be used throughout the annealing process. The process is thus repeated for each possible layer ordering and the solution with the highest likelihood identifies both the correct region labeling and the correct layer ordering. Fig. 8 shows the two different solutions in the "Foreman" case, the first solution has a higher likelihood, so is selected as the final segmentation. The entire maximization of (9), over $\boldsymbol{R}$ and $\boldsymbol{F}$, takes around two seconds on a 300MHz Pentium II for a 352 × 240 image.



(a) (b)

Fig. 8. "Foreman" solutions under different layer orderings. The most likely region labelings, showing the foreground as magenta and the background as yellow (a) with red as the foreground motion and (b) with green as the foreground motion. Case (a) has a higher posterior probability and, so, is the maximum likelihood segmentation over $\boldsymbol{R}$ and

4.4 Results

The two-motion, two-frame implementation has been tested on a wide range of real video sequences.[6] Fig. 4 shows the segmentation from the standard "Foreman" sequence. Edges are extracted and then EM run between this frame and the

[6] The segmentation software developed for this paper may be downloaded from http://mi.eng.cam.ac.uk/~pas1001/Research/edgesegment.html.

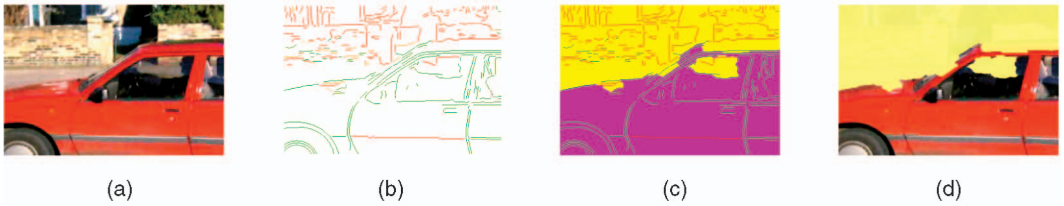


Fig. 9. “Car” segmentation from two frames. (a) Original frame. (b) Edge labels after EM. (c) Most likely region labeling. (d) Final foreground segmentation.

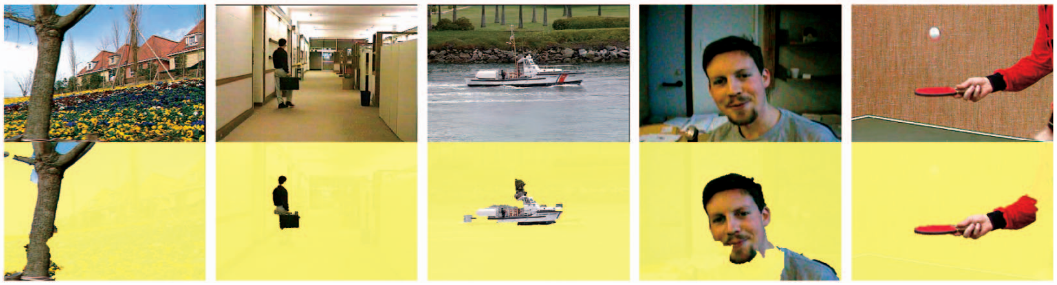


Fig. 10. A sample of the 34 test sequences and their segmentations.

next to estimate the motions. Fig. 4b shows the edges labeled according to how well they fit each motion after convergence. It can be seen that this process labels most of the edges correctly, even though the motion is small (about two pixels). The edges on his shoulders are poorly-labeled, but this is due to the shoulders' motion being even smaller than that of the head. The correct motion is selected as foreground with very high confidence (> 99 percent) and the final segmentation, Fig. 4d, is excellent despite some poor edge labels. In this case the MRF region prior is a great help in producing a plausible segmentation. Compared with the hand-picked segmentation shown in Fig. 4c, 98 percent of the regions are labeled correctly. On a 300MHz Pentium II, it takes a total of around eight seconds to produce the motion segmentation (the image is 352×288 pixels).

Fig. 9 shows the results from the “Car” sequence, recorded for this work. Here the car moves to the left, and is tracked by the camera. This is a rather unusual sequence since more pixels belong to the foreground than to the background and some dominant-motion techniques may therefore assume the incorrect layer ordering. In this paper, however, the ordering is found from the edge labels and no such assumption is made. Unfortunately, the motion of many of the horizontal edges is ambiguous and also, with few T-junctions, there is less depth ordering information than in the previous cases. Nevertheless, the correct motion is identified as foreground, although with less certainty than in the previous cases. The final segmentation (Fig. 9d) labels 96 percent of all pixels correctly (compared with a hand labeling), and there are two main sources of error. As already noted, with both motions being horizontal, the labeling of the horizontal edges is ambiguous. More serious, however, are the reflections on the bonnet and roof of the car which naturally move with the background motion. The edges are correctly labeled—as background—but this gives the incorrect semantic labeling. Without higher-level processing (a prior model of a car), this problem is difficult to resolve. One pleasing element of the solution is that the view through the car window has been correctly segmented as background.

