Southeast Asian Bulletin of Mathematics (2005) 29: 361–??

Southeast Asian
Bulletin of
Mathematics

Randomized Algorithms for Visual Shape Analysis

Kazuhiko Kawamoto
Faculty of Engineering, 1-1 Sensui-cho, Tobata-ku, Kitakyushu, Fukuoka 804-8550, Japan.
Email: kawa@mms.kyutech.ac.jp

Atsushi Imiya
Institute of Media and Information Technology, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan.
Email: imiya@faculty.chiba-u.jp

Kaoru Hirota
Dept. of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Mail-Box:G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan.
Email: hirota@hrt.dis.titech.ac.jp

AMS Mathematics Subject Classification(2000): 68W20

Abstract. Randomized algorithms, based on random sampling and majority decision by voting, are proposed for 3D reconstruction from a sequence of images. The use of random sampling enables to efficiently estimate the 3D positions of objects from their perspective projections. To evaluate the performance, synthetic and real image sequences are used in experiments where 3D objects consisting of points, lines or planes are reconstructed. The proposed randomized algorithms provides a way of efficiently treating a large amount of image data, which is acquired more and more easily due to recent advances in computer and digital camera technology.

Keywords: Randomized algorithms; Random sampling; 3D reconstruction; Visual shape analysis.

1. Introduction

Three-dimensional (3D) object reconstruction from a sequence of images is one of the most fundamental problems in computer vision, because there are a large number of applications such as visual inspection [?, ?, ?], visual navigation [?, ?], object grasping [?, ?], and geometric modeling for virtual reality [?, ?]. These technologies enable computer systems to actively recognize 3D environments around themselves [?] and then the technologies make computer systems more intelligent and friendly.

Since imaging process lose depth information, a single 2D image gives only limited information about the physical shape and size of a 3D object in a space. This implies that available data on an image does not provide sufficient information about 3D environments. Hence inference of 3D information is required from a collection of insufficient 2D information. For shape reconstruction, the determination of correspondences, which mean the projections of the same 3D feature such as a point and a line in a space, among images is required [?]. Without establishing correspondences among images, 2D information provided by each image frame can not fuse together. As a result, it is generally impossible to reconstruct 3D positions of shapes from a sequence of images. Hence, the correspondence determination is a vital clue for shape reconstruction. The automatic acquisition of unique correspondences is, however, a very complex and time-consuming task, because, if a camera or a object moves around in a space, shape appearance changes and invisible part of the object occurs.

This paper proposes randomized algorithms for shape reconstruction. Most classical algorithms for the determination of correspondences use template matching or correlation between stereo pairs [?]. The proposed algorithms do not assume any preprocessing for the determination of correspondences. The algorithms directly estimates the 3D positions of objects from their perspective projections using a random search among a sequence of images. The random search reduces the computational complexity required for the determination of correspondences among images, which idea is motivated by the randomized Hough transform [?, ?]. To evaluate the performance, synthetic and real image sequences are used in experiments where 3D objects consisting of points, lines or planes are reconstructed from their perspective projections. The proposed randomized algorithms provides a basis of efficiently treating a large amount of image data, which is acquired more and more easily due to recent advances in computer and digital camera technology.

Section 2 describes epipolar geometry, which plays an important role in analyzing 3D objects. Sections 3, 4 and 5 propose randomized algorithms for spatial points, spatial lines, and spatial planes in the space, respectively.

2. Epipolar Geometry

This section refers to epipolar geometry [?], which describes the fundamental geometric relationship between two perspective cameras. This geometry plays an important role in reconstructing the shape of 3D objects from their perspective projections.

Set $\boldsymbol{e}_\kappa, \kappa = 1, \ldots, n$, to be a 3-vector of the κth optical center and $\boldsymbol{p}_{\kappa\alpha}, \alpha = 1, \ldots, m(\kappa)$, to be a 3-vector of the αth point in the κth image, where n and $m(\kappa)$ denote the numbers of images and points in the κth image, respectively. Throughout this paper, the vectors $\boldsymbol{e}_\kappa$ and $\boldsymbol{p}_{\kappa\alpha}$ are assumed to be defined in a world-coordinate system; this is possible because, if the positions and directions of the n cameras are known, these vectors can be transformed into a coordinate system by rigid transformation.

Assume $\boldsymbol{p}_{\kappa\alpha}$ indicates the projection of the αth 3-D point $\boldsymbol{x}_\alpha \in \boldsymbol{R}^3$ onto the κth image plane. For all pairs of κ and κ', $\kappa \neq \kappa'$, three vectors, $\boldsymbol{p}_{\kappa\alpha} - \boldsymbol{e}_\kappa$, $\boldsymbol{p}_{\kappa'\alpha} - \boldsymbol{e}_{\kappa'}$ and $\boldsymbol{t}_{\kappa\kappa'} = \boldsymbol{e}_{\kappa'} - \boldsymbol{e}_\kappa$, lie on the same plane, as shown in Fig. **??**. Therefore, the relation

$$|\, \boldsymbol{p}_{\kappa\alpha} - \boldsymbol{e}_\kappa,\ \boldsymbol{p}_{\kappa'\alpha} - \boldsymbol{e}_{\kappa'},\ \boldsymbol{t}_{\kappa\kappa'} \,| = 0, \tag{1}$$

holds, where $|\, a, b, c \,|$ denotes the scalar triple product. This relation is called *epipolar constraint*, which is rediscovered in [**?**] in the computer vision community. If a pair of vectors $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\alpha})$ satisfies eq. (**??**), the spatial point $\boldsymbol{x}_\alpha$ is uniquely calculated [**?**] by

