Figure–Ground Segmentation from Occlusion

Pedro M. Q. Aguiar, *Member, IEEE,* and José M. F. Moura, *Fellow, IEEE*

Abstract

Layered video representations are increasingly popular, see [1] for a recent review. Segmentation of moving objects is a key step for automating such representations. Current motion segmentation methods either fail to segment moving objects in low textured regions or are computationally very expensive. This paper presents a computationally simple algorithm that segments moving objects even in low texture/low contrast scenes. Our method infers the moving object templates directly from the image intensity values, rather than computing the motion field as an intermediate step. Our model takes into account the *rigidity* of the moving object and the *occlusion* of the background by the moving object. We formulate the segmentation problem as the minimization of a *penalized likelihood* cost-function and present an algorithm to estimate all the unknown parameters: the motions, the template of the moving object, and the intensity levels of the object and of the background pixels. The cost function combines a *maximum likelihood* estimation term with a term that penalizes large templates. The minimization algorithm performs two alternate steps for which we derive closed-form solutions. Relaxation improves the convergence even when low texture makes it very challenging to segment the moving object from the background. Experiments demonstrate the good performance of our method.

EDICS: 2-SEGM (Image and Video Processing—Segmentation), **2-ANAL** (Analysis).

Permission to publish this abstract separately is granted.

Contact author: José M. F. Moura, Carnegie Mellon University, ECE Dep., 5000 Forbes Ave, Pittsburgh, PA 15213-3890. E-mail: moura@ece.cmu.edu. His work was partially supported by ONR grant # N000 14-00-1-0593.

Pedro M. Q. Aguiar is with ISR—Institute for Systems and Robotics, IST, Av. Rovisco Pais, 1049-001 Lisboa, Portugal. E-mail: aguiar@isr.ist.utl.pt. His work was partially supported by FCT project POSI/SRI/41561/2001.

I. INTRODUCTION

Modern content-based video representations demand efficient methods to infer the contents of video sequences, like the shape and texture of objects and their motions. Some existing methods lead to good results, see for example [2], [3], but require extensive human interaction. Fully automatic methods continue to be lacking and are of major interest. This paper considers the automatic segmentation of moving objects from a video sequence.

Motivation Segmentation of image sequences into regions with different motions is of interest to a large number of researchers. There is the need for segmentation methods that are *simple* and perform well, in particular, when the moving objects contain low-textured regions or there is low contrast between the object and the background. We present here a computationally simple method that performs well under these conditions: low-texture and low-contrast. Our algorithms use as a key assumption that the moving objects are *rigid* objects.

Several papers on video coding develop computationally simple algorithms for motion segmentation by processing two consecutive frames only. They predict each frame from the previous one through motion compensation [4]. Because their focus is on compression and not in developing a high level representation, these algorithms fail to provide accurate segmentation, in particular with low textured scenes; regions with no texture are considered to remain unchanged. For example, we applied the algorithm in [5] to segment a low textured moving object, a car, in a traffic video clip; see Fig. 1 where we show on the left two frames of this video clip. The template of the moving car as found by the algorithm in [5] is on the right of Fig. 1. As we see, due to the low texture of the car, the regions in the interior of the car are misclassified as belonging to the background, leading to a highly incomplete car template.

Fig. 1.   Motion segmentation in low texture.

Related work Background-estimation methods are very appealing approaches to segmentation of moving objects due to their simplicity. These methods infer the moving object template by subtracting the input image from a previously estimated background image [6], [7], [8], [9], [10], [11]. They generally estimate the background from the data by attempting to classify each pixel as either foreground or background. Although background-estimation succeeds in many relevant situations, *e.g.*, surveillance applications [12], it requires robust estimation of the background, which limits its application. Their major failing is that generally they do not exploit the structure of the object—they are usually pixel-wise independent.

In computer vision, commonly, motion-based segmentation copes with low textured scenes by coupling motion-based segmentation with prior knowledge about the scenes as in statistical regularization techniques, or by combining

motion with other attributes. For example, [13] uses a Markov Random Field (MRF) prior and a Bayesian Maximum a Posteriori (MAP) criterion to segment moving regions. The authors suggest a multiscale MRF to resolve large regions of uniform intensity. In [14], the contour of a moving object is estimated by fusing motion with color segmentation and edge detection. In general, these methods lead to complex and time consuming algorithms. Another approach to object segmentation uses active contours [15], [16], including methods that describe the contour as the level set of a real-valued function defined in the image domain [17], see also [18], [19] for applications to bioimaging. Besides edge information, some of these methods also account for prior models on the intensity values of the image inside and outside the contour [20], [21]. These methods, as the pioneering work of Mumford and Shah [22], estimate the contour of the object by minimizing a global cost function, thus leading to robust estimates. The computational cost is their major drawback—the minimization of the cost function resorts to calculus of variations [23] with the contour evolving according to partial differential equations [24], which makes the algorithms rather expensive.

Irani, Rousso, and Peleg use temporal integration. They average the images by registering them according to the motion of the different objects in the scene [25], [26]. After processing a number of frames, each of these averaged images should show only one sharp region corresponding to the tracked object. This region is found by detecting the stationary regions between the corresponding averaged image and the current frame. Unless the background is textured enough to blur completely the averaged images, some regions of the background can be classified as stationary. In this situation, the method in [25], [26] overestimates the template of the moving object. This is particularly likely to happen when the background has large regions with almost constant color or intensity level.

Layered models [27], [28], [29], [30], [31], [32], [33], [34] brought new approaches to the segmentation of moving objects. Tao, Sawhney, and Kumar proposed a filtering approach where a 2-D Gaussian shape model is updated from frame to frame [30]. This work was extended to the case where the background is described by a set of layers rather than a single one [34]. In contrast to online filtering, Jojic and Frey proposed an offline approach to infer *flexible* templates [32]. They use probabilistic learning to estimate robustly the state of the system. Since the exact posterior for the problem results intractable, they use variational inference to compute a factorized approximation and non-linear optimization techniques coupled into an EM algorithm [35].

Proposed approach Like the simple background-estimation algorithms, our approach exploits the fact that the moving object occludes the background. We formulate segmentation in a global way, as a parameter estimation problem and derive a computationally simple algorithm. Because in many interesting situations the 2-D shape of the moving object does not change significantly across a number of consecutive frames, *e.g.*, moving cars, see Fig. 1, we exploit the object *rigidity*. In the paper we show how *occlusion+rigidity* enable a computationally simple algorithm to jointly estimate the unknown background and rigid shape of the moving object directly from the image intensity values.

Our segmentation algorithm is derived as an approximation to a *penalized likelihood* (PL) estimate of the unknown parameters in the image sequence model: the motions; the template of the moving object; and the intensity levels of the object pixels (object texture) and of the background pixels (background texture). The joint estimation of

this complete set of parameters is a very complex task. Motivated by our experience with real video sequences, we decouple the estimation of the motions (moving objects and camera) from that of the remaining parameters. The motions are estimated on a frame by frame basis and then these estimates are used in the estimation of the remaining parameters. Then, we introduce the motion estimates into the penalized likelihood cost function and minimize it with respect to the remaining parameters.

The estimate of the texture of the object is obtained in closed form. To estimate the texture of the background and the template of the moving object, we develop a fast two-step iterative algorithm. The first step estimates the background texture for a fixed template—the solution is obtained in closed form. The second step estimates the object template for a fixed background—the solution is given by a simple binary test evaluated at each pixel. The algorithm converges in a few iterations, typically three to five iterations.

Our penalized likelihood cost function balances two terms. The first term is the *Maximum Likelihood* (ML) cost function. It is a measure of the error between the observed data and the model. The second term measures the size of the moving object, *i.e.*, the area of its template. The minimum of the first term, *i.e.*, the ML estimate, is not always sharply defined. In fact, for regions with low texture, the likelihood that this region belongs to the background is very similar to the likelihood that it belongs to the moving object. The penalization term addresses this difficulty and makes the segmentation problem well-posed: we look for the *smallest* template that describes well the observed data.

The penalization term has a second very relevant impact—it improves the convergence of the two-step iterative segmentation algorithm. Usually, with iterative minimization algorithms, it is important to have a good initial guess in order for the algorithm to exhibit good convergence. In our algorithm, we adopt a relaxation strategy for the weight of the penalization term. This enables us to avoid computationally expensive methods to compute the initial estimates. Our experience shows that this strategy makes the behavior of the algorithm quite insensitive to the initial guess, so much so that it suffices to initialize the process with the trivial guess of having no moving object, *i.e.*, every pixel is assumed to belong to the background.

Although related to the work of Irani, Rousso, and Peleg [25], [26], our approach models explicitly the *occlusion* of the background by the moving object, and we use the frames available to estimate the moving object template rather than just a single frame. Even when there is little contrast and the color of the moving object is very similar to the color of the background, our algorithm resolves accurately the moving object from the background, because it integrates over time existing small differences. Our approach also relates to the work of Jojic and Frey [32] in the sense that both approaches model the occlusion of the background by the moving object. However, our work is concerned with *rigid* shape, in contrast with [32] that is concerned with *flexible* shape. We can then exploit the *rigidity* of the object to derive a very *simple* algorithm that estimates with high accuracy the shape of the moving object, where all steps admit closed-form solutions. Although our work applies only to rigid moving objects, the simplicity of our algorithm enables us to consider more general class of motions—translations and rotations—than [32] that restricts the motions to single pixel translations. A final comment on our approach regards offline versus online and real time. Our approach, as [32], builds the object template by processing several frames. This

leads to an inherent delay so that we can accumulate a sufficient number of frames to resolve template ambiguities and to achieve high accuracy. The number of frames, and the corresponding delay, depends on the level of contrast between the moving object and the background and on the object texture; it may be acceptable or not acceptable in close-to-real time applications. In several sequences we tested, this number is on the order of tens of frames, requiring buffering the video from a fraction of a second to a few seconds. For example, with the "road traffic" video clip in section V, the maximum delay is 14 frames.

Paper organization In section II, we state the segmentation problem. We define the notation, develop the observation model, and derive the penalized likelihood cost function. In section III, we address the minimization of the cost function. To provide insight into the problem, we start by studying the ML estimation problem, *i.e.*, when no penalizing term is present; we detail a two-step iterative method that minimizes this ML term of the cost function. Section IV deals with penalized likelihood estimation. We discuss when ML estimation is ill-posed and address the minimization of the complete penalized likelihood cost function. In section V, we describe experiments that demonstrate the performance of our algorithm. Section VI concludes the paper.

The model used in the paper and described in section II was introduced in [36]. Preliminary versions of the ML-estimation step were presented in [36], [37], [38].

II. Problem Formulation

We discuss motion segmentation in the context of *Generative Video* (GV), see [39], [40], [41], [42]. GV is a framework for the analysis and synthesis of video sequences. In GV the operational units are not the individual images in the original sequence, as in standard methods, but rather the world images and the ancillary data. The world images encode the non-redundant information about the video sequence. They are augmented views of the world—background world image—and complete views of moving objects—figure world images. The ancillary data registers the world images, stratifies them at each time instant, and positions the camera with respect to the layering of world images. The world images and the ancillary data are the generative video representation, the information that is needed to regenerate the original video sequence. We formulate the moving object segmentation task as the problem of generating the world images and ancillary data for the generative video, [39], [40], [41], [42], representation of a video clip.

Motion analysis toward three-dimensional model-based video representations are treated in [43], [44].

