Lecture Notes in Computer Science 1654
International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition
July 1999, York, UK

Maximum Likelihood Inference of 3D Structure from Image Sequences

Pedro M. Q. Aguiar* and José M. F. Moura

Carnegie Mellon University
{aguiar,moura}@ece.cmu.edu

Abstract. The paper presents a new approach to recovering the 3D rigid shape of rigid objects from a 2D image sequence. The method has two distinguishing features: it exploits the rigidity of the object over the sequence of images, rather than over a pair of images; and, it estimates the 3D structure directly from the image intensity values, avoiding the common intermediate step of first estimating the motion induced on the image plane. The approach constructs the maximum likelihood (ML) estimate of all the shape and motion unknowns. We do not attempt the minimization of the ML energy function with respect to the entire set of unknown parameters. Rather, we start by computing the 3D motion parameters by using a robust factorization appraoch. Then, we refine the estimate of the object shape along the image sequence, by minimizing the ML-based energy function by a continuation-type method. Experimental results illustrate the performance of the method.

1 Introduction

The recovery of three-dimensional (3D) structure (3D shape and 3D motion) from a two-dimensional (2D) video sequence has been widely considered by the computer vision community. Methods that infer 3D shape from a single frame are based on cues such as shading and defocus. These methods fail to give reliable 3D shape estimates for unconstrained real-world scenes.

If no prior knowledge about the scene is available, the cue to estimating the 3D structure is the 2D motion of the brightness pattern in the image plane. For this reason, the problem is generally referred to as *structure from motion*. The two major steps in *structure from motion* are usually the following: compute the 2D motion in the image plane; and estimate the 3D shape and the 3D motion from the computed 2D motion.

Structure from motion Early approaches to *structure from motion* processed a single pair of consecutive frames and provided existence and uniqueness results to the problem of estimating 3D motion and absolute depth from the 2D motion in the camera plane between two frames, see for example [10]. Two-frame based algorithms are highly sensitive to image noise, and, when the object is far from

* The first author is also affiliated with Instituto Superior Técnico, Instituto de Sistemas e Robótica, Lisboa, Portugal. His work was partially supported by INVOTAN.

the camera, i.e., at a large distance when compared to the object depth, they fail even at low level image noise. More recent research has been oriented towards the use of longer image sequences. For example, in [8], the authors use a Kalman filter to integrate along time a set of two-frame depth estimates, while reference [4] uses nonlinear optimization to solve for the rigid 3D motion and the set of 3D positions of feature points tracked along a set of frames. References [9] and [2] estimate the 3D shape and motion by factorizing a measurement matrix whose entries are the set of trajectories of the feature point projections.

The approaches of the references above rely on the matching of a set of features along the image sequence. This task can be very difficult when processing noisy videos. In general, only distinguished points, as brightness corners, can be used as "trackable" feature points. As a consequence, those approaches do not provide dense depth estimates. In [1], we extended the factorization approach of [9] to recover 3D structure from a sequence of optical flow parameters. Instead of tracking pointwise features, we track regions where the optical flow is described by a single set of parameters. The approach of [1] is well suited to the analysis of scenes that can be well approximated with polyhedral surfaces. In this paper we seek dense depth estimates for general shaped surfaces.

To overcome the difficulties in estimating 3D structure through the 2D motion induced onto the image plane, some researchers have used techniques that infer 3D structure directly from the image intensity values. For example [6] estimates directly the 3D structure parameters by using the *brightness change constraint* between two consecutive frames. Reference [5] builds on this work by using a Kalman filter to update the estimates over time.

Proposed approach To formulate the problem of inferring the 3D structure from a video sequence, we use the analogy between the visual perception mechanism and a classical communication system. This analogy has been used to deal with perception tasks involving a single image, such as texture segmentation, and the recovering of shape from texture, see for example [7]. In a communication system, the transmitter receives a message S to be sent to the receiver. The transmitter codes the message and sends the resulting signal I^* through the channel, to the receiver. The receiver gets the signal I, a noisy version of the signal I^*. The receiver decodes I obtaining the estimate $\widehat{S}$ of the message S. In statistical communications theory, we describe statistically the channel distortion and design the receiver according to a statistically optimal criteria. For example, we can estimate $\widehat{S}$ as the message S that maximizes the probability of receiving the signal I, conditioned on the message S sent. This is the *Maximum Likelihood* (ML) estimate.

The communication system is a good metaphor for the problem of recovering 3D structure from video. The message source is the 3D environment. The transmitter is the geometric projection mechanism that transforms the real world S into an ideal image I^*. The channel is the camera that captures the image I, a noisy version of I^*. The receiver is the video analysis system. The task of this system is to recover the real world that has originated the image sequence captured.

According to the analogy above, we recover the 3D structure from the video sequence by computing the ML estimate of all the unknowns: the parameters describing the 3D motion, the object shape, and the object texture. A distinguishing feature of our work is the formulation of the estimate from a set of images, rather than a single pair. This provides accurate estimates for the 3D structure, due to the 3D rigidity of the scene. The formulation of the ML estimate from a set of frames leads to the minimization of a complex energy function. To minimize the ML energy function, we solve for the object texture in terms of the 3D shape and the 3D motion parameters. By replacing the texture estimate, we are left with the minimization of the ML energy function with respect to the 3D shape and 3D motion. We do not attempt the minimization of the ML energy function with respect to the entire set of unknown parameters by using generic optimization methods. Rather, we exploit the specific characteristics of the problem to develop a computationally feasible approximation to the ML solution. We compute the 3D motion by using the factorization method detailed in [2]. In fact, experiments with real videos show that the 3D rigid motion can be computed with accuracy through the optical flow computed across a set of frames for a small number of distinguished points or regions. After estimating the 3D motion, we are left with the minimization of the ML energy function with respect to the 3D shape. We propose a computationally simple continuation method to solve this non-linear minimization. Our algorithm starts by estimating coarse approximations to the 3D shape. Then, it refines the estimate as more images are being taken into account. The computational simplicity of our algorithm comes from the fact that each refinement stage, although non-linear, is solved by a simple Gauss-Newton method that requires no more than one or two iterations.

Our approach provides an efficient way to cope with the ill-posedness of estimating the motion in the image plane. In fact, the local *brightness change constraint* leads to a single restriction, which is insufficient to determine the two components of the local image motion (the so called *aperture problem*). Our method of estimating directly the 3D shape overcomes the *aperture problem* because we are left with the local depth as a single unknown, after computing the 3D motion in a first step.

In this paper we model the image formation process by assuming orthogonal projections. Orthogonal projections have been used as a good approximation to the perspective projection when the object is far from the camera [9, 1, 2]. With this type of scenes, two-frame based methods fail to estimate the absolute depth. Although formulated assuming orthogonal projections, which leads to estimates of the relative depth, our method can be easily extended to cope with perspective projections, which then leads to estimates of the absolute depth.

Paper organization Section 2 formulates the problem. Section 3 discusses the ML estimate. Section 4 summarizes the factorization method used to estimate the 3D motion. Section 5 describes the continuation method used to minimize the ML energy function. Experiments are in section 6. Section 7 concludes the paper.