$$\boldsymbol{x}_\alpha = \boldsymbol{e}_\kappa + \frac{\boldsymbol{t}_{\kappa\kappa'}{}^\top \boldsymbol{n}_{\kappa\alpha} - \boldsymbol{n}_{\kappa\alpha}{}^\top \boldsymbol{n}_{\kappa'\alpha} \boldsymbol{t}_{\kappa\kappa'}{}^\top \boldsymbol{n}_{\kappa'\alpha}}{1 - (\boldsymbol{n}_{\kappa\alpha}{}^\top \boldsymbol{n}_{\kappa'\alpha})^2} \boldsymbol{n}_{\kappa\alpha}, \tag{2}$$

or

$$\boldsymbol{x}'_\alpha = \boldsymbol{e}_{\kappa'} - \frac{\boldsymbol{n}_{\kappa\alpha}{}^\top \boldsymbol{n}_{\kappa'\alpha} \boldsymbol{t}_{\kappa\kappa'}{}^\top \boldsymbol{n}_{\kappa\alpha} - \boldsymbol{t}_{\kappa\kappa'}{}^\top \boldsymbol{n}_{\kappa'\alpha}}{1 - (\boldsymbol{n}_{\kappa\alpha}{}^\top \boldsymbol{n}_{\kappa'\alpha})^2} \boldsymbol{n}_{\kappa'\alpha}, \tag{3}$$

where

$$\boldsymbol{n}_{\kappa\alpha} = \frac{(\boldsymbol{p}_{\kappa\alpha} - \boldsymbol{e}_\kappa)}{||\boldsymbol{p}_{\kappa\alpha} - \boldsymbol{e}_\kappa||}, \quad \boldsymbol{n}_{\kappa'\alpha} = \frac{(\boldsymbol{p}_{\kappa'\alpha} - \boldsymbol{e}_{\kappa'})}{||\boldsymbol{p}_{\kappa'\alpha} - \boldsymbol{e}_{\kappa'}||}. \tag{4}$$

Since the position of point $\boldsymbol{p}_{\kappa\alpha}$ is always noisy due to digitalization and errors, even if two points $\boldsymbol{p}_{\kappa\alpha}$ and $\boldsymbol{p}_{\kappa'\alpha}$ are the projections of the same spatial point, these points do not satisfy the epipolar constraint. Therefore, if the relation

$$|\, \boldsymbol{p}_{\kappa\alpha} - \boldsymbol{e}_\kappa,\ \boldsymbol{p}_{\kappa'\alpha} - \boldsymbol{e}_{\kappa'},\ \boldsymbol{t}_{\kappa\kappa'} \,| \leq \epsilon, \tag{5}$$

holds, where ϵ is a small constant positive number, the epipolar constraint should be considered to be satisfied, If eq. (**??**) holds, then the spatial point $\boldsymbol{x}_\alpha$ is computed by $\boldsymbol{x}_\alpha = (\boldsymbol{x}_\alpha + \boldsymbol{x}'_\alpha)/2$, where $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}'_\alpha$ are computed by eqs. (**??**) and (**??**), respectively. Furthermore, if

$$||\boldsymbol{x}_\alpha - \boldsymbol{x}_\beta|| \leq \delta, \tag{6}$$

Figure 1: Epipolar geometry between two perspective cameras.

$\boldsymbol{x}_\alpha$ and $\boldsymbol{x}_\beta$ are should be considered as the same point in the space.

The constants ϵ and δ mainly depend on quantization errors on image planes. To determine ϵ and δ, noisy vectors of $\boldsymbol{n}_{\kappa\alpha}$ and $\boldsymbol{n}_{\kappa'\beta}$ in eq. (**??**) are set to be

$$\boldsymbol{n}'_{\kappa\alpha} = \boldsymbol{n}_{\kappa\alpha} + \Delta\boldsymbol{n}_{\kappa\alpha},\ \boldsymbol{n}'_{\kappa'\beta} = \boldsymbol{n}_{\kappa'\beta} + \Delta\boldsymbol{n}_{\kappa'\beta}, \tag{7}$$

where $||\Delta\boldsymbol{n}_{\kappa\alpha}|| \leq \Delta$ holds for a positive real number Δ. Accordingly, the left side of eq. (**??**) is bounded by

$$\begin{aligned} |\boldsymbol{n}'_{\kappa\alpha}\,\boldsymbol{n}'_{\kappa'\beta}\,\boldsymbol{t}_{\kappa\kappa'}| &\cong |\boldsymbol{n}_{\kappa\alpha}\,\boldsymbol{n}_{\kappa'\beta}\,\boldsymbol{t}_{\kappa\kappa'}| + |\Delta\boldsymbol{n}_{\kappa\alpha}\,\boldsymbol{n}_{\kappa'\beta}\,\boldsymbol{t}_{\kappa\kappa'}| + |\boldsymbol{n}_{\kappa\alpha}\,\Delta\boldsymbol{n}_{\kappa'\beta}\,\boldsymbol{t}_{\kappa\kappa'}| \\ &\leq (|\boldsymbol{n}_{\kappa\alpha}| + |\boldsymbol{n}_{\kappa'\beta}|)|\boldsymbol{t}_{\kappa\kappa'}|\Delta \cong 2fR\Delta, \end{aligned} \tag{8}$$

where f is the focal length of the camera and $R = \max||\boldsymbol{t}_{\kappa\kappa'}||$ for any pair of κ and κ' [**?**]. Furthermore, from the relations

