Proc IEEE Conf. on Computer Vision and Pattern Recognition, 1986

MOTION ESTIMATION FROM 3-D POINT SETS WITH AND WITHOUT CORRESPONDENCES

Zse-Cherng Lin*, Thomas S. Huang, Steven D. Blostein, Hua Lee, and E.A. Margerum**

Coordinated Science Laboratory
University of Illinois
Urbana, IL 61801

ABSTRACT- This paper studies the estimation of rigid-body motion from two 3-D point sets with and without correspondences. In Part I, by utilizing properties of Fourier transformation, a frequency-domain approach for determining the motion without point correspondences is presented. In the algorithm, the problem of estimating the translation is first separated by subtracting the coordinates of the centroids from the 3-D point coordinates in the point sets. The rotation parameters are estimated by correlating the Fourier transforms of two functions defined on the two sets of the 3-D points. In Part II, with point correspondence given, an algorithm for finding the least-squares solution to the motion parameters is presented. First, we prove that the least-squares solution is such that the centroids of the two point sets coincide. The determination of rotation and translation can therefore be decoupled. An iterative procedure for obtaining the rotation parameters is then described.

INTRODUCTION

The estimation of 3-D motion parameters is an important problem in motion analysis. It can be useful in many applications such as scene analysis, motion prediction and trajectory planning. This paper presents motion estimation from 3-D point sets with and without point correspondences. The paper contains two parts. Part I, "A frequency-domain algorithm for determining motion of a rigid object from range data without correspondences," was supported in part by the National Science Foundation under Grants ENG-84-51484 and ECS-83-19509, in part by Motorola Inc., and in part by Hughes Aircraft Co. Part II, "Least-squares estimation of motion parameters from 3-D point correspondences," was supported in part by the National Science Foundation under Grant DCR-84-15325, and in part by the Battelle Columbus Laboratories under the Scientific Service Program Contract No. DAAG29-81-D-0100.

PART I

A FREQUENCY-DOMAIN ALGORITHM FOR DETERMINING MOTION OF A RIGID OBJECT FROM RANGE DATA WITHOUT CORRESPONDENCES

Zse-Cherng Lin, Hua Lee and Thomas S. Huang

I. INTRODUCTION

The estimation of three-dimensional motion parameters of a rigid body is an important problem in motion analysis. It can be useful in many applications such as scene analysis, motion prediction and trajectory planning. In general, to solve the problem requires the matching of two- or three- dimensional data of feature points on the object at two time instances. After the matching of corresponding points has been accomplished, the motion parameters can be estimated by solving the equations which govern the corresponding points at these two time instances [1-6].

The motion parameters of a rigid body can be represented by a rotation around an axis passing through the origin of the coordinate system, followed by a translation. In general, the three-dimensional motion parameters are uniquely determined by the range data of three asymmetrical feature points on the object. When the data contain noise, the correspondences of feature points at two time instances are difficult to determine. The motion parameters estimated from three asymmetrical feature points may not be accurate. In this paper, the problem of estimating the three-dimensional motion parameters from range data without point correspondences is investigated. It is assumed that the 3-D coordinates of a set of feature points on an object at two time instances have been obtained. However, point correspondences between the two time instances are not established. We present a systematic algorithm to find the motion parameters. Two functions are defined on the 3-D coordinates of the two sets of feature points. The motion estimation is based on the Fourier transforms of the two functions. The principle that a function and its Fourier transform must experience the same rotation is utilized. Since the computation of the Fourier transform does not require the point correspondences, the motion parameters can be determined without knowing the correspondences.

In Section 2, the theoretical background of the algorithm is described. The problem of estimating the motion parameters is formulated. The determination of the translation parameter is separated first by subtracting the coordinates of the centroid from the 3-D coordinates of the feature points in each set of data. Two functions

* Zse-Cherng Lin is on leave from the Department of Electrical and Computer Engineering, University of California, Santa Barbara.

** E.A. Margerum is with the U.S. Army Engineer Topographic Laboratories, Ft. Belvoir, VA 22060.

are defined. The effect of the object rotation on the Fourier transforms of the functions is discussed. Section 3 establishes a cost function for the search of the rotation axis by using the fact that the axis is stationary in the domain of the Fourier transform during the rotation. In Section 4, the estimation of the rotation angle based on the correlation of the values of the two Fourier transforms on a circle is presented. A cost function based on the correlation is derived. Some simulation results and practical aspects in the realization of the algorithm are discussed in Section 5.