2 Problem Formulation

We consider a rigid object $\mathcal{O}$ moving in front of a camera. We define the 3D motion of the object by specifying the position of the object coordinate system relative to the camera coordinate system. The position and orientation of $\mathcal{O}$ at time instant f is represented by $\boldsymbol{m}_f = \left\{t_{uf}, t_{vf}, t_{wf}, \theta_f, \phi_f, \psi_f\right\}$ where $\left(t_{uf}, t_{vf}, t_{wf}\right)$ are the coordinates of the origin of the object coordinate system with respect to the camera coordinate system (3D translation), and $(\theta_f, \phi_f, \psi_f)$ are the Euler angles that determine the orientation of the object coordinate system relative to the camera coordinate system (3D rotation).

Observation model The frame $\boldsymbol{I}_f$ captured at time f, $1 \leq f \leq F$, is modeled as a noisy observation of the projection of the object

$$\boldsymbol{I}_f = \mathcal{P}\left(\mathcal{O}, \boldsymbol{m}_f\right) + \boldsymbol{W}_f. \tag{1}$$

We assume that $\mathcal{P}$ is the orthogonal projection operator. For simplicity, the observation noise $\boldsymbol{W}_f$ is zero mean, white, and Gaussian.

The object $\mathcal{O}$ is described by its 3D shape $\mathcal{S}$ and texture $\mathcal{T}$. The texture $\mathcal{T}$ represents the light received by the camera after reflecting on the object surface, i.e., the texture $\mathcal{T}$ is the object brightness as perceived by the camera. The texture depends on the object surface photometric properties, as well as on the environment illumination conditions. We assume that the texture does not change with time.

The operator $\mathcal{P}$ returns the texture $\mathcal{T}$ as a real valued function defined over the image plane. This function is a nonlinear mapping that depends on the object shape $\mathcal{S}$ and the object position $\boldsymbol{m}_f$. The intensity level of the projection of the object at pixel $\boldsymbol{u}$ on the image plane is

$$\mathcal{P}\left(\mathcal{O}, \boldsymbol{m}_f\right)(\boldsymbol{u}) = \mathcal{T}\left(\boldsymbol{s}_f(\mathcal{S}, \boldsymbol{m}_f; \boldsymbol{u})\right), \tag{2}$$

where $\boldsymbol{s}_f(\mathcal{S}, \boldsymbol{m}_f; \boldsymbol{u})$ is the nonlinear mapping that lifts the point $\boldsymbol{u}$ on the image $\boldsymbol{I}_f$ to the corresponding point on the 3D object surface. This mapping $\boldsymbol{s}_f(\mathcal{S}, \boldsymbol{m}_f; \boldsymbol{u})$ is determined by the object shape $\mathcal{S}$, and the position $\boldsymbol{m}_f$. To simplify the notation, we will usually write explicitly only the dependence on f, i.e., $\boldsymbol{s}_f(\boldsymbol{u})$. Figure 1 illustrates the lifting mapping $\boldsymbol{s}_f(\boldsymbol{u})$ and the direct mapping $\boldsymbol{u}_f(\boldsymbol{s})$ for the orthogonal projection of a two-dimensional object. The inverse mapping $\boldsymbol{u}_f(\boldsymbol{s})$ also depends on $\mathcal{S}$ and $\boldsymbol{m}_f$, but we will, again, usually show only explicitly the dependence on f. On the left of figure 1, the point $\boldsymbol{s}$ on the surface of the object projects onto $\boldsymbol{u}_f(\boldsymbol{s})$ on the image plane. On the right, pixel $\boldsymbol{u}$ on the image plane is lifted to $\boldsymbol{s}_f(\boldsymbol{u})$ on the object surface. We assume that the object does not occlude itself, i.e., we have $\boldsymbol{u}_f(\boldsymbol{s}_f(\boldsymbol{u})) = \boldsymbol{u}$ and $\boldsymbol{s}_f(\boldsymbol{u}_f(\boldsymbol{s})) = \boldsymbol{s}$. The mapping $\boldsymbol{u}_f(\boldsymbol{s})$, seen as a function of the frame index f, for a particular surface point $\boldsymbol{s}$, is the trajectory of the projection of that point in the image plane, i.e., it is the motion induced in the image plane, usually referred to as *optical flow*.

The observation model (1) is rewritten by using (2) as

$$\boldsymbol{I}_f(\boldsymbol{u}) = \mathcal{T}\left(\boldsymbol{s}_f(\boldsymbol{u})\right) + \boldsymbol{W}_f(\boldsymbol{u}). \tag{3}$$

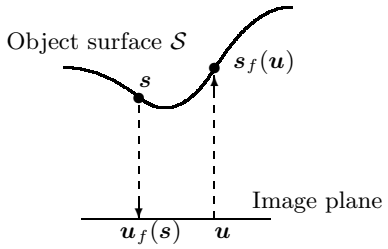

Fig. 1. Mappings $\boldsymbol{u}_f(\boldsymbol{s})$ and $\boldsymbol{s}_f(\boldsymbol{u})$.

We consider the estimation of the 3D shape $\mathcal{S}$ and the 3D motion $\{\boldsymbol{m}_f, 1 \leq f \leq F\}$ of the object $\mathcal{O}$ given the video sequence $\{\boldsymbol{I}_f, 1 \leq f \leq F\}$ of F frames.

Maximum Likelihood estimate formulation Given the observation model, the 3D shape and the 3D motion of the object $\mathcal{O}$ are recovered from the video sequence $\{\boldsymbol{I}_f, 1 \leq f \leq F\}$ by estimating all the unknowns: the 3D shape $\mathcal{S}$; the texture $\mathcal{T}$; and the set of 3D positions of the object $\{\boldsymbol{m}_f, 1 \leq f \leq F\}$ with respect to the camera. We formulate the ML solution. When the noise sequence $\{\boldsymbol{W}_f(\boldsymbol{u})\}$ is zero mean, spatially and temporally white, and Gaussian, the ML estimate minimizes the sum over all the frames of the integral over the image plane of the squared errors between the observations and the model[1],

$$C_{\mathrm{ML}}\left(\mathcal{S}, \mathcal{T}, \{\boldsymbol{m}_f\}\right) = \sum_{f=1}^{F} \int \left[\boldsymbol{I}_f(\boldsymbol{u}) - \mathcal{T}\left(\boldsymbol{s}_f(\boldsymbol{u})\right)\right]^2 \, d\boldsymbol{u}, \tag{4}$$

$$\left\{\hat{\mathcal{S}}, \widehat{\mathcal{T}}, \{\hat{\boldsymbol{m}}_f\}\right\} = \arg \min_{\mathcal{S}, \mathcal{T}, \{\boldsymbol{m}_f\}} C_{\mathrm{ML}}\left(\mathcal{S}, \mathcal{T}, \{\boldsymbol{m}_f\}\right). \tag{5}$$

In (4), we make explicit the dependence of the cost function C_{ML} on the object texture $\mathcal{T}$. Note that C_{ML} depends on the object shape $\mathcal{S}$ and the object positions $\{\boldsymbol{m}_f\}$ through the mappings $\{\boldsymbol{s}_f(\boldsymbol{u})\}$.

3 Maximum Likelihood Estimation

We address the minimization of $C_{\mathrm{ML}}\left(\mathcal{S}, \mathcal{T}, \{\boldsymbol{m}_f\}\right)$ by first solving for the texture estimate $\widehat{\mathcal{T}}$ in terms of the 3D object shape $\mathcal{S}$ and the object positions $\{\boldsymbol{m}_f\}$.