This implementation has been tested on a total of 34 real sequences. Full results can be seen in [36], but Fig. 10 shows a selection of these further results. Compared with a manual labeling of regions, a third of all the sequences tested are segmented near-perfectly by the system (> 95 percent of pixels correct), and a further third are good or very good (> 75 percent). In the cases where the segmentation fails, this is either because the motion between frames is extremely nonaffine, or is ambiguous, resulting in a poor edge labeling.

For the algorithm to succeed, the edges merely have to fit *better* under one motion than the other—an exact match is not necessary. As a result, even when one or both of the motions are significantly nonaffine, as in the first two examples in Fig. 10, a good segmentation can still be generated. It is a testament to the sufficiency of edges that where the edges are labeled correctly, the segmentation is invariably good. The principal way to improve a poor edge labeling is to continue to track the edges over additional frames until the two motions can be better distinguished.

5 Extension to Multiple Frames

Accumulating evidence over a number of frames can resolve ambiguities that may be present between the first two frames, and also makes the labeling more robust. This section first describes how evidence can be accumulated over frames to improve the segmentation of one frame, and then outlines how the techniques can be extended to segment a whole sequence.

5.1 Accumulating Evidence to Improve Segmentations

While the segmentation of frame 1 using a pair of frames is often very good, a simple extension allows this to be improved. The two-frame algorithm of Section 4 can be run between frame 1 and other frames in the sequence to gather

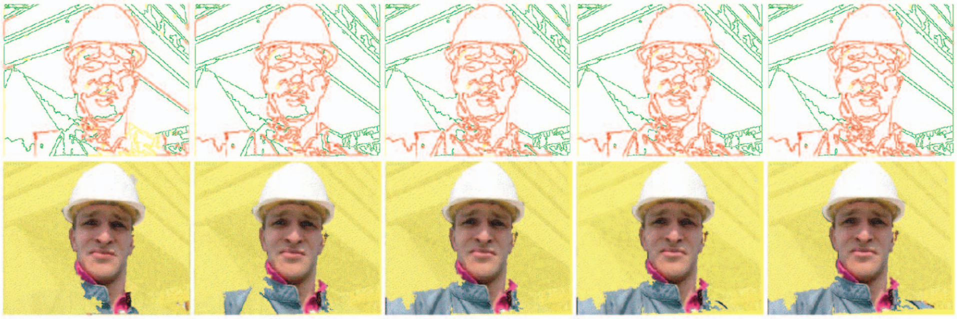


Fig. 11. Evolution of the "Foreman" segmentation, showing the edge probabilities and segmentations of frame 1 as the evidence is accumulated over five successive frames. The edge probabilities become more certain and small errors are removed, resulting in an improved region segmentation.

more evidence about the segmentation of frame 1. The efficiency of this process can be improved by using the results from one frame to initialize the next and, in particular, the EM stage can be given a better initialization. The initial motion estimate is that for the previous frame incremented by the velocity between the previous two frames. The edge labeling is initialized to be that implied by the region labeling of the previous frame and the EM begins at the M-stage.

5.1.1 Combining Statistics

The probability that an edge obeys a particular motion over a sequence is the probability that it obeyed that motion between each of the frames. This can be calculated from the product of the probabilities for that edge over all those frames, if it is assumed that the image data yielding information about the labeling of each edge is independent in each frame. The EM is performed only on the edge probabilities and the motion between the frame in question and frame 1, but after convergence the final probabilities are multiplied together with the probabilities from the previous frames to give the cumulative edge statistics. The region and foreground labeling is then performed as described in Section 4, but using the cumulative edge statistics.

5.1.2 Occlusion

The problem of occlusion was ignored when considering only two frames since the effects are minimal, but occlusion becomes a significant problem when tracking over multiple frames. Knowing the foreground/background labeling for edges and regions in frame 1, and the motions between frames, enables this to be overcome. For each edge labeled as background, its sample points are projected into frame 2 under the background motion and are then projected back into frame 1 according to the foreground motion. If a sample point falls into a region currently labeled as foreground, this foreground region must move on top of that point in frame 2. If this is the case, the sample point is marked as occluded and does not contribute to the tracking of its edge into frame 3. All sample points are also tested to see if they project outside the frame under their motions and if so they are also ignored. This process can be repeated for as many frames as is necessary.

5.2 Results

The success of the multiple frame approach can be seen in Fig. 11, showing the "Foreman" example. Accumulating the edge probabilities over several frames allows random errors to be removed and edge probabilities to be reinforced. The larger motions between more widely separated frames also removes ambiguity. It can be seen that, over time, the consensus among many edges on the shoulders is towards the foreground motion and the accumulated edge probabilities have a positive effect on the region segmentation, which settles down after a few frames to a very accurate solution.

Over the 34 test sequences considered in this work, including a second frame in the labeling process increases the average number of pixels correct from 76 to 86 percent, with 14 sequences labeled near-perfectly, and only six with less than 75 percent of pixels correct. The sequences which still failed either had very large nonaffine motions (e.g., dancing), or very few edge features, but many challenging sequences are very well segmented.

