IEEE Signal Processing Society International Conference on Image Processing
October 1999, Kobe, Japan

A Fast Algorithm for Rigid Structure from Image Sequences

Pedro M. Q. Aguiar [*] José M. F. Moura

Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh PA, USA
{aguiar,moura}@ece.cmu.edu

Abstract

The factorization method [1] is a feature-based approach to recover 3D rigid structure from motion. In [2], we extended their framework to recover a parametric description of the 3D shape. In [1, 2], the 3D shape and 3D motion are computed by using an SVD to approximate a matrix that is rank 3 *in a noiseless situation. In this paper, we develop a new algorithm that has two relevant advantages over the algorithms of [1, 2]. First, instead of imposing a common origin for the parametric representation of the 3D surface patches, as in [2], we allow the specification of different origins for different patches. This improves the numerical stability of the image motion estimation algorithm and the accuracy of the 3D structure recovery algorithm. Second, we show how to compute the 3D shape and 3D motion by a simple factorization of a modified matrix that is rank 1 in a noiseless situation, instead of a rank 3 matrix as in [1, 2]. This allows the use of very fast algorithms even when using a large number of features (or regions) and large number of frames.*

1 Introduction

The factorization method [1] is an elegant method to recover 3D rigid structure from an image sequence. In [1], the 3D positions of the feature points are expressed in terms of cartesian coordinates in a world-centered coordinate system, and the images are modeled as orthographic projections. The 2D projection of each feature point is tracked along the image sequence. The 3D shape and motion are then estimated by factorizing a measurement matrix whose entries are the set of trajectories of the feature point projections. The factorization of the measurement matrix, which is rank 3 in a noiseless situation, is computed by using the Singular Value Decomposition (SVD). The factorization method was extended to the scaled-orthography and the paraperspective projections in [3].

When the goal is the recovery of a dense representation of the 3D shape, the factorization approach of [1, 3] may not solve the problem satisfactorily because of two drawbacks. First, being feature-based, it would be necessary to track a huge number of features to obtain a dense description of the 3D shape. This is usually impossible because only distinguished points, as brightness corners, can by accurately tracked. Second, even if it is possible to track a large number of features, the computational cost of the SVD involved in the factorization of the measurement matrix would be very high. These drawbacks motivated the extension of the factorization approach to recover a parametric description of the 3D shape, as we did in [2]. Instead of tracking pointwise features, we track regions for which the motion induced on the image plane is described by a single set of parameters.

In this paper, we reformulate the problem. The reformulation leads to a new algorithm that has two relevant advantages over the algorithms of [1, 2]. First, instead of imposing a common origin for the parametric representation of the 3D surface patches, as in [2], we allow the specification of different origins for different patches. This improves the numerical stability of the image motion estimation algorithm and the accuracy of the 3D structure recovery algorithm, as will become clear later. Second, we show how to compute the 3D shape and 3D motion by a simple factorization of a modified matrix that is rank 1 in a noiseless situation, instead of a rank 3 matrix as in [1, 2]. This simplifies the decomposition and normalization stages involved in the factorization approach. We avoid the computation of the SVD by using a fast iterative method to compute the rank 1 matrix that best matches the data. Reference [4] compares the computational cost of the rank 1 factorization with the computational cost of the original rank 3 factorization method [1] for the feature-based case.

[*] The first author is also affiliated with ISR-IST, Lisboa, Portugal. The first author was partially supported by INVOTAN.

Paper Overview Section 2 describes the scenario. In section 3, we characterize the motion induced in the image plane. Sections 4 and 5 deal with the recovery of the 3D struacture from the image motion. In section 6, we detail a real video experiment. Section 7 concludes the paper.

2 Scenario

We consider a rigid body moving in front of the camera. We attach coordinate systems to the object and to the camera. The object coordinate system (o.c.s.) has axes labeled by x, y, and z. The camera coordinate system (c.c.s.) has axes labeled by u, v, and w. We consider that the o.c.s. coincides with the c.c.s. on the first frame. The image plane is defined by the axes u and v. The images are modeled as orthographic projections of the object texture. Our algorithm is easily extended to the scaled-orthography and the paraperspective projections by proceeding as [3] does for the original factorization method.

