IMPROVED MULTISTAGE LEARNING FOR MULTIBODY MOTION SEGMENTATION

Yasuyuki Sugaya
Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, Aichi 441-8580, Japan

Kenichi Kanatani
Department of Computer Science, Okayama University, Okayama 700-8530, Japan

Keywords: Multibody motion segmentation, GCAP, Taubin method, EM algorithm, Multistage optimization.

Abstract: We present an improved version of the MSL method of Sugaya and Kanatani for multibody motion segmentation. We replace their initial segmentation based on heuristic clustering by an analytical computation based on GPCA, fitting two 2-D affine spaces in 3-D by the Taubin method. This initial segmentation alone can segment most of the motions in natural scenes fairly correctly, and the result is successively optimized by the EM algorithm in 3-D, 5-D, and 7-D. Using simulated and real videos, we demonstrate that our method outperforms the previous MSL and other existing methods. We also illustrate its mechanism by our visualization technique.

1 INTRODUCTION

Separating independently moving objects in a video stream has attracted attention of many researchers in the last decade, and today we are witnessing a new surge of interest in this problem. The most classical work is by Costeira and Kanade (1998), who showed that, under affine camera modeling, trajectories of image points in the same motion belong to a common subspace of a high-dimensional space. They segmented trajectories into different subspaces by zero-nonzero thresholding of the elements of the "interaction matrix" computed in relation to the "factorization method" for affine structure from motion (Poelman and Kanade, 1997; Tomasi and Kanade, 1992). Since then, various modifications and extensions have been proposed. Gear (1998) used the reduced row echelon form and graph matching. Ichimura (1999) used the Otsu discrimination criterion. He also used the QR decomposition (Ichimura, 2000). Inoue and Urahama (2001) introduced fuzzy clustering. Kanatani (2001, 2002, 2002a) combined the geometric AIC (kaike Information Criterion) (Kanatani, 1996) and robust clustering. Wu et al. (2001) introduced orthogonal subspace decomposition. Sugaya and Kanatani (2004) proposed a multistage learning strategy using multiple models. Vidal et al. (2005, 2008) applied their GPCA (Generalized Principal Component Analysis), which fits a high-degree polynomial to multiple subspaces. Fan et al. (2006) and Yan and Pollefeys (2006) introduced new voting schemes for classifying points into different subspaces in high dimensions. Schindler et al. (2008) and Rao et al. (2008) incorporated model selection based on the MDL (Minimum Description Length) principle.

At present, it is difficult to say which is the best among all these methods. Their performance has been tested, using real videos, but the result depends on the test videos and the type of the motion that is taking place (planar, translational, rotational, etc.). If such distinctions are disregarded and simply the gross correct classification ratio is measured using a particular database, typically the Hopkins155 (Tron and Vial, 2007), all the methods exhibit more or less similar performance.

A common view behind existing methods seems to be that the problem is intricate because the segmentation takes place in a high-dimensional space, which is difficult to visualize. This way of thinking has lead to introducing sophisticated mathematics one after another and simply testing the performance using the Hopkins155 database. In this paper, we show that the problem is not difficult at all and that the basis of segmentation lies in low dimensions. Indeed,

Sugaya Y. and Kanatani K. (2010).
IMPROVED MULTISTAGE LEARNING FOR MULTIBODY MOTION SEGMENTATION.
In *Proceedings of the International Conference on Computer Vision Theory and Applications*, pages 199-206
DOI: 10.5220/0002814601990206

we can visualize what is going on in 3-D. This reveals that what is crucial is the type of motion and that different motions can be easily segmented if the motion type is known.

Sugaya and Kanatani (2004) assumed multiple candidate motion types and presented the MSL (MultiStage Learning) strategy, which does not require identification of the motion type. To do this, they exploited the hierarchy of motions (e.g., translations are included in affine motions) and applied the EM algorithm by progressively assuming motion models from particular to general: Once one tested motion type agrees with the true one, the segmentation is unchanged in the subsequent stages because general motions include particular ones. Tron and Vidal (2007) did extensive comparative experiments and reported that MSL is highly effective. In this paper, we present an improved version of MSL.

Since MSL uses the EM algorithm, we need to provide an appropriate initial segmentation, which is the key to the performance of the subsequent stages, in which the segmentation in the preceding stage is input and the output is sent to the next stage. For computing the initial segmentation, MSL used a rather heuristic clustering that combines the interaction matrix of Costeira and Kanade (1998) and model selection using the geometric AIC (Kanatani, 1996). In this paper, we replace this by the GPCA of Vidal et al. (2005, 2008): we fit a degenerate quadric in 3-D by the method of Taubin (1991). Then, we successively apply the EM algorithm and demonstrate, using the Hopkins155 database, that our method outperforms MSL and other existing methods. We also show, using our visualization technique, why and how good segmentation results.

2 AFFINE CAMERAS

Suppose N feature points $\{p_\alpha\}$ are tracked over M image frames. Let $(x_{\kappa\alpha}, y_{\kappa\alpha})$, κ = 1, ..., M, be the image coordinates of the αth point p_α in the κth frame. We call the 2M-D vector

$$\mathbf{p}_\alpha = (x_{1\alpha},\ y_{1\alpha},\ x_{2\alpha},\ y_{2\alpha},\ \cdots\ x_{M\alpha},\ y_{M\alpha})^\top, \qquad (1)$$

the *trajectory* of p_α. Thus, an image motion of each point is identified with a point in 2M-D. We define a camera-based XYZ coordinate system such that the Z-axis coincides with the camera optical axis and regard the scene as moving relative to a stationary camera. We also define a coordinate system fixed to each of the moving objects. Let $(a_\alpha, b_\alpha, c_\alpha)$ be the coordinates of point p_α with respect to the coordinate system of the object it belongs to. Let $\mathbf{t}_\kappa$ be the origin of that coordinate system and $\{\mathbf{i}_\kappa, \mathbf{j}_\kappa, \mathbf{k}_\kappa\}$ the basis vectors in the κth frame. Then, the 3-D position $\mathbf{r}_{\kappa\alpha}$ of the point p_α in the κth frame with respect to the camera coordinate system is

$$\mathbf{r}_{\kappa\alpha} = \mathbf{t}_\kappa + a_\alpha \mathbf{i}_\kappa + b_\alpha \mathbf{j}_\kappa + c_\alpha \mathbf{k}_\kappa. \qquad (2)$$