5.3 Templated Segmentation of a Sequence

The use of multiple frames has been motivated as a means of improving the segmentation of a single frame, using the extended sequence to label edges more robustly. The segmentation scheme generates a final segmentation of frame 1, and the foreground and background motions between frames. However, this information can enable the segmentation of the sequence to be approximated. The foreground regions from frame 1 may be projected into the other frames of the sequence according to the foreground motion at each frame. These regions may then be used as a template to cut out the object in each of the subsequent frames.

Fig. 12 shows such a segmentation and it can be seen that this provides a very good approximation. This accuracy is not restricted to obviously rigid objects; the segmentations in Fig. 11 were also performed by this technique and the cut-out even in frame 5 (using the segmentation from frame 1 warped by the estimated affine transformation) is still excellent. These results demonstrate that the affine motion model is appropriate for these sequences, and that the motion parameters are estimated well by the EM process.

5.4 Frame-by-Frame Segmentation of a Sequence

A more general approach to segmenting a sequence is to perform a new static segmentation, and then labeling, for

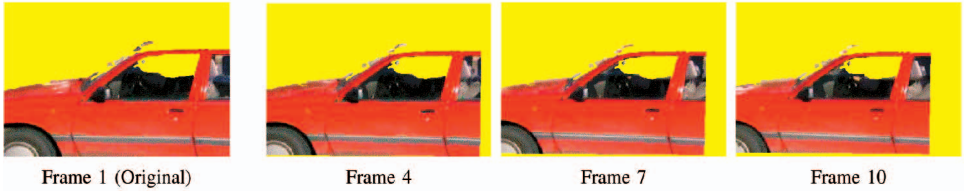


Fig. 12. Templated segmentation of the "Car" sequence. The foreground segmentation for the original frame is transformed under the foreground motion model and used as a template to segment subsequent frames.

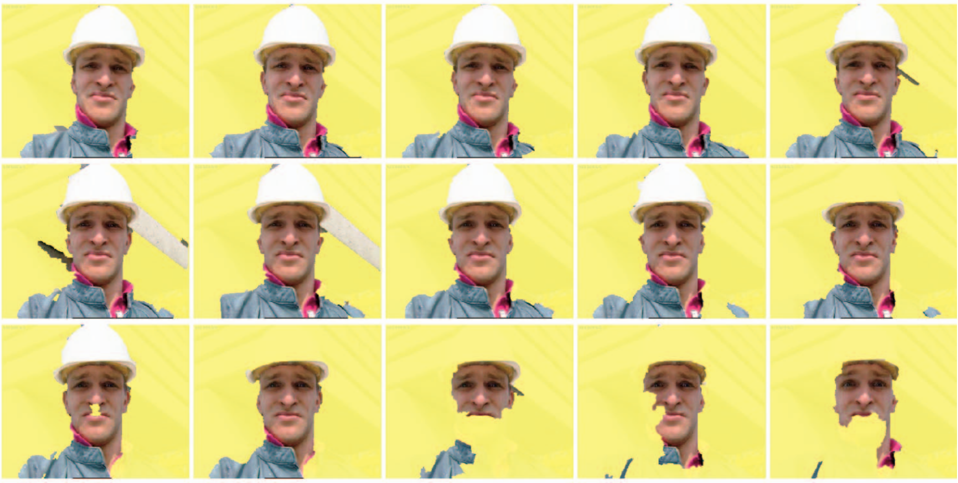


Fig. 13. Segmentation of the "Foreman" sequence. Segmentation of 10 consecutive frames.

each frame (i.e., to run the algorithm between consecutive frames of the sequence). Finding edges and segmenting anew in each frame ensures that the structure of each image is best represented, but presents difficulties in propagating statistics. The statistics in each frame are driven by the sample points, and so the sample points on the edges in the new frame are created in two stages. First, sample points from the previous frame are transformed into the new frame according to their most likely motion. If they land near an edge (within two pixels), they are allocated to that edge and store their previous label probabilities as their prior for this frame. New sample points are then created on any empty edges, with flat priors. These priors are used to initialize EM but, as before, this proceeds just with the probabilities from the current pair of frames, and then the previous probabilities are included when calculating the region labeling.

Fig. 13 shows a segmentation of 15 consecutive frames from the "Foreman" sequence, segmented in this manner. It can be seen that the first 10 frames or so are very well segmented, apart from the occasional mislabeled region. The failures in the last row are due to rapid motions which do not fit the motion model at all well. Problems such as this would be alleviated by a robust technique for propagating statistics between subsequent frames and improving the region labeling priors to reduce fragmentation. These are both areas for future research.

