A FACTORIZATION METHOD FOR 3D MULTI-BODY MOTION ESTIMATION AND SEGMENTATION

by

René Vidal, Stefano Soatto and Shankar Sastry

Memorandum No. UCB/ERL M02/3

26 February 2002

A FACTORIZATION METHOD FOR 3D MULTI-BODY MOTION ESTIMATION AND SEGMENTATION

by

René Vidal, Stefano Soatto and Shankar Sastry

Memorandum No. UCB/ERL M02/3

26 February 2002

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A Factorization Method for 3D Multi-body Motion Estimation and Segmentation *

René Vidal† Stefano Soatto‡ Shankar Sastry†

†Department of EECS, UC Berkeley
301 Cory Hall, Berkeley CA 94710
{rvidal,sastry}@eecs.berkeley.edu

‡Department of Computer Sciences, UCLA
Boelter Hall 4531, Los Angeles CA 90095
soatto@cs.ucla.edu

February 26, 2002

Abstract

We study the problem of estimating the motion of independently moving objects observed by a moving perspective camera. Given a set of image points and their optical flow in multiple frames, we show how to estimate the number of independent motions, the segmentation of the image points, the motion of the camera and that of each object. We do so by combining the so-called *subspace constraints* with the Costeira and Kanade algorithm for orthographic cameras [3]. We evaluate the proposed algorithm on synthetic and real image sequences.

1 Introduction

In this paper, we study the problem of estimating the motion of independently moving objects observed by a moving perspective camera in multiple views. We do not assume prior segmentation of the points, nor do we restrict the motion of the objects to be linear or constant. Also, we do not assume previous knowledge of the number of independent motions. Our approach is based on the fact that infinitesimal image measurements corresponding to independent motions lie on orthogonal six dimensional subspaces of a high dimensional linear space. Therefore, one can estimate the number independent motions, the segmentation of the image points, the motion of the camera and that of each object from a set of image points and their optical flows. Our motion segmentation approach naturally integrates information over time, since the so-called *subspace constraints* [8] are applied to image measurements from multiple frames.

The problem of estimating the 3D motion of a moving camera observing a single static object is well studied in the computer vision community [4, 7] (see, for example, reviews of batch methods [17], recursive methods [12, 16], orthographic case [18] and projective reconstruction [20]). The problem of estimating the 3D motion of multiple moving objects observed by a moving camera is more recent and has received a lot of attention over the past few years [1, 3, 5, 6, 15, 19, 21].

Costeira and Kanade [3] proposed an algorithm to estimate the motion of multiple moving objects relative to a static orthographic camera, based on discrete image measurements for each object. They use a factorization method based on the fact that, under orthographic projection, discrete image measurements lie on a low-dimensional linear variety. Unfortunately, under full perspective projection such a variety is nonlinear [19], hence factorization methods cannot be used. However, Irani [8] showed that infinitesimal image measurements do lie on a low-dimensional linear variety. She used subspace constraints on the motion field to obtain a multi-frame algorithm for the estimation of the optical flow of a moving camera observing a static scene. She did not use those constraints for 3D motion segmentation or estimation.

*Research supported by ONR grant N00014-00-1-0621 and ARO grant DAAD19-99-1-0139.

Han and Kanade [6] proposed an algorithm for reconstructing a scene containing multiple moving points, some of them static and the others moving linearly with constant speed. The algorithm assumes a moving orthographic camera and does not require previous segmentation of the points. The case of a perspective camera was studied by Shashua and Levin [15], again under the assumption that points move linearly with constant speed.

1.1 Notation and Problem Statement

The **motion** of the camera and that of the objects is modeled as a rigid body motion in $\mathbb{R}^3$, *i.e.,* as an element of the special Euclidean group

$$SE(3) = \{(R,T) \mid R \in SO(3), T \in \mathbb{R}^3\}$$

and its Lie algebra

$$se(3) = \{([\omega]_\times, v) \mid [\omega]_\times \in so(3), v \in \mathbb{R}^3\},$$

where SO(3) and so(3) are the space of rotation and skew-symmetric matrices in $\mathbb{R}^{3\times 3}$, respectively.

The **image** $\mathbf{x} = [x, y, 1]^T \in \mathbb{R}^3$ of a point $q = [q_1, q_2, q_3]^T \in \mathbb{R}^3$ (with respect to the camera frame), is assumed to satisfy the *perspective projection* equation:

$$\mathbf{x} = q/Z, \tag{1}$$

where $Z = q_3 > 0$ encodes the (unknown positive) **depth** of the point q with respect to its image $\mathbf{x}$.

The **optical flow** $\mathbf{u}$ at point q is defined as the velocity of $\mathbf{x}$ in the image plane, *i.e.,*

$$[\mathbf{u}^T, 0]^T = \dot{\mathbf{x}}.$$

Problem Statement: Let $\mathbf{x}_j^i$ be the image of point $i = 1, \ldots n$ in frame $j = 0, \ldots, m$, with $j = 0$ being the reference frame. Let $\{\mathbf{u}_j^i\}$ be the optical flow of point $\mathbf{x}_0^i$ between frames 0 and $j = 1, \ldots, m$. Given the images $\{\mathbf{x}_j^i\}$ and the flows $\{\mathbf{u}_j^i\}$, recover the number of moving objects, the object to which each point belongs to, the depth of the n points, the motion of the camera and that of the objects.

To be consistent with the notation, we always use the superscript to enumerate the n different points and/or the object to which the point belongs to. We omit the superscript when we refer to a generic single point and/or object. The subscript is always used to enumerate the m different camera frames.