The *affine camera*, which generalizes orthographic, weak perspective, and paraperspective projections (Poelman and Kanade, 1997), models the camera imaging by

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \mathbf{A}_\kappa \mathbf{r}_{\kappa\alpha} + \mathbf{b}_\kappa, \qquad (3)$$

where the 2 × 2 matrix $\mathbf{A}_\kappa$ and the 2-D vector $\mathbf{b}_\kappa$ are determined by the intrinsic and extrinsic camera parameters of the κth frame. By substitution of Eq. (2), Eq. (3) is written in the form

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \tilde{\mathbf{m}}_{0\kappa} + a_\alpha \tilde{\mathbf{m}}_{1\kappa} + b_\alpha \tilde{\mathbf{m}}_{2\kappa} + c_\alpha \tilde{\mathbf{m}}_{3\kappa}, \qquad (4)$$

where $\tilde{\mathbf{m}}_{0\kappa}$, $\tilde{\mathbf{m}}_{1\kappa}$, $\tilde{\mathbf{m}}_{2\kappa}$, and $\tilde{\mathbf{m}}_{3\kappa}$ are 2-D vectors determined by the intrinsic and extrinsic camera parameters of the κth frame. The trajectory in Eq. (1) is expressed as the vertical concatenation of Eq. (4) for κ = 1, ..., M, in the form

$$\mathbf{p}_\alpha = \mathbf{m}_0 + a_\alpha \mathbf{m}_1 + b_\alpha \mathbf{m}_2 + c_\alpha \mathbf{m}_3, \qquad (5)$$

where $\mathbf{m}_i$, i = 0, 1, 2, 3, are the 2M-D vectors consisting of $\tilde{\mathbf{m}}_{i\kappa}$ for κ = 1, ..., M.

3 GEOMETRIC CONSTRAINTS

Equation (5) states that the trajectories of points that belong to the same object are in a common "4-D subspace" spanned by $\{\mathbf{m}_0,\ \mathbf{m}_1,\ \mathbf{m}_2,\ \mathbf{m}_3\}$. Hence, segmenting trajectories into different motions can be done by classifying them into different 4-D subspaces in 2M-D. However, the coefficient of $\mathbf{m}_0$ in Eq. (5) is identically 1, which means that the trajectories of points that belong to the same object are in a common "3-D affine space" passing through $\mathbf{m}_0$ and spanned by $\{\mathbf{m}_1,\ \mathbf{m}_2,\ \mathbf{m}_3\}$. Thus, segmentation can also be done by classifying trajectories into different 3-D affine spaces in 2M-D.

In real situations, however, objects and a background often translate with rotations only around an axis vertical to the image plane. We say such a motion is *planar*; translations in the depth direction can take place, but they are invisible under the affine camera modeling, so we can regard translations as constrained to be in the XY plane. It follows that if we take the basis vector $\mathbf{k}_\kappa$ in Eq. (2) to be in the Z direction, it is invisible to the camera, and hence $\mathbf{m}_3$ =

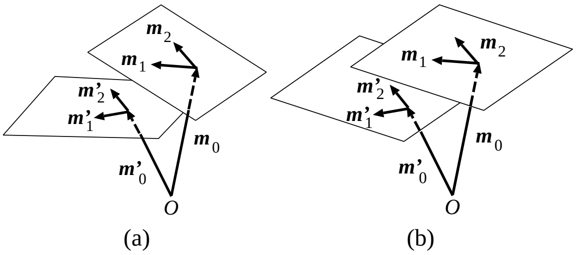

Figure 1: (a) If the motions are planar, object and background trajectories belong to different 2-D affine spaces. (b) If the motions are translational, object and background trajectories belong to 2-D affine spaces that are parallel to each other.

$\mathbf{0}$ in Eq. (5). Thus, the trajectories of points undergoing the same motion are in a common "2-D affine space" passing through $\mathbf{m}_0$ and spanned by $\{\mathbf{m}_1, \mathbf{m}_2\}$ (Fig. 1(a))

If, moreover, objects and a background merely translate without rotation, we can fix the basis vectors $\mathbf{i}_\kappa$ and $\mathbf{j}_\kappa$ in the X and Y directions, respectively. This means that the vectors $\mathbf{m}_1$ and $\mathbf{m}_2$ in Eq. (5) are common to all the objects and the background. Thus, the 2-D affine spaces are *parallel* to each other (Fig. 1(b)).

It is well known that the interaction-matrix-based method of Costeira and Kanade (1998) fails if the motion is planar. Furthermore, if there exist two 2-D affine spaces parallel to each other, they are both contained in some 3-D affine space, and hence in some 4-D subspace. This means classification of different motions into 3-D affine spaces or into 4-D subspaces is impossible. Yet, this type of degeneracy is very frequent in real situations. In fact, almost all "natural" scenes in the Hopkins155 database undergo such degeneracy to some extent[1]. This may be the main reason that many researchers have regarded multibody motion segmentation as difficult and tried various sophisticated mathematics one after another.

The MSL of Sugaya and Kanatani (2004) resolved this by starting from the translational motion assumption and progressively applying more general assumptions so that any degeneracy is not untested. In this paper, we improve their method by introducing new analytical initial segmentation and going on to successive upgrading in slightly different dimensions.