3D Shape The shape of the rigid object is a parametric description of the object surface. Although our approach is general enough to cope with general parameterizations, we consider in this paper objects whose shape is given by a piecewise planar surface with K patches. The shape parameter vector is $\boldsymbol{a} = \left\{a_{00}^k, a_{10}^k, a_{01}^k, 1 \leq k \leq K\right\}$ where

$$z = a_{00}^k + a_{10}^k(x - x_0^k) + a_{01}^k(y - y_0^k) \tag{1}$$

describes the shape of the patch k in the o.c.s.. In what respects to the representation of the planar patches, the parameters x_0^k and y_0^k can have any value, for example they can be made zero as we did in [2]. In this paper, we allow the specification of general parameters x_0^k, y_0^k. The relevance of this generalization is obvious: the shape of a small patch k with support region $\{(x, y)\}$ located far from the the point (x_0^k, y_0^k) has an high sensibility with respect to the shape parameters a_{10}^k and a_{01}^k. To minimize this sensibility, we choose for (x_0^k, y_0^k) the centroid of the support region of patch k. With this choice, we improve the numerical stability of the image motion estimation algorithm and the accuracy of the 3D structure recovery algorithm.

To simplify the notation, we define the vectors $\boldsymbol{a}_k = [a_{10}^k, a_{01}^k]^T$, $\boldsymbol{s} = [x, y]^T$, and $\boldsymbol{s}_{k0} = [x_0^k, y_0^k]^T$, and rewrite the shape of the patch k as

$$z = a_{00}^k + \boldsymbol{a}_k^T(\boldsymbol{s} - \boldsymbol{s}_{k0}^T). \tag{2}$$

3D Motion We define the 3D motion of the object by specifying the position of the o.c.s. relative to the c.c.s. in terms of $\left(t_{uf}, t_{vf}, t_{wf}, \boldsymbol{\Theta}_f\right)$ where $\left(t_{uf}, t_{vf}, t_{wf}\right)$ are the coordinates of the origin of the o.c.s. with respect to the c.c.s. (3D translation), and $\boldsymbol{\Theta}_f$ is the rotation matrix that determine the orientation of the o.c.s. relative to the c.c.s. (3D rotation).

A point with coordinates $[x, y, z]^T$ in the o.c.s. has the following coordinates in the c.c.s., at frame f,

$$\begin{bmatrix} u_f \\ v_f \\ w_f \end{bmatrix} = \begin{bmatrix} i_{xf} & i_{yf} & i_{zf} \\ j_{xf} & j_{yf} & j_{zf} \\ k_{xf} & k_{yf} & k_{zf} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_{uf} \\ t_{vf} \\ t_{wf} \end{bmatrix}, \tag{3}$$

where the matrix above is the 3D rotation matrix $\boldsymbol{\Theta}_f$.

3 Image Motion

In this section we show that the motion induced in the image plane by the body-camera 3D motion is affine with different parameterizations for regions corresponding to different patches. We relate the parameters of the affine motion model to the 3D shape and 3D motion parameters.

Under orthography, the point with coordinates (x, y, z) in the o.c.s. projects in frame f to the image point (u_f, v_f) given by

$$\begin{bmatrix} u_f \\ v_f \end{bmatrix} = \boldsymbol{M}_f \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \boldsymbol{t}_f, \tag{4}$$

where $\boldsymbol{M}_f$ collects the first and second rows of the 3D rotation matrix introduced in expression (3) and $\boldsymbol{t}_f = [t_{uf}, t_{vf}]^T$.

Consider a generic point in the object surface with coordinates $\boldsymbol{s} = [x, y]^T$ and z given by expression (2). We denote by $\boldsymbol{u}_f(\boldsymbol{s}) = [u_f(\boldsymbol{s}), v_f(\boldsymbol{s})]^T$ the trajectory of the projection of the point $\boldsymbol{s}$ in the image plane. Since we have chosen the coordinate systems to coincide on the first frame, we have $\boldsymbol{u}_1(\boldsymbol{s}) = \boldsymbol{s}$. At frame f, the point $\boldsymbol{s}$ projects according to expression (4), to the image point

$$\boldsymbol{u}_f(\boldsymbol{s}) = \boldsymbol{N}_f\boldsymbol{s} + \boldsymbol{n}_f z + \boldsymbol{t}_f, \tag{5}$$

where we have decomposed the matrix $\boldsymbol{M}_f$ as $\boldsymbol{M}_f = [\boldsymbol{N}_f, \boldsymbol{n}_f]$ where $\boldsymbol{N}_f$ collects the first and second columns of $\boldsymbol{M}_f$ and $\boldsymbol{n}_f$ is the third column of $\boldsymbol{M}_f$.