$$\boldsymbol{n}'^{\top}_{\kappa\alpha}\boldsymbol{n}'_{\kappa'\beta} \cong \boldsymbol{n}^{\top}_{\kappa\alpha}\boldsymbol{n}_{\kappa'\beta} + \boldsymbol{n}^{\top}_{\kappa\alpha}\Delta\boldsymbol{n}_{\kappa'\beta} + \boldsymbol{n}^{\top}_{\kappa'\beta}\Delta\boldsymbol{n}_{\kappa\alpha} \tag{9}$$

and

$$\frac{1}{1-(\boldsymbol{n}'^{\top}_{\kappa\alpha}\boldsymbol{n}'_{\kappa'\beta})^2} \cong \frac{1}{1-(\boldsymbol{n}^{\top}_{\kappa\alpha}\boldsymbol{n}_{\kappa'\beta})^2}\left(1+2\frac{\boldsymbol{n}^{\top}_{\kappa\alpha}\Delta\boldsymbol{n}_{\kappa'\beta}+\boldsymbol{n}^{\top}_{\kappa'\beta}\Delta\boldsymbol{n}_{\kappa\alpha}}{1-(\boldsymbol{n}^{\top}_{\kappa\alpha}\boldsymbol{n}_{\kappa'\beta})^2}\right), \tag{10}$$

the following bound is obtained:

$$|\boldsymbol{x}' - \boldsymbol{x}| \leq 4\frac{R}{f}\Delta, \tag{11}$$

where

$$\boldsymbol{x}' = \boldsymbol{e}_i + \frac{\boldsymbol{e}^{\top}_{ij}\boldsymbol{n}'_{\kappa\alpha} - \boldsymbol{n}'^{\top}_{\kappa\alpha}\boldsymbol{n}'_{\kappa'\beta}\boldsymbol{t}^{\top}_{\kappa\kappa'}\boldsymbol{n}'_{\kappa'\beta}}{1-(\boldsymbol{n}'^{\top}_{\kappa\alpha}\boldsymbol{n}'_{\kappa'\beta})^2}\boldsymbol{n}'_{\kappa'\beta}. \tag{12}$$

These relations imply that it is possible to detect an accurate solution by increasing the image resolution. Furthermore, the resolution of accumulator space δ depends on the resolution of imaging planes.

3. 3D Reconstruction of Points from Multiple Views

3.1. Randomized Algorithm

Even if a pair $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\beta})$ satisfies eq. (**??**), $\alpha = \beta$ is not always concluded, i.e., two points $\boldsymbol{p}_{\kappa\alpha}$ and $\boldsymbol{p}_{\kappa'\beta}$ do not share the same spatial point in the space. In other words, if $\boldsymbol{x}_{\alpha\beta}$ denotes the spatial point calculated from $\boldsymbol{p}_{\kappa\alpha}$ and $\boldsymbol{p}_{\kappa'\beta}$, $\boldsymbol{x}_{\alpha\beta}$ is not always equal to both $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}_\beta$, that is, $\boldsymbol{x}_{\alpha\beta}$ does not exist in the scene. If $\boldsymbol{x}_{\alpha\beta}$ is observed, however, from many pairs of points in a sequence of images, it can be concluded that the spatial point exist in the scene. This fact leads to the following voting procedure [**?**] for the reconstruction of spatial points:

1. Select a pair of points $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\beta})$ such that $\kappa \neq \kappa'$.

2. Check whether $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\beta})$ satisfies the epipolar constraint defined by eq. (**??**). If the pair does not satisfy, then go back to **1**.
3. Calculate the spatial point $\boldsymbol{x}_{\alpha\beta}$ according to eq. (**??**) and vote 1 to $\boldsymbol{x}_{\alpha\beta}$.

If $\boldsymbol{x}_{\alpha\beta}$ receives a lot of votes after the voting procedure, it is concluded that $\boldsymbol{x}_{\alpha\beta}$ exists in the scene. This procedure is powerful for solving this kind of inverse problems but requires much computational requirement. Thus a method for reducing the complexity is required.

The combinations of two images from n images and two samples from each image with m points are $\binom{n}{2}$ and m^2, respectively. Therefore, the above voting procedure requires $\binom{n}{2} \cdot m^2$ iterations. The number of iterations, however, for all combinations rapidly increases with the number of images and points. In order to avoid the combinatorial complexity, random sampling is introduced into the procedure:

Algorithm

1. Randomly select a pair of points $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\alpha'})$ such that $\kappa \neq \kappa'$.
2. Check whether $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\alpha'})$ satisfies the epipolar constraint defined by eq.(**??**). If the pair does not satisfy, go back to step **1**.
3. Calculate the spatial point $\boldsymbol{x}_{\alpha\beta}$ according to eq. (**??**) and increase its vote score by one in the accumulator space.
4. If the number of iterations reaches a predefined number, detect peaks in the accumulator space. Otherwise go back to step **1**.

3.2. Experiment for Estimating Grid Points

Using synthetic data, the performance of the proposed algorithm is evaluated. Figure **??**(a) shows one example image of 20 views and Fig. **??**(b) shows the relation between the spatial positions of the 20 cameras and the spatial grid pattern. The size of one pixel in the image is 0.02 units and the focal length of each camera is 8.0 units. The images are digitalized with 256×256 pixels. The parameters ϵ in eq. (**??**) and δ in eq. (**??**) are set as $\epsilon = 0.004$ units and $\delta = 1.0$ units.

