Voting-based Approach to Nullspace Search for Correspondence Matching and Shape Recovery

Kazuhiko Kawamoto[1], Atsushi Imiya[2,3], and Kaoru Hirota[1]

[1]Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan

[2]Media Technology Division, IMIT, Chiba University
1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan

[3]Software Research Division, National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

e-mail: kawa@hrt.dis.titech.ac.jp

Abstract— **Two problems of correspondence matching among the images in a sequence and shape recovery from the image sequence are dealt with. To solve these two problems, grouping and model fitting in a spatio-temporal domain are required. For the achievement of the two tasks, a voting-based method is proposed. In general, voting provides an unified framework for grouping and model fitting from a bottom-up point of view. The aim of this work is to construct a framework for correspondence matching and shape recovery from a sequence of images using voting. Furthermore, in order to derive an algorithm for solving the two problems, the fomulation, called *nullspace search*, is given from a algebraic point of view. The algorithm is demonstrated on synthetic and real image sequences.**

Keywords— **Correspondence Matching, Shape Recovery, Voting, Hough Transform.**

I. Introduction

Grouping and model fitting play an important role in inductive or bottom-up reasoning in a variety of fields such as artificial intelligence, pattern recognition and computer vision. These two processes produce compact descriptions that emphasize relevant distribution pattern structures. The compact descriptions enable us to perform reasoning at higher levels than that at the raw data level. For example, polyhedral approximation of objects obtained by fitting straight line segments to a set of edge points in an image is used for object identification and three-dimensional scene interpretation [1]. From a bottom-up point of view, it is very important to construct a framework for achievement of grouping and model fitting [2].

Three-dimensional shape recovery from a sequence of images involves grouping and model fitting; *grouping* is matching of corresponding features such as planar points and lines among the images in a sequence, and *model fitting* is parameter estimation of geometric features such as spatial points and lines that represent a three-dimensional shape [3]. In most of the proposed methods for shape recovery, correspondence matching among the images in a sequence is performed as a pre-processing using correlation [4][5] or assumed to be done [6][7]. However, since correlation matching is based on similar appearance in the images in a sequence, it does not work well for drastic appearance changes between two consecutive images.

This paper formulates the two problems of correspondence matching and shape recovery as *nullspace search*, and proposes a voting-based method for nullspace search. The formulation converts these two problems into searching the basic vectors spanning the nullspaces that represent geometric features such as spatial points and lines. Mathematically, the formulation is equivalent to solving a set of simultaneous equations

$$\boldsymbol{A}_i \boldsymbol{x}_i = \boldsymbol{0}, \quad i = 1, \ldots, m. \tag{1}$$

The reason why the formulation is called nullspace search is that, in addition to $\boldsymbol{x}_i$, the number of simultaneous equations m and the coefficient matrices $\boldsymbol{A}_i$, $i = 1, 2, \ldots, m$ must be determined because both are also unknown. Our voting method performs the searching and determination of the solutions to eq. (1). In general, voting repeatedly generates relevant model parameters from randomly sampled data as a series of hypotheses, and finally produces the solutions supported by a large number of the hypotheses. This process enables us to remove mismatching pairs of perspective projections among the images in a sequence even if the sequence has appearance changes. In pattern recognition, the idea can be traced to Hough's work in 1962 for the detection of straight line segments on an images [8], which is called the Hough transform [9][10][11]. The formulation and method can be applied to many grouping and model fitting problems in pattern recognition and computer vision, and hence the approach provides an unified approach for bottom-up methodology.

Section II reviews multiple view geometry of points and lines, and then provides linear equations describing the perspective projection process. Section III formulates correspondence matching and shape recovery from a sequence of images using linear algebra. The formulation is called nullspace search. Section IV presents a voting-based algorithm for nullspace search. Section V demonstrates the performance of our algorithm with synthetic and real image sequences.

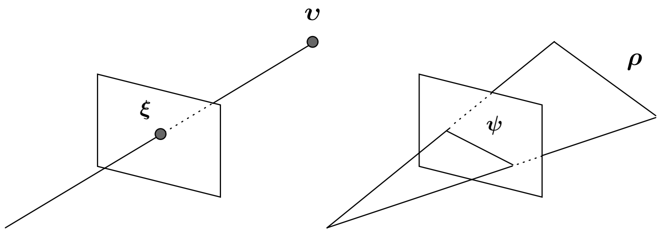


Fig. 1. Perspective projection of a point (left) and a line (right).