Affine Motion Model By inserting expression (2) into expression (5), we express the image displacement between frame 1 and frame f in terms of the 3D shape and 3D motion parameters, for the points $\boldsymbol{s}$ that fall into patch k of the object surface. After simple manipulations, we obtain

$$\boldsymbol{u}_f^k(\boldsymbol{s}) = (\boldsymbol{N}_f + \boldsymbol{n}_f\boldsymbol{a}_k^T)(\boldsymbol{s} - \boldsymbol{s}_0^k) + \boldsymbol{N}_f\boldsymbol{s}_0^k + \boldsymbol{n}_f a_{00}^k + \boldsymbol{t}_f. \tag{6}$$

Denoting the matrix that multiplies $(\boldsymbol{s}-\boldsymbol{s}_0^k)$ and the vector corresponding to the term independent of $\boldsymbol{s}$ by

$$\left\{\begin{array}{l}\boldsymbol{D}_f^k=\boldsymbol{N}_f+\boldsymbol{n}_f\boldsymbol{a}_k^T\\ \boldsymbol{d}_f^k=\boldsymbol{N}_f\boldsymbol{s}_0^k+\boldsymbol{n}_f a_{00}^k+\boldsymbol{t}_f\end{array}\right., \tag{7}$$

we rewrite expression (6) as

$$\boldsymbol{u}_f^k(\boldsymbol{s})=\boldsymbol{D}_f^K(\boldsymbol{s}-\boldsymbol{s}_0^k)+\boldsymbol{d}_f^k. \tag{8}$$

Expression (8) shows that the image coordinates at frame f, $\boldsymbol{u}_f$, of the the points belonging to the object surface are affine mappings of their image coordinates frame 1, $\boldsymbol{u}_1=\boldsymbol{s}$. Expression (7) relates the coefficients of the affine motion models for each patch k to the 3D motion parameters and the 3D shape parameters corresponding to patch k.

Image Motion Estimation Except for particular 3D motions, the image motion corresponding to different surface patches is described by different affine parameterizations. The problem of estimating the support regions of the surface patches leads to the segmentation of the image motion field. The segmentation according to image motion has been widely addressed in the past, see for example [5, 6]. We use the simple method of sliding a rectangular window across the image and detect abrupt changes in the affine motion parameters.

Another possible way to use our structure from motion approach is to select *a priori* the support regions of the surface patches. In fact, our framework is general enough to accommodate the feature tracking approach because it corresponds to selecting *a priori* a set of small (pointwise) support regions with shape described by z = constant in each region. In reference [4] we exploit the feature-based approach.

4 Rigid Structure from Motion

The problem of inferring 3D rigid structure from the image motion is formulated as estimating the 3D motion parameters $\{\boldsymbol{N}_f,\boldsymbol{n}_f,\boldsymbol{t}_f, 2\leq f\leq F\}$ and the 3D shape parameters $\{a_{00}^k,\boldsymbol{a}_k, 1\leq k\leq K\}$ from the image motion parameters $\{\boldsymbol{D}_f^k,\boldsymbol{d}_f^k, 2\leq f\leq F, 1\leq k\leq K\}$ by inverting the overconstrained set of equations of expression (7).