6 Extension to Multiple Motions

The theory of Section 3 applies to any number of motions, and the implementation has been developed so as to be extensible to more than two motions. Tracking and separating three or more motions, however, is a nontrivial task. With more motions for edges to belong to, there is less information with which to estimate each motion and edges can be assigned to a particular model with less certainty. In estimating the edge labels, the EM stage is found to have a large number of local minima, and so an accurate initialization is particularly important. The layer ordering is also more difficult to establish. As the number of motions increase, the number of possible layer hypotheses increases factorially. Also, with fewer regions per motion, fewer regions interact with those of another layer, leading to fewer T-junctions, which are the essential ingredient in determining the layer ordering. These factors all contribute to the difficulty of the multiple motion case and this section proposes some extensions which make the problem easier.

6.1 EM Initialization

The EM algorithm is guaranteed to converge to a maximum, but there is no guarantee that this will be the global maximum. The most important element in EM is always the initialization and, for more than two motions, the EM algorithm will get trapped in a local maximum unless started with a good solution. The best solution to these local maxima problems in EM remains an open question.

The approach adopted in this paper is hierarchical—the gross arrangement is estimated by fitting a small number of models and then these are split to see if any finer detail can be fitted. The one case where local minima does not present a significant problem is when there are only two motions, where it has been found that any reasonable initialization can be used. Therefore, two motions are fitted first and then three-motion initializations are considered near to this solution. It is worth considering what happens in the case of labeling a three-motion scene with only two motions. There are two likely outcomes:

1. One (or both) of the models adjusts to absorb edges which belong to the third motion.
2. The edges belonging to the third motion are discarded as outliers.

This provides a principled method for generating a set of three-motion initializations. First fit two motions, then:

1. Take the set of edges which best fit one motion and try to fit two motions to these by splitting the edges into two random groups and performing EM on just these edges to optimize the split. The original motion is then replaced with these two. Each of the two initial motions can be split in this way, providing two different initializations.
2. A third initialization is given from the outliers by calculating the motion of the outlier edges and adding it to the list of motions. Outlier edges are detected by comparing the likelihood under the "correct motion" statistics of Section 4 with the likelihood under an "incorrect motion" model, also gathered from example data.

From each of these three initializations, EM is run to find the most likely edge labeling and motions. The likelihood of each solution is given by the product of the edge likelihoods (under their most likely motion) and best solution is the one with the highest likelihood. This solution may then be split further into more motions in the same manner.

6.2 Determining the Best Number of Motions

This hierarchical approach can also be used to identify the best number of motions to fit. Increasing the number of models is guaranteed to improve the fit to the data and increase the likelihood of the solution, but this must be balanced against the cost of using a large number of motions. This is addressed by applying the Minimum Description Length (MDL) principle, one of many model selection methods available [41]. This considers the cost of encoding the observations in terms of the model and any residual error. A large number of models or a large residual both give rise to a high cost.

The cost of encoding the model consists of two parts. First, the parameters of the model: Each number is assumed to be encoded to 10-bit precision, and with six parameters per model (2D affine), the cost is $60n_m$ (for n_m models). Second, each edge must be labeled as belonging to one of the models, which costs $\log_2 n_m$ for each of the n_e edges. The edge residuals must also be encoded, and the cost for an optimal coding is equal to the total negative logarithm (to base two) of the edge likelihoods, L_e, giving

$$C = 60n_m + n_e \log_2 n_m + \sum_e \log_2 L_e. \tag{16}$$

The cost C is be evaluated after each attempted initialization, and the smallest cost indicates the best solution and the best number of models.

6.3 Global Optimization: Expectation-Maximization-Constrain (EMC)

The region labeling is determined via two independent optimizations which use edges as an intermediate representation: first the best edge labeling is determined, and then the best region labeling given these edges. It has thus far been assumed that this is a good approximation to the global optimum, but unfortunately this is not always the case, particularly with more than two motions.

In the first EM stage, the edges are assigned purely on the basis of how well they fit each motion, with no consideration given to how likely that edge labeling is in the context of the wider segmentation. There are always a number of edges which are mislabeled and these can have an adverse effect on both the region segmentation and the accuracy of the motion estimate. In order to resolve this, the logical constraints implied by the region labeling stage are used to produce a discrete, constrained edge labeling before the motions are estimated. This is referred to as Expectation-Maximization-Constrain or EMC. Once again, initialization is an important consideration. The constraints (i.e., a sensible segmentation) cannot be applied until near the solution, so the EMC is used as a final global optimization stage after the basic segmentation scheme has completed.

The EMC algorithm follows the following steps:

- **Constrain.** Calculate the most likely region labeling and use this, via the Edge Labeling Rule, to label each edge with a definite motion.
- **Maximization.** Calculate the motions, in each case using just the edges assigned to that motion.
- **Expectation.** Estimate the probable edge labels given the set of motions.

The process is iterated until the region labeling probability is maximized.

In dividing up (3), it was assumed that the motions could be estimated without reference to the region labeling, because of the large number of edges representing each motion. This assumption is less valid for multiple motions, and EMC places the region labeling back into the motion estimation loop, ensuring estimated motions which reflect a self-consistent (and, thus, more likely) edge and region labeling. As a result, EMC helps the system better reach the global maximum.