Texture estimate We rewrite the cost function C_{ML} given by (4) by changing the integration variable from the image plane coordinate $\boldsymbol{u}$ to the object surface coordinate $\boldsymbol{s}$. We obtain

$$C_{\mathrm{ML}}\left(\mathcal{S}, \mathcal{T}, \{\boldsymbol{m}_f\}\right) = \sum_{f=1}^{F} \int \left[\boldsymbol{I}_f(\boldsymbol{u}_f(\boldsymbol{s})) - \mathcal{T}(\boldsymbol{s})\right]^2 J_f(\boldsymbol{s}) \, d\boldsymbol{s}, \tag{6}$$

[1] We use a continuous spatial dependence for commodity. The variables $\boldsymbol{u}$ and $\boldsymbol{s}$ are continuous while f is discrete.

where $\boldsymbol{u}_f(\boldsymbol{s})$ is the mapping that projects the point $\boldsymbol{s}$ on the object surface onto the image plane at instant f, see figure 1. The function $J_f(\boldsymbol{s})$ is the Jacobian of the mapping $\boldsymbol{u}_f(\boldsymbol{s})$, $J_f(\boldsymbol{s}) = |\nabla \boldsymbol{u}_f(\boldsymbol{s})|$.

Expression (6) shows that the cost function C_{ML} is quadratic in each intensity value $\mathcal{T}(\boldsymbol{s})$ of the object texture. The ML estimate $\widehat{\mathcal{T}}(\boldsymbol{s})$ is

$$\widehat{\mathcal{T}}(\boldsymbol{s}) = \frac{\sum_{f=1}^{F} \boldsymbol{I}_f(\boldsymbol{u}_f(\boldsymbol{s}))J_f(\boldsymbol{s})}{\sum_{f=1}^{F} J_f(\boldsymbol{s})} \tag{7}$$

(see appendix A for the proof). Expression (7) states that the estimate of the texture of the object at the surface point $\boldsymbol{s}$ is a weighted average of the measures of the intensity level corresponding to that surface point. A given region around $\boldsymbol{s}$ on the object surface projects at frame $\boldsymbol{I}_f$ to a region around $\boldsymbol{u}_f(\boldsymbol{s})$. The size of this projected region changes with time because of the object motion. The more parallel to the image plane is the tangent to the object surface at point $\boldsymbol{s}$, the larger is the size of the projected region. Expression (7) shows that the larger the Jacobian $J_f(\boldsymbol{s})$ is, i.e., the larger the region around $\boldsymbol{s}$ is magnified at frame $\boldsymbol{I}_f$, the larger is the weight given to that frame when estimating the texture $\mathcal{T}(\boldsymbol{s})$.

Structure from motion as an approximation to ML By inserting the texture estimate $\widehat{\mathcal{T}}$ given by (7) in (6), we can express the cost function C_{ML} in terms of the mappings $\{\boldsymbol{u}_f(\boldsymbol{s})\}$. After manipulations (see appendix B) we get

$$C_{\mathrm{ML}}\left(\mathcal{S}, \{\boldsymbol{m}_f\}\right) = \sum_{f=2}^{F} \sum_{g=1}^{f-1} \int \left[\boldsymbol{I}_f(\boldsymbol{u}_f(\boldsymbol{s})) - \boldsymbol{I}_g(\boldsymbol{u}_g(\boldsymbol{s}))\right]^2 \frac{J_f(\boldsymbol{s})J_g(\boldsymbol{s})}{\sum_{h=1}^{F} J_h(\boldsymbol{s})}\, d\boldsymbol{s}. \tag{8}$$

The cost function C_{ML} in (8) is a weighted sum of the squared differences between all pairs of frames. At each surface point $\boldsymbol{s}$, the frame pair $\{\boldsymbol{I}_f, \boldsymbol{I}_g\}$ is weighted by $\frac{J_f(\boldsymbol{s})J_g(\boldsymbol{s})}{\sum_{h=1}^{F} J_h(\boldsymbol{s})}$. The larger this weight is, i.e., the larger a region around $\boldsymbol{s}$ is magnified in frames $\boldsymbol{I}_f$ and $\boldsymbol{I}_g$, the more the square difference between $\boldsymbol{I}_f$ and $\boldsymbol{I}_g$ affects C_{ML}.

Expression (8) also makes clear why the problem we are addressing is referred to as *structure from motion*: having eliminated the dependence on the texture, we are left with a cost function that depends on the *structure* (3D shape $\mathcal{S}$ and 3D motion $\{\boldsymbol{m}_f\}$) only through the *motion* induced in the image plane, i.e., through the mappings $\{\boldsymbol{u}_f(\boldsymbol{s})\}$. Recall the comment on section 2 that $\boldsymbol{u}_f(\mathcal{S}, \boldsymbol{m}_f; \boldsymbol{s})$ depends on the shape $\mathcal{S}$ and the motion $\boldsymbol{m}_f$. The usual approach to the minimization of the functional (8) is in two steps. The first step estimates the motion in the image plane $\boldsymbol{u}_f(\boldsymbol{s})$ by minimizing an approximation of (8) (in general, only two frames are taken into account). The second step estimates the shape $\mathcal{S}$ and the motion $\boldsymbol{m}_f$ from $\{\boldsymbol{m}_f\}$. Since the motion in the image plane can not be reliably computed in the entire image, these methods cannot provide a reliable dense shape estimate.

Our approach combines the good performance of the factorization method in estimating the 3D motion with the robustness of minimizing the ML energy function with respect to the object shape.

4 Rank 1 Factorization

This section summarizes the factorization method used to estimate the 3D motion. For a detailed description, see [2]. The factorization approach is robust due to the modelization of the rigidity of the moving object along time. This method is also computationally simple because it uses a fast algorithm to factorize a measurement matrix that is rank 1 in a noiseless situation.

A set of N feature points are tracked along an image sequence of F frames. Under orthography, the projection of feature n in frame f, $[u_{fn}, v_{fn}]^T$, is

$$\begin{bmatrix} u_{fn} \\ v_{fn} \end{bmatrix} = \begin{bmatrix} i_{xf} & i_{yf} & i_{zf} \\ j_{xf} & j_{yf} & j_{zf} \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} + \begin{bmatrix} t_{uf} \\ t_{vf} \end{bmatrix} \tag{9}$$

where $i_{xf}, i_{yf}, i_{zf}, j_{xf}, j_{yf}$, and j_{zf} are entries of the well known 3D rotation matrix, uniquely determined by the Euler angles θ_f, ϕ_f, and ψ_f, see [3], and t_{uf} and t_{vf} are the components of the object translation along the camera plane. We make the object coordinate system and camera coordinate system coincide in the first frame, so we have $u_{1n} = x_n$ and $v_{1n} = y_n$. Thus, the coordinates of the feature points along the camera plane $\{x_n, y_n\}$ are given by their projections in the first frame. The goal of the factorization method is to solve the overconstrained equation system (9) with respect to the following set of unknowns: the 3D positions of the object for $2 \leq f \leq F$, and the relative depths $\{z_n, 1 \leq n \leq N\}$.