II. Multiple View Geometry of Points and Lines

The relationship between spatial points and their perspective projections as well as spatial lines onto images is reviewed. The multiple view geometry derive linear equations for the relationship. The expressions derived in this section are used in formulating correspondence matching and shape recovery in Section III.

A. Points

Let $\boldsymbol{v} = (X, Y, Z, W)^\top$ be a three-dimensional point and $\boldsymbol{\xi} = (x, y, z)^\top$ be a perspective projection of $\boldsymbol{v}$ onto an image in homogeneous coordinates. Fig. 1 (left) shows the configuration between $\boldsymbol{v}$ and $\boldsymbol{\xi}$. The relationship between a three-dimensional point $\boldsymbol{v}$ and its perspective projection $\boldsymbol{\xi}$ can be written as

$$\lambda \boldsymbol{\xi} = \boldsymbol{P}\boldsymbol{v}, \tag{2}$$

where λ is an arbitrary nonzero scalar and $\boldsymbol{P}$ is a 3×4 matrix called a perspective projection matrix [12]. By eliminating a scale factor λ in eq. (2), a pair of linear equations in $\boldsymbol{v}$

$$\begin{cases} (x\boldsymbol{p}_3 - z\boldsymbol{p}_1)^\top \boldsymbol{v} = 0, \\ (y\boldsymbol{p}_3 - z\boldsymbol{p}_2)^\top \boldsymbol{v} = 0, \end{cases} \tag{3}$$

is obtained, where $\boldsymbol{p}_i^\top, i = 1, 2, 3$, are the rows of the matrix $\boldsymbol{P}$. If at least two projections of the same three-dimensional point are observed, a system of equations in $\boldsymbol{v}$ can be uniquely solved except for degenerate cases.

B. Lines

Setting $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ to be two distinct points on an image plane, the two-dimensional line passing through these two points is calculated by

$$\lambda \boldsymbol{\psi} = \boldsymbol{\xi}_1 \times \boldsymbol{\xi}_2 \tag{4}$$

up to a scale factor. This equation and eq. (2) lead to the following relationship:

$$\lambda \boldsymbol{\psi} = \boldsymbol{P}\boldsymbol{v}_1 \times \boldsymbol{P}\boldsymbol{v}_2 = \begin{bmatrix} (\boldsymbol{p}_2 \wedge \boldsymbol{p}_3)^\top \\ (\boldsymbol{p}_3 \wedge \boldsymbol{p}_1)^\top \\ (\boldsymbol{p}_1 \wedge \boldsymbol{p}_2)^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \wedge \boldsymbol{v}_2 \end{bmatrix}, \tag{5}$$

where $\wedge$ is the exterior product [13][14] (see, Appendix). In eq. (5), a 6×1 vector $\boldsymbol{\rho} = \boldsymbol{v}_1 \wedge \boldsymbol{v}_2$ expresses the three-dimensional line passing through the two three-dimensional points $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ [13][14]. The coordinates of $\boldsymbol{\rho}$ are called the *plücker coordinates* of the three-dimensional line. Equation (5) is rewritten as

$$\lambda \boldsymbol{\psi} = \boldsymbol{P}_l \boldsymbol{\rho}, \tag{6}$$

where $\boldsymbol{P}_l$ is the 3×6 matrix defined by

$$\boldsymbol{P}_l = \begin{bmatrix} (\boldsymbol{p}_2 \wedge \boldsymbol{p}_3)^\top \\ (\boldsymbol{p}_3 \wedge \boldsymbol{p}_1)^\top \\ (\boldsymbol{p}_1 \wedge \boldsymbol{p}_2)^\top \end{bmatrix}. \tag{7}$$

Since eq. (6) describes the relationship between a three-dimensional line and its perspective projection, $\boldsymbol{P}_l$ models perspective projection of lines. Fig. 1 (right) shows the configuration between $\boldsymbol{\rho}$ and $\boldsymbol{\psi}$. By eliminating a scale factor λ in eq. (6), a pair of linear equations in $\boldsymbol{\rho}$

$$\begin{cases} (a\,\boldsymbol{p}_1 \wedge \boldsymbol{p}_2 - c\,\boldsymbol{p}_2 \wedge \boldsymbol{p}_3)^\top \boldsymbol{\rho} = 0, \\ (b\,\boldsymbol{p}_1 \wedge \boldsymbol{p}_2 - c\,\boldsymbol{p}_3 \wedge \boldsymbol{p}_1)^\top \boldsymbol{\rho} = 0, \end{cases} \tag{8}$$

is obtained, where $\boldsymbol{\psi} = (a, b, c)^\top$ expresses a straight line on an image plane. If at least three projections of the same three-dimensional line are observed, a system of equations in $\boldsymbol{\rho}$ can be uniquely solved except for degenerate cases.