2 Single-Body Multi-View Geometry

Let us start with the simplest case in which the moving camera observes a single moving object. Let $(R_o(t), T_o(t)) \in SE(3)$ and $(R_c(t), T_c(t)) \in SE(3)$ be the poses of the object and the camera at time t with respect to a fixed reference frame. Let Q be a point located on the object with coordinates $q \in \mathbb{R}^3$ relative to the object frame. The coordinates of Q relative to the reference frame are:

$$q_o(t) = R_o(t)q + T_o(t)$$

and the coordinates of Q relative to the camera frame are:

$$q_{oc}(t) = R_c^T(t)R_o(t)q + R_c^T(t)(T_o(t) - T_c(t)). \tag{2}$$

2.1 Differential Case

Differentiating (2) yields:

$$\dot{q}_{oc} = (\dot{R}_c^T R_o + R_c^T \dot{R}_o)q + \dot{R}_c^T(T_o - T_c) + R_c^T(\dot{T}_o - \dot{T}_c). \tag{3}$$

Combining (2) and (3) gives:

$$\dot{q}_{oc} = (\dot{R}_c^T R_c + R_c^T \dot{R}_o R_o^T R_c) q_{oc} + R_c^T (\dot{T}_o - \dot{T}_c - \dot{R}_o R_o^T (T_o - T_c)). \quad (4)$$

Since $\dot{R}R^T \in so(3)$, $[R^T\omega]_\times = R^T[\omega]_\times R$ and $\dot{R}^T R = -R^T \dot{R} R^T R$ [13], we may define the angular velocities $\omega_c, \omega_o \in \mathbb{R}^3$ by:

$$[\omega_o]_\times = \dot{R}_o R_o^T \quad \text{and} \quad [\omega_c]_\times = \dot{R}_c R_c^T. \quad (5)$$

Combining (4) and (5) yields:

$$\dot{q}_{oc} = [R_c^T(\omega_o - \omega_c)] \times q_{oc} + R_c^T(\dot{T}_o - \dot{T}_c - [\omega_o]_\times (T_o - T_c)) = [\omega]_\times q_{oc} + v,$$

where ω and v are the angular and translational velocities of the object relative to the camera.

Under perspective projection, the optical flow $\mathbf{u}$ of point Q is then given by:

$$\mathbf{u} = \frac{d}{dt}\left(\frac{q_{oc}}{Z}\right) = \frac{1}{Z}\begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \end{bmatrix} \dot{q}_{oc} = \begin{bmatrix} -xy & 1+x^2 & -y & 1/Z & 0 & -x/Z \\ -(1+y^2) & xy & x & 0 & 1/Z & -y/Z \end{bmatrix} \begin{bmatrix} \omega \\ v \end{bmatrix}$$

where $q_{oc} = (X, Y, Z)^T$ and $(x, y, 1)^T = q_{oc}/Z$.

Given measurements for the optical flow $\mathbf{u}_j^i = (\mathrm{u}_j^i, \mathrm{v}_j^i)^T$ of point $i = 1...n$ in frame $j = 1...m$, define the matrix of *rotational flows* Ψ and the matrix of *translational flows* Φ as:

$$\Psi = \begin{bmatrix} -\{xy\} & \{1+x^2\} & -\{y\} \\ -\{1+y^2\} & \{xy\} & \{x\} \end{bmatrix} \in \mathbb{R}^{2n\times 3} \quad \text{and} \quad \Phi = \begin{bmatrix} \{1/Z\} & 0 & -\{x/Z\} \\ 0 & \{1/Z\} & -\{y/Z\} \end{bmatrix} \in \mathbb{R}^{2n\times 3},$$

where (for example) $\{xy\}^T = [x^1y^1, \cdots, x^ny^n]$.

Also let

$$U = \begin{bmatrix} \mathrm{u}_1^1 & \cdots & \mathrm{u}_m^1 \\ \vdots & & \vdots \\ \mathrm{u}_1^n & \cdots & \mathrm{u}_m^n \end{bmatrix} \quad \text{and} \quad V = \begin{bmatrix} \mathrm{v}_1^1 & \cdots & \mathrm{v}_m^1 \\ \vdots & & \vdots \\ \mathrm{v}_1^n & \cdots & \mathrm{v}_m^n \end{bmatrix}.$$

Then, the *optical flow* matrix $W \in \mathbb{R}^{2n\times m}$ satisfies:

$$W = \begin{bmatrix} U \\ V \end{bmatrix} = [\Psi \ \Phi]_{2n\times 6} \begin{bmatrix} \omega_1 & \cdots & \omega_m \\ v_1 & \cdots & v_m \end{bmatrix}_{6\times m} = SM^T$$

where ω_j and v_j are the velocities of the object relative to the camera in the j^{th} frame. We call $S \in \mathbb{R}^{2n\times 6}$ the *structure* matrix and $M \in \mathbb{R}^{m\times 6}$ the *motion* matrix. We conclude that, for general translation and rotation, the optical flow matrix W has rank 6. This rank constraint, among others, was first derived by Irani [8] who used it to obtain a multi-frame algorithm for the estimation of the optical flow of a moving camera observing a static scene.