By choosing the origin of the object coordinate system to coincide with the centroid of the set of feature points, we get the estimate for the translation as the centroid of the feature point projections. Replacing the translation estimates in the system of equations (9), and defining

$$\begin{bmatrix} \tilde{u}_{fn} \\ \tilde{v}_{fn} \end{bmatrix} = \begin{bmatrix} u_{fn} \\ v_{fn} \end{bmatrix} - \frac{1}{N} \sum_{m=1}^{N} \begin{bmatrix} u_{fm} \\ v_{fm} \end{bmatrix}, \qquad \boldsymbol{R} = \begin{bmatrix} \tilde{u}_{21} & \cdots & \tilde{u}_{2N} \\ \cdots & \cdots & \cdots \\ \tilde{u}_{F1} & \cdots & \tilde{u}_{FN} \\ \tilde{v}_{21} & \cdots & \tilde{v}_{2N} \\ \cdots & \cdots & \cdots \\ \tilde{v}_{F1} & \cdots & \tilde{v}_{FN} \end{bmatrix}, \tag{10}$$

$$\boldsymbol{M} = \begin{bmatrix} i_{x2} & \cdots & i_{xF} & j_{x2} & \cdots & j_{xF} \\ i_{y2} & \cdots & i_{yF} & j_{y2} & \cdots & j_{yF} \\ i_{z2} & \cdots & i_{zF} & j_{z2} & \cdots & j_{zF} \end{bmatrix}^T, \qquad \text{and} \qquad \boldsymbol{S}^T = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \\ z_1 & z_2 & \cdots & z_N \end{bmatrix}, \tag{11}$$

we rewrite (9) in matrix format as

$$\boldsymbol{R} = \boldsymbol{M}\boldsymbol{S}^T. \tag{12}$$

Matrix $\boldsymbol{R}$ is $2(F-1) \times N$ but it is rank deficient. In a noiseless situation, $\boldsymbol{R}$ is rank 3 reflecting the high redundancy in the data, due to the 3D rigidity of the object.

The factorization approach finds a suboptimal solution to the bilinear LS problem of equation (12) where the solution space is constrained by the orthonormality of the rows of the matrix $\boldsymbol{M}$ (11). This nonlinear minimization is solved in two stages. The first stage, *decomposition stage*, solves the unconstrained bilinear problem $\boldsymbol{R} = \boldsymbol{M}\boldsymbol{S}^T$. The second stage, *normalization stage*, computes a set of normalizing parameters by approximating the constraints imposed by the structure of the matrix $\boldsymbol{M}$.

Decomposition stage Define $\boldsymbol{M} = [\boldsymbol{M}_0, \boldsymbol{m}_3]$ and $\boldsymbol{S} = [\boldsymbol{S}_0, \boldsymbol{z}]$. $\boldsymbol{M}_0$ and $\boldsymbol{S}_0$ contain the first two columns of $\boldsymbol{M}$ and $\boldsymbol{S}$, respectively, $\boldsymbol{m}_3$ is the third column of $\boldsymbol{M}$, and $\boldsymbol{z}$ is the third column of $\boldsymbol{S}$. We decompose the relative depth vector $\boldsymbol{z}$ into the component that belongs to the space spanned by the columns of $\boldsymbol{S}_0$ and the component orthogonal to this space as $\boldsymbol{z} = \boldsymbol{S}_0\boldsymbol{b} + \boldsymbol{a}$, with $\boldsymbol{a}^T\boldsymbol{S}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}$. We rewrite $\boldsymbol{R}$ in (12) as $\boldsymbol{R} = \boldsymbol{M}_0\boldsymbol{S}_0^T + \boldsymbol{m}_3\boldsymbol{b}^T\boldsymbol{S}_0^T + \boldsymbol{m}_3\boldsymbol{a}^T$.

The decomposition stage is formulated as

$$\min_{\boldsymbol{M}_0, \boldsymbol{m}_3, \boldsymbol{b}, \boldsymbol{a}} \left\| \boldsymbol{R} - \boldsymbol{M}_0\boldsymbol{S}_0^T - \boldsymbol{m}_3\boldsymbol{b}^T\boldsymbol{S}_0^T - \boldsymbol{m}_3\boldsymbol{a}^T \right\|_F \tag{13}$$

where $\|.\|_F$ denotes the Frobenius norm. The solution for $\boldsymbol{M}_0$ is given by $\widehat{\boldsymbol{M}}_0 = \boldsymbol{R}\boldsymbol{S}_0 \left(\boldsymbol{S}_0^T\boldsymbol{S}_0\right)^{-1} - \boldsymbol{m}_3\boldsymbol{b}^T$. By replacing $\widehat{\boldsymbol{M}}_0$ in (13), we get

$$\min_{\boldsymbol{m}_3, \boldsymbol{a}} \left\| \widetilde{\boldsymbol{R}} - \boldsymbol{m}_3\boldsymbol{a}^T \right\|_F, \qquad \text{where} \qquad \widetilde{\boldsymbol{R}} = \boldsymbol{R}\left[\boldsymbol{I} - \boldsymbol{S}_0 \left(\boldsymbol{S}_0^T\boldsymbol{S}_0\right)^{-1} \boldsymbol{S}_0^T\right]. \tag{14}$$

We see that the decomposition stage does not determine the vector $\boldsymbol{b}$. This is because the component of $\boldsymbol{z}$ that lives in the space spanned by the columns of $\boldsymbol{S}_0$ does not affect the space spanned by the columns of the entire matrix $\boldsymbol{S}$ and the decomposition stage restricts only this last space.

The solution for $\boldsymbol{m}_3$ and $\boldsymbol{a}$ is given by the rank 1 matrix that best approximates $\widetilde{\boldsymbol{R}}$. In a noiseless situation, $\widetilde{\boldsymbol{R}}$ is rank 1, see [2] for the details. By computing the largest singular value of $\widetilde{\boldsymbol{R}}$ and the associated singular vectors, we get

$$\widetilde{\boldsymbol{R}} \simeq \boldsymbol{u}\sigma\boldsymbol{v}^T, \qquad \widehat{\boldsymbol{m}}_3 = \alpha\boldsymbol{u}, \qquad \hat{\boldsymbol{a}}^T = \frac{\sigma}{\alpha}\boldsymbol{v}^T \tag{15}$$

where α is a normalizing scalar different from 0. To compute $\boldsymbol{u}$, σ, and $\boldsymbol{v}$ we could perform an SVD, but the rank deficiency of $\widetilde{\boldsymbol{R}}$ enables the use of less expensive algorithms to compute $\boldsymbol{u}$, σ, and $\boldsymbol{v}$, as detailed in [2].

Normalization stage In this stage, we compute α and $\boldsymbol{b}$ by imposing the constraints that come from the structure of $\boldsymbol{M}$. We express $\widehat{\boldsymbol{M}}$ in terms of α and $\boldsymbol{b}$ as

$$\widehat{\boldsymbol{M}} = \left[\widehat{\boldsymbol{M}}_0\ \widehat{\boldsymbol{m}}_3\right] = \boldsymbol{N}\begin{bmatrix} \boldsymbol{I}_{2\times 2} & \underline{0}_{2\times 1} \\ -\alpha\boldsymbol{b}^T & \alpha \end{bmatrix}, \qquad \boldsymbol{N} = \left[\boldsymbol{R}\boldsymbol{S}_0 \left(\boldsymbol{S}_0^T\boldsymbol{S}_0\right)^{-1} \boldsymbol{u}\right]. \tag{16}$$

The constraints imposed by the structure of $\boldsymbol{M}$ are the unit norm of each row and the orthogonality between row j and row $j + F - 1$, where F is the number

of frames in the sequence. In terms of $\boldsymbol{N}$, α, and $\boldsymbol{b}$, the constraints are

$$\boldsymbol{n}_i^T \begin{bmatrix} \boldsymbol{I}_{2\times 2} & -\alpha\boldsymbol{b} \\ -\alpha\boldsymbol{b}^T & \alpha^2(1+\boldsymbol{b}^T\boldsymbol{b}) \end{bmatrix} \boldsymbol{n}_i = 1, \;\; \boldsymbol{n}_j^T \begin{bmatrix} \boldsymbol{I}_{2\times 2} & -\alpha\boldsymbol{b} \\ -\alpha\boldsymbol{b}^T & \alpha^2(1+\boldsymbol{b}^T\boldsymbol{b}) \end{bmatrix} \boldsymbol{n}_{j+F-1} = 0 \quad (17)$$

where $\boldsymbol{n}_i^T$ denotes the row i of the matrix $\boldsymbol{N}$. We compute α and $\boldsymbol{b}$ from the linear LS solution of the system above in a similar way to the one described in [9].