III. Problem Formulation

In this section, correspondence matching and shape recovery from a sequence of images is formulated as nullspace search using linear algebra. The formulation converts these two problems into searching the basic vectors spanning the nullspaces that represent spatial points and lines from their perspective projections.

A. Shape Recovery

Each perspective projection of a three-dimensional point $\boldsymbol{v}$ provides a pair of linear equations in eq. (3). Such pairs of linear equations in $\boldsymbol{v}$ lead to a homogeneous system of linear equations $\boldsymbol{\Xi}\boldsymbol{v} = \boldsymbol{0}$. If m three-dimensional points $\boldsymbol{v}_i$, $i = 1, \ldots, m$ are observed from n cameras , m homogeneous systems of linear equations

$$\boldsymbol{\Xi}_i \boldsymbol{v}_i = 0, \quad i = 1, \ldots, m. \tag{9}$$

are obtained. Also, a collection of perspective projections of the same three-dimensional line gives a homogeneous system of linear equations $\boldsymbol{\Psi}\boldsymbol{\rho} = \boldsymbol{0}$. If m three-dimensional lines $\boldsymbol{\rho}_i$, $i = 1, \ldots, m$ are observed from n cameras, m homogeneous systems of linear equations

$$\boldsymbol{\Psi}_i \boldsymbol{\rho}_i = 0, \quad i = 1, \ldots, m. \tag{10}$$

are obtained. The solutions to eqs. (9) and (10) provide the positions of three-dimensional points and lines, respectively. To avoid the trivial solutions $\boldsymbol{v}_i = \boldsymbol{0}$ and $\boldsymbol{\rho}_i = \boldsymbol{0}$, the coefficient matrices $\boldsymbol{\Xi}_i$ and $\boldsymbol{\Psi}_i$ are rank-deficient, i.e., the ranks of $\boldsymbol{\Xi}_i$ and $\boldsymbol{\Psi}_i$ are at most 3 and 5, respectively. This means that the solutions $\boldsymbol{v}_i$ and $\boldsymbol{\rho}_i$ are in the nullspaces of the coefficient matrices $\boldsymbol{\Xi}_i$ and $\boldsymbol{\Psi}_i$, respectively, that is,

$$\boldsymbol{v}_i \in \mathcal{N}(\boldsymbol{\Xi}_i), \quad \boldsymbol{\rho}_i \in \mathcal{N}(\boldsymbol{\Psi}_i), \tag{11}$$

where $\mathcal{N}(\boldsymbol{A}) = \{\boldsymbol{x} | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\}$. Therefore, the estimation of three-dimensional positions is generally formulated as follows.

Problem 1 *Setting $\boldsymbol{A}_i$ to be a $M \times N$ matrix such that $M > N$, solve an overdetermined set of homogeneous equations*

$$\boldsymbol{A}_i \boldsymbol{x}_i = \boldsymbol{0}, \quad i = 1, \ldots, m. \tag{12}$$

This problem is solved by searching the solution $\boldsymbol{x}_i$ in the nullspace of the matrix $\boldsymbol{A}_i$. Since the vector spanning the nullspace is defined up to a scale factor, the normalization of the length of the vector is required, that is, $||\boldsymbol{x}_j|| = 1$. From this normalization, the vectors to be estimated are distributed on the $(N-1)$-dimensional unit sphere.

B. Correspondence Matching

If corresponding geometric features among the images are not predetermined, the entries of the coefficient matrices $\boldsymbol{\Xi}_i$ and $\boldsymbol{\Psi}_i$ are unknown. As shown in Section II, the determination of $\boldsymbol{\Xi}_i$ and $\boldsymbol{\Psi}_i$ is equivalent to correspondence matching among the images.