Table **??** shows the original and estimated positions of the grid points in the space. The error is evaluated using the criterion

$$\Delta x = \sqrt{\frac{1}{m}\sum_{\alpha=1}^{m} ||\bar{\boldsymbol{x}}_\alpha - \boldsymbol{x}_\alpha||^2}, \tag{13}$$

where $\bar{\boldsymbol{x}}_\alpha$ and $\boldsymbol{x}_\alpha$ are the original and detected spatial points on the grid pattern, respectively. According to eq. (**??**), the error for values in Table **??** is obtained as $\Delta x = 1.726041$. Since the size of one pixel is 0.02 units, one pixel has $0.01 \times \sqrt{2} \approx$

1.41×10^{-2} units of ambiguity on imaging planes if the noise is assumed to be mainly caused by digitalization on imaging planes. These ambiguities of points on imaging planes cause ambiguity of the corresponding spatial points in the space. Setting δx to be the ambiguity in the space, $\delta x = \sqrt{2} \cdot \delta p \cdot D/f$ on the plane perpendicular to the view line is obtained, where δp, D and f are half of one pixel edge, the distance between the viewpoint and the spatial point, and the focal length, respectively. By replacing D by the radius of the orbit of the camera center, which is 300 units in this experiment, in Table 1, for simplicity, $\Delta x \approx \sqrt{2} \cdot 0.01 \cdot 300.0/8.0 = 0.54$ is obtained. Thus, each spatial point has 0.54 units of ambiguity on the plane perpendicular to each corresponding view line. Furthermore, the depth error of the detected spatial point is usually larger than the error in the perpendicular direction of the view line, that is, the value of the depth error is larger than 0.54. From the evaluated error, the detected spatial points exist within about three times the 0.54 units of ambiguity.

4. 3D Reconstruction of Lines from Multiple Views

4.1. Plücker Coordinates Representation of Lines

For preliminary, a positive-unit semisphere is first defined in the following [**?**]. Let $\boldsymbol{S}^{n-1}$ be the unit sphere in $\boldsymbol{R}^n$ consisting of all points $\boldsymbol{x}$ with distance 1 from the origin. $\boldsymbol{S}^{n-1}$ is called the *n-sphere*. Here, the n-dimensional positive-unit semisphere $\boldsymbol{S}_+^{n-1}$ is defined as follows. For $n = 1$ and $\boldsymbol{S}^0 = [-1, 1]$, the positive

(a) (b)

Figure 2: (a) Example of grid pattern image. (b) Position of cameras and grid pattern.

Table 1: Estimation of grid points

Original			Estimated		
x	y	z	x	y	z
75.881905	-46.592583	50.0	76.658516	-46.526644	50.541542
75.881905	-46.592583	0.0	77.034992	-46.353581	0.427662
75.881905	-46.592583	-50.0	77.930947	-45.571753	-49.450551
62.940953	1.703709	50.0	63.535994	2.228122	50.456488
62.940953	1.703709	0.0	64.019634	2.296311	0.407401
62.940953	1.703709	-50.0	64.800258	2.952339	-49.659979
50.000000	50.000000	50.0	50.502252	50.545951	50.413020
50.000000	50.000000	0.0	51.049373	51.142913	0.709442
50.000000	50.000000	-50.0	51.552550	51.644017	-49.971516
1.703709	62.940952	50.0	2.204014	63.616967	50.483920
1.703709	62.940952	0.0	2.236144	64.438005	0.402550
1.703709	62.940952	-50.0	3.318760	65.084969	-49.054663
-46.592583	75.881904	50.0	-46.397053	76.479394	50.402371
-46.592583	75.881904	0.0	-46.814347	77.508448	0.362568
-46.592583	75.881904	-50.0	-46.047401	78.069815	-49.708130

half-space is defined as

$$\boldsymbol{R}^n_+ = \{\boldsymbol{x} \mid x_n > 0\}, \quad n \geq 1. \tag{14}$$

Now, setting

$$\boldsymbol{H}^{n-1}_+ = \boldsymbol{S}^{n-1} \cap \boldsymbol{R}^n_+, \quad n \geq 1, \tag{15}$$

the positive-unit semisphere is recursively defined by

$$\boldsymbol{S}^{n-1}_+ = \boldsymbol{S}^{n-2}_+ \cap \boldsymbol{H}^{n-1}_+, \quad n \geq 2. \tag{16}$$

Figure **??** shows an example of the 2-sphere and the positive-unit semisphere in one dimension.

Let $\boldsymbol{x} = (x_1, x_2, x_3)^\top$ and $\boldsymbol{y} = (y_1, y_2, y_3)^\top$ be two points in $\boldsymbol{R}^3$. A spatial line passing through these points are written by

$$\boldsymbol{r} = \boldsymbol{x} + t\,\frac{\boldsymbol{y} - \boldsymbol{x}}{||\boldsymbol{y} - \boldsymbol{x}||}, \tag{17}$$

(a)(b)

Figure 3: (a) the 2-sphere S^1 and (b) the positive-unit semisphere S^1_+ in 1-dimension.

where t is an arbitrary nonzero scalar. By applying the vector product to both sides of eq. (**??**) with $(\boldsymbol{y} - \boldsymbol{x})$, the following relation is obtained:

$$\boldsymbol{r} \times (\boldsymbol{y} - \boldsymbol{x}) = \boldsymbol{x} \times \boldsymbol{y}. \tag{18}$$

Here, two vectors $\boldsymbol{q}_1 = \boldsymbol{y} - \boldsymbol{x}$ and $\boldsymbol{q}_2 = \boldsymbol{x} \times \boldsymbol{y}$ are normalized to $||\boldsymbol{q}_1||^2 + ||\boldsymbol{q}_2||^2 = 1$. Geometrically, vector $\boldsymbol{q}_1 = (q_{01}, q_{02}, q_{03})\top$ indicates the orientation of the spatial line, and vector $\boldsymbol{q}_2 = (q_{23}, q_{31}, q_{12})^\top$ indicates the surface normal to the plane defined by the spatial line and the origin of the coordinate system, as shown in Fig. **??**.