A. Notation

We describe an image by a real-valued function defined on a subset of the real plane. The image space is a set $\{\mathbf{I} : \mathcal{D} \to \mathcal{R}\}$, where $\mathbf{I}$ is an image, $\mathcal{D}$ is the domain of the image, and $\mathcal{R}$ is the range of the image. The domain $\mathcal{D}$ is a compact subset of the real plane $\mathbb{R}^2$, and the range $\mathcal{R}$ is a subset of the real line $\mathbb{R}$. Examples of images in this paper are the frame f in a video sequence, denoted by $\mathbf{I}_f$, the background world image, denoted by $\mathbf{B}$, the moving object world image, denoted by $\mathbf{O}$, and the moving object template, denoted by $\mathbf{T}$. The images $\mathbf{I}_f$,

$\mathbf{B}$, and $\mathbf{O}$ have range $\mathcal{R} = \mathbb{R}$. They code intensity gray levels[1]. The template $\mathbf{T}$ of the moving object is a binary image, *i.e.*, an image with range $\mathcal{R} = \{0, 1\}$, defining the region occupied by the moving object. The domain of the images $\mathbf{I}_f$ and $\mathbf{T}$ is a rectangle corresponding to the support of the frames. The domain of the background world image $\mathbf{B}$ is a subset $\mathcal{D}$ of the plane whose shape and size depends on the camera motion, *i.e.*, $\mathcal{D}$ is the region of the background observed in the entire sequence. The domain $\mathcal{D}$ of the moving object world image is the subset of $\mathbb{R}^2$ where the template $\mathbf{T}$ takes the value 1, *i.e.*, $\mathcal{D} = \{(x, y) : \mathbf{T}(x, y) = 1\}$.

In our implementation, the domain of each image in the video sequence is rectangularly shaped with its size fitting the needs of the corresponding image. Although we use a continuous spatial dependence for commodity, in practice, the domains are discretized and the images are stored as matrices. We index the entries of each of these matrices by the pixels (x, y) of each image and refer to the value of image $\mathbf{I}$ at pixel (x, y) as $\mathbf{I}(x, y)$. Throughout the text, we refer to the image product of two images $\mathbf{A}$ and $\mathbf{B}$, *i.e.*, the image whose value at pixel (x, y) equals $\mathbf{A}(x, y)\mathbf{B}(x, y)$, as the image $\mathbf{AB}$. Note that this corresponds to the Hadamard product, or elementwise product, of the matrices representing images $\mathbf{A}$ and $\mathbf{B}$, not their matrix product.

We consider 2-D parallel motions, *i.e.*, all motions (translations and rotations) are parallel to the camera plane. We represent this type of motions by specifying time varying position vectors. These vectors code rotation-translation pairs that take values in the group of rigid transformations of the plane, the special Euclidean group SE(2). The image obtained by applying the rigid motion coded by the vector $\mathbf{p}$ to the image $\mathbf{I}$ is denoted by $\mathcal{M}(\mathbf{p})\mathbf{I}$. The image $\mathcal{M}(\mathbf{p})\mathbf{I}$ is also usually called the registration of the image $\mathbf{I}$ according to the position vector $\mathbf{p}$. The entity represented by $\mathcal{M}(\mathbf{p})$ is seen as a motion operator. In practice, the (x, y) entry of the matrix representing the image $\mathcal{M}(\mathbf{p})\mathbf{I}$ is given by $\mathcal{M}(\mathbf{p})\mathbf{I}(x, y) = \mathbf{I}(f_x(\mathbf{p}; x, y), f_y(\mathbf{p}; x, y))$ where $f_x(\mathbf{p}; x, y)$ and $f_y(\mathbf{p}; x, y)$ represent the coordinate transformation imposed by the 2D rigid motion. We use bilinear interpolation to compute the intensity values at points that fall in between the stored samples of an image.

The motion operators can be composed. The registration of the image $\mathcal{M}(\mathbf{p})\mathbf{I}$ according to the position vector $\mathbf{q}$ is denoted by $\mathcal{M}(\mathbf{qp})\mathbf{I}$. By doing this we are using the notation $\mathbf{qp}$ for the composition of the two elements of SE(2), $\mathbf{q}$ and $\mathbf{p}$. We denote the inverse of $\mathbf{p}$ by $\mathbf{p}^{\#}$, *i.e.*, the vector $\mathbf{p}^{\#}$ is such that when composed with $\mathbf{p}$ we obtain the identity element of SE(2). Thus, the registration of the image $\mathcal{M}(\mathbf{p})\mathbf{I}$ according to the position vector $\mathbf{p}^{\#}$ obtains the original image $\mathbf{I}$, so we have $\mathcal{M}(\mathbf{p}^{\#}\mathbf{p})\mathbf{I} = \mathcal{M}(\mathbf{pp}^{\#})\mathbf{I} = \mathbf{I}$. Note that, in general, the elements of SE(2) do not commute, *i.e.*, we have $\mathbf{qp} \neq \mathbf{pq}$, and $\mathcal{M}(\mathbf{qp})\mathbf{I} \neq \mathcal{M}(\mathbf{pq})\mathbf{I}$. Only in special cases is the composition of the motion operators not affected by the order of their application, as for example when the motions $\mathbf{p}$ and $\mathbf{q}$ are pure translations or pure rotations.

The notation for the position vectors involved in the segmentation problem is as follows. The vector $\mathbf{p}_f$ represents the position of the background world image relative to the camera in frame f. The vector $\mathbf{q}_f$ represents the position of the moving object relative to the camera in frame f.

---

[1]For simplicity, we take the pixel intensities to be real valued, although, in practice, they are integer valued in the range $[0 \cdots 255]$. The analysis in the paper is easily extended to color images by specifying color either by the perceptual attributes *brightness*, *hue*, and *saturation* or by the primary colors *red*, *green*, and *blue*, see [45].

B. Observation model

The observation model considers a scene with a moving object in front of a moving camera with 2-D parallel motions. The pixel (x, y) of the image $\mathbf{I}_f$ belongs either to the background world image $\mathbf{B}$ or to the object world image $\mathbf{O}$. The intensity $\mathbf{I}_f(x, y)$ of the pixel (x, y) is modeled as

$$\mathbf{I}_f(x, y) = \mathcal{M}(\mathbf{p}_f^{\#})\mathbf{B}(x, y) \left[1 - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x, y)\right] + \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{O}(x, y)\,\mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x, y) + \mathbf{W}_f(x, y)\,, \qquad (1)$$

where $\mathbf{T}$ is the moving object template, $\mathbf{p}_f$ and $\mathbf{q}_f$ are the camera pose and the object position, and $\mathbf{W}_f$ stands for the observation noise, assumed Gaussian, zero mean, and white.

Equation (1) states that the intensity of the pixel (x, y) on frame f, $\mathbf{I}_f(x, y)$, is a noisy version of the true value of the intensity level of the pixel (x, y). If the pixel (x, y) of the current image belongs to the template of the object, $\mathbf{T}$, after the template is compensated by the object position, *i.e.*, registered according to the vector $\mathbf{q}_f^{\#}$, then $\mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x, y) = 1$. In this case, the first term of the right hand side of (1) is zero, and the image intensity $\mathbf{I}_f(x, y)$ reduces to a noisy version of the second term. This second term, $\mathcal{M}(\mathbf{q}_f^{\#})\mathbf{O}(x, y)$, is the intensity of the pixel (x, y) of the moving object. In other words, the intensity $\mathbf{I}_f(x, y)$ equals the object intensity $\mathcal{M}(\mathbf{q}_f^{\#})\mathbf{O}(x, y)$ corrupted by the noise $\mathbf{W}_f(x, y)$. On the other hand, if the pixel (x, y) does not belong to the template of the object, $\mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x, y) = 0$, the pixel belongs then to the background world image $\mathbf{B}$, registered according to the inverse $\mathbf{p}_f^{\#}$ of the camera position. In this case, the intensity $\mathbf{I}_f(x, y)$ is a noisy version of the background intensity $\mathcal{M}(\mathbf{p}_f^{\#})\mathbf{B}(x, y)$. We want to emphasize that, rather than modeling simply the two different motions, as usually done in other approaches that process only two consecutive frames, expression (1) models the *occlusion* of the background by the moving object explicitly. Also, equation (1), which composites the image in the sequence by overlaying on the background image the image of the moving object at the appropriate position, assumes that the object is opaque. Transparency could be taken into consideration by affecting the middle term in (1) with a transparency index. We do not pursue this here.

Expression (1) is rewritten in compact form as

$$\mathbf{I}_f = \left\{\mathcal{M}(\mathbf{p}_f^{\#})\mathbf{B}\left[\mathbf{1} - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}\right] + \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{O}\,\mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T} + \mathbf{W}_f\right\}\mathbf{H}\,, \qquad (2)$$

where we assume that $\mathbf{I}_f(x, y) = 0$ for (x, y) outside the region observed by the camera. This is taken care of in equation (2) by the binary image $\mathbf{H}$ whose (x, y) entry is such that $\mathbf{H}(x, y) = 1$ if pixel (x, y) is in the observed images $\mathbf{I}_f$ or $\mathbf{H}(x, y) = 0$ if otherwise. The image $\mathbf{1}$ is constant with value 1.

Basically, the model in (2) describes the images in the sequence as a noisy version of a collage of 2-D images: the background, described by the background world image $\mathbf{B}$, and the moving object, described by the object world image $\mathbf{O}$. This model, which we have proposed in [36], [37], [38] is similar to the one used by Jojic and Frey in [32] to capture flexible moving objects. We will see that modeling the template $\mathbf{T}$ of the moving object as a fixed binary matrix, i.e., that the object is rigid, enables us to develop a very simple segmentation algorithm.

In (2), each image in the sequence is obtained by first registering the background with respect to the camera position, as given by $\mathbf{p}_f$, then registering the object with respect to the background, as given by $\mathbf{q}_f$, and, finally,

by clipping the composite of the background plus object by the field of view of the camera—operator $\mathbf{H}$. Since the background is first registered according to the camera motion, the clipping operator $\mathbf{H}$ does not depend on the frame index f.

Fig. 2 illustrates model (2) for 1-D frames $\mathbf{I}_f(x)$, where x is now a scalar. The top plot, a sinusoid, is the intensity $\mathbf{B}(x)$ of the background world image. The template $\mathbf{T}(x)$ of the moving object is shown on the left of the second level as the union of two disjoint intervals. The intensity level $\mathbf{O}(x)$ of the moving object is also sinusoidal, and is shown on the right plot on the second level. The frequency of this sinusoidal intensity is higher than the frequency of the background intensity $\mathbf{B}(x)$. The camera window $\mathbf{H}$ is the interval shown in the third level. It clips the region observed by the camera. The two bottom curves show two frames $\mathbf{I}_1$ and $\mathbf{I}_2$. They are given by a noise-free version of the model in expression (2). In between these two frames, both the camera and the object moved: the camera moved 2 pixels to the right, corresponding to the background motion in the opposite direction, while the object moved 3 pixels to the right relative to the camera. The observation model of expression (2) and the illustration of Fig. 2 emphasize the role of the building blocks involved in representing an image sequence $\{\mathbf{I}_f, 1 \le f \le F\}$ according to the generative video framework, [39], [40], [41], [42].
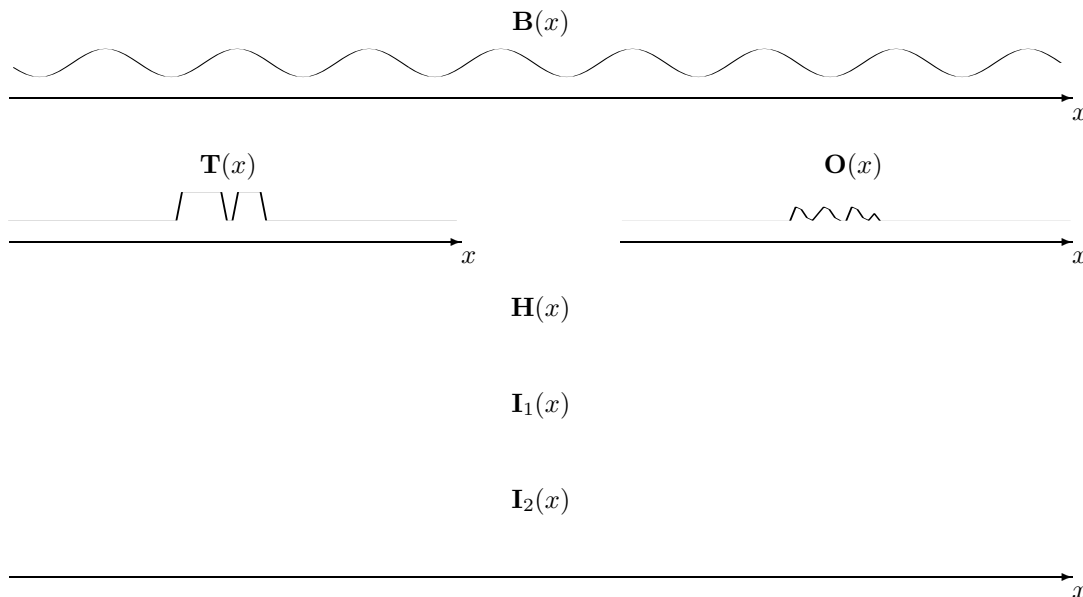

Fig. 2.   Illustration of the 1-D generative video image formation and observation model.