The ranks of $\boldsymbol{\Xi}_i$ and $\boldsymbol{\Psi}_i$ are at most 3 and 5, respectively. Except for some degenerate configurations, each solution is in an one-dimensional nullspace, and then

$$\dim \mathcal{N}(\boldsymbol{\Xi}_i) = 1, \quad \dim \mathcal{N}(\boldsymbol{\Psi}_i) = 1. \tag{13}$$

These geometrical properties of $\boldsymbol{\Xi}_i$ and $\boldsymbol{\Psi}_i$ lead to the following problem for correspondence matching.

Problem 2 *Let $\boldsymbol{a}_i$ be a N-dimensional homogeneous vector. From given data $\boldsymbol{a}_i$, $i = 1, \ldots, k$, find $M \times N$ matrices $\boldsymbol{A}_j$, $j = 1, \ldots, m$, $M > N$ such that*

$$\boldsymbol{A}_j^\top = [\boldsymbol{a}_{1(j)} \boldsymbol{a}_{2(j)} \ldots \boldsymbol{a}_{M(j)}], \qquad \text{s.t. } \dim \mathcal{N}(\boldsymbol{A}_\mathrm{j}) = 1. \tag{14}$$

The formulation includes many model detection problems in computer vision, e.g., straight line detection in an image since a subset of sample points which lies on one line has the same form as eqs. (9) and (10).

IV. Voting Method for Nullspace Search

In the formulation in Section III, Problem 1 is an inverse problem, since it is solved by fitting a model to given data points. This problem can be solved by a least-squares method, and then its solution can be uniquely obtained. Problem 2 is also an inverse problem but different from Problem 1. The difference between Problems 1 and 2 is that the solution of Problem 2 is not uniquely determined. There are many combinations for the selection of the vectors $\{\boldsymbol{a}_i\}$ in Problem 2. In this section, a voting method is proposed for solving the two inverse problems.

A. Algorithm

As mentioned above, the nullspaces to be estimated are distributed on the $(N-1)$-dimensional unit sphere. For searching the nullspaces, our voting method repeatedly generates a hypothesis onto the $(N-1)$-dimensional unit sphere, and finally the solutions are accepted by selecting the hypotheses supported by a large number of given data. This hypothesis generation is based on the following proposition.

Proposition 1 *Let $\boldsymbol{A}$ be a $M \times N$ matrix with $M > N$ and $\boldsymbol{B}$ be any $N \times N$ matrix which is obtained by selecting N rows from $\boldsymbol{A}$. If $\mathrm{rank}(\boldsymbol{A}) = \mathrm{N} - 1$, then*

$$\mathrm{rank}(\boldsymbol{B}) = \mathrm{N} - 1, \tag{15}$$

or equivalently

$$\dim(\mathcal{N}(\boldsymbol{B})) = 1, \tag{16}$$

and the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ share a one-dimensional nullspace.

We show an example for this proposition.

Example 1 *Setting $\boldsymbol{A}$ to be a 4×3 matrix such that*

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 3 & 3 & 1 \\ 4 & 4 & 1 \end{pmatrix}, \tag{17}$$

the rank of $\boldsymbol{A}$ is two and the nullspace of $\boldsymbol{A}$ is spanned by a vector $(-1/\sqrt{2}, 1/\sqrt{2}, 0)^\top$. Furthermore, setting $\boldsymbol{B}_i, i = 1, \ldots 4$ to be the 3×3 matrices which are obtained by selecting 3 rows from $\boldsymbol{A}$ as follows

$$\begin{aligned} \boldsymbol{B}_1 &= \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 3 & 3 & 1 \end{bmatrix}, \quad \boldsymbol{B}_2 = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 4 & 4 & 1 \end{bmatrix}, \\ \boldsymbol{B}_3 &= \begin{bmatrix} 1 & 1 & 1 \\ 3 & 3 & 1 \\ 4 & 4 & 1 \end{bmatrix}, \quad \boldsymbol{B}_4 = \begin{bmatrix} 2 & 2 & 1 \\ 3 & 3 & 1 \\ 4 & 4 & 1 \end{bmatrix}, \end{aligned} \tag{18}$$

the rank of $\boldsymbol{B}_i$ is also two and the nullspace of $\boldsymbol{B}_i$ is also spanned by a vector $(-1/\sqrt{2}, 1/\sqrt{2}, 0)^\top$ for $i = 1, \ldots, 4$.