4 DIMENSION COMPRESSION

In the following, we concentrate on two motions: an object is moving relative to a background, which is also moving. If the two motions are both general, the observed trajectories belong to two 3-D affine spaces in $2M$-D. There exists a 7-D affine space that contains both. Hence, segmentation of trajectories can be done in a 7-D affine space: noise components in the outward directions do not affect the segmentation. If we translate the 7-D affine space so that it passes through the origin, take seven basis vectors in it, and express all the trajectories in their linear combinations, each trajectory can be identified with a point in 7-D. Similarly, if the observed trajectories are in two 2-D affine spaces in $2M$-D, there exists a 5-D affine space that contains both. Then, each trajectory can be identified with a point in 5-D. If, moreover, the two 2-D affine spaces in $2M$-D are parallel to each other, there exists a 3-D affine space that contains both, and each trajectory can be identified with a point in 3-D.

A trajectory in $2M$-D can be identified with a point in d-D by the following PCA (Principal Component Analysis):

1. Compute the centroid $\mathbf{p}_C$ of all the trajectories $\{\mathbf{p}_\alpha\}$ and the deviations $\tilde{\mathbf{p}}_\alpha$ from it:
$$\mathbf{p}_C = \frac{1}{N}\sum_{\alpha=1}^{N}\mathbf{p}_\alpha, \qquad \tilde{\mathbf{p}}_\alpha = \mathbf{p}_\alpha - \mathbf{p}_C. \tag{6}$$
2. Compute the SVD (Singular Value Decomposition) of the following $2M \times N$ matrix in the form
$$\left(\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_N\right) = \mathbf{U}\mathrm{diag}(\sigma_1, \ldots, \sigma_r)\mathbf{V}^\top, \tag{7}$$
where $r = \min(2M, N)$, and $\mathbf{U}$ and $\mathbf{V}$ are $2M \times r$ and $N \times r$ matrices, respectively, having r orthonormal columns.
3. Let $\mathbf{u}_i$ be the ith column of $\mathbf{U}$, and compute the following d-D vectors $\mathbf{r}_\alpha$, α = 1, ..., N:
$$\mathbf{r}_\alpha = \Big((\tilde{\mathbf{p}}_\alpha, \mathbf{u}_1), \ldots, (\tilde{\mathbf{p}}_\alpha, \mathbf{u}_d)\Big)^\top. \tag{8}$$

In this paper, we denote the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$ by $(\mathbf{a}, \mathbf{b})$.

5 INITIAL SEGMENTATION

Now, we describe our analytical initial segmentation that replaces the heuristic clustering of MSL. We identify trajectories with points in 3-D by the above procedure and fit two planes (= 2-D affine spaces). If the object and the background are both in translational motions, all the 3-D points belong to two parallel planes. This may not hold if the data are noisy or rotational components exist, but if the noise is small and the motions are nearly translational, which is the

[1] The exceptions are the artificial "box" scenes, in which boxes autonomously undergo unnatural 3-D translations and rotations. For these, segmentation is very easy.

case in most natural scenes, we can expect that two planes can fit to all the points fairly well.

A plane $Ax+By+Cz+D=0$ in 3-D can be written as $(\mathbf{n},\mathbf{x})=0$, where we put

$$\mathbf{n}=(A,B,C,D)^\top, \qquad \mathbf{x}=(x,y,z,1)^\top. \qquad (9)$$

Two planes $(\mathbf{n}_1,\mathbf{x})=0$ and $(\mathbf{n}_2,\mathbf{x})=0$ can be combined into one in the form

$$(\mathbf{n}_1,\mathbf{x})(\mathbf{n}_2,\mathbf{x})=(\mathbf{x},\mathbf{n}_1\mathbf{n}_2^\top\mathbf{x})=(\mathbf{x},\mathbf{Q}\mathbf{x})=0, \qquad (10)$$

where we define the following symmetric matrix $\mathbf{Q}$:

$$\mathbf{Q}=\frac{\mathbf{n}_1\mathbf{n}_2^\top+\mathbf{n}_2\mathbf{n}_1^\top}{2}. \qquad (11)$$

Note that it is a symmetric matrix that defines a quadratic form. Equation (11) implies that $\mathbf{Q}$ has rank 2 with two multiple zero eigenvalues and that the remaining eigenvalues have different signs. Let these eigenvalues be λ_1, 0, 0, $-\lambda_2$ in descending order, and $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$, $\mathbf{u}_4$ the corresponding unit eigenvectors. Then, $\mathbf{Q}$ has the following spectral decomposition:

$$\begin{aligned}\mathbf{Q}&=\lambda_1\mathbf{u}_1\mathbf{u}_1^\top-\lambda_2\mathbf{u}_4\mathbf{u}_4^\top\\ &=\Big(\sqrt{\frac{\lambda_1}{2}}\mathbf{u}_1+\sqrt{\frac{\lambda_2}{2}}\mathbf{u}_4\Big)\Big(\sqrt{\frac{\lambda_1}{2}}\mathbf{u}_1-\sqrt{\frac{\lambda_2}{2}}\mathbf{u}_4\Big)^\top\\ &\quad+\Big(\sqrt{\frac{\lambda_1}{2}}\mathbf{u}_1-\sqrt{\frac{\lambda_2}{2}}\mathbf{u}_4\Big)\Big(\sqrt{\frac{\lambda_1}{2}}\mathbf{u}_1+\sqrt{\frac{\lambda_2}{2}}\mathbf{u}_4\Big)^\top. \qquad (12)\end{aligned}$$

Comparing this with Eq. (11) and noting that vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ (hence the matrix $\mathbf{Q}$) have scale indeterminacy, we can determine $\mathbf{n}_1$ and $\mathbf{n}_2$ up to scale as follows:

$$\mathbf{n}_1=\sqrt{\lambda_1}\mathbf{u}_1+\sqrt{\lambda_2}\mathbf{u}_4, \quad \mathbf{n}_2=\sqrt{\lambda_1}\mathbf{u}_1-\sqrt{\lambda_2}\mathbf{u}_4. \quad (13)$$