6.4 "One Object" Constraint

The Markov Random Field used for the region prior $\mathrm{P}(R)$ only considers the neighboring regions, and does not consider the wider context of the frame. This makes the simulated annealing tractable, but does not enforce the belief that there should, in general, be only one connected group of regions representing each foreground object. It is common for a few small background regions to be mislabeled as foreground and these can again have an adverse effect on the solution when this labeling has to be used to estimate a new motion (for example, when using multiple frames or EMC).

A simple solution may be employed after the region labeling. For each foreground object with a segmentation which consists of more than one connected group, region labelings are hypothesized which label all but one of these

TABLE 1
MDL Values

	Foreman				Car				Car & Van				Library			
n_m	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Motion	60	120	180	240	60	120	180	240	60	120	180	240	60	120	180	240
Edge	0	482	764	964	0	518	821	1036	0	322	510	644	0	133	211	266
Residual	5067	3733	3467	3334	10491	5167	4931	4763	4158	3669	3109	2944	2341	1691	1450	1400
Total	5127	**4335**	4411	4538	10551	**5805**	5932	6039	4218	4131	**3799**	3828	2401	1944	**1841**	1906

(For different numbers of motions (n_m), the total cost is that of encoding the motion parameters ("Motion"), edge labeling ("Edge") and the residual ("Residual").)

groups as belonging to a lower layer (i.e., further back). The most likely of these "one object" region labelings is the one kept.

6.5 Results

The extended algorithm, featuring all three extensions (multiple motions, EMC and the global region constraint) has been tested on a number of two and three-motion sequences. Table 1 shows the results of the model selection stage. The first two sequences are expected to be fitted by two motions, and the other two by three motions. All the sequences are correctly identified, although in the "Foreman" case there is some support for fitting the girders in the bottom right corner as a third motion. The use of EMC and the global-region constraint has little effect on the two-motion solutions, which, as seen in Section 4, are already excellent. This indicates that the basic two-frame, two-motion algorithm already reaches a solution close to the global optimum.

It is the three-motion sequences which present a more difficult challenge. Fig. 14 shows a sequence where the camera is stationary, and the white car in the foreground begins to pull out (to the left) as the yellow van speeds by. The size of the van's motion means that under two motions, the van's edges are mainly outliers and it is here that the value of fitting a third motion to the outliers becomes apparent. The MDL process is clearly in favor of fitting three motions, as seen in Table 1.

When the edges are labeled, the car motion also fits parts of the building well, particularly due to the repeating nature of the classical architecture. This presents a few problems to the region labeling stage, as can be seen in Fig. 14d where there are a few regions on the columns which are labeled with the car. It is in cases such as this that the "one region" constraint is needed, in conjunction with EMC to produce the clean results seen in Figs. 14e, 14f, and 14g. The region labeling with the car in front, and the van at the middepth is significantly more likely (i.e., better-supported by the edge labels) than any other orders, so this correctly identifies the layer ordering.

Another three-motion sequence is shown in Fig. 15. In this case, the scene is static but the camera moves from right to left. The books, statue and background are at different depths and so have different image motions. Labeling the motion of the horizontal lines in the scene is difficult given the horizontal camera motion and it can be seen that the edge marking the top of the books has been incorrectly labeled as a result and then the initial region segmentation has incorrectly merged some of the books with the statue (see Fig. 15c and 15d). The EMC loop is then entered, performing the constrained global optimization. The edge labels in Fig. 15c can be seen to have a number of logically inconsistent labels. The EMC loop resolves these and gives

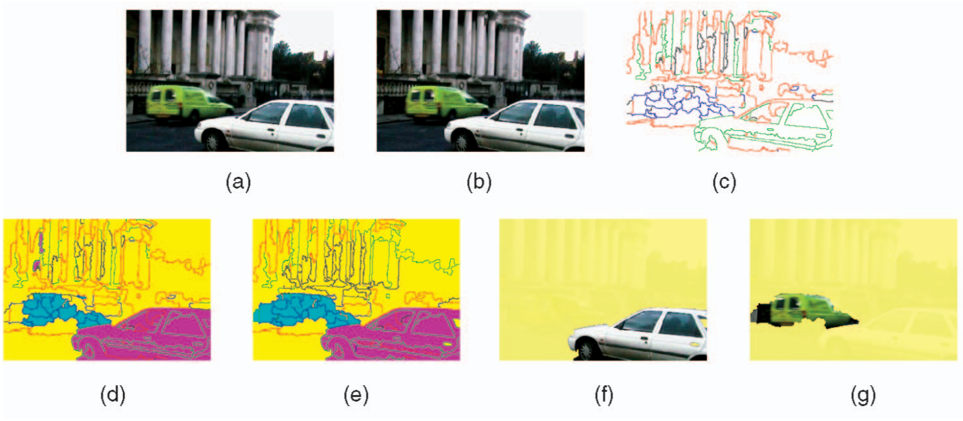

(a) (b) (c)
(d) (e) (f) (g)

Fig. 14. "Car & Van" segmentation. (a) and (b) The two frames used for the segmentation. The car moves to the left, the van to the right. (c) Region edges, labeled by EM. (d) and (e) Edge and region labels before and after EMC. (f) and (g) The two foreground layers. The car is labeled as being in front of the van.