The rank constraint $\mathrm{rank}(W) = 6$ can be naturally used to derive a factorization method for estimating the relative velocities (ω_j, v_j) and depth Z^i from image points $\mathbf{x}_1^i$ and optical flows $\mathbf{u}_j^i$. We can do so by factorizing W into its motion and structure components. For, consider the SVD of $W = \mathcal{U}\mathcal{S}\mathcal{V}^T$ and let $\tilde{S} = \mathcal{U}$ and $\tilde{M} = \mathcal{V}\mathcal{S}$. Then we have $S = \tilde{S}A$ and $M = \tilde{M}A^{-T}$ for some $A \in \mathbb{R}^{6\times 6}$. Let A_k be the k-th column of A. Then the columns of A must satisfy:

$$\tilde{S}A_{1-3} = \Psi \quad \text{and} \quad \tilde{S}A_{4-6} = \Phi.$$

Since Ψ is known, A_{1-3} can be immediately computed. The remaining columns of A and the vector of depths $\{1/Z\}$ can be obtained up to scale from:

$$\begin{bmatrix} -I & \tilde{S}_\mathrm{u} & 0 & 0 \\ -I & 0 & \tilde{S}_\mathrm{v} & 0 \\ \mathrm{diag}(\{x\}) & 0 & 0 & \tilde{S}_\mathrm{u} \\ \mathrm{diag}(\{y\}) & 0 & 0 & \tilde{S}_\mathrm{v} \\ 0 & \tilde{S}_\mathrm{v} & 0 & 0 \\ 0 & 0 & \tilde{S}_\mathrm{u} & 0 \end{bmatrix} \begin{bmatrix} \{1/Z\} \\ A_4 \\ A_5 \\ A_6 \end{bmatrix} = 0.$$

where $\tilde{S}_\mathrm{u} \in \mathbb{R}^{n\times 6}$ and $\tilde{S}_\mathrm{v} \in \mathbb{R}^{n\times 6}$ are the upper and lower part of $\tilde{S}$, respectively.

2.2 Discrete Case

We consider equation (2) at two time instants, t and t_0 and eliminate q to obtain:

$$\begin{aligned} q_{oc}(t) =& R_c(t)^T R_o(t) R_o(t_0)^T R_c(t_0) q_{oc}(t_0) + \\ & R_c(t)^T (T_o(t) - T_c(t)) - \\ & R_c(t)^T R_o(t) R_o(t_0)^T (T_o(t_0) - T_c(t_0)) \\ =& R(t,t_0) q_{oc}(t_0) + T(t,t_0) \end{aligned}$$

where $(R(t,t_0), T(t,t_0))$ can be interpreted as the change in the relative pose of the object with respect to the camera between times t_0 and t.

There are a number of methods to estimate (R,T) from image measurements [17, 12, 16, 18,.20]. Here we choose a simple linear method based on rank constraints on the multiple view matrix [11], because it exploits the fact that the depth vector is known from the factorization method of the previous section.

Assume that we take measurements at discrete time instants $t = t_1, \ldots, t_m$ and let $R_j = R(t_j, t_0)$, $T_j = T(t_j, t_0)$ and $q_j = q_{oc}(t_j)$. Then we have:

$$\begin{aligned} q_j^i =& R_j q_0^i + T_j \\ Z_j^i \mathbf{x}_j^i =& Z^i R_j \mathbf{x}_0^i + T_j \\ 0 =& Z^i [\mathbf{x}_j^i]_\times R_j \mathbf{x}_0^i + [\mathbf{x}_j^i]_\times T_j. \end{aligned}$$

Solving for (R_j, T_j) is equivalent to finding vectors $\vec{R}_j$ and $\vec{T}_j$, $j = 1, \ldots, m$, such that:

$$P_j \begin{bmatrix} \vec{R}_j \\ \vec{T}_j \end{bmatrix} = \begin{bmatrix} Z^1 [\mathbf{x}_j^1]_\times * \mathbf{x}_0^{1T} & [\mathbf{x}_j^1]_\times \\ Z^2 [\mathbf{x}_j^2]_\times * \mathbf{x}_0^{2T} & [\mathbf{x}_j^2]_\times \\ \vdots & \vdots \\ Z^n [\mathbf{x}_j^n]_\times * \mathbf{x}_0^{nT} & [\mathbf{x}_j^n]_\times \end{bmatrix} \begin{bmatrix} \vec{R}_j \\ \vec{T}_j \end{bmatrix} = 0 \in \mathbb{R}^{3n}, \tag{6}$$

where $\vec{R}_j = [r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}]^T \in \mathbb{R}^9$, $\vec{T}_j = T_j \in \mathbb{R}^3$ and $A * B$ is the *Kronecker product* of A and B.

It can be shown that P_j is of rank 11 if more than $n \geq 6$ points in general position are given. In that case, the kernel of P_j is unique, and so is (R_j, T_j). However, in the presence of noise, R_j may not be an element of $SO(3)$. In order to obtain an element of $SO(3)$ we proceed as follows: Let $\tilde{R}_j \in \mathbb{R}^{3\times 3}$ and $\tilde{T}_j \in \mathbb{R}^3$ be the (unique) solution of (6). Such a solution is obtained as the eigenvector of P_j associated to the smallest eigenvalue. Let $\tilde{R}_j = \mathcal{U}_j \mathcal{S}_j \mathcal{V}_j^T$ be the SVD of $\tilde{R}_j$. Then the solution of (6) in $SO(3) \times \mathbb{R}^3$ is:

$$R_j = \text{sign}(\det(\mathcal{U}_j \mathcal{V}_j^T))\, \mathcal{U}_j \mathcal{V}_j^T \in SO(3) \tag{7}$$

$$T_j = \frac{\text{sign}(\det(\mathcal{U}_j \mathcal{V}_j^T))}{\sqrt[3]{\det(\mathcal{S}_j)}}\, \tilde{T}_j \in \mathbb{R}^3. \tag{8}$$

3 Multi-Body Multi-View Geometry

So far, we have assumed that the scene contains a single moving object. Now, we consider the case in which a single camera observes n_o objects. The new optical flow matrix W will contain additional rows corresponding to measurements from the different objects. However, we cannot directly apply the factorization method of the previous section to solve for the relative motion of each object, because we do not know which measurements in W correspond to which object. We therefore need to consider the segmentation problem first, *i.e.*, the problem of separating all the measurements into n_o classes:

$$\mathcal{I}^k = \{i \in \{1...n\} | \, \forall j \in \{1...m\} \; \mathbf{x}_j^i \in \text{object } k\}.$$

Furthermore, we assume that n_o itself is unknown.