We start by estimating the translation. By choosing the object coordinate system in such a way that $\sum_k a_{00}^k=0$ and the image origin in such a way that $\sum_k \boldsymbol{s}_0^k=[0,0]^T$, we obtain the Least Squares (LS) estimate for the translation vector $\boldsymbol{t}_f$ as the mean of the vectors $\{\boldsymbol{d}_f^k, 1\leq k\leq K\}$,

$$\hat{\boldsymbol{t}}_f=\frac{1}{K}\sum_{k=1}^K\boldsymbol{d}_f^k. \tag{9}$$

To eliminate the dependence of the image motion parameters on the translation, we replace the translation estimates into expression (7) and define a new set of parameters $\{\tilde{\boldsymbol{d}}_f^k\}$ related to $\{\boldsymbol{d}_f^k\}$ by

$$\tilde{\boldsymbol{d}}_f^k=\boldsymbol{d}_f^k-\frac{1}{K}\sum_{l=1}^K\boldsymbol{d}_f^l. \tag{10}$$

Defining the matrices $\boldsymbol{R}_f^k$ and $\boldsymbol{S}_k^T$ as

$$\boldsymbol{R}_f^k=\left[\begin{array}{cc}\boldsymbol{D}_f^k & \tilde{\boldsymbol{d}}_f^k\end{array}\right]\quad\text{and}\quad \boldsymbol{S}_k^T=\left[\begin{array}{cc}I_{2\times 2} & \boldsymbol{s}_0^k\\ \boldsymbol{a}_k^T & a_{00}^k\end{array}\right], \tag{11}$$

we rewrite the equation system (7) in matrix format as

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

Expression (12) relates the image motion parameters at frame f and patch k to the 3D rotation at frame f and the 3D shape parameters for the patch k.

To make explicit the entire set of equations that arise from considering every patch $1\leq k\leq K$ and every frame $2\leq f\leq F$, we define the $2(F-1)\times 3K$ matrix $\boldsymbol{R}$ of image motion parameters, the $2(F-1)\times 3$ matrix $\boldsymbol{M}$ of 3D rotation parameters, and the $3K\times 3$ matrix $\boldsymbol{S}$ of 3D shape parameters as

$$\boldsymbol{R}=\left[\begin{array}{ccc}\boldsymbol{R}_2^1 & \cdots & \boldsymbol{R}_2^K\\ \vdots & \ddots & \vdots\\ \boldsymbol{R}_F^1 & \cdots & \boldsymbol{R}_F^K\end{array}\right],\ \boldsymbol{M}=\left[\begin{array}{c}\boldsymbol{M}_2\\ \vdots\\ \boldsymbol{M}_F\end{array}\right],\ \boldsymbol{S}=\left[\begin{array}{c}\boldsymbol{S}_1\\ \vdots\\ \boldsymbol{S}_K\end{array}\right], \tag{13}$$

and we write the relation between the image motion parameters and the 3D structure parameters as

$$\boldsymbol{R}=\boldsymbol{M}\boldsymbol{S}^T. \tag{14}$$

The matrix $\boldsymbol{R}$ of image motion parameters is highly rank deficient. In a noiseless situation, $\boldsymbol{R}$ is rank 3 reflecting the high redundancy in the data, due to the rigidity of the object.

5 Rank 1 Factorization

Estimating the 3D shape and 3D rotation parameters given the observation matrix $\boldsymbol{R}$ is a nonlinear LS problem. The factorization approach [1, 2] finds a suboptimal solution to this problem in two stages. The first stage, *decomposition stage*, solves $\boldsymbol{R}=\boldsymbol{M}\boldsymbol{S}^T$ in the LS sense by computing the SVD of the matrix $\boldsymbol{R}$ and selecting the 3 largest singular values. From $\boldsymbol{R}\simeq\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, the solution is $\boldsymbol{M}=\boldsymbol{U}\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{A}$ and $\boldsymbol{S}^T=\boldsymbol{A}^{-1}\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{V}^T$ where $\boldsymbol{A}$ is a non-singular 3×3 matrix. The second stage, *normalization stage*, computes $\boldsymbol{A}$ by approximating the constraints imposed by the structure of the matrices $\boldsymbol{M}$ and $\boldsymbol{S}$.

The formulation we adopt in this paper takes advantage of the fact that the first two rows of $\boldsymbol{S}^T$ are known. The problem is then reduced to, given $\boldsymbol{R}$, compute $\boldsymbol{M}$ and $\{a_{mn}^k\}$. We also perform in sequence the decomposition and normalization stages. These stages are as follows. For more details see [4].