Let $\mathbf{x}_1$, ..., $\mathbf{x}_N$ be the 3-D points that represent trajectories. In the presence of noise or rotational components, they may not exactly satisfy Eq. (10), so we fit a quadratic surface $(\mathbf{x},\mathbf{Q}\mathbf{x})=0$ to them in such a way that

$$(\mathbf{x}_\alpha,\mathbf{Q}\mathbf{x}_\alpha)\approx 0, \qquad \alpha=1,...,N. \qquad (14)$$

Once such a $\mathbf{Q}$ is obtained (the computation is described in the next section), we can determine the vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ that specify the two planes by Eqs. (13). The distance d of a point (x,y,z) to a plane $Ax+By+Cz+D=0$ is

$$d=\frac{|Ax+By+Cz+D|}{\sqrt{A^2+B^2+C^2}}. \qquad (15)$$

For each point $\mathbf{x}_\alpha$, we compute the distances to the two planes and classify it to the nearer one. The resulting segmentation is fed to the subsequent learning.

The above computation is a special application of the GPCA of Vidal et al. (2005, 2008), which expresses multiple subspaces as one high-dimensional polynomial and classifies points into different subspaces by fitting the high-dimensional polynomial to all the points. Here, we classify points into two affine spaces using the same principle.

6 HYPERSURFACE FITTING

The matrix $\mathbf{Q}$ that satisfies Eq. (14) is computed as follows. In terms of the homogeneous coordinate vector $\mathbf{x}$ defined in Eqs. (9), the equation $(\mathbf{x},\mathbf{Q}\mathbf{x})=0$ for a symmetric matrix $\mathbf{Q}$ defines a quadric surface, describing an ellipsoid, a hyperboloid, an elliptic/hyperbolic paraboloid, or their degeneracy including a pair of planes. We fit a surface $(\mathbf{x},\mathbf{Q}\mathbf{x})=0$ to the points $\mathbf{x}_\alpha$ in 3-D in the same way as we fit a conic (an ellipse, a hyperbola, a parabola, or their degeneracy) to points in 2-D (Kanatani and Sugaya, 2007). If we define 9-D vectors $\mathbf{z}_\alpha$ and $\mathbf{u}$ by

$$\begin{aligned}\mathbf{z}_\alpha&=(x_\alpha^2,y_\alpha^2,z_\alpha^2,2y_\alpha z_\alpha,2z_\alpha x_\alpha,2x_\alpha y_\alpha,2x_\alpha,2y_\alpha,2z_\alpha)^\top,\\ \mathbf{v}&=(Q_{11},Q_{22},Q_{33},Q_{23},Q_{31},Q_{12},Q_{41},Q_{42},Q_{43})^\top, \quad (16)\end{aligned}$$

Eq. (14) is rewritten as

$$(\mathbf{z}_\alpha,\mathbf{v})+Q_{44}\approx 0, \qquad \alpha=1,...,N. \qquad (17)$$

A well known method for computing such $\mathbf{v}$ and Q_{44} is the method of Taubin (1991), which is known to be highly accurate as compared with naive least squares (Kanatani, 2008; Kanatani and Sugaya 2007). Theoretically, ML (Maximum Likelihood) achieves higher accuracy (Kanatani, 2008; Kanatani and Sugaya, 2007), but the surface $(\mathbf{x},\mathbf{Q}\mathbf{x})=0$ that degenerates into two planes has singularities along their intersection. We have observed that iterations for ML fail to converge when some data points are near the singularities; the corresponding denominators diverge and become ∞ if they coincide with singularities[2].

The Taubin method in this case goes as follows. Assume that x_α, y_α, and z_α are perturbed by Gaussian noise Δx_α, Δy_α, and Δz_α, respectively, of mean 0 and standard deviation σ. Let $\Delta\mathbf{z}_\alpha$ be the perturbation of $\mathbf{z}_\alpha$ in Eqs. (16). By first order expansion, we have

$$\begin{aligned}\Delta\mathbf{z}_\alpha=(&2x_\alpha\Delta x_\alpha,2y_\alpha\Delta y_\alpha,2z_\alpha\Delta z_\alpha,2\Delta y_\alpha z_\alpha+2y_\alpha\Delta z_\alpha,\\ &...,2\Delta z_\alpha)^\top, \qquad (18)\end{aligned}$$

from which we can evaluate the covariance matrix $V[\mathbf{z}_\alpha]=E[\Delta\mathbf{z}_\alpha\Delta\mathbf{z}_\alpha^\top]$ of $\mathbf{z}_\alpha$. Noting the relations $E[\Delta x_\alpha]=E[\Delta y_\alpha]=E[\Delta z_\alpha]=0$, $E[\Delta y_\alpha\Delta z_\alpha]=E[\Delta z_\alpha\Delta x_\alpha]=E[\Delta x_\alpha\Delta y_\alpha]=0$, and $E[\Delta x_\alpha^2]=E[\Delta y_\alpha^2]=E[\Delta z_\alpha^2]=\sigma^2$, we obtain $V[\mathbf{z}_\alpha]=\sigma^2V_0[\mathbf{z}_\alpha]$, where

$$V_0[\mathbf{z}_\alpha]=\begin{pmatrix} x_\alpha^2 & 0 & 0 & 0 & z_\alpha x_\alpha & x_\alpha y_\alpha & x_\alpha & 0 & 0\\ * & y_\alpha^2 & 0 & y_\alpha z_\alpha & 0 & x_\alpha y_\alpha & 0 & y_\alpha & 0\\ * & * & z_\alpha^2 & y_\alpha z_\alpha & z_\alpha x_\alpha & 0 & 0 & 0 & z_\alpha\\ * & * & * & y_\alpha^2+z_\alpha^2 & x_\alpha y_\alpha & z_\alpha x_\alpha & 0 & z_\alpha & y_\alpha\\ * & * & * & * & z_\alpha^2+x_\alpha^2 & y_\alpha z_\alpha & z_\alpha & 0 & x_\alpha\\ * & * & * & * & * & x_\alpha^2+y_\alpha^2 & y_\alpha & x_\alpha & 0\\ * & * & * & * & * & * & 1 & 0 & 0\\ * & * & * & * & * & * & * & 1 & 0\\ * & * & * & * & * & * & * & * & 1 \end{pmatrix}. \quad (19)$$

[2] ML minimizes the sum of the distances, measured in the direction of the surface normals, to the surface, but no surface normals can be defined at singularities.