3.1 Estimating the number of independent motions

Assume that the camera tracks n^k image points for object k and let $n = \sum n^k$ be the total number of points tracked. Also let U_k and V_k be matrices containing the optical flow of object k. If the segmentation of these points were known, then the multi-body optical flow matrix could be written as:

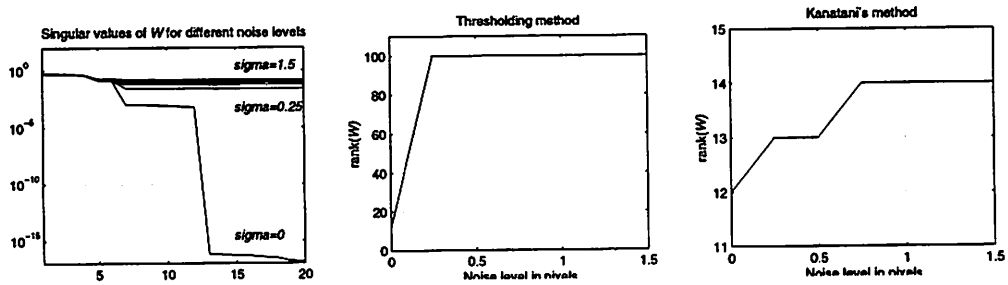
$$
\begin{aligned}
W = \begin{bmatrix} U \\ \hline V \end{bmatrix} &= \begin{bmatrix} U_1 \\ \vdots \\ U_{n_o} \\ \hline V_1 \\ \vdots \\ V_{n_o} \end{bmatrix} = \left[\begin{array}{ccc} \tilde{S}_{u1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \tilde{S}_{un_o} \\ \hline \tilde{S}_{v1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \tilde{S}_{vn_o} \end{array}\right] \begin{bmatrix} \tilde{M}_1^T \\ \vdots \\ \tilde{M}_{n_o}^T \end{bmatrix} = \tilde{S}\tilde{M}^T \\
&= \tilde{S} \begin{bmatrix} A_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_{n_o} \end{bmatrix} \begin{bmatrix} A_1^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_{n_o}^{-1} \end{bmatrix} \tilde{M}^T \\
&= \tilde{S}AA^{-1}\tilde{M}^T = SM^T.
\end{aligned}
$$

where $\tilde{S}_{uk}$ and $\tilde{S}_{vk} \in \mathbb{R}^{n^k \times 6}$, $k = 1...n_o$, $\tilde{S}$ and $S \in \mathbb{R}^{2n \times 6n_o}$, $A \in \mathbb{R}^{6n_o \times 6n_o}$ and $\tilde{M}, M \in \mathbb{R}^{m \times 6n_o}$.

Since we are assuming that the segmentation of the image points is unknown, the rows of W may be in a different order. However, the reordering of the rows of W will not affect its rank. Assuming that $n \geq 6n_o$ and $m \geq 6n_o$, we conclude that the number of independent motions n_o can be estimated as:

$$\boxed{n_o = \text{rank}(W)/6.} \tag{9}$$

In practice, optical flow measurements will be noisy and W will be full rank. Even though one could estimate the number of objects using by thresholding the singular values of W, it is better to use some statistics. Kanatani [10] studied the problem for the orthographic projection model using the geometric information criterion. The same method can be used here for a perspective camera as shown in Figure 1, which plots the singular values of W and the estimated rank as a function of noise.



(a) (b) (c)

Figure 1: Estimating the rank of W for two independent motions. Zero mean Gaussian noise with standard deviation σ in pixels is added to W. (a) Singular values of W for different levels of noise $\sigma \in [0, 1.5]$. (b) rank(W) estimated with a threshold of 10^{-4} and (c) with Kanatani's method.

3.2 Segmenting the image points

Segmenting the image points is equivalent to finding the unknown reordering of the rows of W. We can model such a reordering as a permutation matrix $P \in \mathbb{R}^{n \times n}$, *i.e.*, a matrix such that $P^2 = P$, applied to both

U and V. Such a permutation will affect the rows of $\tilde{S}$, hence those of S, but A, $\tilde{M}$ and M are unaffected. Therefore, from the SVD of $W = \mathcal{U}\mathcal{S}\mathcal{V}^T$ we have

$$\mathcal{U}\mathcal{U}^T = \begin{bmatrix} P\begin{bmatrix} \tilde{S}_{u1}\tilde{S}_{u1}^T & 0 \\ & \ddots \\ 0 & \tilde{S}_{un_o}\tilde{S}_{un_o}^T \end{bmatrix}P^T & P\begin{bmatrix} \tilde{S}_{u1}\tilde{S}_{v1}^T & 0 \\ & \ddots \\ 0 & \tilde{S}_{un_o}\tilde{S}_{vn_o}^T \end{bmatrix}P^T \\ P\begin{bmatrix} \tilde{S}_{v1}\tilde{S}_{u1}^T & 0 \\ & \ddots \\ 0 & \tilde{S}_{vn_o}\tilde{S}_{un_o}^T \end{bmatrix}P^T & P\begin{bmatrix} \tilde{S}_{v1}\tilde{S}_{v1}^T & 0 \\ & \ddots \\ 0 & \tilde{S}_{vn_o}\tilde{S}_{vn_o}^T \end{bmatrix}P^T \end{bmatrix},$$