C. Energy minimization: Penalized likelihood estimation

Given F frames $\{\mathbf{I}_f, 1 \le f \le F\}$, we want to estimate the background world image $\mathbf{B}$, the object world image $\mathbf{O}$, the object template $\mathbf{T}$, the camera poses $\{\mathbf{p}_f, 1 \le f \le F\}$, and the object positions $\{\mathbf{q}_f, 1 \le f \le F\}$. The quantities $\{\mathbf{B}, \mathbf{O}, \mathbf{T}, \{\mathbf{p}_f\}, \{\mathbf{q}_f\}\}$ define the generative video representation, [39], [40], [41], [42], the information that is needed to regenerate the original video sequence.

The problem as stated may be very difficult. As an example, consider that the object moves in front of a constant intensity background, *i.e.*, the background has no texture. This image sequence is indistinguishable from an image sequence where the object template is arbitrarily enlarged with pixels whose intensity equals the intensity

of the background. Without additional knowledge, it is not possible to decide whether a pixel with intensity equal to the background intensity belongs to the background or to the moving object, *i.e.*, no algorithm can segment unambiguously the moving object. Although extreme, this example illustrates the difficulties of segmenting objects from backgrounds that have large patches with low texture (see the example in Fig. 1).

To address this issue, we assume that the object is small. This is in agreement with what the human visual system usually implicitly assumes. We incorporate this constraint into the segmentation problem by minimizing a cost function given by

$$C_{\mathrm{PL}} = C_{\mathrm{ML}} + \alpha \, \mathrm{Area}(\mathbf{T}) \,, \tag{3}$$

where C_{ML} is the ML cost function, derived below, α is a non-negative weight, and $\mathrm{Area}(\mathbf{T})$ is the area of the template. Minimizing the cost C_{PL} balances the agreement between the observations and the model (term C_{ML}) with minimizing the area of the template. The term $\alpha \, \mathrm{Area}(\mathbf{T})$ can be interpreted as a Bayesian prior and the cost function (3) as the negative log posterior probability whose minimization leads to the Maximum a Posteriori estimate, as usual in Bayesian inference approaches [46]. It can also be motivated through information-theoretic criteria like Akaike's AIC [47] or the Minimum Description Length principle [48]. Different basic principles lead to different choices for the parameter α but the structure of the cost function is still as in (3). Statisticians usually call the generic form (3) a *penalized likelihood* (PL) cost function [49]. Our choice for the weight α is discussed in section IV.

From the observation model (2) and the Gaussian white noise assumption, the likelihood is given by

$$p\left(\mathbf{B}, \mathbf{O}, \mathbf{T}, \{\mathbf{p}_f, \mathbf{q}_f\} \,|\, \{\mathbf{I}_f\}\right) = \prod_{f,x,y} \mathcal{N}\left(\mathbf{I}_f(x,y); \mathcal{M}(\mathbf{p}_f^{\#})\mathbf{B}(x,y)\Big[1 - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x,y)\Big] + \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{O}(x,y)\,\mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x,y), \sigma^2\right) \,. \tag{4}$$

By maximizing the logarithm of the likelihood (4), we derive the ML term of the penalized likelihood cost function (3) as[2]

$$C_{\mathrm{ML}} = \iint \sum_{f=1}^{F} \left\{ \mathbf{I}_f(x,y) - \mathcal{M}(\mathbf{p}_f^{\#})\mathbf{B}(x,y)\Big[1 - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x,y)\Big] - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{O}(x,y)\,\mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x,y) \right\}^2 \mathbf{H}(x,y)\, dx\, dy \,, \tag{5}$$

where the inner sum is over the full set of F frames and the outer integral is over all pixels.

The estimation of the parameters of (2) using the F frames rather than a single pair of images is a distinguishing feature of our work. Other techniques usually process only two or three consecutive frames. We use all frames available as needed. The estimation of the parameters through the minimization of a cost function that involves directly the image intensity values is another distinguishing feature of our approach. Other methods try to make some type of post-processing over incomplete template estimates. We process directly the image intensity values, through penalized likelihood estimation.

By describing the shape of the moving object by the binary template $\mathbf{T}$, we are able to express the ML cost function as in (5), *i.e.*, in terms of an integral whose region of integration is independent of the unknown shape. This enables developing a computationally simple algorithm to estimate the shape of the object. The same type of

---

[2]We use a continuous spatial dependence for simplicity. The variables x and y are continuous while f is discrete. In practice, the integral is approximated by the sum over all the pixels.

idea has been used in the context of the single-image intensity-based segmentation problem, for example, Ambrosio and Tortorelli [50] adapted Mumford and Shah theory [22] by using a continuous binary field instead of a binary edge process.

The minimization of the functional C_{PL} in (5) with respect to the set of generative video constructs $\{\mathbf{B}, \mathbf{O}, \mathbf{T}\}$ and to the motions $\{\{\mathbf{p}_f\}, \{\mathbf{q}_f\}, 1 \le f \le F\}$ is still a highly complex task. To obtain a computationally feasible algorithm, we simplify the problem. We decouple the estimation of the motions $\{\{\mathbf{p}_f\}, \{\mathbf{q}_f\}, 1 \le f \le F\}$ from the determination of the generative video constructs $\{\mathbf{B}, \mathbf{O}, \mathbf{T}\}$. This is reasonable from a practical point of view and is well supported by our experimental results with real videos.

The rationale behind the simplification is that the motion of the object (and the motion of the background) can be usually inferred without knowing precisely the object template. To better appreciate the complexity of the problem, consider an image sequence with no prior knowledge available, except that an object moves with respect to an unknown background. Even with no spatial cues, for example, if the background texture and the object texture are spatially white noise random variables, the human visual system can easily infer the motion of the background and the motion of the object from only two consecutive frames. However, this is not the case with respect to the template of the moving object; to infer an accurate template we need a much higher number of frames that enables us to easily capture the *rigidity* of the object across time. This observation motivated our approach of decoupling the estimation of the motions from the estimation of the remaining parameters.

We estimate the motions on a frame by frame basis using a simple sequential method, see [37] for the details. We first compute the dominant motion in the image, which corresponds to the motion of the background. Then, after compensating for the background motion, we compute the object motion. We estimate the parameters describing both motions by using a known motion estimation method, see [51]. After estimating the motions, we introduce the motion estimates into the penalized likelihood cost function and minimize with respect to the remaining parameters. Clearly, this solution is sub-optimal, in the sense that it is an approximation to the penalized likelihood estimate of the entire set of parameters, and it can be thought of as an initial guess for the minimizer of the penalized likelihood cost function given by (5). This initial estimate can then be refined by using a greedy approach. We emphasize that the key problem we address in this paper is finding the initial guess in an expedite way, not the final refinement.

III. Minimization Procedure

In this section, we assume that the motions have been correctly estimated and are known. In reality, the motions are continuously estimated. Assuming the motions are known, the problem becomes the minimization of the penalized likelihood cost function with respect to the remaining parameters, *i.e.*, with respect to the template of the moving object, the texture of the moving object, and the texture of the background.

A. Two-step iterative algorithm

Due to the special structure of the penalized likelihood cost function, we can express explicitly and with no approximations involved the estimate $\widehat{\mathbf{O}}$ of the texture of the object world image in terms of the template $\mathbf{T}$. Doing

this, we are left with the minimization of C_{PL} with respect to the template $\mathbf{T}$ and the texture of the background world image $\mathbf{B}$, still a nonlinear minimization. We approximate this minimization by a two-step iterative algorithm: (i) in step one, we solve for the background $\mathbf{B}$ while the template $\mathbf{T}$ is kept fixed; and (ii) in step two, we solve for the template $\mathbf{T}$ while the background $\mathbf{B}$ is kept fixed. We obtain closed-form solutions for the minimizers in each of the steps (i) and (ii). The two steps are repeated iteratively. The value of C_{PL} decreases along the iterative process. The algorithm proceeds till every pixel has been assigned unambiguously to either the moving object or to the background.

The initial guess in iterative algorithms is very relevant to the convergence of the algorithm—a bad initial guess may lead to convergence to a local optimum. As an initial guess, we may start with an estimate for the background like the average of the images in the sequence, including or not a robust statistic technique like outlier rejection, see for example [52]. The quality of this background estimate depends on the occlusion level of the background in the images processed. In [37], we propose a more elaborate technique that leads to better initial estimates of the background. However, sophisticate ad-hoc methods to recover the background result in computationally complex algorithms. In this paper, instead of using these algorithms, we use a continuation method, *i.e.*, we relax the cost function. We start from a cost for which we know we can find the global minimum, and then we gradually change the cost, keeping track of the minimum, to end at the desired cost function. Due to the structure of the penalized likelihood cost function (3), the continuation method is easily implemented by relaxing the weight α, as in annealing schedules, *e.g.*, stochastic relaxation [53]. We start with a high value for α such that the minimum of the cost (3) occurs at $\widehat{\mathbf{T}}(x, y) = 0, \forall_{x,y}$—it is clear from (3) that this is always possible. Then, we gradually decrease α and minimize the corresponding intermediate costs, till we reach the desired cost and the correct segmentation. In section IV, we discuss the impact of the final value of α.

To provide good insight into the problem, we start by studying the cost function (3) when there is no penalty term, *i.e.*, when $\alpha = 0$. The problem reduces to minimizing the term C_{ML}, *i.e.*, the ML cost function given by (5). This we do in the remaining of this section. In section IV, we come back to the general penalized likelihood cost function; we will see that the ML-analysis extends gracefully to penalized likelihood estimation.

B. Estimation of the moving object world image

We express the estimate $\widehat{\mathbf{O}}$ of the moving object world image in terms of the object template $\mathbf{T}$. By minimizing C_{ML} in (5) with respect to the intensity value $\mathbf{O}(x, y)$, we obtain the average of the pixels that correspond to the point (x, y) of the object. The estimate $\widehat{\mathbf{O}}$ of the moving object world image is then

$$\widehat{\mathbf{O}} = \mathbf{T} \frac{1}{F} \sum_{f=1}^{F} \mathcal{M}(\mathbf{q}_f) \mathbf{I}_f \,. \qquad (6)$$

This compact expression averages the observations $\mathbf{I}$ registered according to the motion $\mathbf{q}_f$ of the object in the region corresponding to the template $\mathbf{T}$ of the moving object.

We consider now separately the two steps of the iterative algorithm described above.