5 Minimization Procedure

After recovering the 3D motion $\boldsymbol{m}_f$ as described is section 4, we insert the 3D motion estimates into the energy function (8) and minimize with respect to the unknown shape $\mathcal{S}$.

We first make explicit the relation between the image trajectories $\boldsymbol{u}_f(\boldsymbol{s})$ and the 3D shape $\mathcal{S}$ and the 3D motion $\boldsymbol{m}_f$. Choose the coordinate $\boldsymbol{s}$ of the generic point in the object surface to coincide with the coordinates $[x, y]^T$ of the object coordinate system. Under orthography, a point with coordinate $\boldsymbol{s}$ in the object surface is projected on coordinate $\boldsymbol{u} = [x, y]^T = \boldsymbol{s}$ in the first frame, so that $\boldsymbol{u}_1(\boldsymbol{s}) = \boldsymbol{s}$ (remember that we have chosen the object coordinate system so that it coincides with the camera coordinate system in the first frame). At instant f, that point is projected to

$$\begin{aligned} \boldsymbol{u}_f(\boldsymbol{s}) = \boldsymbol{u}_f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) &= \begin{bmatrix} i_{xf} & i_{yf} & i_{zf} \\ j_{xf} & j_{yf} & j_{zf} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_{uf} \\ t_{vf} \end{bmatrix} = \begin{bmatrix} \boldsymbol{N}_f & \boldsymbol{n}_f \end{bmatrix} \begin{bmatrix} \boldsymbol{s} \\ z \end{bmatrix} + \boldsymbol{t}_f \\ &= \boldsymbol{N}_f\boldsymbol{s} + \boldsymbol{n}_f z + \boldsymbol{t}_f, \end{aligned} \quad (18)$$

where $i_{xf}, i_{yf}, i_{zf}, j_{xf}, j_{yf}$, and j_{zf} are entries of the 3D rotation matrix [3]. The 3D shape is represented by the unknown relative depth z.

Modified image sequence for known motion The 3D shape and the 3D motion are observed in a coupled way through the 2D motion on the image plane, see expression (18). When the 3D motion is known, the problem of inferring the 3D shape from the image sequence is simplified. In fact, the local *brightness change constraint* leads to a single restriction, which is insufficient to determine the two components of the local image motion (this is the so called *aperture problem*). Our method of estimating directly the 3D shape overcomes the *aperture problem* because we are left with the local depth as a single unknown, after computing the 3D motion in the first step. To better illustrate why the problem becomes much simpler when the 3D motion is known, we introduce a modified image sequence $\{\widetilde{\boldsymbol{I}}_f, 1 \leq f \leq F\}$, obtained from the original sequence $\{\boldsymbol{I}_f, 1 \leq f \leq F\}$ and the 3D motion. We show that the 2D motion of the brightness pattern on image sequence $\widetilde{\boldsymbol{I}}_f$ depends on the 3D shape in a very particular way. This motivates the algorithm we use to minimize (8).

Consider the image $\widetilde{\boldsymbol{I}}_f$ related to $\boldsymbol{I}_f$ by the following affine mapping that depends only on the 3D position at instant f,

$$\widetilde{\boldsymbol{I}}_f(\boldsymbol{s}) = \boldsymbol{I}_f(\boldsymbol{N}_f\boldsymbol{s} + \boldsymbol{t}_f). \quad (19)$$

From this definition it follows that a point $\boldsymbol{s}$, that projects to $\boldsymbol{u}_f(\boldsymbol{s})$ in image $\boldsymbol{I}_f$, is mapped to $\widetilde{\boldsymbol{u}}_f(\boldsymbol{s}) = \boldsymbol{N}_f^{-1}[\boldsymbol{u}_f(\boldsymbol{s}) - \boldsymbol{t}_f]$ in image $\widetilde{\boldsymbol{I}}_f$. Replacing $\boldsymbol{u}_f(\boldsymbol{s})$ by expression (18), we obtain for the image motion of the modified sequence $\{\widetilde{\boldsymbol{I}}_f\}$,

$$\widetilde{\boldsymbol{u}}_f(\boldsymbol{s}) = \boldsymbol{s} + \boldsymbol{N}_f^{-1}\boldsymbol{n}_f z. \tag{20}$$

Expression (20) shows that the trajectory of a point $\boldsymbol{s}$ in image sequence $\{\widetilde{\boldsymbol{I}}_f\}$ depends on the relative depth of that point in a very particular way. In fact, the trajectory has the same shape for every point. The shape of the trajectories is given by the evolution of $\boldsymbol{N}_f^{-1}\boldsymbol{n}_f$ across the frame index f. Thus, the shape of the trajectories depends uniquely on the rotational component of the 3D motion. The relative depth z affects only the magnitude of the trajectory. A point with relative depth $z = 0$ is stationary in $\{\widetilde{\boldsymbol{I}}_f\}$, since we get $\widetilde{\boldsymbol{u}}_f(\boldsymbol{s}) = \boldsymbol{s}$ from (20) for arbitrary 3D motion of the object.

Continuation method By minimizing (8) with respect to the relative depth of each point $\boldsymbol{s}$, we are in fact estimating the magnitude of the trajectory of the point to where the point $\boldsymbol{s}$ maps in image sequence $\{\widetilde{\boldsymbol{I}}_f\}$. The shape of the trajectory is known, since it depends only on the 3D motion. Our algorithm is based on this characteristic of the ML energy function. We use a continuation-type method to estimate the relative depth of each point. The algorithm refines the estimate of the relative depth as more frames are being taken into account. When only a few frames are taken into account, the magnitude of the trajectories on image sequence $\{\widetilde{\boldsymbol{I}}_f\}$ can be only roughly estimated because the length of the trajectories is short and their shape may be quite simple. When enough frames are considered, the trajectories on image sequence $\{\widetilde{\boldsymbol{I}}_f\}$ are long enough, their magnitude is unambiguous, and the relative depth estimates are accurate. Our algorithm does not compute $\{\widetilde{\boldsymbol{I}}_f\}$, it rather uses the corresponding intensity values of $\{\boldsymbol{I}_f\}$.