We define the *segmentation matrix* Σ as the sum of the diagonal blocks of $\mathcal{U}\mathcal{U}^T$, *i.e.*,

$$\Sigma = P\begin{bmatrix} \tilde{S}_{u1}\tilde{S}_{u1}^T + \tilde{S}_{v1}\tilde{S}_{v1}^T & & 0 \\ & \ddots & \\ 0 & & \tilde{S}_{un_o}\tilde{S}_{un_o}^T + \tilde{S}_{vn_o}\tilde{S}_{vn_o}^T \end{bmatrix}P^T.$$

Then, $\Sigma_{ij} > 0$ if and only if image points i and j belong to the same object. In the absence of noise, the matrix Σ can be trivially used to determine the class to which each image point belongs to. One can also use each one of the two diagonal blocks of $\mathcal{U}\mathcal{U}^T$. In the presence of noise, Σ_{ij} will be nonzero even if points i and j correspond to different objects. Techniques that handle this case can be found in [3, 9] for the orthographic case. They can also be applied here to the perspective case.

3.3 Recovering absolute motion from relative motion

Once the segmentation problem has been solved, one can apply the algorithms in Section 2 to estimate the motion of each object separately. Since the camera is moving, this will give the motion of each object relative to the camera. We now consider the problem of obtaining the motion relative to a fixed reference frame. We show that it is not possible to solve the problem from image measurements only, unless some additional assumptions are made.

3.3.1 Camera motion

In practice, some of the image measurements will correspond to static points in 3D space. We define the *background* as the set of image points associated to static 3D points. We will assume that the background is the class with the largest spatial standard deviation in all the frames[1].

The optical flow matrix W will be segmented into $n_o + 1$ classes. We denote the background as zero-th class. Also, let (ω_j^k, v_j^k) and (R_j^k, T_j^k) be the estimates of relative motion for class k in frame j as obtained by the algorithms in Section 2. Given the assumptions, the zero-th class contains information about the motion of the camera only. More explicitly, we have:

$$w_j^0 = -R_{cj}^T\omega_{cj} \qquad \lambda^0 v_j^0 = -R_{cj}^T\dot{T}_{cj} \tag{10}$$

$$R_j^0 = R_{cj}^T R_{c0} \qquad \lambda^0 T_j^0 = -R_{cj}^T(T_{cj} - T_{c0}) \tag{11}$$

where λ^0 is the unknown scale lost under perspective projection. We are now interested in recovering the absolute motion (R_{c0}, T_{c0}) and (ω_{cj}, v_{cj}), (R_{cj}, T_{cj}), $j = 1...m$. We can see from (10) and (11) that this cannot be done, because there are $12m + 7$ unknowns and $12m$ equations. Therefore, the absolute motion of the camera can be estimated up to a 7-parameter family, given (for example) by the initial rotation and translation of the camera and the scale lost under perspective projection. For the case of a single camera, this ambiguity is not relevant, since it is equivalent to choosing the reference frame, which can be chosen to coincide with the initial location of the camera, *i.e.*, $(R_{c0}, T_{c0}) = (I, 0)$.

[1]Even though not all scenes satisfy the assumption, in many practical situations static points are distributed uniformly, while moving points are a collection of connected components, each one corresponding to one object.

3.3.2 Motion of each object

Given λ^0, $(\omega_{cj}, \dot{T}_{cj}), j = 1...m$ and $(R_{cj}, T_{cj}), j = 0...m$ we would like to solve for λ^k, $(\omega_{oj}^k, \dot{T}_{oj}^k), j = 1...m$ and $(R_{oj}^k, T_{o0}^k), j = 0...m, k = 1...n_o$ from:

$$w_j^k = R_{cj}^T(\omega_{oj}^k - \omega_{cj}) \tag{12}$$

$$\lambda^k v_j^k = R_{cj}^T(\dot{T}_{oj}^k - \dot{T}_{cj} - [w_{oj}^k]_\times(T_{oj}^k - T_{cj})) \tag{13}$$

$$R_j^k = R_{cj}^T R_{oj}^k R_{o0}^{kT} R_{c0} \tag{14}$$

$$\lambda^k T_j^k = R_{cj}^T(T_{oj}^k - T_{cj} - R_{oj}^k R_{o0}^{kT}(T_{o0}^k - T_{c0})). \tag{15}$$

Again, we observe that the motion of the objects can be recovered up to a $7n_o$-parameter family given by the initial pose of each object and the unknown scales lost under perspective projection.

In order to resolve the translation ambiguity T_{o0}, as before we assume that image points corresponding to each object are concentrated in a specific region of the image. Therefore, the average of the 3D points associated to those image points well approximates the position of the object relative to the camera (up to scale). We then approximate the initial position of each object as:

$$T_{o0}^k \approx \lambda^k R_{c0} \sum_{i \in \mathcal{I}^k} \frac{\mathbf{x}_0^i Z_0^i}{n^k} + T_{c0} \tag{16}$$

Combining (14), (15) and (16), the position of the objects in the remaining frames is given by:

$$T_{oj}^k \approx \lambda^k R_{cj} \left(T_j^k + R_j^k \sum_{i \in \mathcal{I}^k} \frac{\mathbf{x}_0^i Z_0^i}{n^k} \right) + T_{cj} \tag{17}$$

In relation to the rotation ambiguity, we observe that it is not possible to estimate R_{o0}^k. One can only estimate $R_{oj}^k R_{o0}^{kT}$, which is the orientation of the object relative to its initial configuration. If we assume that the initial orientation of the objects is known, then $(\omega_{oj}^k, \dot{T}_{oj}^k)$ can be trivially obtained from (12) and (13). Therefore, given the assumptions, the motion of each object can be completely solved with $(R_{oj}^k, \omega_{oj}^k)$ obtained uniquely, and $(T_{oj}^k, \dot{T}_{oj}^k)$ obtained up to a scale λ^k.