C. Step (i): estimation of the background for fixed template

To find the estimate $\widehat{\mathbf{B}}$ of the background world image, given the template $\mathbf{T}$, we register each term of the sum of C_{ML} in (5) according to the position of the camera $\mathbf{p}_f$ relative to the background. This is a valid operation because C_{ML} is defined as a sum over all the space $\{(x, y)\}$. We get

$$C_{\mathrm{ML}} = \iint \sum_{f=1}^{F} \left\{ \mathcal{M}(\mathbf{p}_f)\mathbf{I}_f - \mathbf{B}\left[1 - \mathcal{M}(\mathbf{p}_f \mathbf{q}_f^{\#})\mathbf{T}\right] - \mathcal{M}(\mathbf{p}_f \mathbf{q}_f^{\#})\mathbf{O}\,\mathcal{M}(\mathbf{p}_f \mathbf{q}_f^{\#})\mathbf{T}(x, y) \right\}^2 \mathcal{M}(\mathbf{p}_f)\mathbf{H}\, dx\, dy\,. \tag{7}$$

Minimizing the ML cost function C_{ML} given by (7) with respect to the intensity value $\mathbf{B}(x, y)$, we get the estimate $\widehat{\mathbf{B}}(x, y)$ as the average of the observed pixels that correspond to the pixel (x, y) of the background. The background world image estimate $\widehat{\mathbf{B}}$ is then written as

$$\widehat{\mathbf{B}} = \frac{\sum_{f=1}^{F}\left[\mathbf{1} - \mathcal{M}(\mathbf{p}_f \mathbf{q}_f^{\#})\mathbf{T}\right]\,\mathcal{M}(\mathbf{p}_f)\mathbf{I}_f}{\sum_{i=f}^{F}\left[\mathbf{1} - \mathcal{M}(\mathbf{p}_f \mathbf{q}_f^{\#})\mathbf{T}\right]\,\mathcal{M}(\mathbf{p}_f)\mathbf{H}}\,. \tag{8}$$

The estimate $\widehat{\mathbf{B}}$ of the background world image in (8) is the average of the observations $\mathbf{I}_f$ registered according to the background motion $\mathbf{p}_i$, in the regions $\{(x, y)\}$ not occluded by the moving object, *i.e.*, when $\mathcal{M}(\mathbf{p}_f \mathbf{q}_f^{\#})\mathbf{T}(x, y) = 0$. The term $\mathcal{M}(\mathbf{p}_f)\mathbf{H}$ provides the correct averaging normalization in the denominator by accounting only for the pixels seen in the corresponding image.

If we compare the moving object world image estimate $\widehat{\mathbf{O}}$ given by (6) with the background world image estimate $\widehat{\mathbf{B}}$ in (8), we see that $\widehat{\mathbf{O}}$ is linear in the template $\mathbf{T}$, while $\widehat{\mathbf{B}}$ is nonlinear in $\mathbf{T}$. This has implications when estimating the template $\mathbf{T}$ of the moving object, as we see next.

D. Step (ii): estimation of the template for fixed background

Let the background world image $\mathbf{B}$ be given and replace the object world image estimate $\widehat{\mathbf{O}}$ given by (6) in expression (5). The ML cost function C_{ML} becomes linearly related to the object template $\mathbf{T}$. Manipulating C_{ML} as described next, we obtain

$$C_{\mathrm{ML}} = \iint \mathbf{T}(x, y)\,\mathbf{Q}(x, y)\, dx\, dy + \mathrm{Constant}\,, \tag{9}$$

where $\mathbf{Q}$, which we call the *segmentation matrix*, is given by

$$\mathbf{Q}(x, y) = \mathbf{Q}_1(x, y) - \mathbf{Q}_2(x, y)\,, \tag{10}$$

$$\mathbf{Q}_1(x, y) = \frac{1}{F}\sum_{f=2}^{F}\sum_{g=1}^{f-1}\left[\mathcal{M}(\mathbf{q}_f)\mathbf{I}_f(x, y) - \mathcal{M}(\mathbf{q}_g)\mathbf{I}_g(x, y)\right]^2\,, \tag{11}$$

$$\mathbf{Q}_2(x, y) = \sum_{f=1}^{F}\left[\mathcal{M}(\mathbf{q}_f)\mathbf{I}_f(x, y) - \mathcal{M}(\mathbf{q}_f \mathbf{p}_f^{\#})\mathbf{B}(x, y)\right]^2\,. \tag{12}$$

On first reading, the reader may want to skip the derivation of expressions (9) to (12) and proceed till after equation (21) on page 13.

Derivation of expressions (9) to (12) Replace the estimate $\widehat{\mathbf{O}}$ of the moving object world image, given by (6), in expression (5), to obtain

$$C_{\mathrm{ML}} = \iint \sum_{f=1}^{F} \left\{ \mathbf{I} - \mathcal{M}(\mathbf{p}_f^{\#})\mathbf{B} \left[1 - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T} \right] - \frac{1}{F} \sum_{g=1}^{F} \mathcal{M}(\mathbf{q}_f^{\#}\mathbf{q}_g)\mathbf{I}_g \, \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T} \right\}^2 \mathbf{H} \, dx \, dy \, . \qquad (13)$$

Register each term of the sum according to the object position $\mathbf{q}_f$. This is valid because C_{ML} is defined as an integral over all the space $\{(x,y)\}$. The result is

$$C_{\mathrm{ML}} = \iint \sum_{f=1}^{F} \left\{ \left[\mathcal{M}(\mathbf{q}_f)\mathbf{I}_f - \mathcal{M}(\mathbf{q}_f\mathbf{p}_f^{\#})\mathbf{B} \right] + \left[\mathcal{M}(\mathbf{q}_f\mathbf{p}_f^{\#})\mathbf{B} - \frac{1}{F} \sum_{g=1}^{F} \mathcal{M}(\mathbf{q}_g)\mathbf{I}_g \right] \mathbf{T} \right\}^2 \mathcal{M}(\mathbf{q}_f)\mathbf{H} \, dx \, dy \, . \qquad (14)$$

In the remainder of the derivation, the spatial dependence is not important here, and we simplify the notation by omitting (x,y). We rewrite the expression for C_{ML} in compact form as

$$C_{\mathrm{ML}} = \iint \mathbf{C} \, dx \, dy \, , \qquad \text{where} \qquad \mathbf{C} = \sum_{f=1}^{F} \left\{ \left[\mathcal{I}_f - \mathcal{B}_f \right] + \left[\mathcal{B}_f - \frac{1}{F} \sum_{g=1}^{F} \mathcal{I}_g \right] \mathbf{T} \right\}^2 \mathcal{H}_f \, , \qquad (15)$$

$$\mathcal{I}_f = \mathcal{M}(\mathbf{q}_f)\mathbf{I}_f(x,y) \, , \qquad \mathcal{B}_f = \mathcal{M}(\mathbf{q}_f\mathbf{p}_f^{\#})\mathbf{B}(x,y) \, , \qquad \text{and} \qquad \mathcal{H}_f = \mathcal{M}(\mathbf{q}_f)\mathbf{H}(x,y) \, . \qquad (16)$$

Manipulating $\mathbf{C}$ under the assumption that the moving object is completely visible in the F images ($\mathbf{T}\mathcal{H}_f = \mathbf{T}, \forall_f$), and using the left equality in (19), we obtain

$$\mathbf{C} = \mathbf{T} \left\{ \sum_{f=1}^{F} \left[2\mathcal{I}_f\mathcal{B}_f - \mathcal{B}_f^2 \right] - \frac{1}{F} \left[\sum_{g=1}^{F} \mathcal{I}_g \right]^2 \right\} + \sum_{f=1}^{F} \left[\mathcal{I}_f - \mathcal{B}_f \right]^2 \mathcal{H}_f \, . \qquad (17)$$

The second term of $\mathbf{C}$ in (17) is independent of the template $\mathbf{T}$. To show that the sum that multiplies $\mathbf{T}$ is the segmentation matrix $\mathbf{Q}$ as defined by expressions (10), (11), and (12), write $\mathbf{Q}$ using the notation introduced in (16):

$$\mathbf{Q} = \frac{1}{F} \sum_{f=2}^{F} \sum_{g=1}^{f-1} \left[\mathcal{I}_f^2 + \mathcal{I}_g^2 - 2\mathcal{I}_f\mathcal{I}_g \right] - \sum_{f=1}^{F} \left[\mathcal{I}_f^2 + \mathcal{B}_f^2 - 2\mathcal{I}_f\mathcal{B}_f \right] \, . \qquad (18)$$

We now need the following equalities:

$$\left[\sum_{g=1}^{F} \mathcal{I}_g \right]^2 = \sum_{f=1}^{F} \sum_{g=1}^{F} \mathcal{I}_f\mathcal{I}_g \qquad \text{and} \qquad \sum_{f=2}^{F} \sum_{g=1}^{f-1} \left[\mathcal{I}_i^2 + \mathcal{I}_g^2 \right] = (F-1) \sum_{g=1}^{F} \mathcal{I}_g^2 \, . \qquad (19)$$

Manipulating (18), using the two equalities in (19), we obtain

$$\mathbf{Q} = \sum_{f=1}^{F} \left[2\mathcal{I}_f\mathcal{B}_f - \mathcal{B}_f^2 \right] - \frac{1}{F} \left[\sum_{g=1}^{F} \mathcal{I}_g^2 + 2 \sum_{f=2}^{F} \sum_{g=1}^{f-1} \mathcal{I}_f\mathcal{I}_g \right] \, . \qquad (20)$$

The following equality concludes the derivation:

$$\left[\sum_{g=1}^{F} \mathcal{I}_g \right]^2 = \sum_{g=1}^{F} \mathcal{I}_g^2 + 2 \sum_{f=2}^{F} \sum_{g=1}^{f-1} \mathcal{I}_f\mathcal{I}_g \, . \qquad (21)$$

$$\square$$

We estimate the template $\mathbf{T}$ by minimizing the ML cost function given by (9) over the template $\mathbf{T}$, given the background world image $\mathbf{B}$. It is clear from (9), that the minimization of C_{ML} with respect to each spatial location

of $\mathbf{T}$ is independent from the minimization over the other locations. The template $\widehat{\mathbf{T}}$ that minimizes the ML cost function C_{ML} is given by the following test evaluated at each pixel:

$$
\mathbf{Q}_1(x,y) \mathop{\substack{\widehat{\mathbf{T}}(x,y)=0 \\ > \\ < \\ \widehat{\mathbf{T}}(x,y)=1}} \mathbf{Q}_2(x,y) \ . \tag{22}
$$

The estimate $\widehat{\mathbf{T}}$ of the template of the moving object in (22) is obtained by checking which of two accumulated square differences is greater. In the spatial locations where the accumulated differences between each frame $\mathcal{M}(\mathbf{q}_f)\mathbf{I}_f$ and the background $\mathcal{M}(\mathbf{q}_g\mathbf{p}_g^{\#})\mathbf{B}$ are greater than the accumulated differences between each pair of co-registered frames $\mathcal{M}(\mathbf{q}_f)\mathbf{I}_f$ and $\mathcal{M}(\mathbf{q}_g)\mathbf{I}_g$, we estimate $\widehat{\mathbf{T}}(x,y)=1$, meaning that these pixels belong to the moving object. If not, the pixel is assigned to the background.

The reason why we did not replace the background world image estimate $\widehat{\mathbf{B}}$ given by (8) in (5) as we did with the object world image estimate $\widehat{\mathbf{O}}$ is that it leads to an expression for C_{ML} in which the minimization with respect to each different spatial location $\mathbf{T}(x,y)$ is not independent from the other locations. Solving this binary minimization problem by a conventional method is extremely time consuming. In contrast, the minimization of C_{ML} over $\mathbf{T}$ for fixed $\mathbf{B}$ results in a local binary test, which makes our solution computationally very simple. This closed-form solution is rooted on our assumption of *rigid* shape, which contrasts to the *flexible* shape model and the probabilistic learning approach of [32], where the solution is not in closed form.