The advantage of the continuation-type method is that is provides a computationally simple way to estimate the relative depth because each stage of the algorithm updates the estimate by using a Gauss-Newton method, i.e., by solving a linear problem. We consider the relative depth z to be constant in a region $\mathcal{R}$. We estimate z by minimizing the energy resultant from neglecting the weighting factor $\frac{J_f(\boldsymbol{s})J_g(\boldsymbol{s})}{\sum_{h=1}^{F} J_h(\boldsymbol{s})}$ in the ML energy function (8). Thus, we get

$$\hat{z} = \arg\min_z E(z), \qquad E(z) = \sum_{f=2}^{F}\sum_{g=1}^{f-1}\int_{\mathcal{R}} e^2(z)\, d\boldsymbol{s}, \tag{21}$$

where

$$e(z) = \boldsymbol{I}_f(\boldsymbol{N}_f\boldsymbol{s} + \boldsymbol{n}_f z + \boldsymbol{t}_f) - \boldsymbol{I}_g(\boldsymbol{N}_g\boldsymbol{s} + \boldsymbol{n}_g z + \boldsymbol{t}_g). \tag{22}$$

We compute $\hat{z}$ by refining a previous estimate z_0, as

$$\hat{z} = z_0 + \widehat{\delta_z}, \qquad \widehat{\delta_z} = \arg\min_{\delta_z} E(z_0 + \delta_z). \tag{23}$$

The Gauss-Newton method neglects the second and higher order terms of the Taylor series expansion of $e(z_0 + \delta_z)$. By making this approximation, we get

$$\widehat{\delta_z} = -\frac{\sum_{f=2}^{F}\sum_{g=1}^{f-1}\int_{\mathcal{R}} e(z_0)e'(z_0)}{\sum_{f=2}^{F}\sum_{g=1}^{f-1}\int_{\mathcal{R}}\left[e'(z_0)\right]^2}, \tag{24}$$

where e' is the derivative of e with respect to z. By differentiating (22), we get

$$\begin{aligned} e'(z) = {} & \boldsymbol{I}_{f_x}(\boldsymbol{N}_f\boldsymbol{s} + \boldsymbol{n}_f z + \boldsymbol{t}_f)i_{zf} + \boldsymbol{I}_{f_y}(\boldsymbol{N}_f\boldsymbol{s} + \boldsymbol{n}_f z + \boldsymbol{t}_f)j_{zf} \\ & -\boldsymbol{I}_{g_x}(\boldsymbol{N}_g\boldsymbol{s} + \boldsymbol{n}_g z + \boldsymbol{t}_g)i_{zg} - \boldsymbol{I}_{g_y}(\boldsymbol{N}_g\boldsymbol{s} + \boldsymbol{n}_g z + \boldsymbol{t}_g)j_{zg}, \end{aligned} \tag{25}$$

where $\boldsymbol{I}_{f_x}$ and $\boldsymbol{I}_{f_y}$ denote the components of the spatial gradient of image $\boldsymbol{I}_f$.

At the beginning, we start with the initial guess $z_0 = 0$ for any region $\mathcal{R}$. We use square regions where z is estimated as being constant. The size of the regions determines the resolution of the relative depth estimate. We use large regions when processing the first frames and decrease the size of regions as the continuation method takes more frames into account.

6 Experimens

We describe two experiments that illustrate our approach. The first experiment uses a synthetic sequence for which we compare the estimates obtained with the ground truth. The second experiment uses a real video sequence.

Synthetic sequence We consider that the world is 2D and that the images are 1D orthogonal projections of the world. This scenario reflects all the basic properties and difficulties of the *structure from motion* paradigm and corresponds to the real 3D world if we consider only one epipolar plane and assume that the motion occurs on that plane. In figure 2 we show a computer generated sequence of 25 1D images. Time increases from top to bottom. The time evolution of the translational and rotational components of the motion are shown respectively in the left and middle plots of figure 3. The object shape is shown on the right plot of figure 3. The object texture is an intensity function defined over the object contour. We obtained the image sequence in figure 2 by projecting the object texture on the image plane and by adding noise.

In figure 4 we represent the modified image sequence, computed from the original sequence in figure 2, as described in section 5 for the 3D scenario, see expression (19). The motion of the brightness pattern in figure 4 is simpler than the motion in figure 2. In fact, the horizontal positions of the brightness patterns in figure 4 have a time evolution that is equal for the entire image (see, from figure 4 and the left plot of figure 3 that the shape of the trajectories of the brightness patterns is related to the rotational component of the motion). Only the amplitude of the time evolution of the horizontal positions of the brightness patterns in figure 4 is different from an object region to another object region. The amplitude for a given region is proportional to the relative depth of that region. Note that the brightness pattern is almost stationary for regions with

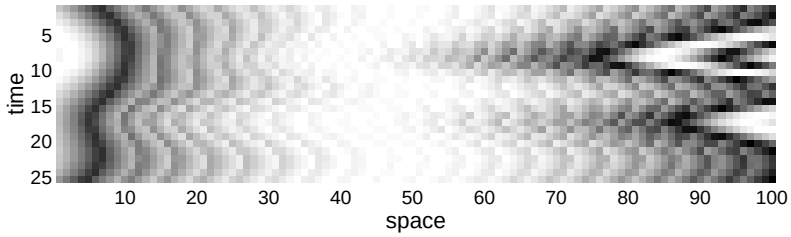

Fig. 2. Sequence of 25 1D images: each horizontal slice is one image.

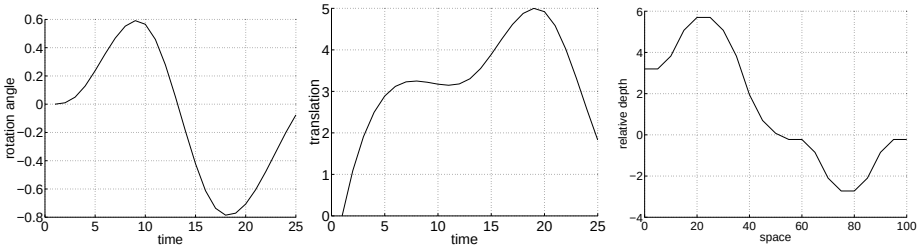

Fig. 3. True motion and true shape. Left: rotational motion; middle: translational motion; right: object shape.

relative depth close to zero (see the regions around pixels 55 and 95 on the right plot of figure 3 and on figure 4). This agrees with the discussion in section 5.

We estimated the relative depth of the object by using the continuation method introduced in section 5. The evolution of the relative depth estimate is represented in the plots of figure 5 for several time instants. The size of the estimation region $\mathcal{R}$ was 10 pixels when processing the first 5 frames, 5 pixels when processing frames 6 to 10, and 3 pixels when processing frames 11 to 25. The true depth shape is shown by the dashed line in the bottom right plot of figure 5. The top left plot was obtained with the first three frames and shows a very coarse estimate of the shape. The bottom right plot was obtained after all 25 frames of the image sequence have been processed. In this plot we made

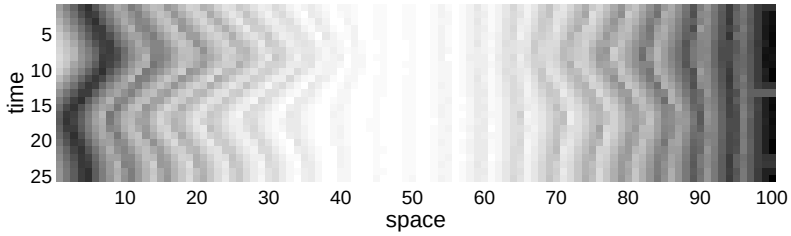

Fig. 4. Modified image sequence for known motion.

a linear interpolation between the central points of consecutive estimation regions. This plot superposes the true and the estimated depths showing a very good agreement between them. The intermediate plots show progressively better estimates of the depth shape.