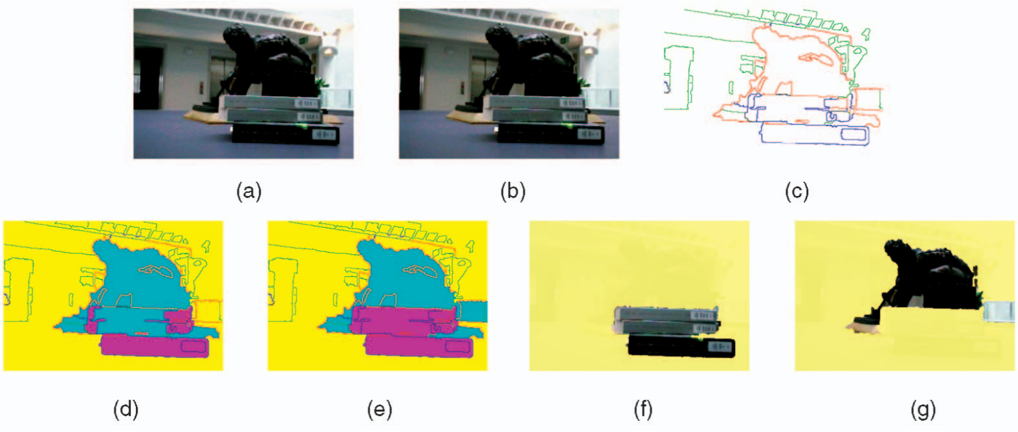


Fig. 15. "Library" segmentation. (a) and (b) The two frames used for the segmentation. The camera moves to the left, and the books, statue, and background move differing amounts due to parallax. (c) Region edges, labeled by EM. (d) and (e) Edge and region labels before and after EMC. (f) and (g) The two foreground layers. The books are identified as being in front.

the labeling shown in Fig. 15e, which is very good. The EMC loop is performed for each possible layer ordering (six in this case) to determine the correct order. While the background layer is confidently labeled, the ordering of the two foreground layers is more ambiguous in this case. The poor labeling of the main horizontal edge dividing the two objects has already been mentioned and there are very few other edges which contribute to the decision. The book is correctly identified as being in front, but with a probability of only 53 percent over the other foreground ordering.

The extended algorithm is somewhat slower than the basic one. On a 300MHz PII, it takes about a minute to segment a two-motion sequence, and about three minutes to segment a three-motion sequence (for 320×240-pixel images). Most of the time is spent in the EMC loop, which has to be repeated four extra times in the three-motion case to consider all possible layer orderings. These results demonstrate that the scheme can successfully be extended to multiple motions, but suggest several avenues for future work.

7 Conclusions and Future Work

This paper develops and demonstrates a novel Bayesian framework for segmenting a video sequence into ordered motion layers based on tracking image edges between frames and segmenting the image into regions along these edges. It is demonstrated that edges can be reliably labeled according to their motion and are sufficient to label regions and determine the motion layer ordering.

The EM algorithm is used to simultaneously estimate the motions and the edge label probabilities. The correct layer ordering and region labeling is identified by hypothesizing and testing to maximize the probability of the model given the edge data and a MRF-style prior. The algorithm runs quickly and the results are very good even when using only two frames. The framework can also be extended to accumulate edge probabilities over multiple frames, which improves robustness and resolves some ambiguities, resulting in a very accurate segmentation. It is shown that many sequences are well-segmented using the affine motion model, even when they contain significant nonaffine motion. However, the extension of this scheme to other motion models is one area for future work.

The framework works best when there are two clear motions (i.e., the background and one large foreground object), where the EM algorithm converges well. Some extensions have been proposed to deal with the case of more than two motions and these have been met with some success. However, the problem of multiple-motion segmentation, and model selection, is a difficult one and is on the limit of the information that can be gathered from edges alone. With multiple motions and smaller foreground objects, errors are much more likely to occur and then, with a higher number of mislabeled edges, the region labeling and layer ordering becomes quite fragile. The main difficulty is in the EM stage, which suffers from many local maxima, and other solutions should be investigated, such as Deterministically Annealed EM [42], or alternative (perhaps optic-flow based) approaches to initializing the motions. More informative region-label priors would also help to resolve the region labeling issues in the presence of poor edge labels, not just in this case, but in all the cases considered in this paper.

The use of multiple frames to improve edge labeling has been shown to be successful. This should be developed further, refining the statistics and enforcing the consistent labeling of edges and regions. In particular, this will resolve many of the ambiguities present in labeling multiple motions. A further extension is that, currently, weak edges are ignored by the system, which can mean that some useful edges can be missed. Information from the motion models could be used to promote weak edges which are consistent with the motion into the model, producing a genuine *motion* segmentation of the sequence.