Here, $*$ means copying the element in the symmetric position. The Taubin method minimizes

$$J_{\rm T} = \frac{\sum_{\alpha=1}^{N}\Big((\mathbf{z}_\alpha, \mathbf{v}) + Q_{44}\Big)^2}{\sum_{\alpha=1}^{N}(\mathbf{v}, V_0[\mathbf{z}_\alpha]\mathbf{v})}. \tag{20}$$

If the denominator is omitted, this becomes the naive least squares, but the existence of the denominator is crucial for improving the accuracy as we show later. The solution $\{\mathbf{v}, Q_{44}\}$ that minimizes Eq. (20) is obtained as follows (Kanatani and Sugaya, 2007):

1. Compute the centroid $\mathbf{z}_C$ of $\{\mathbf{z}_\alpha\}$ and the deviations $\tilde{\mathbf{z}}_\alpha$ from it:

$$\mathbf{z}_C = \frac{1}{N}\sum_{\alpha=1}^{N}\mathbf{z}_\alpha, \qquad \tilde{\mathbf{z}}_\alpha = \mathbf{z}_\alpha - \mathbf{z}_C. \tag{21}$$

2. Compute the following 9×9 matrices:

$$\mathbf{M}_{\rm T} = \sum_{\alpha=1}^{N}\tilde{\mathbf{z}}_\alpha\tilde{\mathbf{z}}_\alpha^\top, \qquad \mathbf{N}_{\rm T} = \sum_{\alpha=1}^{N}V_0[\mathbf{z}_\alpha]. \tag{22}$$

3. Solve the generalized eigenvalue problem

$$\mathbf{M}_{\rm T}\mathbf{v} = \lambda\mathbf{N}_{\rm T}\mathbf{v}, \tag{23}$$

and compute the unit generalized eigenvector $\mathbf{v}$ for the smallest generalized eigenvalue λ.

4. Compute Q_{44} as follows:

$$Q_{44} = -(\mathbf{z}_C, \mathbf{v}). \tag{24}$$

7 MULTISTAGE LEARNING

After an initial segmentation is obtained, we fit affine spaces by the EM algorithm in successively higher dimensions:

1. Two parallel panes in 3-D.
2. Two 2-D affine spaces in 5-D.
3. Two 3-D affine spaces in 7-D.

If the object and the background are in translational motions, an optimal solution is obtained in the first stage, and it is still optimal in the second and the third stages. If the object and the background undergo planar motions with rotations, an optimal solution is obtained in the second stage, and it is still optimal in the third. If the object and the background are in general 3-D motions, an optimal solution is obtained in the third stage. Because a degenerate motion is a special case of general motions, an optimal solution for a degenerate motion is unchanged when optimized by assuming a more general motion. This is the basic principle of MSL of Sugaya and Kanatani (2004).

The EM algorithm for classifying n-D points $\mathbf{r}_\alpha$, α = 1, ..., N, into two d-D affine spaces ($n \geq 2d+1$) is as follows:

1. Using the initial classification, define the membership weight $W_\alpha^{(k)}$ of $\mathbf{r}_\alpha$ to class k (= 1, 2) as follows

$$W_\alpha^{(k)} = \begin{cases} 1 & \text{if } \mathbf{r}_\alpha \text{ belongs to class } k \\ 0 & \text{otherwise} \end{cases}. \tag{25}$$

2. For each class k (= 1, 2), do the following computation:

   (a) Compute the prior $w^{(k)}$ of class k as follows.

$$w^{(k)} = \frac{1}{N}\sum_{\alpha=1}^{N}W_\alpha^{(k)}. \tag{26}$$

   (b) If $w^{(k)} \leq d/N$, stop (the number of points is too small to span a d-D affine space).

   (c) Compute the centroid $\mathbf{r}_C^{(k)}$ of class k:

$$\mathbf{r}_C^{(k)} = \frac{\sum_{\alpha=1}^{N}W_\alpha^{(k)}\mathbf{r}_\alpha}{\sum_{\alpha=1}^{N}W_\alpha^{(k)}}. \tag{27}$$

   (d) Compute the moment $\mathbf{M}^{(k)}$ of class k:

$$\mathbf{M}^{(k)} = \frac{\sum_{\alpha=1}^{N}W_\alpha^{(k)}(\mathbf{r}_\alpha - \mathbf{r}_C^{(k)})(\mathbf{r}_\alpha - \mathbf{r}_C^{(k)})^\top}{\sum_{\alpha=1}^{N}W_\alpha^{(k)}}. \tag{28}$$

   Let $\lambda_1^{(k)} \geq \cdots \geq \lambda_n^{(k)}$ be the n eigenvalues of $\mathbf{M}^{(k)}$, and $\mathbf{u}_1^{(k)}$, ..., $\mathbf{u}_n^{(k)}$ the corresponding unit eigenvectors.