II. THEORETICAL BACKGROUND

Assume that the 3-D coordinates of some feature points on a rigid object at two time instances are given. If there is no noise, the two point sets can be related by

$$\mathbf{p}_{2i} = \mathbf{R}\mathbf{p}_{1i} + \mathbf{T}, \qquad i = 1, 2, \ldots, N, \tag{1}$$

where $\mathbf{p}_{1i}$ and $\mathbf{p}_{2i}$ represent the corresponding points in the two range-data sets before and after the motion, $\mathbf{T}$ is the translation vector and $\mathbf{R}$ is the rotation matrix. The elements of the rotation matrix can be shown as

$$\mathbf{R} = \tag{2}$$

$$\begin{bmatrix} n_1^2+(1-n_1^2)\cos\theta & n_1n_2(1-\cos\theta)+n_3\sin\theta & n_1n_3(1-\cos\theta)-n_2\sin\theta \\ n_1n_2(1-\cos\theta)-n_3\sin\theta & n_2^2+(1-n_2^2)\cos\theta & n_2n_3(1-\cos\theta)+n_1\sin\theta \\ n_1n_3(1-\cos\theta)+n_2\sin\theta & n_2n_3(1-\cos\theta)-n_1\sin\theta & n_3^2+(1-n_3^2)\cos\theta \end{bmatrix},$$

where n_1, n_2 and n_3 are the directional cosines of the rotation axis, and θ is the angle of rotation.

Because the two centroids of the points in the two data sets are also governed by the same rotation matrix and translation vector, the following relation must hold:

$$\frac{1}{N}\sum_{k=1}^{N}\mathbf{p}_{2k} = \mathbf{R}\left(\frac{1}{N}\sum_{k=1}^{N}\mathbf{p}_{1k}\right) + \mathbf{T}, \tag{3}$$

or

$$\mathbf{T} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{p}_{2k} - \mathbf{R}\left(\frac{1}{N}\sum_{k=1}^{N}\mathbf{p}_{1k}\right). \tag{4}$$

If we define

$$\mathbf{q}_{1i} = \mathbf{p}_{1i} - \frac{1}{N}\sum_{k=1}^{N}\mathbf{p}_{1k} \tag{5}$$

and

$$\mathbf{q}_{2i} = \mathbf{p}_{2i} - \frac{1}{N}\sum_{k=1}^{N}\mathbf{p}_{2k}, \tag{6}$$

Eq. (1) can be rewritten as

$$\mathbf{q}_{2i} = \mathbf{R}\mathbf{q}_{1i}, \qquad i = 1, 2, \ldots, N. \tag{7}$$

Note that in calculating the centroids, no point correspondence is required. The problem of motion estimation becomes first finding a rotation matrix $\mathbf{R}$ that satisfies Eq. (7) and then using Eq. (4) to determine the translation vector $\mathbf{T}$. Existing techniques require that the point correspondences be given or determined. Then three asymmetrical points can be used to solve Eq. (7) [6].

Many algorithms have been discussed for estimating the three-dimensional motion from range data in the literature [6],[7]. In the following, the principle of a new algorithm that utilizes the property of the Fourier Transformation to find the rotation parameters will be presented. The advantage of this algorithm is that point correspondences are not needed.

We define two functions $g_1(\mathbf{q})$ and $g_2(\mathbf{q})$ on the new coordinates $\mathbf{q}_{1i}$ and $\mathbf{q}_{2i}$ of the feature points as follows:

$$g_1(\mathbf{q}) = \sum_{i=1}^{N}\delta(\mathbf{q}-\mathbf{q}_{1i}) \tag{8}$$

and

$$g_2(\mathbf{q}) = \sum_{i=1}^{N}\delta(\mathbf{q}-\mathbf{q}_{2i}), \tag{9}$$

where $\delta(\mathbf{q})$ is a Dirac delta function. The Fourier transforms of $g_1(\mathbf{q})$ and $g_2(\mathbf{q})$ can be expressed as