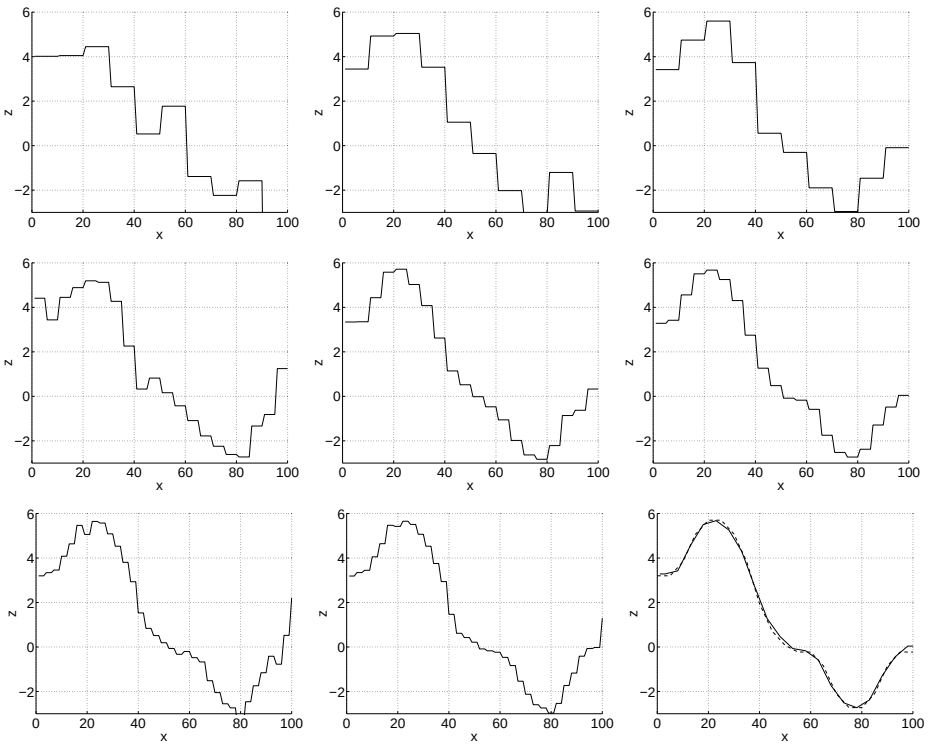

Fig. 5. Continuation method: evolution of the shape estimate. Left to right, top to bottom, after processing F frames where F is successively: $3, 4, 5, 6, 8, 10, 15, 20, 25$. The true shape is shown as the dashed line in the bottom right plot.

Real video We used a sequence of 10 frames from a real video sequence showing a toy clown. Figure 6 shows frames 1 and 5. Each frame has 384×288 pixels. Superimposed on frame 1, we marked with white squares 20 features used in the factorization method. The method used to select the features is reported elsewhere. We tracked the feature points by matching the intensity pattern of each feature along the sequence. Using the factorization approach summarized in section 4 we recovered the 3D motion from the feature trajectories.

We estimated the relative depth of the 3D object by using the continuation method described in section 5. The evolution of the estimate of the relative depth is illustrated by figure 7. The grey level images in this figure code the relative depth estimates. The brighter a pixel is, the closer to the camera it is in the first

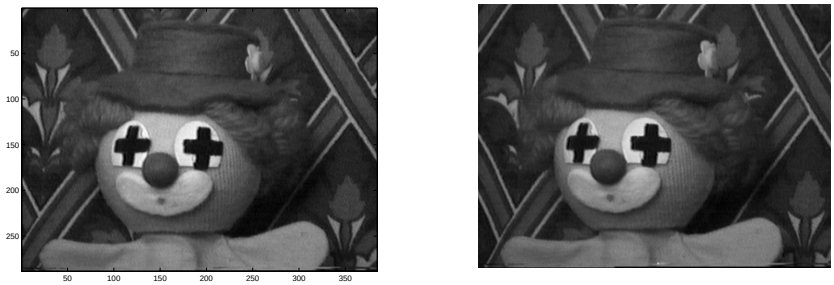

Fig. 6. Clown sequence: frames 1 and 5.

frame. The size of the estimation region $\mathcal{R}$ was 30×30 pixels when processing the first 3 frames, 20×20 pixels when processing frames 4 to 6, and 10×10 pixels when processing frames 7 to 10. The left image was obtained with the first three frames and shows a very coarse estimate of the shape. The right image was obtained after all 10 frames of the image sequence have been processed.

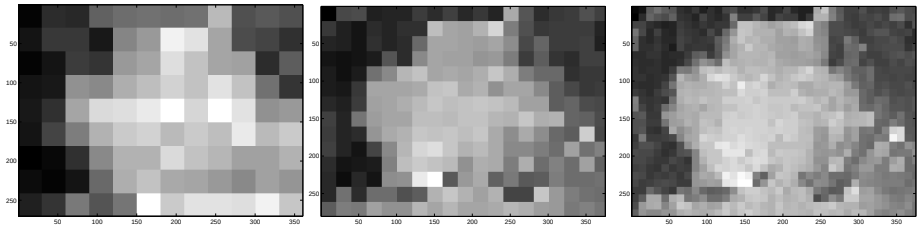

Fig. 7. Relative depth estimate after processing 3, 6, and 10 frames.

7 Conclusion

Final remarks We presented a new approach to the recovery of 3D rigid structure from a 2D video sequence. The problem is formulated as the ML estimation of all the unknowns directly from the intensity values of the set of images in the sequence. We estimate the 3D motion by using a factorization method. We develop a continuation-type algorithm to minimize the ML energy function with respect to the object shape. The experimental results obtained so far are promising and illustrate the good performance of the algorithm.

Future extensions A number of possible extensions of this work are foreseen. First, our methodology can be modified to achieve the estimation of the absolute depth by using the perspective projection model. Other possible extensions include the investigation of different shape models. In this paper we use a dense depth map. Parametric models enable more compact and robust shape representation. The formulation of the problem directly from the image intensity values

provides a robust way of dealing with such issues as segmentation (for piecewise models) and model complexity, by using classic tools such as the Bayesian inference or information-theoretic criteria.

A Texture Estimation

To prove that the ML estimate $\widehat{\mathcal{T}}(\boldsymbol{s})$ of the texture $\mathcal{T}(\boldsymbol{s})$ is given by expression (7), we show that it leads to the minimum of the cost function C_{ML}, given by expression (6), over all texture functions $\mathcal{T}(\boldsymbol{s})$. Consider the candidate $\mathcal{T}(\boldsymbol{s}) = \widehat{\mathcal{T}}(\boldsymbol{s}) + \mathcal{U}(\boldsymbol{s})$. The functional C_{ML} for texture function $\mathcal{T}(\boldsymbol{s})$ is

$$\begin{aligned} C_{\mathrm{ML}}(\mathcal{T}) &= \sum_{f=1}^{F} \int \Big[\boldsymbol{I}_f(\boldsymbol{u}_f(\boldsymbol{s})) - \widehat{\mathcal{T}}(\boldsymbol{s}) - \mathcal{U}(\boldsymbol{s})\Big]^2 J_f(\boldsymbol{s})\, d\boldsymbol{s} \\ &= \sum_{f=1}^{F} \int \Big[\boldsymbol{I}_f(\boldsymbol{u}_f(\boldsymbol{s})) - \widehat{\mathcal{T}}(\boldsymbol{s})\Big]^2 J_f(\boldsymbol{s})\, d\boldsymbol{s} + \sum_{f=1}^{F} \int \mathcal{U}^2(\boldsymbol{s}) J_f(\boldsymbol{s})\, d\boldsymbol{s} \\ &\quad -2 \sum_{f=1}^{F} \int \Big[\boldsymbol{I}_f(\boldsymbol{u}_f(\boldsymbol{s})) - \widehat{\mathcal{T}}(\boldsymbol{s})\Big] \mathcal{U}(\boldsymbol{s}) J_f(\boldsymbol{s})\, d\boldsymbol{s}. \end{aligned} \tag{26}$$