   (e) Compute the "inward" projection matrix $\mathbf{P}^{(k)}$ onto class k and the "outward" projection matrix $\mathbf{P}_\perp^{(k)}$ onto the space orthogonal to it by

$$\mathbf{P}^{(k)} = \sum_{i=1}^{d}\mathbf{u}_i^{(k)}\mathbf{u}_i^{(k)\top}, \quad \mathbf{P}_\perp^{(k)} = \mathbf{I} - \mathbf{P}^{(k)}. \tag{29}$$

3. Estimate the square noise level σ^2 from the square sum of the "outward" noise components in the form

$$\hat{\sigma}^2 = \min[\frac{N}{(n-d)(N-d-1)}{\rm tr}(w^{(1)}\mathbf{P}_\perp^{(1)}\mathbf{M}^{(1)}\mathbf{P}_\perp^{(1)} + w^{(2)}\mathbf{P}_\perp^{(2)}\mathbf{M}^{(2)}\mathbf{P}_\perp^{(2)}), \sigma_{\rm min}^2], \tag{30}$$

where tr denotes the trace, and $\sigma_{\rm min}$ is a small number, say 0.1 pixels, to prevent $\hat{\sigma}^2$ from becoming exactly 0, which would cause computational failure in the subsequent computation, The number $(n-d)(N-d-1)$ accounts for the degree of freedom of the χ^2-distribution of the square sum of the "outward" noise components (Kanatani, 1996).

4. Compute the covariance matrix $\mathbf{V}^{(k)}$ of class k (= 1, 2) as follows:
$$\mathbf{V}^{(k)} = \mathbf{P}^{(k)}\mathbf{M}^{(k)}\mathbf{P}^{(k)} + \hat{\sigma}^2\mathbf{P}_\perp^{(k)}. \qquad (31)$$
The first term on the right-hand side is for the data variations within the affine space; the second accounts for the "outward" noise components.
5. Do the following computation for each point $\mathbf{r}_\alpha$, α = 1, ..., N:
   (a) Compute the conditional likelihood $P(\alpha|k)$, k = 1, 2, of $\mathbf{r}_\alpha$ by
$$P(\alpha|k) = \frac{e^{-(\mathbf{r}_\alpha - \mathbf{r}_C^{(k)}, \mathbf{V}^{(k)-1}(\mathbf{r}_\alpha - \mathbf{r}_C^{(k)}))/2}}{\sqrt{\det \mathbf{V}^{(k)}}}. \qquad (32)$$
   (b) Update the membership weight $W_\alpha^{(k)}$, k = 1, 2, of $\mathbf{r}_\alpha$ as follows:
$$W_\alpha^{(k)} = \frac{w^{(k)}P(\alpha|k)}{w^{(1)}P(\alpha|1) + w^{(2)}P(\alpha|2)}. \qquad (33)$$
6. Go back to Step 2 and iterate the computation until $\{W_\alpha^{(k)}\}$ converges.
7. After convergence (or interruption), classify each $\mathbf{r}_\alpha$ to the class k for which $W_\alpha^{(k)}$, k = 1, 2, is larger.

If we let n = 5 and d = 2, the above procedure is the second stage of the multistage learning, and if we let n = 7 and d = 3, it is the third stage. The first stage requires an additional constraint that the two planes be parallel. For this, we let n = 3 and d = 2 and compute from the two matrices $\mathbf{M}^{(k)}$, k = 1, 2, their weighted average
$$\mathbf{M} = w^{(1)}\mathbf{M}^{(1)} + w^{(2)}\mathbf{M}^{(2)}. \qquad (34)$$
Let $\lambda_1 \geq \cdots \geq \lambda_n$ be its n eigenvalues, and $\mathbf{u}_1$, ..., $\mathbf{u}_n$ the corresponding unit eigenvectors. We let the projection matrices $\mathbf{P}^{(k)}$ and $\mathbf{P}_\perp^{(k)}$ coincide in the form $\mathbf{P}^{(1)}$ = $\mathbf{P}^{(2)}$ = $\mathbf{P}$ and $\mathbf{P}_\perp^{(1)}$ = $\mathbf{P}_\perp^{(2)}$ = $\mathbf{P}_\perp$, where
$$\mathbf{P} = \sum_{i=1}^{d}\mathbf{u}_i\mathbf{u}_i^\top, \qquad \mathbf{P}_\perp = \mathbf{I} - \mathbf{P}. \qquad (35)$$
The estimation of the square noise level σ^2 in Step 3 is replaced by
$$\hat{\sigma}^2 = \min[\frac{N}{(n-d)(N-d-2)}\mathrm{tr}(\mathbf{P}_\perp\mathbf{M}\mathbf{P}_\perp), \sigma^2_{\min}]. \qquad (36)$$
The rest is unchanged.

However, there is an inherent problem in EM-based learning: If there is no noise, its distribution cannot be stably estimated. This causes no problem in real situations but may result in computational failure when ideal data are used for a testing purpose. This phenomenon was reported by Tron and Vidal (2007) for MSL. In the above procedure, this occurs when points are exactly in a 2-D affine space in 7-D, in which case the covariance matrix degenerates to have rank 2 and hence the likelihood cannot be defined: To define $P(\alpha|k)$, the matrix $\mathbf{V}^{(k)}$ in Eq. (31) must have rank n, and $\det\mathbf{V}^{(k)}$ in the denominator of Eq. (32) must be positive. To cope with this, our system checks if such a degeneracy exists by using the geometric AIC (Kanatani, 1996), and if so judged, the 3-D affine space is replaced by a 2-D affine space (we omit the details). Such a treatment does not affect the performance when real data are used.

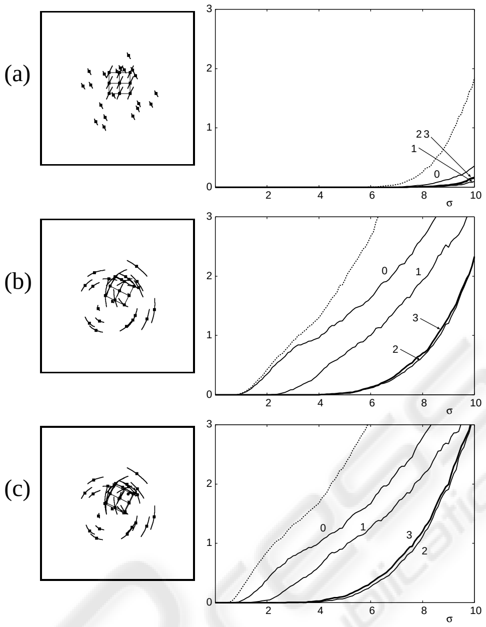

Figure 2: Left column: 20 background points and 14 object points. (a) Translational motion. (b) Planar motion. (c) General 3-D motion. Right column: Average misclassification ratio over 5000 trials. The horizontal axis is for the standard deviation σ of added noise. 0) Initial segmentation by the Taubin method. 1) Parallel plane fitting in 3-D. 2) 2-D affine space fitting in 5-D. 3) 3-D affine space fitting in 7-D. The dotted lines are for initial segmentation by least squares.