We illustrate the template estimation step for a sequence of 1-D frames obtained with the generative video building blocks of Fig. 2. We synthesized an image sequence by using the model in (2). The camera position was chosen constant and the object position was set to increase linearly with time. The frame sequence obtained is represented in Fig. 3. Time increases from bottom to top. From the plot of Fig. 3 we can see that the background is stationary and the object moves from left to right.
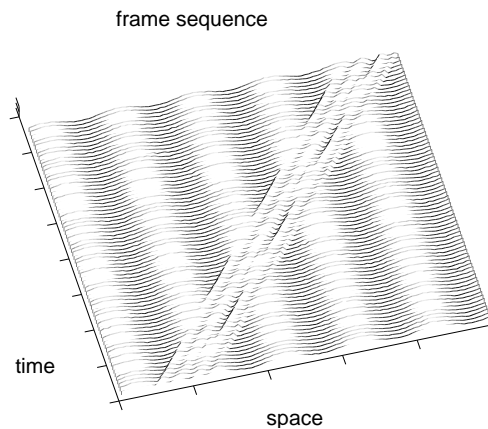

Fig. 3.   1-D image sequence synthesized with the generative video constructs of Fig. 2. Time increases from bottom to top.

The evolutions of the matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ (in this experiment, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are vectors because the frames are 1-D) are represented by the plots in Fig. 4. The left plot represents the evolution of $\mathbf{Q}_1$, while the right plot represents $\mathbf{Q}_2$. Time increases from bottom to top. At the beginning, when only a few frames have been taken into account, the values of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are small and the test (22) is inconclusive. As more observations are processed,

the absolute value of the difference between $\mathbf{Q}_1$ and $\mathbf{Q}_2$ rises and the test becomes unambiguous, see the evolution of the segmentation matrix $\mathbf{Q} = \mathbf{Q}_1 - \mathbf{Q}_2$ shown on the left plot of Fig. 5. When enough frames were processed, $\mathbf{Q}$ takes high positive values for pixels that do not belong to the template of the moving object, and negative values for pixels belonging to the template, see the shape of $\mathbf{Q}$ in the top of the left plot of Fig. 5 (the straight line at the bottom represents $\mathbf{Q} = \mathbf{0}$) and the template of the moving object in Fig. 2.
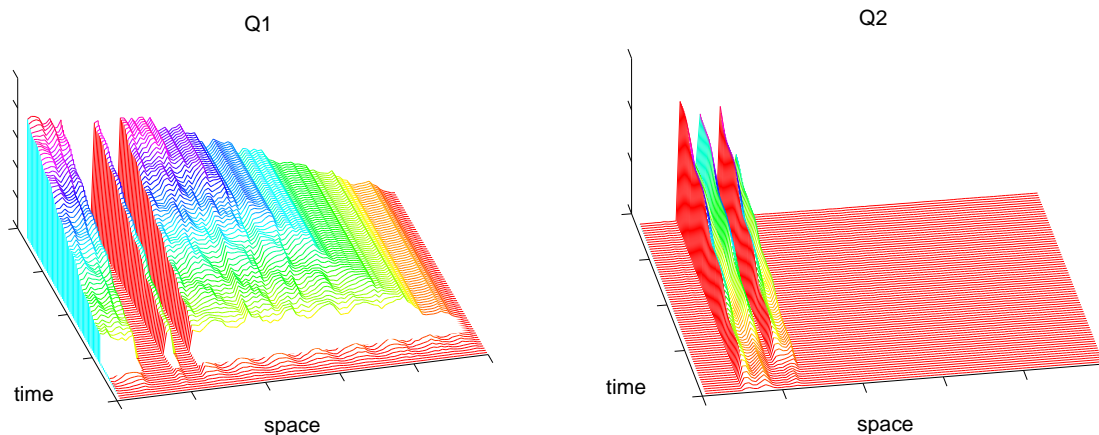
Fig. 4. Evolution of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ for the image sequence of Fig. 3. Time increases from bottom to top.
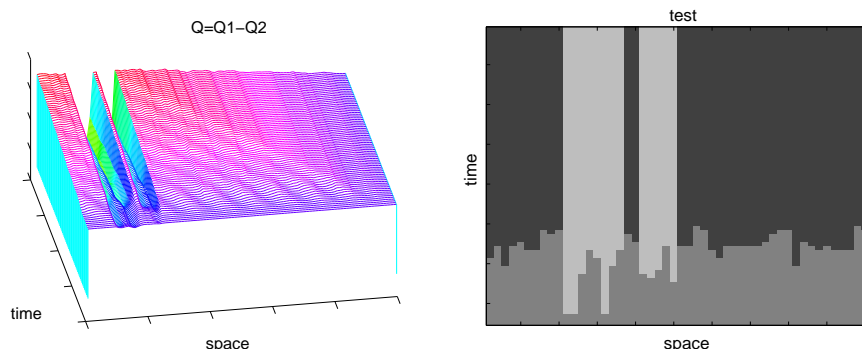
Fig. 5. Template estimation for the image sequence of Fig. 3. Left: evolution of the segmentation matrix $\mathbf{Q}$. Right: template estimates. Regions classified as belonging to the object template are light. Regions classified as not belonging to the template are dark. Middle grey regions correspond to the test (22) being inconclusive. In both plots, time increases from bottom to top.

On the right plot of Fig. 5, we show a grey level representation of the evolution of the result of the test (22). Time increases from bottom to top. Regions classified as belonging to the object template are light. Regions classified as not belonging to the template are dark. Middle grey regions correspond to the test (22) being inconclusive. Note that, after processing a number of frames, the regions are either light or dark, meaning that the test (22) is unambiguous at every spatial location. The right plot of Fig. 5 illustrates the convergence behavior of the template test—the estimates of the template of the moving object confirm the statement above about the evolution of the segmentation matrix $\mathbf{Q}$ in the left plot of Fig. 5, *i.e.*, we see that the sequence of estimates of the template converges to the true template, represented in Fig. 2.

The top row of the right plot in Fig. 5 shows the final estimate of the template of the moving object. It is equal to the actual template, represented in Fig. 2. In this example, the template of the moving object is the union of two

disjoint intervals. We see that the segmentation algorithm recovers successfully the template of the moving object even when it is a disconnected set of pixels.

IV. PENALIZED LIKELIHOOD

As anticipated in section II when we formulated the problem, it may happen that, after processing the F available frames, the test (22) remains inconclusive at a given pixel (x, y), *i.e.*, $\mathbf{Q}_1(x, y) \simeq \mathbf{Q}_2(x, y)$. In other words, it is not possible to decide if this pixel belongs to the moving object or to the background. This ambiguity comes naturally from the fact that the available observations are in agreement with both hypothesis. We make the decision unambiguous by looking for the *smallest* template that describes well the observations, through penalized likelihood estimation. Minimizing the penalized likelihood cost function (3), introduced in section II, balances the agreement between the observations and the model, with minimizing the area of the template.

A. Penalized likelihood estimation algorithm

We now modify the algorithm described in the previous section to address the minimization of the penalized likelihood cost function C_{PL} in (3). Re-write C_{PL} as

$$C_{\mathrm{PL}} = C_{\mathrm{ML}} + \alpha \operatorname{Area}(\mathbf{T}) = C_{\mathrm{ML}} + \alpha \iint \mathbf{T}(x, y) \, dx \, dy \,, \tag{23}$$

where C_{ML} is as in (5), α is non-negative, and $\operatorname{Area}(\mathbf{T})$ is the area of the template. Carrying out the minimization, first note that the second term in (23) does not depend on $\mathbf{O}$, neither on $\mathbf{B}$, so we get $\widehat{\mathbf{O}}_{\mathrm{PL}} = \widehat{\mathbf{O}}$ and $\widehat{\mathbf{B}}_{\mathrm{PL}} = \widehat{\mathbf{B}}$. By replacing $\widehat{\mathbf{O}}$ in C_{PL}, we get a modified version of (9),

$$C_{\mathrm{PL}} = \iint \mathbf{T}(x, y) \left[\mathbf{Q}(x, y) + \alpha \right] \, dx \, dy + \text{Constant} \,, \tag{24}$$

where the segmentation matrix $\mathbf{Q}$ is as defined in (10), (11), and (12). The penalized likelihood estimate of the template is then given by the following test, which extends test (22),

$$\mathbf{Q}(x, y) \begin{array}{c} \widehat{\mathbf{T}}_{\mathrm{PL}}(x, y) = 0 \\ > \\ < \\ \widehat{\mathbf{T}}_{\mathrm{PL}}(x, y) = 1 \end{array} - \alpha \,. \tag{25}$$

B. Relaxation

It is now clear that the strategy of relaxing the parameter α has an advantage with respect to the ML-only two-step algorithm of [38]. To emphasize this point, consider using ML as in section III-A initialized by estimating the background as the average of the co-registered input images, *i.e.*, the initial estimate of the background is contaminated by the moving object intensity values. It may happen that the next estimate of the template, obtained from test (22) is, erroneously, so large that, in the next step, the estimate of the background can not be computed at all pixels and the algorithm freezes and can not proceed. Consider now using the same initialization but with a relaxation scheme for the parameter α. Using the penalized likelihood test (25), with a large value for α, the next estimate of the template will be very small (the parameter α can even be set to a value such that the template

estimate will contain a single pixel). Using this template estimate, the next estimate of the background will be less contaminated by the moving object intensity values and thus closer to the true background. The next penalized likelihood estimate of the template, obtained from test (25) with a slightly smaller α, will then be slightly larger and closer to the true template of the moving object. This relaxation proceeds until the parameter α reaches either zero, leading to the ML estimate minimizing (5), or a value previously chosen, leading to the penalized likelihood estimate minimizing (23).

We illustrate the impact of the relaxation procedure by using again a 1-D example. The moving object template is represented on the left plot of Fig. 6. It is composed by four segments of different lengths. We synthesized eleven 1-D images by moving the object from left to right with a constant speed of two pixels per frame. Each of the line of the right plot of Fig. 6, labeled from top to bottom from 1 to 11, shows one resulting image and the full right plot shows the image sequence. As this plot clearly shows, the noise and the similarity between the textures of the background and the object makes it very challenging to obtain an accurate segmentation.
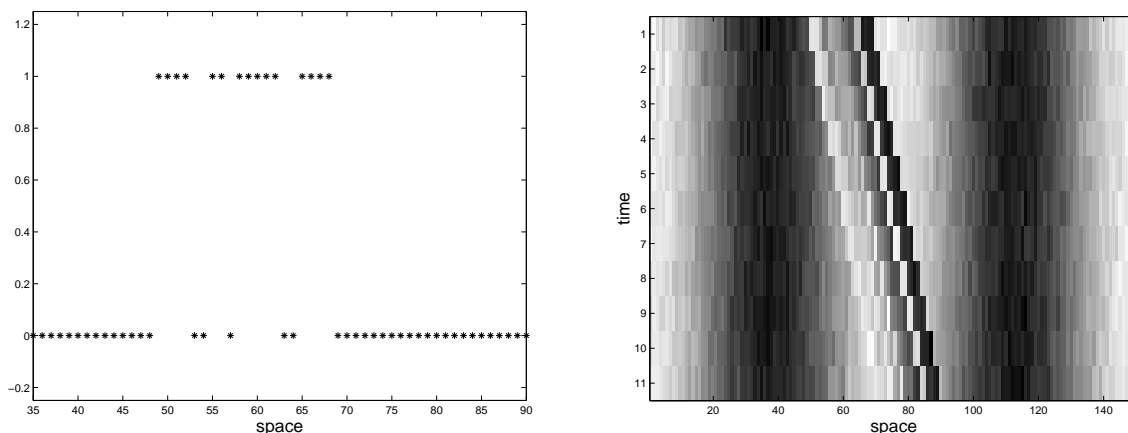

Fig. 6.   Left: 1-D template $[1\,1\,1\,1\,0\,0\,1\,1\,0\,1\,1\,1\,1\,1\,1\,0\,0\,1\,1\,1\,1]$. Right: 1-D image sequence. Time increases from top to bottom.