Compound vector $\boldsymbol{q}^\top = (\boldsymbol{q}_1^\top, \boldsymbol{q}_2^\top) = (q_{01}, q_{02}, q_{03}, q_{23}, q_{31}, q_{12})$ represents the homogeneous coordinates of the spatial line and is called the *Plücker coordinates* of the spatial line [**?**]. (F. Klein and J. Plücker prove the fact that there is one-to-one mapping between lines in $\boldsymbol{R}^3$ and vector $\boldsymbol{q}$ in $\boldsymbol{PR}^5$ in 1930s.) From the definition of the Plücker coordinates, two vectors $\boldsymbol{q}$ and $-\boldsymbol{q}$ define the same spatial line. Accordingly, the sign of the sixth element of vector $\boldsymbol{q}$ is adjusted to be positive. With this adjustment, the Plücker coordinates uniquely determine a single point in $\boldsymbol{S}^5_+$.

Conversely, a single point on $\boldsymbol{S}^5_+$ determines the original spatial line. When $\boldsymbol{r}$ is set to be a point on the spatial line, eq.(**??**) implies the relation $\boldsymbol{r} \times \boldsymbol{q}_1 = \boldsymbol{q}_2$. Therefore, the relation $\boldsymbol{q}_1 \times \boldsymbol{r} \times \boldsymbol{q}_1 = \boldsymbol{q}_1 \times \boldsymbol{q}_2$ is obtained. This relation is expressed by

$$\boldsymbol{r} = \frac{\boldsymbol{q}_1 \times \boldsymbol{q}_2}{||\boldsymbol{q}_1||^2} + \frac{(\boldsymbol{r}^\top \boldsymbol{q}_1)}{||\boldsymbol{q}_1||} \frac{\boldsymbol{q}_1}{||\boldsymbol{q}_1||}. \tag{19}$$

Equation (**??**) is equivalent to eq. (**??**). Let $\boldsymbol{r}_0$ be the point which is on the spatial line defined by eq. (**??**) and closest to the coordinate origin; then eq. (**??**) is rewritten by

$$\boldsymbol{r_0} = \frac{\boldsymbol{q}_1 \times \boldsymbol{q}_2}{||\boldsymbol{q}_1||^2}. \tag{20}$$

Consequently, the parameter space of spatial lines is the six-dimensional positive-unit semisphere $\boldsymbol{S}^5_+$.

4.2. Randomized Algorithm

For the computation of a spatial line from image data, the algorithm in Sec. 3.1 is used. Since a pair of points in $\boldsymbol{R}^3$ uniquely determines a spatial line, $\boldsymbol{q}$ in $\boldsymbol{S}^5_+$ is computed from two pairs of points which satisfy the epipolar constraint. Similar to the algorithm in Sec. 3.1, the following algorithm uses random sampling and a dynamic data structure.

Algorithm

Figure 4: Geometrical configuration of vectors $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, which are the first and second of three elements of Plücker coordinates of the spatial line.

1. Randomly select two pairs of points $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\alpha'})$ and $(\boldsymbol{p}_{\kappa\beta}, \boldsymbol{p}_{\kappa'\beta'})$, such that $\kappa \neq \kappa'$.
2. Check whether $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\alpha'})$ and $(\boldsymbol{p}_{\kappa\beta}, \boldsymbol{p}_{\kappa'\beta'})$ satisfy the epipolar constraint. If both pairs of points satisfy it, then go to step **3**, otherwise return to step **1**.
3. Compute the spatial points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Compute point $\boldsymbol{q}$ in $\boldsymbol{S}^5_+$.
4. Check whether there exists the same element with $\boldsymbol{q}$ in the accumulator space. If there exists the same element, increase its voting score by one. If none is found, insert $\boldsymbol{x}$ into the accumulator space as a new element.
5. If the number of iterations reaches a predefined number, then detect peaks in the accumulator space. Otherwise go to step **1**.

This algorithm is considered to be an extension of the Hough transform [**?**]. Thus, the algorithm has both of the advantages and disadvantages of the Hough transform. The Hough transform can detect curve segments from sparse and occluded data. However, it is usually difficult for the Hough transform to detect relatively short curve segments, since the votes scores of short curve segments are lower than those of longer curve segments in the accumulator space. For the detection of short spatial lines, the following stepwise procedure is proposed; the step 5 of the algorithm is rewritten as follows.

5.1 If the voting score of a single element in the accumulator space reaches a predefined threshold, then detect the element in the accumulator space and go to Step 5.2. Otherwise go to Step 1.
5.2 Remove all points from images which lie on the projection of the detected spatial line.
5.3 Set the accumulator space to null.
5.4 Return to Step 1.

This stepwise procedure removes meaningless selections of sampled points. As a result, the voting scores of short spatial lines are increased. Therefore, the stepwise procedure enables us to detect short spatial lines from a sequence of images.