4 Experimental Results

In this section, we evaluate the proposed algorithm on real and synthetic image sequences. Each pixel of each frame is considered as a feature and segmentation is performed using the segmentation matrix associated to the optical flow of those pixels.

Figure 2 shows the *street* sequence [14], which contains two independent motions: the motion of the car and the motion of the camera that is panning to the right. Figure 4(a) shows frames 3, 8 12 and 16 of the sequence with the corresponding optical flow superimposed. Optical flow is computed using Black's algorithm [2]. Figures 4(b)-(c) show the segmentation results. In frame 4 the car is partially occluded, thus only the frontal part of the car is segmented from the background. The door is incorrectly segmented because it is in a region with low texture. As time proceeds, motion information is integrated over time by incorporating optical flow from many frames in the optical flow matrix, thus the door is correctly segmented. In frame 16 the car is fully visible and correctly segmented from the moving background.

Figure 3 shows the *sphere-cube* sequence [14], which contains a sphere rotating along a vertical axis and translating to the right, a cube rotating counter clock-wise and translating to the left, and a static background. Even though the optical flow of the sphere appears to be noisy, its motion is correctly segmented. The top left (when visible), top and right sides of the square are also correctly segmented in spite of the fact that only normal flow is available. The left bottom side of the cube is confused with the background,

because its optical flow is approximately zero, since the translational motion of the cube cancels its rotational motion. The center of the cube is never segmented correctly since it corresponds to a region with low texture. Integrating motion information over many frames does not help here since those pixels are in a region with low texture during the whole sequence.

Figure 4(a) shows the *two-robot* sequence with the corresponding optical flow superimposed. Figures 4(b) and 4(c) show the results of the segmentation. Groups 1 and 2 correspond to the each one of the moving objects, while group 3 corresponds to the background, which is the correct segmentation.

5 Conclusions

We have proposed an algorithm for estimating the motion of multiple moving objects as observed by a moving camera in multiple frames. Our algorithm is based on the fact that image measurements from independent motions lie on orthogonal subspaces of a high dimensional space, thus it does not require prior segmentation or previous knowledge of the number of independent motions. Experimental results show how segmentation is correctly obtained by integrating image measurements from multiple frames.

References

[1] S. Avidan and A. Shashua. Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(4):348–357, 2000.

[2] M. Black. `http://www.cs.brown.edu/people/black/ignc.html`, 1996.

[3] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[4] O. Faugeras and Q.-T. Luong. *Geometry of Multiple Images.* The MIT Press, 2001.

[5] A. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3D reconstruction of independently moving objects. In *European Conference on Computer Vision*, 2000.

[6] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 542–549, 2000.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge, 2000.

[8] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *IEEE International Conference on Computer Vision*, pages 626–633, 1999.

[9] K. Kanatani. Motion segmentation by subspace separation and model selection. In *IEEE International Conference on Computer Vision*, volume 2, pages 586–591, 2001.

[10] K. Kanatani and C. Matsunaga. Estimating the number of independent motions for multibody motion segmentation. In *Asian Conference on Computer Vision*, 2002.

[11] Y. Ma, J. Košecká, and K. Huang. Rank deficiency condition of the multiple view matrix for mixed point and line features. In *Asian Conference on Computer Vision*, 2002.

[12] P. McLauchlan and D. Murray. A unifying framework for structure and motion recovery from image sequences. In *International Conference on Computer Vision and Pattern Recognition*, pages 314–20, 1995.

[13] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC press Inc., 1994.

[14] University of Otago New Zeland. http://www.cs.otago.ac.nz/research/vision/Research/OpticalFlow/opticalflow.html#Sequences.

[15] A. Shashua and A. Levin. Multi-frame infinitesimal motion model for the reconstruction of (dynamic) scenes with multiple linearly moving objects. In *IEEE International Conference on Computer Vision*, volume 2, pages 592–599, 2001.

[16] S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3):393–413, March 1996.