The plots of Fig. 7 illustrate the behavior of the algorithm with the relaxation procedure just outlined. Evolution occurs from the top-left plot to the bottom-right plot. Each plot shows: i) the symmetry[3] of the entries of the segmentation matrix $\mathbf{Q}$, marked with a solid line; ii) the value of the threshold parameter α, marked with a dashed line; and iii) the estimate $\widehat{\mathbf{T}}_{\mathrm{PL}}$ of the template, marked with stars ("*"). The top-left plot represents the first penalized likelihood test (25) after initializing the background estimate by averaging the images in the sequence. From this plot we see that the threshold parameter α is high enough such that only one pixel is classified as belonging to the object template, *i.e.*, only one entry of the symmetric segmentation matrix is above the threshold α. The values of the segmentation matrix in this plot make clear that, if α was set to zero at this early stage, the template would be clearly overestimated (compare with the true template in the left plot Fig. 6), the next background estimate would be incomplete, and the two-step algorithm could not proceed. On the other hand, by choosing the value of α in such a way that only one pixel is classified as belong to the template, the algorithm is able to refine the background estimate, leading to the second template test, represented on the second plot from left on the top of Fig. 7. Here, we

---

[3]We represent the entries of negative $\mathbf{Q}$, *i.e.*, $-\mathbf{Q}$, because those are the values to be compared with the weight α through (25).

decrease the value of the threshold α, enabling a second pixel to be classified as belonging to the template. In this example, the relaxation process continues until α reaches zero, leading to the ML estimate. To ease visualization, we use a different vertical scale for the last eight plots. The final estimate, represented on the bottom-right plot, shows that our method successfully segmented the moving object from the image sequence on the right of Fig. 6 (compare with the left plot of Fig. 6).
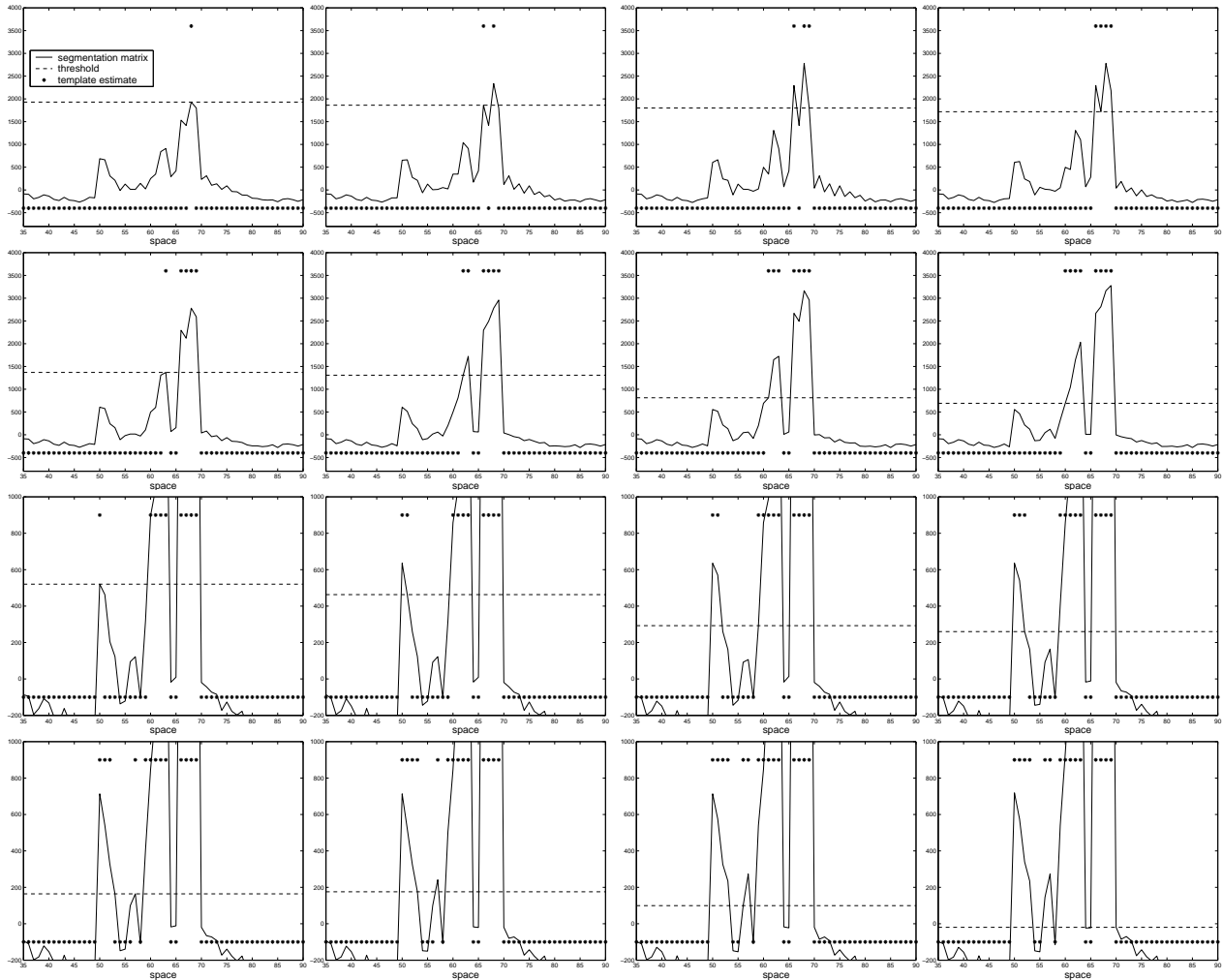
Fig. 7. Relaxation for the 1-D image sequence of Fig. 6. Evolution occurs from the top-left plot to the bottom-right plot. Each plot shows: i) the symmetry of the entries of the segmentation matrix $\mathbf{Q}$, marked by a solid line; ii) the value of the threshold parameter α, marked by a dashed line; and iii) the estimate $\widehat{\mathbf{T}}_{\mathrm{PL}}$ of the template, marked by stars ("*"). The final estimate, on the bottom-right plot, shows that our method successfully segmented the moving object (compare with the left plot of Fig. 6).

In general, the relaxation of α can be made faster than we did for this example, *i.e.*, at each step several pixels can be added to the estimate of the template. Anyway, any relaxation procedure for our segmentation algorithm should stop and decrease the relaxation rate whenever a background estimate returns incomplete. In our experiments with real video, we decreased α linearly, in four to five steps.

C. Stopping criteria

To stop the relaxation process we could adopt as strategy to stop as soon as the estimate $\widehat{\mathbf{T}}_{\mathrm{PL}}$ of the template of the moving object stabilizes, *i.e.*, as soon as no more pixels are added to it. However, to resolve the problems with low contrast background that motivated the use of penalized likelihood estimation, we stop the relaxation when α reaches a pre-specified minimum value α_{MIN}. This α_{MIN} can be chosen by experimentation, but we can actually predict from the observation model (1) what are good choices for it. If the minimum value α_{MIN} is chosen very high, we risk that some pixel (x, y) of the moving object, *i.e.*, with $\mathbf{T}(x, y) = 1$, is erroneously classified as belonging to the background, since from test (25), $\mathbf{Q}(x, y) > -\alpha_{\mathrm{MIN}} \Rightarrow \widehat{\mathbf{T}}_{\mathrm{PL}}(x, y) = 0$. In [37], using the observation model (1) and the definition of the segmentation matrix $\mathbf{Q}$ in (10), (11), and (12), we show that the expected value of the entry $\mathbf{Q}(x, y)$ for a pixel (x, y) of the moving object, *i.e.*, with $\mathbf{T}(x, y) = 1$, can be approximated as

$$\mathrm{E}_{\mathbf{T=1}}\left\{\mathbf{Q}(x, y)\right\} \simeq -\sum_{f=1}^{F}\left[\mathbf{O}(x, y) - \mathcal{M}(\mathbf{q}_f \mathbf{p}_f^{\#})\mathbf{B}(x, y)\right]^2 . \tag{26}$$

This expression shows that, as we process more frames, $\mathrm{E}_{\mathbf{T=1}}\left\{\mathbf{Q}(x, y)\right\}$ becomes more negative, reducing the probability of $\mathbf{Q}(x, y) > -\alpha_{\mathrm{MIN}}$, and so of misclassifying the pixel (x, y) as belonging to the background[4]. Expression (26) then suggests that good choices for the threshold α_{MIN} are in the interval $]\,0, -\mathrm{E}_{\mathbf{T=1}}\left\{\mathbf{Q}\right\}\,[$. Since in practice we can not compute $\mathrm{E}_{\mathbf{T=1}}\left\{\mathbf{Q}\right\}$ because we do not know before hand what are the intensity levels of the object and the background, we assume a value S^2 for their average square difference and chose α_{MIN} in the middle of the interval, $]\,0, FS^2\,[$, where F is a constant. In our experiments, with 1-byte per pixel gray level images, *i.e.*, with intensities in the interval $[0, 255]$, we used $\alpha_{\mathrm{MIN}} = 20$, obtained by setting $S = 2$ and $F = 10$. Our experience has shown that any other value α_{MIN} not to close to the extremes of the above interval would lead to the same estimates.

V. EXPERIMENTS

We describe four experiments. The first two use challenging computer generated image sequences to illustrate the convergence of our algorithm and its capability to segment complex shaped moving objects and low contrast video. In the third and fourth experiments we use real video sequences. The third experiment illustrates the time evolution of the segmentation matrix. The fourth experiment segments a traffic video clip.

Complex shape We synthesized an image sequence, the "IST" sequence, according to the model described in section II. Fig. 8 shows the world images used. The left frame, from a real video, is the background world image. The moving object template is the logo of the *Instituto Superior Técnico* (IST) which is transparent between the letters. Its world image, shown on the right frame, is obtained by clipping with the IST logo a portion of one of the frames in the sequence. The task of reconstructing the object template is particularly challenging with this video sequence due to the low contrast between the object and the background and the complexity of the template. We synthesized a sequence of 20 images where the background is static and the IST logo moves around.

---

[4]In [37], using Tchebycheff inequality [54], we derive upper bounds for the probability of misclassification.
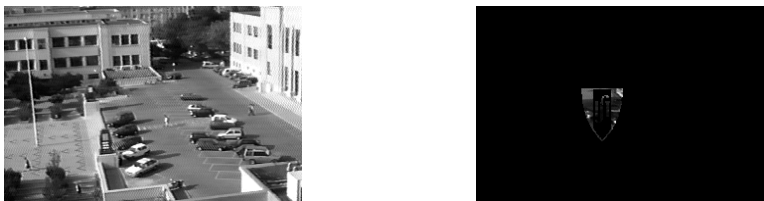
Fig. 8.   Constructs for the synthetic image sequence. Left: background. Right: moving object.

Fig. 9 shows three frames of the sequence obtained according to the image formation model introduced in section II, expression (2), with noise variance $\sigma^2 = 4$ (the intensity values are in the interval $[0, 255]$). The object moves from the center (left frame) down by translational and rotational motion. It is difficult to recognize the logo in the right frame because its texture is confused with the texture of the background.

Fig. 9.   Three frames of the image sequence synthesized with the constructs of Fig. 8.