8 EXPERIMENTS

8.1 Simulation

The left column of Fig. 2 shows simulated 512 × 512-pixel images of 14 object points and 20 background points in (a) translational motion, (b) planar motion, and (c) general 3-D motion. These are the 5th of 10 frames; the curves in them are trajectories over the

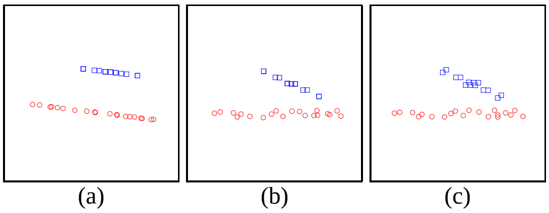

Figure 3: 3-D visualization of image motions in Fig. 2.

10 frames. We added Gaussian noise of mean 0 and standard deviation σ to the x and y coordinates of each point in each frame independently, and evaluated the average misclassification ratio over 5000 independent trials for each σ. The result is shown in the right column. The plots 0 – 3 correspond to the initial segmentation by the Taubin method, parallel plane fitting in 3-D, 2-D affine space fitting in 5-D, and 3-D affine space fitting in 7-D, respectively. For comparison, we plot in dotted lines the initial segmentation we would obtain if naive least squares were used.

We can observe that for the translational motion (a), the initial segmentation is already correct enough; an almost complete segmentation is obtained in the first stage. For the planar motion (b), we obtain an almost correct segmentation in the second stage, and for the general 3-D motion in the third. We can also confirm that the Taubin method (plots 0) for initial segmentation is more accurate than the naive least squares (dotted lines).

Figure 3 shows motion trajectories compressed to 3-D by Eq. (13) ($d = 3$) viewed from a particular angle. For the translational motion (a), all the points belong to two parallel planes, as predicted. For the planar motion (b) and the general 3-D motion (c), the points still belong to nearly parallel and nearly planar surfaces. This fact explains the high performance of our Taubin initial segmentation.

8.2 Real Video Experiments

The upper row of Fig. 4 shows six videos from the Hopkins155 database[3] (Tron and Vidal, 2007). The lower row shows our 3-D visualization of the trajectories. Table 1 lists the correct classification ratios at each stage of our method[4] and some others: the MSL of Sugaya and Kanatani[5] (2004); the method of Vidal et al.[6] (Vidal et al., 2005); RANSAC[5]; the method of Yan and Pollefeys[5] (2006). We can see that for all the videos, our method reach high classification ratios in relatively early stages and 100% in the end, while other methods do not necessarily achieve 100%. This is because we focus on the motion type and take degeneracies into account, while other methods do not pay so much attention to them. As the bottom row of Fig. 4 shows, even when the visible motions look complicated, it is common for the trajectories to be in nearly parallel planes. The high performance of our method is based on this observation.

Table 1: Correct classification ratios (%) for the data in Fig. 4 in each stage of our method, and comparisons with other methods: MSL of Sugaya and Kanatani (2004), Vidal et al. (2005, 2008), RANSAC, and Yan and Pollefeys (2006).

	(a)	(b)	(c)	(d)	(e)	(f)
Initial	88.8	99.1	98.0	100.0	100.0	98.6
1st stage	99.7	99.6	100.0	100.0	100.0	100.0
2nd stage	98.8	99.6	100.0	100.0	100.0	100.0
3rd stage	100.0	100.0	100.0	100.0	100.0	100.0
MSL	99.7	99.6	100.0	100.0	100.0	100.0
Vidal et al.	88.2	99.6	99.2	99.4	100.0	100.0
RANSAC	91.8	99.6	96.6	97.5	100.0	100.0
Yan-Pollefeys	98.5	98.2	97.4	94.3	99.8	80.8

9 CONCLUSIONS

We presented an improved version of the MSL of Sugaya and Kanatani (2004). First, we replaced their initial segmentation based on heuristic clustering using the interaction matrix of Costeira and Kanade (1998) and the geometric AIC (Kanatani, 1996) by an analytical computation based on the GPCA of Vidal et al. (2005, 2008), fitting two 2-D affine spaces in 3-D by the method of Taubin (1991). The resulting initial segmentation alone can segment most of the motions we frequently encounter in natural scenes fairly correctly, and the result is successively optimized by the EM algorithm in 3-D, 5-D, and 7-D. Using simulated and real videos, we demonstrated that our method behaves as predicted and illustrated the mechanism underneath using our visualization technique. This is a big contrast to all existing methods, whose behavior is difficult to predict unless tested using a particular database.

ACKNOWLEDGEMENTS

The authors thank Réne Vidal of Johns Hopkins University for helpful discussions. This work was supported in part by the Ministry of Education, Culture, Sports, Science, and Technology, Japan, under Grant in Aid for Scientific Research (C 21500172) and for Young Scientists (B 18700181).

[3] http://www.vision.jhu.edu/data/hopkins155

[4] http://www.iim.cs.tut.ac.jp/~sugaya/public-e.html

[5] The code is at the cite in the footnote 4.

[6] We used the code placed at the cite in footnote 3.