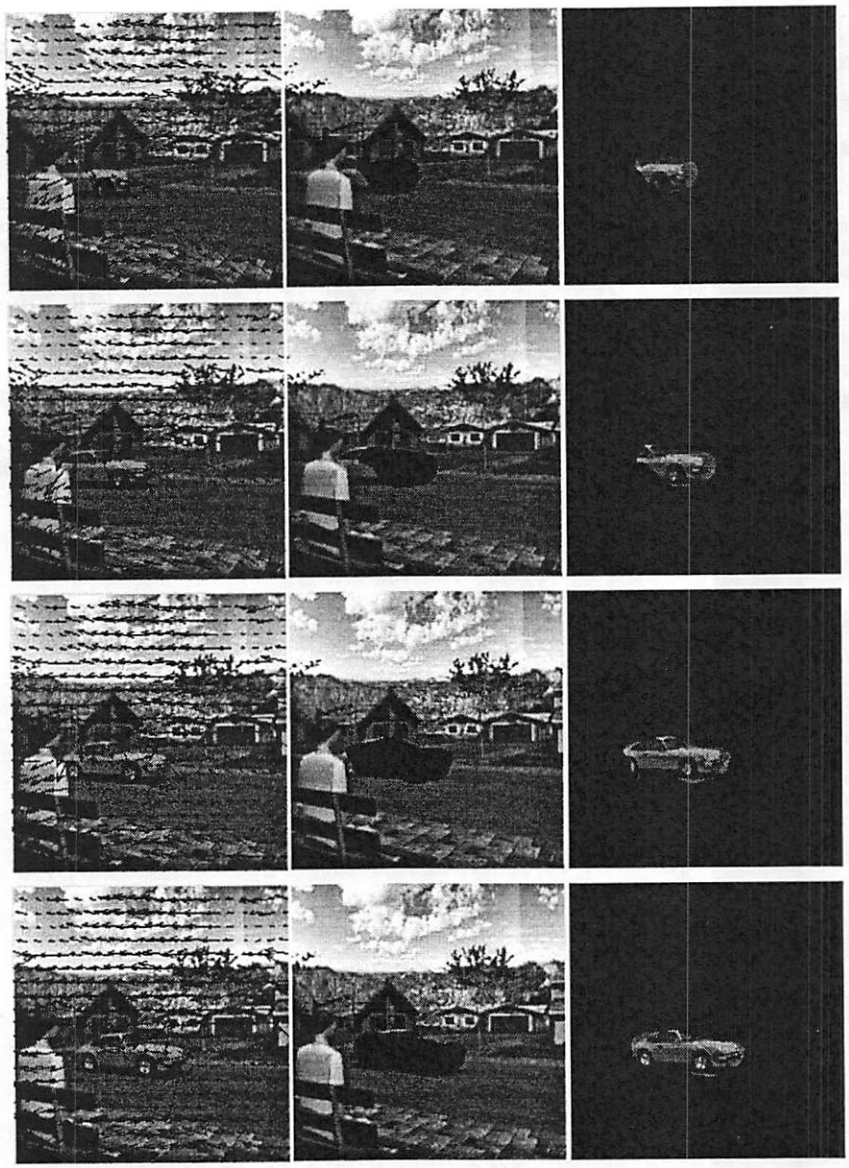
[17] R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, 1994.

[18] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[19] P. H. S. Torr. Geometric motion segmentation and model selection. *Phil. Trans. Royal Society of London A*, 356(1740):1321–1340, 1998.

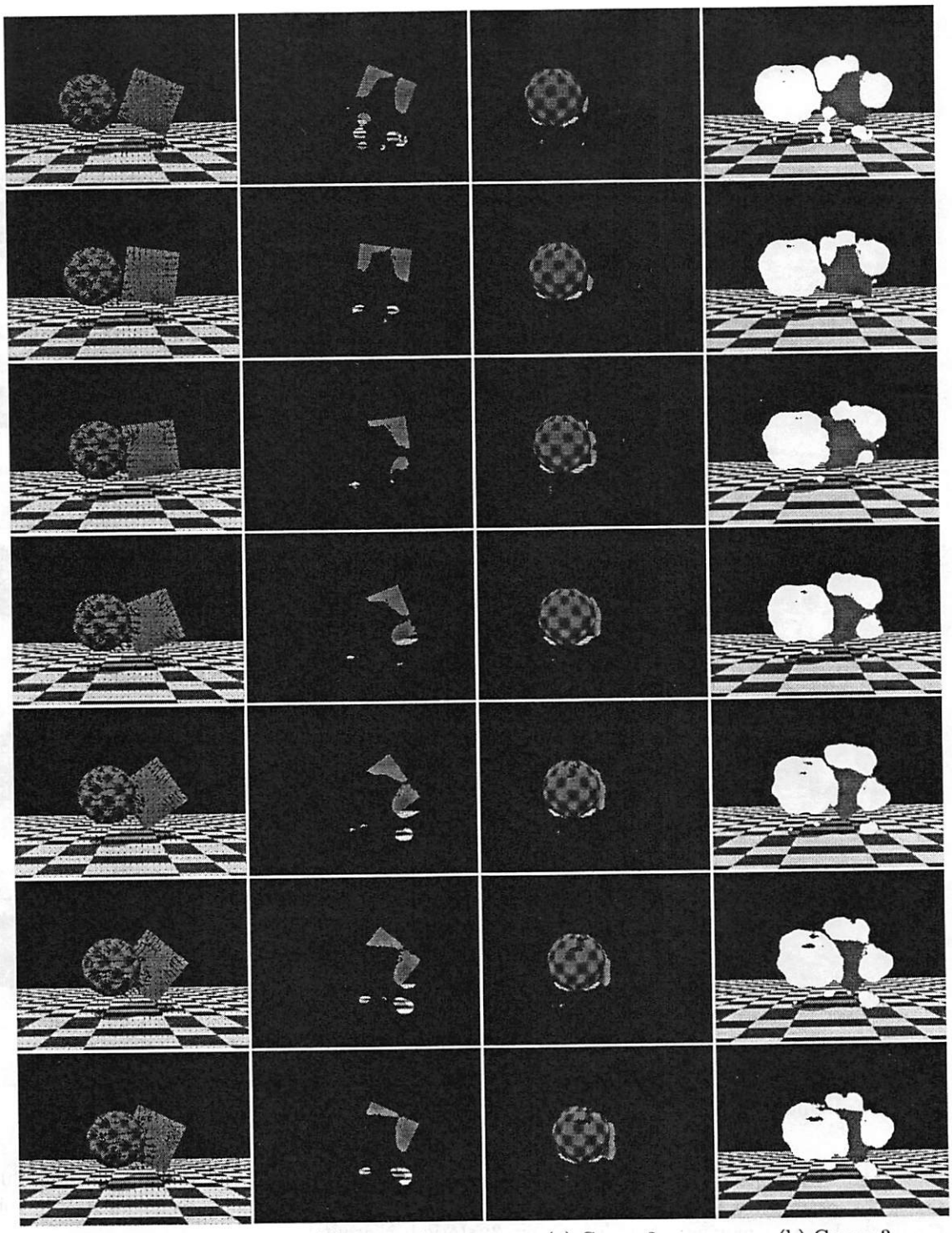
[20] B. Triggs. Factorization methods for projective structure and motion. In *International Conference on Computer Vision and Pattern Recognition*, pages 845–51, 1996.