Fig. 10 illustrates the four iterations it took for the two-step estimation method of our algorithm to converge. The template estimate is initialized to zero (top left frame). Each background estimate in the bottom was obtained using the template estimate on the top of it. Each template estimate was obtained using the previous background estimate. The arrows in Fig. 10 indicate the flow of the algorithm. The good template estimate obtained, see bottom left image, illustrates that our algorithm can estimate complex templates in low contrast background.
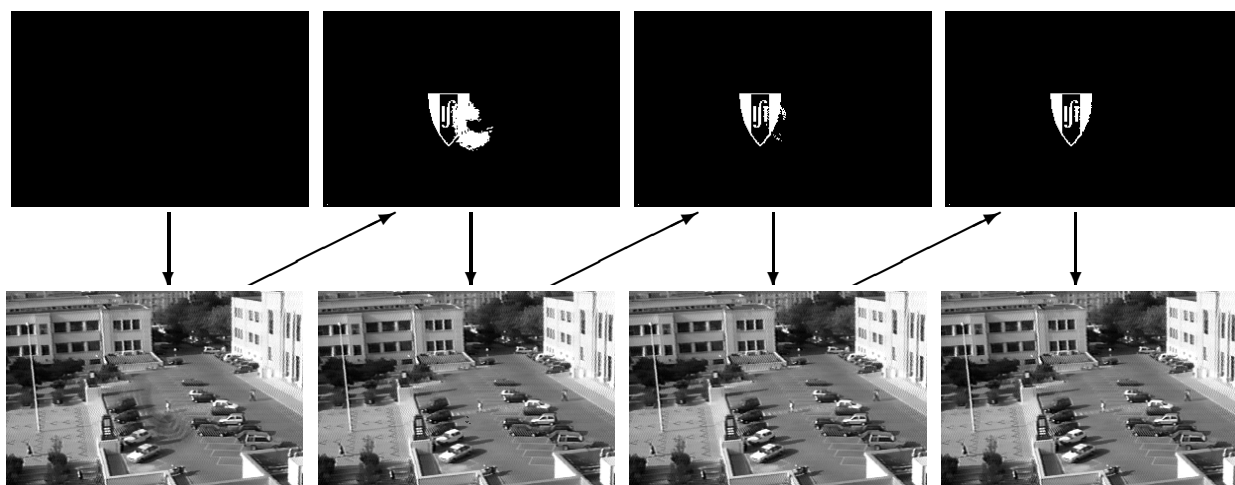

Fig. 10.   Two-step iterative method: template estimates and background estimates for the image sequence of Fig. 9.

Note that this type of complex templates (objects with transparent regions) is much easier to describe by using a binary matrix than by using contour based descriptions, like splines, Fourier descriptors, or snakes. Our algorithm

overcomes the difficulty arising from the higher number of degrees of freedom of the binary template by integrating over time the small intensity differences between the background and the object. The two-step iterative algorithm performs this integration in an expedite way.

Low contrast video By rotating and translating the object shown on the left image of Fig. 11, we synthesized 20 frames, two of which are shown in the middle and right images of Fig. 11. As these images clearly show, the noise and the similarity between the textures of the background and the object makes it very challenging to obtain an accurate segmentation.
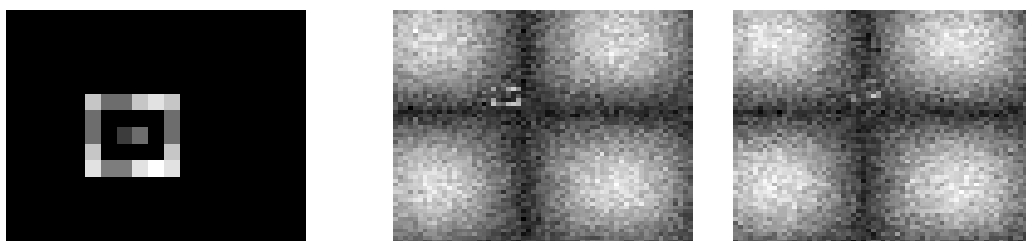
Fig. 11. Left: moving object. Middle and right: noisy video frames.

Fig. 12 describes the evolution of the estimate of the moving object template through the relaxation process described in section IV. The final estimate, shown in the bottom-right image of Fig. 12, shows that our algorithm was able to recover the true shape of the moving object (left image of Fig. 11).
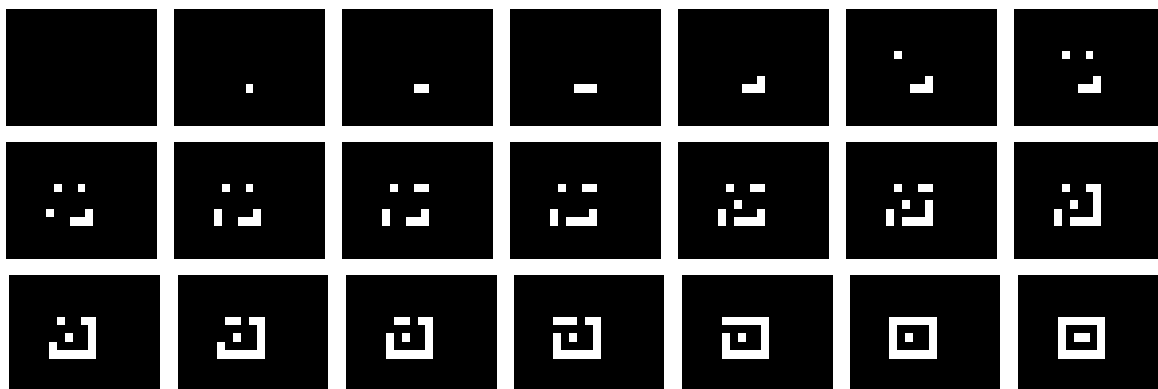

Fig. 12. Relaxation. Evolution of the estimate of the moving object template for the image sequence in Fig. 11. The final estimate (bottom-right) coincides with the true shape of the moving object in the left image of Fig. 11.

Robot soccer We used a sequence of 20 images, the "robot soccer" sequence, obtained from a robot soccer game, see [55]. It shows a white robot pursuing the ball. Frames $1, 4, 8$, and 16 of the robot soccer video sequence are in Fig. 13.

Although it is an easy task for humans to segment correctly the video sequence in Fig. 13, even looking at a single frame, this is not the case when motion is the only cue taken into account. In fact, due to the low texture of the regions of both the field and the robot, the robot template is ambiguous during the first frames of the sequence. This is because several regions belonging to the field can be incorrectly classified as belonging to the robot, since
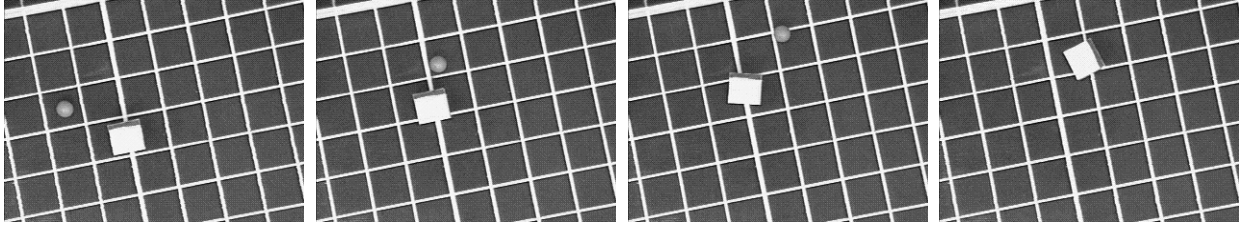
Fig. 13.   Robot soccer video sequence. Frames $1, 4, 8$, and 16.

the motion of the robot during the first frames is such that the video sequence would be the same whether or not those regions move rigidly with the robot. The same happens to regions of the robot that can be interpreted as being stationary with respect to the field. Only after the robot rotates, it is possible to determine, without ambiguity, its template.

Multiple objects The robot soccer video sequence contains two moving objects. Our algorithm deals with multiple moving objects by running the segmentation procedure, independently, for each of them. This basically requires estimating the motions of the independently moving objects. Since the algorithm does not require an accurate segmentation when estimating the image motion (in fact it does not require any segmentation at all since the algorithm uses in further steps only the motion estimates), we resolve the simultaneous estimation of the support regions and the corresponding motion parameters by using a fast and simple sequential method. We first estimate the motion parameters that best describe the motion of the entire image. Then, the images are co-registered according to the estimated motion. The pixels where the registered frame difference is below a threshold are considered to belong to the dominant region, which we assume is the background. Then, the dominant region is discarded and the process is repeated with the remaining pixels.

Applying the moving object template test, in expression (22), see section III-A, the ball template becomes unambiguous after 5 frames. Figure 14 shows the evolution of the robot template. Regions where the test is inconclusive are grey, regions classified as being part of the robot template are white, and regions classified as being part of the background are black. The robot template is unambiguous after 10 frames. The final robot template estimate is shown on the right side of Fig. 14.

Fig. 14.   Estimate of the robot template after frames $2, 4, 6$, and 10 of the video sequence of Fig. 13.

Figure 15 illustrates the evolution of the segmentation matrix $\mathbf{Q}$ introduced in section III-A. The curves on the left side plot the value of $\mathbf{Q}(x, y)$ for representative pixels (x, y) in the template of the robot. These curves start close to zero and decrease with the number of frames processed, as predicted by the analysis in section III. The curves on the right side plot of Fig. 15 represent the evolution of $\mathbf{Q}(x, y)$ for pixels not in the template of the

robot. For these pixels, $\mathbf{Q}(x, y)$ increases with the number of frames, again according to the analysis in section III. Thus, while during the first frames the value of $\mathbf{Q}(x, y)$ is close to zero and the template test is ambiguous (due to the low texture of the scene), after processing enough images the absolute value of $\mathbf{Q}(x, y)$ increases and the robot template becomes unambiguous.
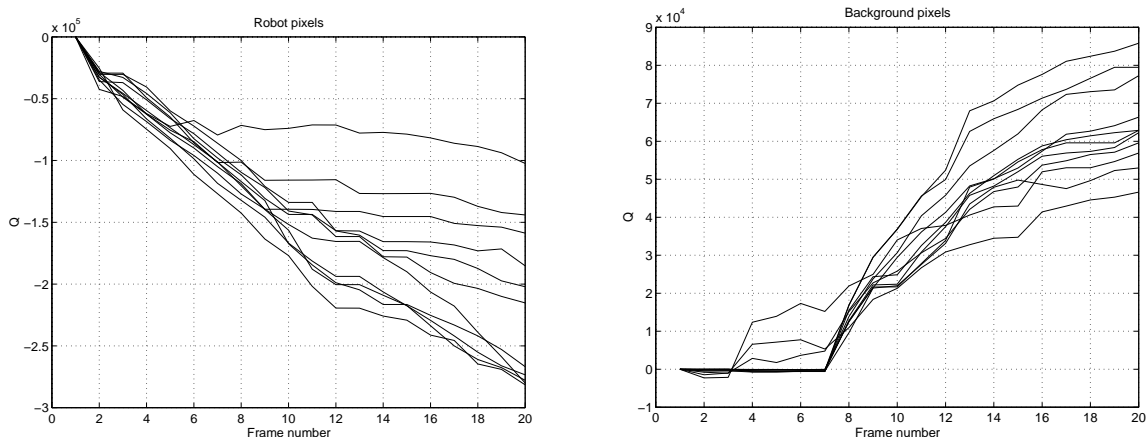

Fig. 15. Evolution of the entries $\mathbf{Q}(x, y)$ of the segmentation matrix $\mathbf{Q}$ for representative pixels: left plots are for pixels (x, y) in the robot template; right plots are for pixels (x, y) not in the robot template.