4.3. Experiment for Estimating Grid Objects

The spatial lines on the grid pattern used in Sec. 3.2 is reconstructed using points on line segments in the images. Figure **??** shows the result by the proposed algorithm for line reconstruction. The resolution of the accumulator space δ is set at $\delta = 0.004$ units. Table **??** shows the original and detected positions of points $\boldsymbol{r}_0$ in eq. (**??**) which are on each spatial line and are closest to the coordinate origin. The error of $\boldsymbol{r}_0$ is evaluated using the criterion

$$\Delta r = \sqrt{\frac{1}{m}\sum_{\alpha=1}^{m} ||\bar{\boldsymbol{r}}_{0\alpha} - \boldsymbol{r}_{0\alpha}||^2}, \tag{21}$$

where $\bar{\boldsymbol{r}}_{0\alpha}$ and $\boldsymbol{r}_{0\alpha}$ are the original and detected α-th spatial points $\boldsymbol{r}_0$, respectively. According to eq. (**??**), the error for values in Table **??** is obtained as $\Delta r = 1.844175$. Table **??** shows the original and detected orientation of spatial lines which is computed by eq.(**??**). The error of $\boldsymbol{u} = \boldsymbol{q_1}/||\boldsymbol{q_1}||$ is evaluated using the criterion

$$\Delta u = \sqrt{\frac{1}{m}\sum_{\alpha=1}^{m}(\bar{\boldsymbol{u}}_\alpha^\top \boldsymbol{u}_\alpha)^2}, \tag{22}$$

where $\bar{\boldsymbol{u}}_\alpha$ and $\boldsymbol{u}_\alpha$ are original and detected orientation of α-th spatial line, respectively. According to eq. (**??**), the error for values in Table **??** is obtained as $\Delta u = 0.999895$.

In this experiment, sample points on the grid pattern viewed from various directions are used. The length of the projected planar line segments on the images depends on the direction of views. Therefore the number of the sample points on planar line segments is not invariant among images even if no part is occluded. Hence, one-to-one point correspondences on planar line segments among images are not obtained. The algorithm uses the epipolar constraint to find the candidates of point correspondences among images. In the experiment, if the distance between sample points and epipolar lines is less than 0.2 pixels, the epipolar constraint is considered to be satisfied. Hence the pairs of points for which the distance between samples points and epipolar lines are shorter than 1.0 pixel are selected. Therefore, instead of finding all candidates of point correspondences on planar line segments, some candidates of point correspondences on planar line segments are only found.

Since the epipolar constraint is checked for sample points to avoid a meaningless sampling process, the algorithm reconstructs 3D positions from sparse data

Figure 5: The experimental result of the detection of spatial lines.

Table 2: The original and detected points $\boldsymbol{r}_0$

Original			Estimated		
x	y	z	x	y	z
75.879341	-46.591006	0.000000	77.162248	-44.846738	0.019543
62.941310	1.703715	0.000000	63.967091	2.983187	0.184057
49.999010	49.999010	0.000000	51.655587	51.484986	0.285862
1.703715	62.941310	0.000000	3.192047	63.630430	-0.140889
-46.591006	75.879341	0.000000	-44.925165	76.572184	0.463579
59.149757	15.851380	50.000008	58.550691	18.041296	50.274073
59.151828	15.850335	0.000000	60.089139	15.976696	0.262321
59.150139	15.851482	-50.000008	60.221615	16.543689	-50.131953
15.851482	59.150139	50.000008	13.019607	58.710254	49.783143
15.850335	59.151828	-0.000000	16.473921	59.562074	0.206351
15.851380	59.149757	-50.000008	16.115218	57.959300	-49.224928

on images. Although sparse data reduce the size of input data, sparse data do not preserve the epipolar constraint. Therefore, the effect of sparseness of data to the time and domain complexities is evaluated. For this evaluation, images with 90%,80%,70%,60%, and 50% sampled points on each image of Fig. **??** are prepared. These sample points are randomly selected from each original image. Table **??** shows the number of iterations, the execution time and the memory size. From Table **??**, both the execution time and the memory consumption does not depend on the number of sample points on images. These results show that the algorithm is stable for variance of the number of sample points on images.

The proposed algorithm directly reconstructs spatial lines from points on images, i.e., it does not include planar line detection on images. Thus the algorithm differs from other traditional methods including planar curve detection for the recovery of spatial lines [**?**, **?**, **?**]. The advantage of the algorithm is the reduction of the domain complexity compared with the traditional methods, because the domain for planar line detection is not required.

5. 3D Reconstruction of Planes from Multiple View

5.1. Representation of Planes

Let $\boldsymbol{x} = (x_1, x_2, x_3)^\top$, $\boldsymbol{y} = (y_1, y_2, y_3)^\top$, and $\boldsymbol{z} = (z_1, z_2, z_3)^\top$ be three points in $\boldsymbol{R}^3$. A spatial plane passing through these points are written by

$$\boldsymbol{n}^\top \boldsymbol{x} = h, \quad ||\boldsymbol{n}|| = 1, \tag{23}$$

where

$$\boldsymbol{n} = \frac{(\boldsymbol{y} - \boldsymbol{x}) \times (\boldsymbol{z} - \boldsymbol{x})}{||(\boldsymbol{y} - \boldsymbol{x}) \times (\boldsymbol{z} - \boldsymbol{x})||}, \quad h = \boldsymbol{n}^\top \boldsymbol{x}. \tag{24}$$