Proposition 1 enables us to generate a hypothesis onto a nullspace on the $(N-1)$-dimensional unit sphere from sampled data points as follows:

Procedure: Hypothesis Generation

1. Randomly select N homogeneous vectors $\boldsymbol{a}_{i(1)}$, $\boldsymbol{a}_{i(2)}$, …, $\boldsymbol{a}_{i(N)}$ from all sample data $\boldsymbol{a}_i$, $i = 1, \ldots, k$.

2. If the matrix $\boldsymbol{B}$ such that $\boldsymbol{B}^\top = [\boldsymbol{a}_{i(1)}, \boldsymbol{a}_{i(2)}, \ldots, \boldsymbol{a}_{i(N)}]$ has a one-dimensional nullspace, vote 1 to the nullspace of $\boldsymbol{B}$.

If this hypothesis generation is iterated until an appropriate number, the nullspaces are estimated by detecting peaks of the votes on the $(N-1)$-dimensional unit sphere. Therefore, the estimation of three-dimensional positions can be solved by this iteration.

In Procedure Nullspace Search, the computation of the nullspaces of given matrices is required. In our implementation, the singular value decomposition (SVD) (see, e.g.,[15]) has been applied. If matrix $\boldsymbol{A}$ has only a zero singular value, then the nullspace of $\boldsymbol{A}$ is spanned by the right singular vector associated with the zero singular value. Therefore, setting $\sigma_1 \geq \ldots \geq \sigma_N$ to be the singular values of the sampled $N \times N$ matrix $\boldsymbol{B}$ and $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$ to be the corresponding right singular vectors, if the relationship $\sigma_1 \geq \dots > \sigma_N = 0$ holds, the nullspace to be generated is the vector $\boldsymbol{v}_N$.

These properties of matrices lead to the following algorithm for nullspace search.

Algorithm: Voting for Nullspace Search

1. Repeat the following steps from Step **2** to Step **6** until a predefined number.

2. Randomly select N homogeneous vectors $\boldsymbol{a}_{i(1)}$, $\boldsymbol{a}_{i(2)}$, …, $\boldsymbol{a}_{i(N)}$ from $\boldsymbol{a}_i$, $i = 1, \dots, k$.

3. Construct the $N \times N$ matrix $\boldsymbol{B}$ such that $\boldsymbol{B}^\top = [\boldsymbol{a}_{i(1)}, \boldsymbol{a}_{i(2)}, \dots, \boldsymbol{a}_{i(N)}]$.

4. Compute the SVD of $\boldsymbol{B}$ and let $\sigma_1 \geq \dots \geq \sigma_N$ be its singular values and $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$ be the corresponding right singular vectors.

5. If the smallest singular value is not equal to 0, that is, $\sigma_1 \geq \dots \geq \sigma_N > 0$, then go to Step **2**.

6. Add 1 to the accumulator of the right singular vector $\boldsymbol{v}_N$ associated with the smallest singular value σ_N.

7. Detect the vectors whose values of the accumulators are larger than a predefined constant.

B. Efficient Algorithm using Multilinear Constraints

Rigid motion leads to multilinear constraints on a set of corresponding perspective projections between two, three and four images [16]. Conversely, non-rigid motion does not provides the constraints and a set of mismatching perspective projections does not obey the constraints. Note that some of mismatching perspective projections satisfy the constraints because they are a necessary condition. Therefore the multilinear constraints can be used to reduce a set of mismatching perspective projections in number. This enables us to reduce the computational cost of the algorithm proposed in the previous subsection. In the following, the bilinear and trilinear constraints are used for points and lines, respectively.

B.1 Bilinear constraint

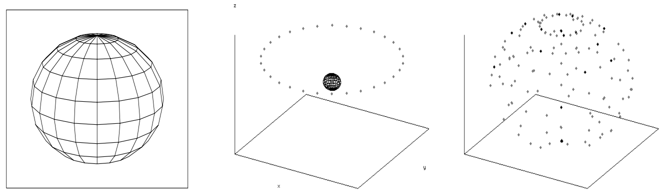
Two perspective projections of the same three-dimensional point lead to a 4×4 matrix containing the entries of perspective projection matrices and the coordinates of the projections. Setting $\boldsymbol{\Xi}_i$ to be the 4×4 matrix, and $\boldsymbol{\xi}_i^j$ and $\boldsymbol{\xi}_{i'}^{j'}$ to be two perspective projections of the same three-dimensional point, if the 4×4 matrix $\boldsymbol{\Xi}_i$ is a singular matrix, the singular condition is written in the following bilinear form