$$G_1(\mathbf{f}) = \int_{-\infty}^{\infty} g_1(\mathbf{q})\exp(-j\,2\pi\mathbf{f}^T\mathbf{q})\,d\mathbf{q} = \sum_{i=1}^{N}\exp(-j\,2\pi\mathbf{f}^T\mathbf{q}_{1i}), \tag{10}$$

and

$$G_2(\mathbf{f}) = \sum_{i=1}^{N}\exp(-j\,2\pi\mathbf{f}^T\mathbf{q}_{2i}), \tag{11}$$

in which $\mathbf{f}^T\mathbf{q}$ shows the inner product of the two vectors $\mathbf{f}$ and $\mathbf{q}$. By substituting Eq. (7) into Eq. (11), it follows that

$$G_2(\mathbf{f}) = \sum_{i=1}^{N}\exp(-j\,2\pi\mathbf{f}^T\mathbf{q}_{2i}) = \sum_{i=1}^{N}\exp\left\{-j\,2\pi\mathbf{f}^T(\mathbf{R}\mathbf{q}_{1i})\right\} = \sum_{i=1}^{N}\exp\left\{-j\,2\pi(\mathbf{R}^T\mathbf{f})^T\mathbf{q}_{1i}\right\} = G_1(\mathbf{R}^T\mathbf{f}). \tag{12}$$

From Eq. (12), it is seen that because $\mathbf{q}_{1i}$ and $\mathbf{q}_{2i}$ are related by the rotation matrix $\mathbf{R}$, the Fourier transforms $G_1(\mathbf{f})$ and $G_2(\mathbf{f})$ are also related by $\mathbf{R}$. Therefore, the estimation of the rotation matrix can be resolved by finding a matrix $\mathbf{R}$ that satisfies Eq. (12). This implies that the rotation matrix can be found by correlating the values of the two Fourier transforms. Since the computations of the two transforms do not require the point correspondences, the correlation can also be determined without the point correspondences. However, it is necessary to have a systematic procedure such that the correlation can be computed efficiently and accurately. In principle, it requires exhaustive search through all possible rotation matrices to determine the rotation parameters by direct correlation of the two Fourier transforms. A more efficient approach would be to find the rotation axis first and then to determine the rotation angle. This is also an advantage of correlating the Fourier transforms G_1 and G_2 instead of g_1 and g_2. In the next section, we will present a systematic approach for finding the rotation axis.

III. DETERMINATION OF THE ROTATION AXIS

The values of the Fourier transform along the rotation axis are not changed because it is stationary during the rotation. This property can be used to search for the axis. From Eq. (12), it can be seen that the values of $G_1(\mathbf{f})$ and $G_2(\mathbf{f})$ along the rotation axis are identical because the values $\mathbf{f}$ and $\mathbf{R}^T\mathbf{f}$ are equal along the axis. Hence, the rotation axis can be determined by correlating the values of $G_1(\mathbf{f})$ and $G_2(\mathbf{f})$ along possible axes. For this purpose, we define three functions $G_{11}(a,b,c)$, $G_{12}(a,b,c)$ and $G_{22}(a,b,c)$ of the three variables a, b and c as follows:

$$G_{11}(a,b,c) = \int_{-A}^{A} G_1(\mathbf{f})G_1^*(\mathbf{f})dr \Bigg|_{\mathbf{f}=r\begin{bmatrix}a\\b\\c\end{bmatrix}} \quad (13)$$

$$G_{22}(a,b,c) = \int_{-A}^{A} G_2(\mathbf{f})G_2^*(\mathbf{f})dr \Bigg|_{\mathbf{f}=r\begin{bmatrix}a\\b\\c\end{bmatrix}} \quad (14)$$

and

$$G_{12}(a,b,c) = \int_{-A}^{A} G_1(\mathbf{f})G_2^*(\mathbf{f})dr \Bigg|_{\mathbf{f}=r\begin{bmatrix}a\\b\\c\end{bmatrix}}, \quad (15)$$

where $c=\sqrt{1-a^2-b^2}$ and A is a constant which has to be chosen properly and will be discussed in a later section. Note that a, b and c represent the directional cosines of the axis to be determined. The cost function for determining the directional cosines of the rotation axis can then be defined as

$$C_1(a,b,c) = \frac{G_{12}(a,b,c)}{[G_{11}(a,b,c)G_{22}(a,b,c)]^{1/2}}. \quad (16)$$