Figure 16 shows the recovered world images for the two moving objects and background, after processing the entire sequence of 20 frames.
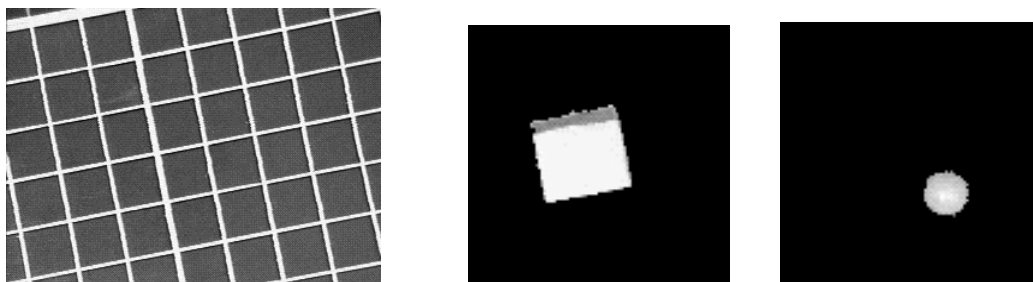

Fig. 16. Background, robot, and ball world images recovered from the robot soccer video sequence of Fig. 13.

Road traffic In this experiment we use a road traffic video clip. The road traffic sequence has 250 frames. Figure 17 shows frames 15, 166, and 225. The example given in section I to motivate the study of the segmentation of low textured scenes, see Fig. 1, also uses frames 76 and 77 from the road traffic video clip.

Fig. 17. Road traffic video sequence. Frames 15, 166, and 225.

In this video sequence, the camera exhibits a pronounced panning motion, while four different cars enter and leave the scene. The cars and the background have regions of low texture. The intensity of some of the cars is very similar to the intensity of parts of the background.

Figs. 18, top and bottom, show the good results obtained after segmenting the sequence with our algorithm. Fig. 18, bottom, displays the background world image, while Fig. 18, top, shows the world images of each of the moving cars. The estimates of the templates for the cars in Fig. 18 become, from left to right, unambiguous after 10, 10, and 14 frames, respectively.
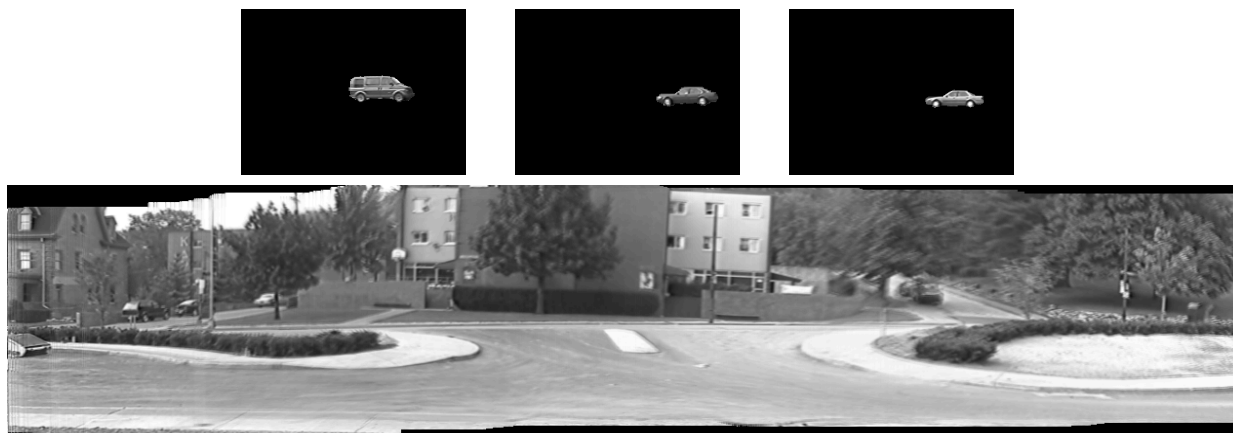

Fig. 18.   Top: Moving objects recovered from the road traffic video sequence of Fig. 17.; Bottom: Background world image recovered from the the road traffic video sequence of Fig. 17.

The CPU time taken by our algorithm to process a sequence of images depends on several factors, in particular, the level of relaxation used. With little or no relaxation, as used in our experiments with the IST, the robot soccer, and the road traffic sequences, to process a typical sequence of 20 video frames of 160×120 pixels takes about 1.75 sec with a non-optimized MATLAB implementation, running on a 2.4 GHz Pentium IV laptop. Using the same implementation of the algorithm but using a high degree of relaxation where the threshold is decreased very slowly this time can increase to 20 sec or even 30 sec.

VI. CONCLUSION

We develop an algorithm for segmenting 2-D rigid moving objects from an image sequence. Our method recovers the template of the moving object by processing *directly the image intensity values*. We model both the *rigidity* of the moving object over a set of frames and the *occlusion* of the background by the moving object. We estimate all unknowns (object and camera motions and textures) by an algorithm that approximates the minimization of a penalized likelihood (PL) energy functional. We first estimate the motion estimates, and then use a two-step iterative algorithm to approximate the minimization of the resulting cost function. The solutions for both steps are in closed form and so computationally very simple. Convergence is achieved in a small number of iterations (typically three to five iterations). Experiments show that the proposed algorithm can recover complex templates in low contrast scenes.

REFERENCES

[1] P. M. Q. Aguiar, R. S. Jasinschi, J. M. F. Moura, and C. Pluempitiwiriyawej, "Content-based image sequence representation," in *Video Processing*, T. Reed, Ed. Boca Raton: CRC Press, 2004, pp. 7–72, chapter 2.

[2] C. Gu and M.-C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, 1998.

[3] R. Castango, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features for interactive multimedia applications," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, 1998.

[4] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: A review," *IEEE Trans. on Image Processing*, vol. 3, no. 5, 1994.

[5] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," *Signal Processing: Image Communication*, vol. 3, no. 1, pp. 23–56, 1991.

[6] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[7] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Fort Collins CO, USA, 1999.

[8] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *IEEE Int. Conf. on Computer Vision*, Kerkyra, Greece, 1999.

[9] K.-P. Karmann, A. Brandt, and R. Gerl, "Moving object segmentation based on adaptive reference images," in *Signal Processing V: Theories and Application*, Elsevier Science Publishers, 2000.

[10] B. Li and M. I. Sezan, "Adaptive video background replacement," in *IEEE Int. Conf. on Multimedia*, Tokio, Japan, 2001.

[11] J. Pan, C.-W. Lin, C. Gu, and M.-T. Sun, "A robust spatio-temporal video object segmentation scheme with prestored background information," in *IEEE Int. Symp. on Circuits and Systems*, Arizona, USA, 2002.

[12] I. Haritaoglu, D. Harwood, and L. Davis, "W^4: Real-time surveillance of people and their activities," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–930, 2000.

[13] P. Bouthemy and E. François, "Motion segmentation and qualitative dynamic scene analysis from an image sequence," *Int. Jounal of Computer Vision*, vol. 10, no. 2, 1993.

[14] M.-P. Dubuisson and A. Jain, "Contour extraction of moving objects in complex outdoor scenes," *Int. Jounal of Computer Vision*, vol. 14, no. 1, 1995.

[15] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[16] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic snakes," *Int. Journal of Computer Vision*, vol. 22, pp. 61–79, 1997.

[17] R. Malladi, J. A. Sethian, and B. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–175, 1995.

[18] C. Pluempitiwiriyawej, J. M. F. Moura, Y.-J. L. Wu, S. Kanno, and C. Ho, "Stochastic active contour for cardiac MR image segmentation," in *ICIP'03, IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain, September 2003.

[19] ——, "Cardiac MR image segmentation: Quality assessment of STACS," in *ISBI'04, IEEE International Symposium on BioImaging*, Crystal City, VA, April 2004.

[20] A. Chakraborty and J. Duncan, "Game-theoretic integration for image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 1, pp. 12–30, 1999.

[21] T. Chan and L. Vese, "Active contours without edges," *IEEE Trans. on Image Processing*, vol. 10, no. 2, 2001.

[22] D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 1985.

[23] J. Morel and S. Solimini, *Variational Methods in Image Segmentation*. Boston: Birkhäuser, 1995.

[24] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.

[25] M. Irani and S. Peleg, "Motion analysis for image enhancement: Resolution, occlusion, and transparency," *Journal of Visual Communications and Image Representation*, vol. 4, no. 4, pp. 324–335, 1993.

[26] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions," *Int. Journal of Computer Vision*, vol. 12, no. 1, 1994.

[27] J. Wang and E. Adelson, "Representing moving images with layers," *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp. 625–638, 1994.

[28] H. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, 1996.

[29] S. Baker, R. Szeliski, and P. Anandan, "Hierarchical model-based motion estimation," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Santa Barbara CA, USA, 1998.

[30] H. Tao, H. Sawhney, and R. Kumar, "Dynamic layer representation with applications to tracking," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, 2000.

[31] N. Jojic, N. Petrovic, B. Frey, and T. Huang, "Transformed hidden markov models: Estimating mixture models of images and inferring spatial transformations in video sequences," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, 2000.

[32] N. Jojic and B. Frey, "Learning flexible sprites in video layers," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001.

[33] B. Frey, N. Jojic, and Kannan, "Learning subspace models of occluded objects in layers," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Madison, Wisconsin, 2003.

[34] Y. Zhou and H. Tao, "A background layer model for object tracking through occlusion," in *IEEE Int. Conf. on Computer Vision*, Nice, France, 2003.

[35] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*.   New York: John Wiley & Sons, 1997.

[36] P. M. Q. Aguiar and J. M. F. Moura, "Detecting and solving template ambiguities in motion segmentation," in *IEEE Int. Conf. on Image Processing*, Santa Barbara, CA, USA, 1997.

[37] P. M. Q. Aguiar, "Rigid structure from video," Ph.D. dissertation, Instituto Superior Técnico, Lisboa, Portugal, 2000, available at `http://www.isr.ist.utl.pt/~aguiar`.

[38] P. M. Q. Aguiar and J. M. F. Moura, "Maximum likelihood estimation of the template of a rigid moving object," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Sophia Antipolis, France, 2001, Springer–Verlag, LNCS 2134.

[39] R. S. Jasinschi and J. M. F. Moura, "Content-based video sequence representation," in *IEEE Int. Conf. on Image Processing*, Washigton D.C., USA, 1995.

[40] ——, *Generative Video: Very Low Bit Rate Video Compression*.   U.S. Patent and Trademark Office, S.N. 5,854,856, 1998.

[41] R. S. Jasinschi, J. M. F. Moura, J.-C. Cheng, and A. Asif, "Video compression via constructs," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, USA, 1995.

[42] R. S. Jasinschi, "Generative video: A meta video representation," Ph.D. dissertation, Carnegie Mellon University, USA, 1995.

[43] P. M. Q. Aguiar and J. M. F. Moura, "Three-dimensional modeling from two-dimensional video," *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1541–1551, 2001.

[44] ——, "Rank 1 weighted factorization for 3-D structure recovery: Algorithms and performance analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1134–1149, 2003.

[45] A. Jain, *Fundamentals of Digital Image Processing*.   Prentice-Hall International Inc., 1989.

[46] J. Berger, *Statistical Decision Theory and Bayesian Analysis*.   New York: Springer-Verlag, 1993.

[47] B. Ripley, *Pattern Recognition and Neural Networs*.   Cambridge University Press, 1996.

[48] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. on Information Theory*, vol. 44, no. 6, 1998.

[49] P. Green, "Penalized likelihood," in *Encyclopedia of Statistical Sciences*.   New York: John Wiley & Sons, 1998.

[50] L. Ambrosio and V. M. Tortorelli, "Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence," *Communications on Pure and Applied Mathematics*, vol. 43, no. 8, pp. 999–1036, 1990.

[51] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *European Conf. on Computer Vision*, Santa Margherita Ligure, Italy, 1992.

[52] M. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *Int. Jounal of Computer Vision*, vol. 19, no. 1, 1996.

[53] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions and the bayesian restoration of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 16, pp. 721–741, 1984.

[54] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw Hill, 1991.

[55] M. Veloso, P. Stone, S. Achim, and M. Bowling, "A layered approach for an autonomous robotic soccer system," in *Int. Conf. on Autonomous Agents*, Marina del Rey, CA, USA, 1997.