$$\boldsymbol{\xi}_i^{j\top} \boldsymbol{F} \boldsymbol{\xi}_{i'}^{j'} = 0, \tag{19}$$

where $\boldsymbol{F}$ is a fundamental matrix containing only the entries of perspective projection matrices [12]. This means

$$\mathrm{rank}(\Xi_\mathrm{i}) \leq 3 \quad \text{iff } \boldsymbol{\xi}_\mathrm{i}^{\mathrm{j}\top} \boldsymbol{F} \boldsymbol{\xi}_{\mathrm{i}'}^{\mathrm{j}'} = 0. \tag{20}$$

Therefore, if eq. (19) does not holds for a sampled pair of two-dimensional points, the SVD computation at Step 4 in the algorithm can be omitted, i.e., if a sampled pair of points does not satisfy eq. (19), another pair of points is selected.



(a) (b) (c)

Fig. 2. Figures (a) and (b) show an example of input images of "Sphere Object" and 3D configurations between the spherical object and the cameras, respectively. Figure (c) shows the reconstructed result of our algorithm.

Type	Time (s)
Algorithm 1	60.06
Algorithm 2	5.62

TABLE I

The execution time for the spherical object.

B.2 Trilinear constraint

In the case of lines, there are no constraints for perspective projections between two images. Three perspective projections of the same three-dimensional line are required. Setting $\boldsymbol{\Psi}_i$ to be a 6×6 matrix containing a selected triplet of lines on three different images and $\boldsymbol{\psi}_i^j$, $\boldsymbol{\psi}_{i'}^{j'}$ and $\boldsymbol{\psi}_{i''}^{j''}$ to be the selected lines, if the 6×6 matrix $\boldsymbol{\Psi}_i$ is a singular matrix, the singular condition is written in the following trilinear form

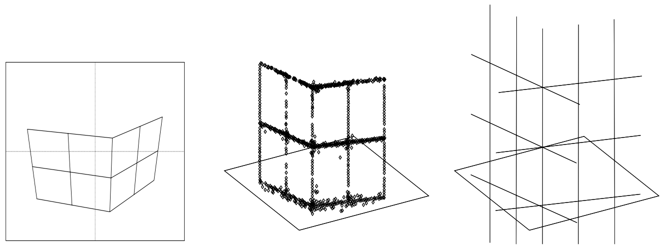
$$\boldsymbol{\psi}_i^j \times \begin{bmatrix} \boldsymbol{\psi}_{i'}^{j'\top} \boldsymbol{T}_1^1 \boldsymbol{\psi}_{i''}^{j''} \\ \boldsymbol{\psi}_{i'}^{j'\top} \boldsymbol{T}_1^2 \boldsymbol{\psi}_{i''}^{j''} \\ \boldsymbol{\psi}_{i'}^{j'\top} \boldsymbol{T}_1^3 \boldsymbol{\psi}_{i''}^{j''} \end{bmatrix} = 0, \tag{21}$$

where $\boldsymbol{T}_1^1$, $\boldsymbol{T}_1^2$ and $\boldsymbol{T}_1^3$ are trifocal tensors containing only the entries of perspective projection matrices. As well as the case of points, the SVD computation for a meaningless triplet of lines can be omitted by checking whether eq. (21) holds or not.

V. Experiments

A. Synthetic Data

The performance of the proposed algorithms is evaluated using two synthetic data "Sphere Object" and "Grid-Object", shown in Figures 2 and 3, both of which are digitized in 256×256 pixels. The spherical object and grid-object are measured from 30 views and 20 views, respectively. For the spherical object, the spatial configuration between the object and the cameras is shown in Figures 2 (b).



(a) (b) (c)

Fig. 3. Figure (a) shows an example of input images of "Grid-Object". Figures (b) and (c) show the reconstructed results of our algorithm for three-dimensional points and three-dimensional lines, respectively.

A.1 Sphere Object:

Grid points on the spherical object in the image sequence are predetected and the homogeneous coordinates of the grid points are given to the algorithm. In this experiment, the number of iterations is set to be 10^6 times and the threshold for detecting peaks in the parameter space is set to be 10.