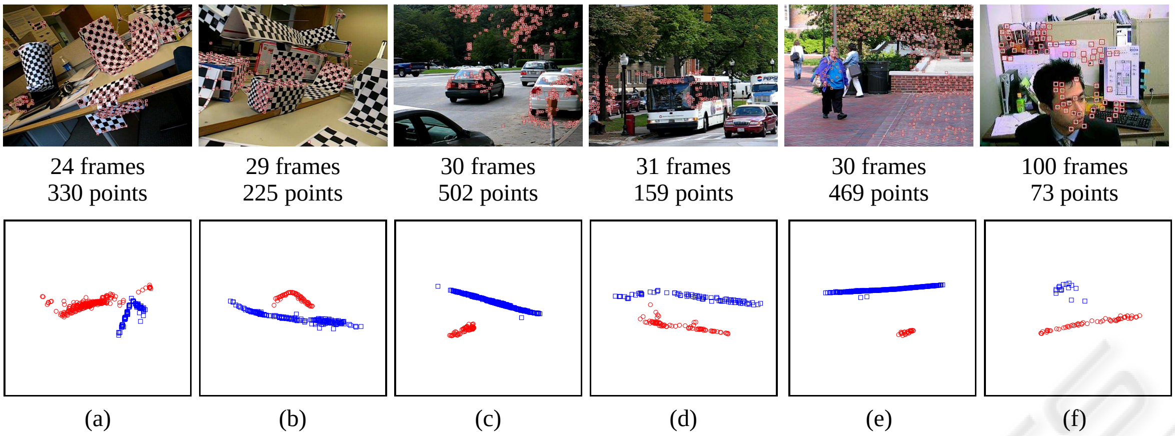

Figure 4: Top: Feature points detected from 6 video streams of the Hopkins155 database. Bottom: Their their 3-D representation.

REFERENCES

Costeira, J. P. and Kanade, T. (1998). A multibody factorization method for independently moving objects, *Int. J. Computer Vision*, 29, 159–179.

Fan, Z., Zhou, J. and Wu, Y. (2006). Multibody grouping by inference of multiple subspace from high-dimensional data using oriented-frames, *IEEE Trans Patt. Anal. Mach. Intell.*, 28, 91–105.

Gear, C. W. (1998). Multibody grouping from motion images, *Int. J. Comput. Vision*, 29, 133–150.

Ichimura, N. (1999). Motion segmentation based on factorization method and discriminant criterion, *Proc. 7th Int. Conf. Comput. Vis.*, Vol. 1, Kerkyra, Greece, 600–605.

Ichimura, N. (2000). Motion segmentation using feature selection and subspace method based on shape space, *Proc. 15th Int. Conf. Pattern Recog.*, Vol. 3, Barcelona, 858–864.

Inoue, K. and Urahama, K. (2001). Separation of multiple objects in motion images by clustering, *Proc. 8th Int. Conf. Comput. Vis.*, Vol. 1, Vancouver, 219–224.

Kanatani, K. (2002a). Evaluation and selection of models for motion segmentation, *Proc. 7th Euro. Conf. Comput. Vis.*, Vol. 3, Copenhagen, 335–349.

Kanatani, K. (2001). Motion segmentation by subspace separation and model selection, *Proc. 8th Int. Conf. Comput. Vis.*, Vol. 2, Vancouver, 301–306.

Kanatani, K. (2002b). Motion segmentation by subspace separation: Model selection and reliability evaluation, *Int. J. Image Graphics*, 2, 179–197.

Kanatani, K. (1996). *Statistical Optimization for Geometric Computation: Theory and Practice*, Amsterdam: Elsevier. Reprinted (2005) New York: Dover.

Kanatani, K. (2008). Statistical optimization for geometric fitting: Theoretical accuracy analysis and high order error analysis, *Int. J. Comput. Vision*, 80, 167–188.

Kanatani, K. and Sugaya, Y. (2007). Performance evaluation of iterative geometric fitting algorithms, *Comp. Stat. Data Anal.*, 52, 1208–1222.

Poelman, C. J. and Kanade, T. (1997). A paraperspective factorization method for shape and motion recovery, *IEEE Trans. Pattern Anal. Mach. Intell.*, 19, 206–218.

Rao, S. R., Tron, R., Vidal R. and Ma, Y. (2008). Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Anchorage, AK.

Schindler, K., Suter, D. and Wang, H. A model-selection framework for multibody structure-and-motion of image sequences, *Int. J. Comput. Vision*, 79, 159–177.

Sugaya, Y. and Kanatani, Y. (2004). Multi-stage optimization for multi-body motion segmentation. *IEICE Trans. Inf. & Syst.*, E87-D, 1935–1942.

Taubin, G. (1991). Estimation of planer curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Trans. Patt. Anal. Mach. Intell.*, 13, 1115–1138.

Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography—A factorization method, *Int. J. Comput. Vision*, 9, 137–154.

Tron, R. and Vidal, R. (2007). A benchmark for the comparison of 3-D motion segmentation algorithms, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Minneapolis, MN.

Vidal, R., Ma, Y. and Sastry, S. (2005). Generalized principal component analysis (GPCA), *IEEE Trans. Patt. Anal. Mach. Intell.*, 27, 1945–1959.

Vidal, R. Tron, R. and Hartley, R. (2008). Multiframe motion segmentation with missing data using PowerFactorization and GPCA, *Int. J. Comput. Vision*, 79, 85–105.

Wu, Y. Zhang, Z., Huang, T. S. and Lin, J. Y. (2001). Multibody grouping via orthogonal subspace decomposition, sequences under affine projection, *Proc. IEEE Conf. Computer Vision Pattern Recog.*, Vol. 2, Kauai, HI, 695–701.

Yan, J. and Pollefeys, M. (2006). A general framework for motion segmentation: Independent, articulate, rigid, non-rigid, degenerate and nondegenerate, *Proc. 9th Euro. Conf. Comput. Vision.*, Vol. 4, Graz, Austria, 94–104.