Table 3: The original and detected $\boldsymbol{u}$

Original			Estimation		
u_x	u_y	u_z	u_x	u_y	u_z
0.000000	0.000000	-1.000000	0.000357	0.000179	-0.999999
0.000000	0.000000	-1.000000	0.002946	-0.001473	-0.999995
0.000000	0.000000	-1.000000	0.002699	0.002845	-0.999992
0.000000	0.000000	-1.000000	0.000319	-0.002230	-0.999997
0.000000	0.000000	1.000000	0.002752	-0.004439	0.999986
-0.258853	0.965917	0.000000	-0.284860	0.958491	-0.012206
-0.258829	0.965923	0.000000	-0.256950	0.966424	-0.001306
-0.258853	0.965917	0.000000	-0.263659	0.964615	0.001602
-0.965917	0.258853	0.000000	-0.973522	0.228136	-0.014444
-0.965923	0.258829	0.000000	-0.963814	0.266573	0.000927
-0.965917	0.258853	0.000000	-0.964407	0.264384	-0.004431

Table 4: Experiment conditions of the detection of spatial lines

Cases	Iterations	Time(sec)	Memory
original	870598506	17600.41	168286
90%	769115189	22423.40	191261
80%	828713076	19854.98	181553
70%	915380276	17835.13	165667
60%	980371002	17028.31	155826
50%	829385152	19297.89	174409

Geometrically, vector $\boldsymbol{n}$ and scalar h indicate the surface normal and the distance from the origin to the plane, respectively.

Vector $\boldsymbol{q} = (n_x, n_y, n_z, q_h)^\top$ represents the homogeneous coordinates of the spatial plane [?], where $\boldsymbol{n} = (n_x, n_y, n_z)^\top$. From the definition, two vectors $\boldsymbol{q}$ and $-\boldsymbol{q}$ define the same spatial plane. Therefore the sign of the fourth element of the vector $\boldsymbol{q}$ is adjusted to be positive. With this adjustment, the homogeneous coordinates determine a single point in $\boldsymbol{S}^3_+$. Conversely, a single point on $\boldsymbol{S}^3_+$ determines the original spatial plane. Consequently, the accumulator space of spatial planes is the four-dimensional positive-unit semisphere $\boldsymbol{S}^4_+$.

5.2. Randomized Algorithm

For the computation of spatial planes from image data, the algorithm in Sec. 3.1 is used to generate three spatial points. Since a pair of points in $\boldsymbol{R}^3$ uniquely determines a spatial point, $\boldsymbol{q}$ in $\boldsymbol{S}^3_+$ is computed from three pairs of points which satisfy the epipolar constraint. In the following, the algorithm for reconstructing spatial planes are proposed using random sampling and the dynamic data structure.

Algorithm

1. Randomly select three pairs of points $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\alpha'})$, $(\boldsymbol{p}_{\kappa\beta}, \boldsymbol{p}_{\kappa'\beta'})$, and $(\boldsymbol{p}_{\kappa\gamma}, \boldsymbol{p}_{\kappa'\gamma'})$, such that $\kappa \neq \kappa'$.
2. Check whether $(\boldsymbol{p}_{\kappa\alpha}, \boldsymbol{p}_{\kappa'\alpha'})$, $(\boldsymbol{p}_{\kappa\beta}, \boldsymbol{p}_{\kappa'\beta'})$, and $(\boldsymbol{p}_{\kappa\gamma}, \boldsymbol{p}_{\kappa'\gamma'})$ satisfy the epipolar constraints. If every pair of the points does not satisfy it, go back to step **1**.
3. Compute the spatial points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$. Compute point $\boldsymbol{q}$ in $\boldsymbol{S}^3_+$.
4. Check whether there exists the same element with $\boldsymbol{q}$ in the accumulator space. If there exists the same element, increase its voting score by one. If none is found, insert $\boldsymbol{x}$ into the accumulator space as a new element.
5. If the number of iterations reaches a predefined number, then detect peaks in the accumulator space. Otherwise go back to step **1**.

5.3. Experiment for 3D Plane Reconstruction Using Synthetic and Real Images

Using synthetic data, the performance of the algorithm is evaluated. Figure **??** shows the example images of 5 views for three spatial planes. The algorithm is given 24 projections of spatial points on the three planes for each image. The error for the detected spatial planes using is evaluated using the criterion

$$\Delta n = \sqrt{\frac{1}{m}\sum_{\alpha=1}^{m}(\bar{\boldsymbol{n}}_\alpha{}^\top \boldsymbol{n}_\alpha)^2}, \quad \Delta h = \sqrt{\frac{1}{m}\sum_{\alpha=1}^{m}(\bar{h}_\alpha - h_\alpha)^2}, \tag{25}$$

where $\bar{\boldsymbol{n}}_\alpha$ and $\bar{h}_\alpha$ are the true parameters, and $\boldsymbol{n}_\alpha$ and h_α are α-th estimated parameters. According to eq. (**??**), $\Delta n = 0.995534$ and $\Delta h = 54.626280$ are obtained.

Figure 6: Example synthetic images for plane reconstruction.

Figure 7: Example real images for plane reconstruction.

Real data are next used to show the performance of the algorithm. Figure ?? shows the example images of 2 views for the three spatial planes of a box. The algorithm is given 7 projections of corner points on the box for each image. These images and the calibrated data are created by INRIA-Syntim. Table ?? shows the detected parameters of the spatial planes. Since it is impossible to obtain the true parameters of the spatial planes, the angles (ideally 90 degrees) between the pairs of the detected spatial planes are evaluated. From Table ??, the angles between the first and the second, the first and the third, and the second and the third planes are 89.93, 90.30 and 88.23 degrees, respectively. Therefore the algorithm also works well for real data.