Figure 2 (c) shows the recovered result of the algorithm. The relative average error between the true and recovered points is 0.207 units. The result shows the algorithm recovers the three-dimensional points on the sphere from the images without knowing a set of corresponding points among the images. Also, since the spherical three-dimensional object produces a similar pattern over an image sequence, it has been difficult to determine correspondences among the images. The algorithm can recover such three-dimensional objects

For the evaluation of the efficiency of the algorithm, that uses the multilinear constraint, two algorithms are tested for the same spherical object. One algorithm does not use the multilinear constraint, as described in Section IV-A, and another algorithm uses the constraint, as explained in Section IV-B. Table I shows the execution time of the two algorithms. In the table, Algorithms 1 and 2 indicate the former and the latter algorithms, respectively. The experiment is done using UltraSPARC-II 297MHz processor. For both algorithms, the number of iterations and thresholds for peak detection are the same as in the above experiment. Table 1 shows that the computation speed of Algorithm 2 is about eleven times faster than that of Algorithm 1, that indicates the efficiency of the computation using multilinear constraint in the algorithm.

A.2 Gird-Object:

In the previous experiment, sparse feature points on the three-dimensional spherical object are recovered. In this experiment, a set of edge points on the images of the three-dimensional grid-object is used as the fed data, that is, the proposed algorithm is tested for a dense data set. Figure 3 (b) shows the reconstructed result of the algorithm. The result shows the algorithm recover the three-dimensional points on the lines of the grid-object. Most existing recovery algorithms are applied to sparse image features such as corners on images. Unlike the existing algorithms, the proposed algorithm works for the case of dense data. This enables us to apply the proposed algorithm to a three-dimensional object on which curve segments appear, because a curve segment consists of a series of points.

Next, straight lines on the grid-object in the image sequence are predetected and the homogeneous coordinates of the straight lines are given to the algorithm. Figure 3 (c) shows the reconstructed result of the algorithm. Unlike the recovery of three-dimensional points, the nullspaces to be estimated are distributed on the five-dimensional unit sphere, since a three-dimensional line is expressed by the six-dimensional homogeneous coordinates. In the previous and these experiments, the results show that the algorithm described in Section IV-A has a possibility to apply for both the 3- and 6-dimensional cases.

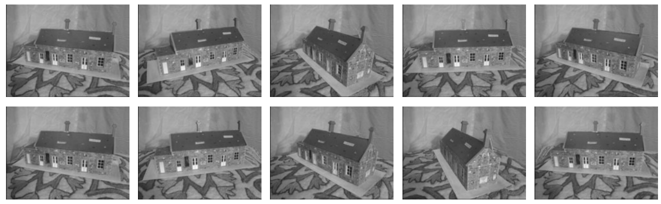
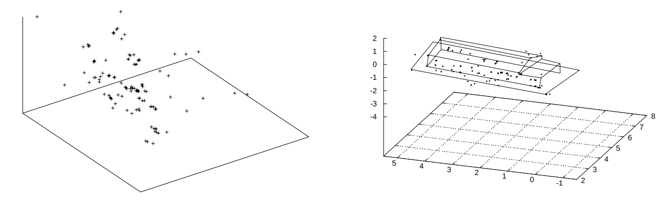


Fig. 4. The figures show the image sequence "Model House".

B. Real Data

The performance of the proposed algorithms is also evaluated using real data "Model House", shown in Figure 4, each of which is digitized in 768×576 pixels. This image sequence is created at Visual Geometry Group, University of Oxford. In the experiment, corner points on the images are predetected using SUSAN corner detector [17], and then a part of the detected points is manually selected because of the removal of the points which do not correspond to actual corners on the images.

In the experiment, the images are given at random, as shown in Figure 4. Since the algorithm can find a set of corresponding points among images in a sequence without knowing the order of the image sequence, the experiment is intended to confirm the characteristics of the algorithm. Figure 5 shows the reconstructed result of the algorithm for the image sequence "Model House". In this experiment, the number of iterations is set to be 10^7 and the threshold for detecting peaks in the accumulator space is set to be 55. In Figure 5 (b), the wireframe model of the three-dimensional house object is superimposed on the reconstructed corner points in Figure 5 (a), to clearly show the relationship between the reconstructed points and the three-dimensional house object. The result shows the algorithm recovers corner points on the house object. Furthermore, since the order of the images is assumed to be unknown, the result shows that the algorithm organizes image data.