Decomposition Because the first two rows of $\boldsymbol{S}^T$ are known (see expressions (11), and (13)), we show that the unconstrained bilinear problem $\boldsymbol{R} = \boldsymbol{M}\boldsymbol{S}^T$ is solved by the factorization of a rank 1 matrix, rather than a rank 3 matrix like in [1, 2]. Define $\boldsymbol{M} = [\boldsymbol{M}_0, \boldsymbol{m}_3]$ and $\boldsymbol{S} = [\boldsymbol{S}_0, \boldsymbol{a}]$. Matrices $\boldsymbol{M}_0$ 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 $\boldsymbol{a}$ is the third column of $\boldsymbol{S}$. We decompose the shape parameter vector $\boldsymbol{a}$ into the component that belongs to the space spanned by the columns of $\boldsymbol{S}_0$ and the component orthogonal to this space as $\boldsymbol{a} = \boldsymbol{S}_0\boldsymbol{b} + \boldsymbol{a}_1$, with $\boldsymbol{a}_1^T\boldsymbol{S}_0 = [0, 0]$. Using these definitions, we rewrite $\boldsymbol{R}$ as

$$\boldsymbol{R} = \boldsymbol{M}_0\boldsymbol{S}_0^T + \boldsymbol{m}_3\boldsymbol{b}^T\boldsymbol{S}_0^T + \boldsymbol{m}_3\boldsymbol{a}_1^T. \qquad (15)$$

The decomposition stage is formulated as

$$\min_{\boldsymbol{M}_0,\boldsymbol{m}_3,\boldsymbol{b},\boldsymbol{a}_1} \left\| \boldsymbol{R} - \boldsymbol{M}_0\boldsymbol{S}_0^T - \boldsymbol{m}_3\boldsymbol{b}^T\boldsymbol{S}_0^T - \boldsymbol{m}_3\boldsymbol{a}_1^T \right\|_F, \qquad (16)$$

where $\|.\|_F$ denotes the Frobenius norm. By solving the linear LS for $\boldsymbol{M}_0$ in terms of the other variables, we get

$$\widehat{\boldsymbol{M}}_0 = \boldsymbol{R}\boldsymbol{S}_0\left(\boldsymbol{S}_0^T\boldsymbol{S}_0\right)^{-1} - \boldsymbol{m}_3\boldsymbol{b}^T, \qquad (17)$$

where we used the orthogonality between $\boldsymbol{a}_1$ and $\boldsymbol{S}_0$. By replacing $\widehat{\boldsymbol{M}}_0$ in (16), we get

$$\min_{\boldsymbol{m}_3,\boldsymbol{a}_1} \left\| \widetilde{\boldsymbol{R}} - \boldsymbol{m}_3\boldsymbol{a}_1^T \right\|_F, \qquad (18)$$

$$\text{where} \qquad \widetilde{\boldsymbol{R}} = \boldsymbol{R}\left[\boldsymbol{I} - \boldsymbol{S}_0\left(\boldsymbol{S}_0^T\boldsymbol{S}_0\right)^{-1}\boldsymbol{S}_0^T\right]. \qquad (19)$$

We see that the decomposition stage does not determine the vector $\boldsymbol{b}$. This is because the component of $\boldsymbol{a}$ 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 $\boldsymbol{a}_1$ is given by the rank 1 matrix that best approximates $\widetilde{\boldsymbol{R}}$. In a noiseless situation, $\widetilde{\boldsymbol{R}}$ is rank 1 (we get $\widetilde{\boldsymbol{R}} = \boldsymbol{m}_3\boldsymbol{a}_1^T$ by replacing (15) in (19)). By computing the largest singular value of $\widetilde{\boldsymbol{R}}$ and the associated singular vectors, we get

$$\widetilde{\boldsymbol{R}} \simeq \boldsymbol{u}\sigma\boldsymbol{v}^T, \qquad \widehat{\boldsymbol{m}}_3 = \alpha\boldsymbol{u}, \qquad \hat{\boldsymbol{a}}_1^T = \frac{\sigma}{\alpha}\boldsymbol{v}^T, \qquad (20)$$

where α is a normalizing scalar different from 0. To compute $\boldsymbol{u}$, σ, and $\boldsymbol{v}$ we use a fast algorithm outlined in [4]. This makes our decomposition stage simpler than the one in [1, 2]. In fact, $\widetilde{\boldsymbol{R}}$ in (19) is $\boldsymbol{R}$ multiplied by the orthogonal projector onto the orthogonal complement of the space spanned by the columns of $\boldsymbol{S}_0$. This projection reduces the rank of the problem from 3 (matrix $\boldsymbol{R}$) to 1 (matrix $\widetilde{\boldsymbol{R}}$).