Acknowledgments

This research was funded by a United Kingdom EPSRC studentship, with a CASE award from AT&T Laboratories, Cambridge, UK. Thanks go to David Sinclair for the use of his image segmentation code and to Ken Wood for many useful discussions.

References

[1] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, "Efficient Representations of Video Sequences and Their Representations," *Signal Processing: Image Comm.*, vol. 8, no. 4, pp. 327-351, May 1996.

[2] H.S. Sawhney and S. Ayer, "Compact Representations of Videos through Dominant and Multiple Motion Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 814-830, Aug. 1996.

[3] Information Technology—Coding of Audio-Visual Objects, ISO/IEC 14496, MPEG-4 Standard, 1999-2002.

[4] M. Gelgon and P. Bouthemy, "Determining a Structured Spatio-Temporal Representation of Video Content for Efficient Visualisation and Indexing," *Proc. Fifth European Conf. Computer Vision (ECCV '98),* pp. 595-609, 1998.

[5] M. Irani and P. Anandan, "Video Indexing Based on Mosaic Representations," *Proc. IEEE,* vol. 86, no. 5, pp. 905-921, May 1998.

[6] B.K.P. Horn and B.G. Schunk, "Determining Optical Flow," *Artificial Intelligence,* vol. 17, nos. 1-3, pp. 185-203, Aug. 1981.

[7] G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 4, pp. 384-401, July 1985.

[8] D.W. Murray and B.F. Buxton, "Scene Segmentation from Visual Motion Using Global Optimization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, no. 2, pp. 220-228, Mar. 1987.

[9] A. Jepson and M. Black, "Mixture Models for Optical Flow Computation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 93),* pp. 760-761, 1993.

[10] J.Y.A. Wang and E.H. Adelson, "Layered Representation for Motion Analysis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 93),* pp. 361-366, 1993.

[11] J.Y.A. Wang and E.H. Adelson, "Representing Moving Images with Layers," *Trans. Image Processing,* vol. 3, no. 5, pp. 625-638, Sept. 1994.

[12] S. Ayer, P. Schroeter, and J. Bigün, "Segmentation of Moving Objects by Robust Motion Parameter Estimation Over Multiple Frames," *Proc. Third European Conf. Computer Vision (ECCV '94),* pp. 317-327, 1994.

[13] M. Irani, B. Rousso, and S. Peleg, "Computing Occluding and Transparent Motions," *Int'l J. Computer Vision,* vol. 12, no. 1, pp. 5-16, Jan. 1994.

[14] J.M. Odobez and P. Bouthemy, "Separation of Moving Regions from Background in an Image Sequence Acquired with a Mobile Camera," *Video Data Compression for Multimedia Computing.* pp. 283-311, Dordrecht, The Netherlands, Kluwer Academic Publishers, 1997.

[15] G. Csurka and P. Bouthemy, "Direct Identification of Moving Objects and Background from 2D Motion Models," *Proc. Seventh Int'l Conf. Computer Vision (ICCV '99),* pp. 566-571, 1999.

[16] J.Y.A. Wang and E.H. Adelson, "Spatio-Temporal Segmentation of Video Data," *Proc. SPIE: Image and Video Processing II,* pp. 130-131, 1994.

[17] A.P. Dempster, H.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. of Royal Statistical Soc.: Series B (Methodological),* vol. 39, no. 1, pp. 1-38, Jan. 1977.

[18] S. Ayer and H.S. Sawhney, "Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding," *Proc. Fifth Int'l Conf. Computer Vision (ICCV '95),* pp. 777-784, 1995.

[19] Y. Weiss and E.H. Adelson, "A Unified Mixture Framework for Motion Segmentation: Incorporating Spatial Coherence and Estimating the Number of Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '96),* pp. 321-326, 1996.

[20] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 721-741, Nov. 1984.

[21] J.M. Odobez and P. Bouthemy, "Direct Incremental Model-Based Image Motion Segmentation for Video Analysis," *Signal Processing,* vol. 66, no. 2, pp. 143-155, Apr. 1998.

[22] J. Shi and J. Malik, "Motion Segmentation and Tracking Using Normalized Cuts," *Proc. Sixth Int'l Conf. Computer Vision (ICCV '98),* pp. 1154-1160, 1998.

[23] P. Giaccone and G. Jones, "Segmentation of Global Motion Using Temporal Probabilistic Classification," *Proc. Ninth British Machine Vision Conference (BMVC '98),* vol. 2, pp. 619-628, 1998.

[24] W.B. Thompson, "Combining Motion and Contrast for Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 2, no. 6, pp. 543-549, Nov. 1980.

[25] F. Dufaux, F. Moscheni, and A. Lippman, "Spatio-Temporal Segmentation Based on Motion and Static Segmentation," *Proc. Int'l Conf. Image Processing (ICIP),* vol. 1, pp. 306-309, 1995.