6. Conclusion

This paper addresses the problem of reconstructing 3D shape from a sequence of images without the predetermination of correspondences among the images. Since the correspondence determination is a very complex and time-consuming task, many classical algorithms assume that correspondences among images are predetermined. For the achievement of shape reconstruction from a sequence of images, 2D information provided by each image fuses together. For the fusion of 2D information, a random sampling and voting process is introduced, which determines a solution by selecting hypotheses supported by a large number of given data. The experimental results show that the process is useful for solving inverse problems such as the 3D reconstruction problem. In general, there are many inverse problems to be solved in computer vision and computational intelligence, where the solution is not uniquely determined. The proposed approach gives a way of determining a reasonable solution from insufficient information. In this sense, it plays a role as an engine for solving inverse problems. The paper aims to build such an engine in computer vision and computational intelligence from a theoretical point of view.

The human visual system easily determines correspondences among images [?]. On the other hand, robot control technology recently becomes more and

Table 5: The detected spatial planes of the real object "Box".

No.	n_1	n_2	n_3	h
1	0.992099	-0.069374	-0.104528	258.0
2	0.091142	-0.186870	0.978148	44.0
3	0.102244	0.972789	0.207912	116.0

more accurate and stable, for example, robots walking with two legs are realized. Such robots are, however, usually controlled by a person. Thus, the determination of correspondences among image frames is very important for the design of autonomous systems. The aim of this paper is to provide one of the fundamental technologies for this purpose.

References

[1] Aloimonos, J.Y., Weiss, I. and Bandyopadhyay, A.: Active Vision, *Int. J. Computer Vision* **1**, 333–356 (1988).

[2] Baylou, P., Hadj Amor, B.El., Monsion, M., Bauvet, C. and Bousseau, G.: Detection and three-dimensional localization by stereoscopic visual sensor and its application to a robot for picking asparagus, *Pattern Recognition* **17**, 377–384 (1984).

[3] Chin, R.T. and Harlow, C.A.: Automated visual inspection: A survey, *IEEE Trans. Pattern Analysis and Machine Intelligence* **4**(6), 557–573 (1982).

[4] Duda, R.O. and Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures, *Comm. ACM*, **15**(1), 11–15 (1972).

[5] Gottesfeld, B.L.: A Survey of Image Registration Techniques, *ACM Computing Surveys* **24**(4), 325–376 (1992).

[6] Imiya, A.: Metric for spatial lines, Pattern Recognition Letters, *Pattern Recognition Letters* **17**, 1265–1269 (1996).

[7] Imiya, A. and Kawamoto, K.: Shape Reconstruction by Voting Process: Randomized Combinatorial Tomography, *Technical Report of IEICE*, **PRMU96-134**, 41–48 (1997).

[8] Ikeuchi, K., Nishihara, H.K., Horn, B.K.P., Sobalvarro, P. and Nagata, S.: Determining Grasp Con.gurations using Photometric Stereo and the PRISM Binocular Stereo System, *Int. J. Robotics Research* **5**(1), 46–65 (1986).

[9] Kanade, T., Rander, P. and Narayanan, P.: Virtualized Reality: Constructing Virtual Worlds From Real Scenes, *IEEE Trans. on MultiMedia* **4**(1), 34-47 (1997).

[10] Kanatani, K.: *Geometric Computation for Machine Vision*. Oxford Univ. Press, Oxford, U.K., 1993.

[11] Kawamoto, K. and Imiya, A.: The detection of spatial points and lines by random sampling and voting procedure, *Pattern Recognition Letters* **22**(2), 199–207 (2001).

[12] Liu, Y. and Huang, T.S.: A Linear Algorithm for Motion Estimation using Straight Line Correspondences, *Computer Vision, Graphics, and Image Processing* **44**, 35–57 (1988).

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

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

[15] Moravec, H.: Obstacle Avoidance and Navigation in the Real World by A Seeing Robot Rover, *CMU-RI-TR-3*, 1980.

[16] Newman, T.S. and Jain, A.K.: A Survey of Automated Visual Inspection, *Computer Vision and Image UnderStanding* **61**(2), 231–262 (1995).

[17] Nitzan, D.: Three-dimensional vision structure for robot applications, *IEEE Trans. Pattern Analysis and Machine Intelligence* **10**(3), 291–309 (1988).

[18] Selig, J.M.: *Geometrical Methods in Robotics*, Springer-Verlag, Berlin Heidelberg, 1996.

[19] Spetsakis, M.E. and Aloimonos, J.: Structure from Motion using Line Correspondences, *Int. J. Computer Vision* **4**, 171–183 (1990).

[20] Thorpe, C.E., Hebert, M., Shafer, S.A. and Kanade, T.: Vision and Navigation for the Carnegie-Mellon NAVLAB, *IEEE Trans. on Pattern Analysis and Machine*

Intelligence **10**(3), 361–372 (1988).

[21] Weng, J., Huang, T.S. and Ahuja, N.: Motion and Structure from Line Correspondences: Closed-form Solution, Uniqueness, and Optimization, *IEEE Trans. Pattern Analysis and Machine Intelligence* **14**(3), 318–336 (1992).

[22] Xu, G. and Zhang, Z.: *Epipolar Geometry in Stereo, Motion and Object Recognition*, Kluwer Academic Publishers, 1996.

[23] Xu, L., Oja, E., Kultanen, P.: A new curve detection method: randomized Hough transform (RHT), *Pattern Recognition Lett.* **11**(5), 331–338 (1990).

[24] Xu, L. and Oja, E.: Randomized Hough Transform: Basic Mechanisms, Algorithms, and Computational Complexities, *CVGIP: Image Understanding* **57**(2), 131–154 (1993).

[25] Zisserman, A., Fitzgibbon, A.W. and Cross, G.: VHS to VRML: 3D Graphical Models from Video Sequences, *IEEE Int. Conf. on Multimedia and Systems*, 51–57 (1999).