Normalization We compute α and $\boldsymbol{b}$ by imposing the constraints that come from the structure of $\boldsymbol{M}$. By replacing $\widehat{\boldsymbol{m}}_3$ in (17), we get

$$\widehat{\boldsymbol{M}} = \begin{bmatrix} \widehat{\boldsymbol{M}}_0 & \widehat{\boldsymbol{m}}_3 \end{bmatrix} = \boldsymbol{N}\begin{bmatrix} \boldsymbol{I}_{2\times 2} & 0_{2\times 1} \\ -\alpha\boldsymbol{b}^T & \alpha \end{bmatrix}, \qquad (21)$$

$$\text{where} \qquad \boldsymbol{N} = \begin{bmatrix} \boldsymbol{R}\boldsymbol{S}_0\left(\boldsymbol{S}_0^T\boldsymbol{S}_0\right)^{-1} & \boldsymbol{u} \end{bmatrix}. \qquad (22)$$

The constrains imposed by the structure of $\boldsymbol{M}$ are the unit norm of each row and the orthogonality between row $2j$ and row $2j-1$. In terms of $\boldsymbol{N}$, α, and $\boldsymbol{b}$, the constraints are

$$\boldsymbol{n}_i^T\begin{bmatrix} \boldsymbol{I}_{2\times 2} & -\alpha\boldsymbol{b} \\ -\alpha\boldsymbol{b}^T & \alpha^2(1+\boldsymbol{b}^T\boldsymbol{b}) \end{bmatrix}\boldsymbol{n}_i = 1 \qquad \text{and} \qquad (23)$$

$$\boldsymbol{n}_{2j}^T\begin{bmatrix} \boldsymbol{I}_{2\times 2} & -\alpha\boldsymbol{b} \\ -\alpha\boldsymbol{b}^T & \alpha^2(1+\boldsymbol{b}^T\boldsymbol{b}) \end{bmatrix}\boldsymbol{n}_{2j-1} = 0, \qquad (24)$$

where $\boldsymbol{n}_i^T$ denotes the row i of 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 [1]. The normalization stage is also simpler than the one in [1] because the number of unknowns is 3 (α and $\boldsymbol{b} = [b_1, b_2]^T$) as opposed to the 9 entries of a generic 3×3 normalization matrix.

6 Experiment

We used a hand hold taped video sequence of 30 frames showing a box over a carpet. Figure 1 shows frames 1 and 10 of the video sequence. The 3D shape of the scene is well described in terms of four planar patches. One corresponds to the floor, and the other three correspond to the three visible faces of the box. The camera motion was approximately a rotation around the box.

We processed the box video sequence by using the method described above. We start by estimating the affine motion parameters. The plots in figure 2 represent the time evolution of the affine motion parameters. The 6 affine motion parameters are the entries of the 2×2 matrix $\boldsymbol{D}_f^k$ and the 2×1 vector $\boldsymbol{d}_f^k$, see expression (8). The top four plots of figure 2 represent the entries of $\boldsymbol{D}_f^k$ as a function of f for each of the four

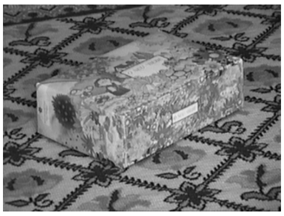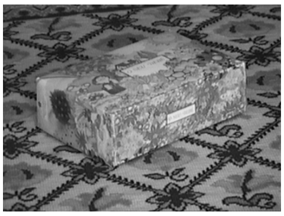

Figure 1: Frames 1 and 10 of the box video sequence.

planar patches. The bottom two plots represent $\boldsymbol{d}_f^k$. We used four different line types to identify each of the planar patches. The solid line corresponds to patch 1 (the left side vertical face of the box in the frames of figure 1). The dotted line corresponds to patch 2 (the right side vertical face of the box). The dash-dotted line corresponds to patch 3 (the top of the box). The dashed line corresponds to patch 4 (the floor). We see the evolution of the set of affine parameters is distinct for each surface patch, in particular see the evolution of $\boldsymbol{D}_{11}, \boldsymbol{D}_{12}$, and $\boldsymbol{d}_1$.