[26] F. Moscheni and S. Bhattacharjee, "Robust Region Merging for Spatio-Temporal Segmentation," *Proc. Int'l Conf. Image Processing (ICIP),* vol. 1, pp. 501-504, 1996.

[27] F. Moscheni and F. Dufaux, "Region Merging Based on Robust Statistical Testing," *Proc. SPIE Int'l Conf. Visual Communications and Image Processing (VCIP '96),* 1996.

[28] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatiotemporal Segmentation Based on Region Merging," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 9, pp. 897-915, Sept. 1998.

[29] L. Bergen and F. Meyer, "Motion Segmentation and Depth Ordering Based on Morphological Segmentation," *Proc. Fifth European Conf. Computer Vision (ECCV '98),* pp. 531-547, 1998.

[30] D. Tweed and A. Calway, "Integrated Segmentation and Depth Ordering of Motion Layers in Image Sequences," *Proc. 11th British Machine Vision Conference (BMVC 2000),* pp. 322-331, 2000.

[31] M.J. Black and D.J. Fleet, "Probabilistic Detection and Tracking of Motion Boundaries," *Int'l J. Computer Vision,* vol. 38, no. 3 pp. 229-243, July 2000.

[32] L. Gaucher and G. Medioni, "Accurate Motion Flow Estimation with Discontinuities," *Proc. Seventh Int'l Conf. Computer Vision (ICCV '99),* vol. 2, pp. 695-702, 1999.

[33] J.F. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679-698, Nov. 1986.

[34] T. Drummond and R. Cipolla, "Application of Lie Algebras to Visual Servoing," *Int'l J. Computer Vision,* vol. 37, no. 1, pp. 21-41, June 2000.

[35] W.J.J. Rey, *Introduction to Robust and Quasi-Robust Statistical Methods.* Springer-Verlag, Berlin 1978.

[36] P.A. Smith, "Edge-Based Motion Segmentation," PhD dissertation, Univ. of Cambridge, UK, Aug. 2001.

[37] P. Smith, T. Drummond, and R. Cipolla, "Motion Segmentation by Tracking Edge Information over Multiple Frames," *Proc. Sixth European Conf. Computer Vision (ECCV 2000),* pp. 396-410, 2000.

[38] D. Sinclair, "Voronoi Seeded Colour Image Segmentation," Technical Report, 1999.3, AT&T Laboratories, Cambridge, UK, 1999.

[39] L. Vincent and P. Soille, "Watersheds in Digital spaces: An Efficient Algorithm Based on Immersion Simulations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 6, pp. 583-589, June 1991.

[40] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4598, pp. 671-680, May 1983.

[41] P.H.S. Torr, "An Assessment of Information Criteria for Motion Model Selection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '97),* pp. 47-53, 1997.

[42] N. Ueda and R. Nakano, "Deterministic Annealing EM Algorithm," *Neural Networks,* vol. 11, no. 2, pp. 271-282, Apr. 1998.

Paul Smith received the BA and MEng degrees (electrical and information sciences) from the University of Cambridge in 1996. In 2002, he was awarded the PhD degree (computer vision) from the same institution. In 2000, he joined Robinson College, Cambridge, as a teaching fellow, a post also affiliated to the Department of Engineering, University of Cambridge. His research interests are in computer vision and robotics, and include robust statistics, video analysis, and visual tracking. He is a member of the IEEE Computer Society.

Tom Drummond received the BA (mathematics) degree from the University of Cambridge in 1988. From 1989 to 1998, he studied and worked in Australia and in 1998 was awarded the PhD degree from Curtin University in Perth, Western Australia. In 1998, he joined the Department of Engineering in the University of Cambridge as a research associate. In 2001, he was appointed as a university lecturer. His research interests are in computer vision and robotics and include real-time visual tracking, visual servoing, and augmented reality. He is a member of the IEEE Computer Society.

Roberto Cipolla received the BA degree (engineering) from the University of Cambridge, England, in 1984, and the MSEE degree (electrical engineering) from the University of Pennsylvania in 1985. From 1985 to 1988, he studied and worked in Japan at the Osaka University of Foreign Studies (Japanese language) and then obtained the MEng degree (robotics) from the University of Electro-Communications in Tokyo in 1988. In 1991, he was awarded the DPhil (computer vision) degree by the University of Oxford, England, and from 1991–1992, he was a Toshiba fellow and engineer at the Toshiba Corporation Research and Development Center in Kawasaki, Japan. He joined the Department of Engineering, University of Cambridge, in 1992, as a lecturer and a fellow of Jesus College. He became a reader in 1997 and a professor of information engineering in 2000. His research interests are in computer vision and robotics and include recovery of motion and 3D shape of visible surfaces from image sequences, visual tracking and navigation, robot hand-eye coordination, algebraic and geometric invariants for object recognition and perceptual grouping, and novel man-machine interfaces using visual gestures and visual inspection. He is the author of two books, editor of five volumes, and coauthor of more than 175 papers. He is a member of the IEEE.