[21] L. Wolf and A. Shashua. Two-body segmentation from two perspective views. In *International Conference on Computer Vision and Pattern Recognition*, 2001.



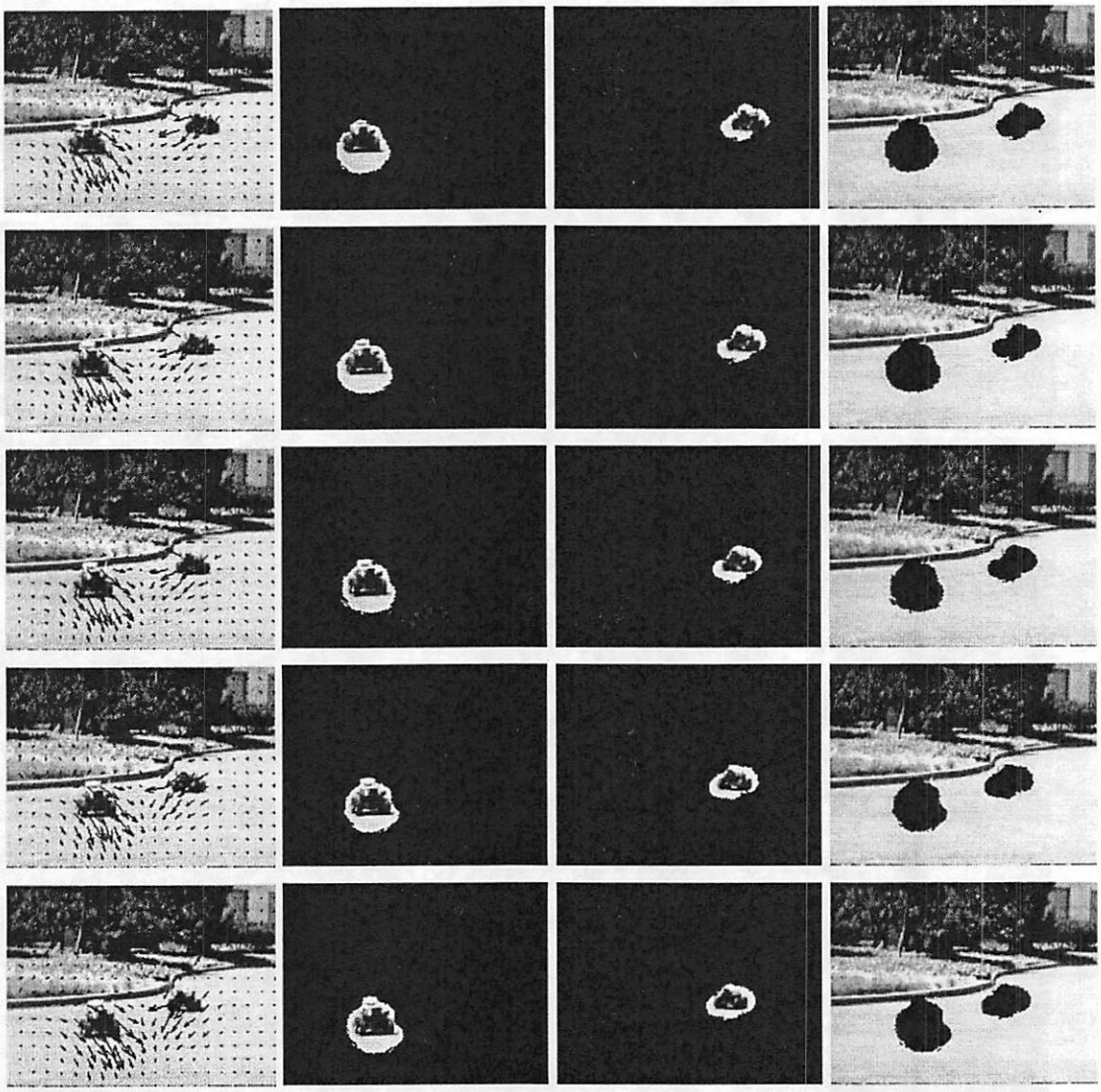
(a) Optical flow (b) Group 1 (c) Group 2

Figure 2: Segmentation results for the *street* sequence. The sequence has 18 frames and 200×200 pixels. The camera is panning to the right while the car is also moving to the right. (a) Frames 3, 8 12 and 16 of the sequence with the corresponding optical flow superimposed. (b) Group 1: motion of the camera. (c) Group 2: motion of the car.



(a) Optical flow (b) Group 1 (a) Group 2 (b) Group 3

Figure 3: Segmentation results for the *sphere-cube* sequence. The sequence contains 10 frames and 400×300 pixels. The sphere is rotating along a vertical axis and translating to the right. The cube is rotating counter clock-wise and translating to the left. The background is static. (a) Frames 2-8 with corresponding optical flow superimposed. (b) Group 1: cube motion. (c) Group 2: sphere motion. (d) Group 3: static background.



(a) Optical flow (b) Group 1 (c) Group 2 (d) Group 3

Figure 4: Segmentation results for the *two-robot* sequence. The sequence contains 6 frames and 200×150 pixels. (a) Frames 1-5 of the sequence with optical flow superimposed. (b) Group 1: one moving robot. (c) Group 2: the other moving robot. (d): Group 3: static background.