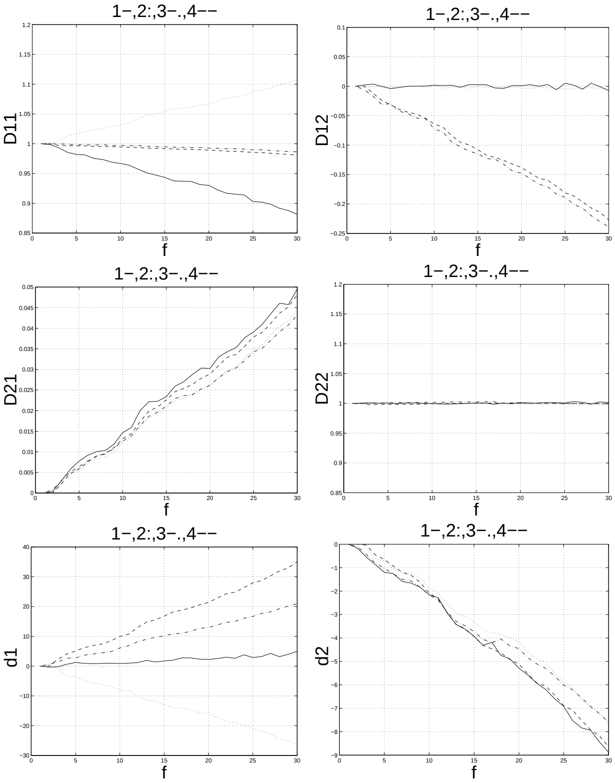

Figure 2: Estimates of the image motion parameters.

From the affine motion parameters of figure 2, we have recovered the 3D structure of the scene by using the rank 1 factorization method. Figure 3 shows two perspective views of the reconstructed 3D shape with the scene texture mapped on it. We see that the angles between the planar patches are correctly recovered.

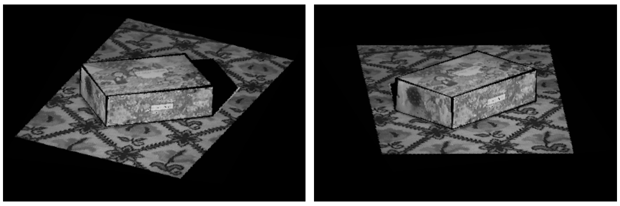

Figure 3: Two perspective views of the reconstructed 3D shape and texture.

7 Conclusion

We proposed a fast method to recover 3D rigid structure from motion. Rather than relying on the tracking of pointwise features, the image motion estimation step makes use of the affine motion parameterization for larger regions, leading to robust estimates of the image motion parameters. The 3D structure from motion step is robust because it takes into account the rigidity of scene over a set of frames. This step is accomplished with very low computational cost by factorizing a rank 1 matrix. The experimental results illustrate the performance of the method.

References

[1] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, 1992.

[2] Pedro M. Q. Aguiar and José M. F. Moura. Video representation via 3D shaped mosaics. In *IEEE ICIP*, Chicago, USA, October 1998.

[3] Conrad J. Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE PAMI*, 19(3):206–218, 1997.

[4] Pedro M. Q. Aguiar and José M. F. Moura. Factorization as a rank 1 problem. In *IEEE CVPR*, Fort Collins, USA, June 1999.

[5] H. Zheng and S. Blostein. Motion-based object segmentation and estimation using the MDL principle. *IEEE Trans. Image Processing*, 4(9), 1995.

[6] M. M. Chang, A. M. Tekalp, and M. I. Sezan. Simultaneous motion estimation and segmentation. *IEEE Trans. Image Processing*, 6(9), 1997.