(a) (b)

Fig. 5. Figure (a) shows the reconstructed result for the image sequence “Model House”. In Figure (b), the wireframe model of the house object is superimposed on the result.

VI. Conclusions

The two problems of correspondence matching and shape recovery from a sequence of images are addressed. In order to solve these two problems, a voting-based algorithm is proposed. The algorithm repeatedly generates relevant geometric parameters from randomly sampled projections as a series of hypotheses, and finally produces the solutions supported by a large number of the hypotheses. Although the idea of voting is simple and its mechanism requires a large number of iterations, it can solve, in principle, a wide variety of grouping and model fitting problems. This paper tests the potential in the case of three-dimensional shape recovery from a sequence of images, which is a common problem in computer vision. Furthermore, using multilinear constraints, more efficient voting-based algorithm is developed. The use of the multilinear constraints reduces the mismatching of correspondences among the images in a sequence by checking whether or not sampled perspective projections satisfy the constraint equations. Finally, to evaluate the performance of the algorithms, the experiments with synthetic and real image sequences are demonstrated. The results indicate the algorithms are able to recover the three-dimensional objects for the synthetic and real image sequences.

References

[1] K. Sugihara. *Interpretation of Line Drawings.* MIT Press, Cambridge, 1986.

[2] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual infomation.* W. H. Freeman, New York, U.S.A., 1982.

[3] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice.* Elsevier Science, Amsterdam. 1996.

[4] Z. Zhang, R. Deriche, O. Faugeras and Q. -T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, **78**, 87-119, 1995.

[5] M. Okutomiand T. Kanade. A Multiple-Baseline Stereo. *IEEE Trans. on PAMI*, **15**-4, 353-363, 1993.

[6] H. C. Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, **293**, 133-135, 1981.

[7] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: A Factorization Method. *International Journal of Computer Vision*, **9**-2, 137-154, 1992.

[8] P. V. C. Hough. A Method and Means for Recognizing Complex Patterns. U.S. Patent. 3,069,654, 1962.

[9] J. Illingworth and J. Kittler. A Survey of the Hough Transform. *Computer Vision, Graphics, and Image Processing*, **44**-1, 87-116, 1988.

[10] V. F. Leavers. Which Hough Transform? *Computer Vision, Graphics, and Image Processing: Image Understanding*, **58**-2, 250-264, 1993.

[11] L. Xu and E. Oja. Randomized Hough Transform: Basic Mechanisms, Algorithms, and Computational Complexities. *Computer Vision, Graphics, and Image Processing: Image Understanding*, **57**-2, 131-154, 1993.

[12] O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, Cambridge, MA, 1993.

[13] S. Carlsson. Multiple Image Invariance using the Double Algebra. *Proceedings of the 2nd Joint European–US Workshop on Application of Invariance in Computer Vision, Lecture Notes in Computer Science*, **825**, 145-164, Springer-Verlag, Berlin Heidelberg, 1994.

[14] O. D. Faugeras and B. Mourrain. On the Geometry and Algebra of the Point and Line Correspondences between N Images. *Proceedings of the 5th International Conference on Computer Vision*, 951-956, 1995.

[15] G. H. Golub and C. F. Van Loan. *Matrix Computations, Third Edition.* The Johns Hopkins University Press, Baltimore and London, 1996.

[16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[17] S.M. Smith and J.M. Brady. SUSAN - A New Approach to Low Level Image Processing. *International Journal of Computer Vision*, **23**-1, 45-78, 1997.

Appendix

I. Exterior Product

Let $\boldsymbol{a} = (a_0, a_1, a_2, a_3)^\top$ and $\boldsymbol{b} = (b_0, b_1, b_2, b_3)^\top$ be two homogeneous vectors and $\boldsymbol{Q}$ be the matrix such that

$$\boldsymbol{Q} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \end{bmatrix}, \tag{22}$$

where $a_3 = b_3 = 1$. Furthermore, the 2×2 minors of the matrix $\mathbf{Q}$ are denoted by Q_{ij} as

$$Q_{ij} = \begin{vmatrix} a_i & a_j \\ b_i & b_j \end{vmatrix}.$$

The coordinates of $\boldsymbol{a} \wedge \boldsymbol{b}$ are defined as follows:

$$\boldsymbol{a} \wedge \boldsymbol{b} = (Q_{03}, Q_{13}, Q_{23}, Q_{01}Q_{12}, Q_{13})^\top.$$