The first term of the expression above is $C_{\mathrm{ML}}(\widehat{\mathcal{T}})$. The third term is 0, as comes immediately by replacing $\widehat{\mathcal{T}}(\boldsymbol{s})$ by expression (7). We have

$$C_{\mathrm{ML}}(\mathcal{T}) = C_{\mathrm{ML}}(\widehat{\mathcal{T}}) + \sum_{f=1}^{F} \int \mathcal{U}^2(\boldsymbol{s}) J_f(\boldsymbol{s})\, d\boldsymbol{s} \geq C_{\mathrm{ML}}(\widehat{\mathcal{T}}), \tag{27}$$

which concludes the proof. The inequality comes from the fact that we can always choose the texture coordinates $\boldsymbol{s}$ in such a way that the mappings $\boldsymbol{u}_f(\boldsymbol{s})$ are such that the determinants $J_f(\boldsymbol{s}) = |\nabla \boldsymbol{u}_f(\boldsymbol{s})|$ are positive. For example, make the texture coordinate $\boldsymbol{s}$ equal to the image plane coordinate $\boldsymbol{u}$ in the first frame $\boldsymbol{I}_1$. The mapping $\boldsymbol{u}_1(\boldsymbol{s})$ is the identity mapping $\boldsymbol{u}_1(\boldsymbol{s}) = \boldsymbol{s}$ and we have a positive Jacobian $J_1(\boldsymbol{s}) = 1$. Now, draw an oriented closed contour on the surface $\mathcal{S}$, in the neighborhood of $\boldsymbol{s}$, and containing $\boldsymbol{s}$ in its interior. This contour, which we call $\mathcal{C}_{\boldsymbol{S}}$ is projected in image $\boldsymbol{I}_1$ in an oriented closed planar contour $\mathcal{C}_{\boldsymbol{u}1}$. It is geometrically evident that the same contour $\mathcal{C}_{\boldsymbol{S}}$ projects in image $\boldsymbol{I}_f$, in a contour $\mathcal{C}_{\boldsymbol{u}f}$ that has, in general, different shape but the same orientation that the contour $\mathcal{C}_{\boldsymbol{u}1}$ (remember that we are assuming the object does not occlude itself). For this reason, the Jacobian $J_f(\boldsymbol{s})$ of the function that maps from $\boldsymbol{s}$ to $\boldsymbol{u}_f(\boldsymbol{s})$ for $2 \leq f \leq F$, has the same signal as the Jacobian $J_1(\boldsymbol{s})$ of the function that maps from $\boldsymbol{s}$ to $\boldsymbol{u}_1(\boldsymbol{s})$, so we get $J_f(\boldsymbol{s}) > 0$ for $1 \leq f \leq F$.

B C_{ML} in terms of $\{\boldsymbol{u}_f(\boldsymbol{s})\}$

We show that the ML-based cost function is expressed in terms of the motion in the image plane as in expression (8). Replace the texture estimate $\widehat{\mathcal{T}}(\boldsymbol{s})$, given

by (7), into the ML-based cost function, given by (6). After simple algebraic manipulations, we get

$$C_{\mathrm{ML}} = \int \sum_{f=1}^{F} \left[\frac{\sum_{g=1}^{F} \left[\boldsymbol{I}_f(\boldsymbol{u}_f(\boldsymbol{s})) - \boldsymbol{I}_g(\boldsymbol{u}_g(\boldsymbol{s})) \right] J_g(\boldsymbol{s})}{\sum_{h=1}^{F} J_h(\boldsymbol{s})} \right]^2 J_f(\boldsymbol{s})\, d\boldsymbol{s}. \tag{28}$$

Expressing the square above in terms of a sum of products and carrying out the products, after algebraic manipulations, we get

$$C_{\mathrm{ML}} = \int \frac{\sum_{f=1}^{F} \sum_{g=1}^{F} \sum_{h=1}^{F} \left[\boldsymbol{I}_f^2(\boldsymbol{s}) - \boldsymbol{I}_f(\boldsymbol{s}) \boldsymbol{I}_g(\boldsymbol{s}) \right] J_f(\boldsymbol{s}) J_g(\boldsymbol{s}) J_h(\boldsymbol{s})}{\left[\sum_{h=1}^{F} J_h(\boldsymbol{s}) \right]^2} \, d\boldsymbol{s}. \tag{29}$$

Now we divide by $\sum_{h=1}^{F} J_h(\boldsymbol{s})$ both the numerator and the denominator of the integrand function. By using the equality

$$\sum_{f=1}^{F} \sum_{g=1}^{F} \left[\boldsymbol{I}_f^2(\boldsymbol{s}) - \boldsymbol{I}_f(\boldsymbol{s}) \boldsymbol{I}_g(\boldsymbol{s}) \right] J_f(\boldsymbol{s}) J_g(\boldsymbol{s}) = \sum_{f=2}^{F} \sum_{g=1}^{f-1} \left[\boldsymbol{I}_f(\boldsymbol{s}) - \boldsymbol{I}_g(\boldsymbol{s}) \right]^2 J_f(\boldsymbol{s}) J_g(\boldsymbol{s}), \tag{30}$$

we get

$$C_{\mathrm{ML}} = \int \frac{\sum_{f=2}^{F} \sum_{g=1}^{f-1} \left[\boldsymbol{I}_f(\boldsymbol{s}) - \boldsymbol{I}_g(\boldsymbol{s}) \right]^2 J_f(\boldsymbol{s}) J_g(\boldsymbol{s})}{\sum_{h=1}^{F} J_h(\boldsymbol{s})} \, d\boldsymbol{s}, \tag{31}$$

and conclude the derivation. Note that by interchanging the integral and the sum in (31), we get the ML-based cost function C_{ML} as in expression (8).

References

[1] P. M. Q. Aguiar and J. M. F. Moura. Video representation via 3D shaped mosaics. In *IEEE International Conference on Image Processing*, Chicago, USA, 1998.

[2] P. M. Q. Aguiar and J. M. F. Moura. Factorization as a rank 1 problem. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, June 1999.

[3] N. Ayache. *Artificial Vision for Mobile Robots*. The MIT Press, MA, USA, 1991.

[4] T. Broida and R. Chellappa. Estimating the kinematics and structure of a rigid object from a sequence of monocular images. *IEEE Trans. on PAMI*, 13(6), 1991.

[5] J. Heel. Direct estimation of structure and motion from multiple frames. A. I. Memo 1190, MIT AI Laboratory, MA, USA, 1990.

[6] B. Horn and E. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1), 1988.

[7] D. C. Knill, D. Kersten, and A. Yuille. A Bayesian formulation of visual perception. In *Perception as Bayesian Inference*. Cambridge University Press, 1996.

[8] S. Soatto, P. Perona, R. Frezza, and G. Picci. Recursive motion and structure estimations with complete error characterization. In *IEEE CVPR*, June 1993.

[9] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. Journal of Computer Vision*, 9(2), 1992.

[10] R. Tsai and T. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. PAMI*, 6(1), 1984.