It is clear that when a, b and c are the exact directional cosines of the rotation axis, the cost function C_1 has value one because of $G_{11}(a,b,c) = G_{22}(a,b,c) = G_{12}(a,b,c)$. In general, the correlation has value less than one. By substituting the definition of $G_1(\mathbf{f})$ and $G_2(\mathbf{f})$ into Eqs. (13), (14) and (15), the integrations can be performed and they can be simplified as follows:

$$G_{11}(a,b,c) \quad (17)$$

$$= \int_{-A}^{A} \sum_{i=1}^{N} \exp(-j\,2\pi\mathbf{f}^T\mathbf{q}_{1i}) \sum_{k=1}^{N} \exp(j\,2\pi\mathbf{f}^T\mathbf{q}_{1k})dr \Bigg|_{\mathbf{f}=r\begin{bmatrix}a\\b\\c\end{bmatrix}}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{N} \frac{\sin\left[2\pi A\,[a(x_{1i}-x_{1k})+b(y_{1i}-y_{1k})+c(z_{1i}-z_{1k})]\right]}{\pi[a(x_{1i}-x_{1k})+b(y_{1i}-y_{1k})+c(z_{1i}-z_{1k})]}.$$

$$G_{22}(a,b,c) \quad (18)$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{N} \frac{\sin\left[2\pi A\,[a(x_{2i}-x_{2k})+b(y_{2i}-y_{2k})+c(z_{2i}-z_{2k})]\right]}{\pi[a(x_{2i}-x_{2k})+b(y_{2i}-y_{2k})+c(z_{2i}-z_{2k})]}.$$

and

$$G_{12}(a,b,c) \quad (19)$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{N} \frac{\sin\left[2\pi A\,[a(x_{1i}-x_{2k})+b(y_{1i}-y_{2k})+c(z_{1i}-z_{2k})]\right]}{\pi[a(x_{1i}-x_{2k})+b(y_{1i}-y_{2k})+c(z_{1i}-z_{2k})]}.$$

where x_{mn}, y_{mn} and z_{mn} are the 3-D coordinates of $\mathbf{q}_{mn}$. Since Eqs. (17), (18) and (19) all have simple representations, the rotation axis can be found by an efficient search procedure based on Eq. (16). Although it may seem to have three variables to be determined, the variables must satisfy the relation $a^2+b^2+c^2=1$ and one of the variables is not necessary in the search.

IV. ESTIMATING THE ROTATION ANGLE

From the derivation shown in Section 2, it is clear that if we evaluate the values of the two Fourier transforms, we will see that $G_2(\mathbf{f})$ is identical to $G_1(\mathbf{f})$ rotated by $\mathbf{R}$. This suggests that if the values of $G_1(\mathbf{f})$ are evaluated on a sphere, after the rotation the values become those of $G_2(\mathbf{f})$ on the same sphere. Furthermore, it follows that if we evaluate the values of $G_1(\mathbf{f})$ and $G_2(\mathbf{f})$ on the circle which is the intersection of the sphere and a plane passing through the origin and perpendicular to the rotation axis, $G_2(\mathbf{f})$ should be a circular shifted version of $G_1(\mathbf{f})$. Because the rotation axis can be determined by the approach described in the previous section, in this section we will present a systematic procedure for determining the rotation angle.

Since the rotation matrix can be specified by three parameters n_1, n_2 and θ as described in Eq. (2), in the following we will include the parameters to be determined in the rotation matrix $\mathbf{R}(n_1, n_2, \theta)$. For simplicity, the circle noted in the previous paragraph is assumed to have a radius B, then the variable on this circle in the transformed domain can be expressed as

$$\mathbf{f} = \begin{bmatrix}f_x\\f_y\\f_z\end{bmatrix} = \mathbf{R}(n_1, n_2, \phi)\begin{bmatrix}\frac{n_2B}{\sqrt{n_1^2+n_2^2}}\\ \frac{-n_1B}{\sqrt{n_1^2+n_2^2}}\\ 0\end{bmatrix}, \quad (20)$$

where the angle ϕ becomes the variable and $0 < \phi \leqslant 2\pi$. The values of $G_1(\mathbf{f})$ and $G_2(\mathbf{f})$ on the circle can be rewritten as functions of n_1, n_2 and ϕ as follows:

$$G_1(\mathbf{f}) = G_1(n_1,n_2,\phi) \quad (21)$$

$$= \sum_{i=1}^{N} \exp\left\{-j\,2\pi\begin{bmatrix}\frac{n_2B}{\sqrt{n_1^2+n_2^2}} & \frac{-n_1B}{\sqrt{n_1^2+n_2^2}} & 0\end{bmatrix}\mathbf{R}^T(n_1,n_2,\phi)\begin{bmatrix}x_{1i}\\y_{1i}\\z_{1i}\end{bmatrix}\right\}$$

and

$$G_2(\mathbf{f}) = G_2(n_1,n_2,\phi) \quad (22)$$

$$= \sum_{i=1}^{N} \exp\left\{-j\,2\pi\begin{bmatrix}\frac{n_2B}{\sqrt{n_1^2+n_2^2}} & \frac{-n_1B}{\sqrt{n_1^2+n_2^2}} & 0\end{bmatrix}\mathbf{R}^T(n_1,n_2,\phi)\begin{bmatrix}x_{2i}\\y_{2i}\\z_{2i}\end{bmatrix}\right\},$$

where x_{ki}, y_{ki} and z_{ki} are the three coordinates of the vector $\mathbf{q}_{ki}$. From Eq. (12), if the actual rotation matrix is $\mathbf{R}(n_1,n_2,\theta)$, the values of $G_2(n_1,n_2,\phi)$ on the circle can also be expressed in terms of values of $G_1(n_1,n_2,\phi)$ as

$$G_2(n_1,n_2,\phi) = G_2(\mathbf{f}) = G_1(\mathbf{R}^T(n_1,n_2,\theta)\mathbf{f})$$

$$= \sum_{i=1}^{N}\exp\left\{-j\,2\pi\begin{bmatrix}\frac{n_2B}{\sqrt{n_1^2+n_2^2}} & \frac{-n_1B}{\sqrt{n_1^2+n_2^2}} & 0\end{bmatrix}\mathbf{R}^T(n_1,n_2,\phi-\theta)\begin{bmatrix}x_{1i}\\y_{1i}\\z_{1i}\end{bmatrix}\right\}$$

$$= G_1(n_1,n_2,\phi-\theta). \quad (23)$$

Eq. (23) proves that the values of $G_1(\mathbf{f})$ and $G_2(\mathbf{f})$ on the circle are the circular shifted version of each other if the rotation axis has directional cosines n_1, n_2 and $\sqrt{1-n_1^2-n_2^2}$. We can define a cost function to be maximized for the search procedure as

$$C_2(\gamma)=\frac{\int_0^{2\pi} G_1(n_1,n_2,\phi)G_2^*(n_1,n_2,\phi+\gamma)\,d\phi}{\left[\int_0^{2\pi} |G_1(n_1,n_2,\phi)|^2 d\phi\right]^{1/2} \cdot \left[\int_0^{2\pi} |G_2(n_1,n_2,\phi)|^2 d\phi\right]^{1/2}} \quad (24)$$

for $0<\gamma\leqslant 2\pi$. From Eq. (24), it can be easily seen that

$$C_2(\gamma) \leqslant C_2(\theta) = 1 \,. \quad (25)$$

Therefore, a systematic search based on this cost function can be designed. The values of $G_1(n_1,n_2,\phi)$ and $G_2(n_1,n_2,\phi)$ are computed and the cost function $C_2(\gamma)$ is evaluated. When the cost function reaches its maximum, the rotation angle is determined. This process is virtually a process of computing the correlation of the two Fourier transforms $G_1(n_1,n_2,\phi)$ and $G_2(n_1,n_2,\phi)$. Theoretically, the maximum value of the cost function should be one if there is no noise, n_1 and n_2 are correct, and the search can reach all possible values γ.

From the definition of the cost function in Eq. (24), one can understand that if the directional cosines n_1 and n_2 are not known in advance, it is still possible to find the correct rotation parameters by using this cost function because it has only a global maximum. However, an exhaustive search has to be done by computing the values of cost functions C_2 for all possible n_1, n_2 and γ. It is far more efficient if the directional cosines n_1 and n_2 of the rotation axis can be found first by using the approach described previously.

V. DISCUSSION AND SIMULATION RESULTS

As shown in Section 4, the determination of the rotation angle is a process of correlation in the Fourier transforms. The correlation can be accomplished by taking advantage of the fast Fourier transform. The search for the rotation axis requires more computation because the cost function C_1 has to be evaluated for all possible directional cosines a and b. The values a and b need to satisfy $a^2+b^2 \leqslant 1$. The fast Fourier transform can not be used. Fortunately, the computations of functions G_{11}, G_{12} and G_{22} are easy because of their simple representations. Overall, the computation time greatly depends on the accuracies required in tne estimation of the motion parameters. It increases approximately with a factor of the square of that increased in the levels of the parameters.

The two constants A and B have significant effects on the smoothness of the cost functions C_1 and C_2. To assure that the global maximum values of the cost functions be reached when the motion parameters can have only finite levels, A and B must be chosen carefully. In general, the smaller the values of A and B, the smoother the cost functions C_1 and C_2. When the values of A and B become bigger, the global maximum values of the cost functions become distinctly biggest values in their nearby region in the domain of the parameters. They may be missed easily if the numbers of the levels in the motion parameters are few. When the values of A and B are smaller, there exist more local maximums which have values closer to the global maximums. Therefore, A and B must be chosen properly so that the algorithm provides optimal accuracy. The relations between the cost functions, the coordinates of the feature points and the two constants A and B are under investigation.

Many simulations have been performed with artificial data to evaluate the efficiency and accuracy of this algorithm. Two sets of 3-D data of feature points on an object were generated. Each set contained 10 feature points. The feature points before the motion were assumed to be randomly distributed in the cubic object region of size 50 on each side and centered at (50, 50, 50). The object went through a three-dimensional motion. The two sets of data were used to determine the motion parameters. Different smallest levels that were available in the search of the motion parameters were tried in the simulation to investigate the performance of this algorithm.

From the results of the simulations, it was found that about 25 uniformly distributed levels in the directional cosines n_1 and n_2 are required to approximately locate the rotation axis. If the number of levels were fewer, an incorrect rotation axis might be determined. Two sets of typical results are shown in Tables 1 and 2. The numbers of possible levels in both n_1 and n_2 of the rotation axis shown in the tables are 50 and 100. Since the search of the rotation angle contained only one variable and was efficient, 512 levels were assumed for the purpose of using the FFT subroutine. In the simulation 0.1 is used for the values of both A and B.

Table 1 shows the results of the case in which the rotation parameters are n_1=0.32, n_2=-0.42 and θ=61.171875°, and the translation parameter is (54, 63, 47). When 100 uniformly distributed levels were available for n_1 and n_2, the parameters could be found without error. If the number of available levels was 50, only approximate values close to the exact parameters were estimated. The accuracy of the rotation angle and the translation parameter were also affected. When the data were quantized to integer numbers, quantization noises were included in the simulation. The results are also shown in the table. It can be seen that the estimation is not as accurate as without quantization noises. In Table 2, we show the case in which all parameters were different from those levels available for the search. The parameters are n_1=0.33, n_2=-0.41, θ=61.0° and T=(54, 63, 47). Although the estimated values were not completely accurate, they were very close to the true values. In both Tables 1 and 2, (a) shows the true motion parameters. The results of 50 levels in n_1 and n_2 with data not quantized and quantized to integers are shown in (b) and (c) respectively. The results of 100 levels in n_1 and n_2 with data not quantized and quantized to integers are shown in (d) and (e) respectively.

Estimation of Motion Parameters Object Size: 50^3, Object Center: (50, 50, 50)						
	n_1	n_2	θ	Δx	Δy	Δz
(a)	0.32	-0.42	61.17	54.0	63.0	47.0
(b)	0.32	-0.44	61.17	53.87	62.55	48.65
(c)	0.32	-0.40	63.28	54.18	65.58	46.65
(d)	0.32	-0.42	61.17	54.0	63.0	47.0
(e)	0.32	-0.42	62.58	54.00	64.37	47.94

Table 1.

Estimation of Motion Parameters Object Size: 50^3, Object Center: (50, 50, 50)						
	n_1	n_2	θ	Δx	Δy	Δz
(a)	0.33	-0.41	61.0	54.0	63.0	47.0
(b)	0.32	-0.40	61.17	54.13	63.89	45.88
(c)	0.36	-0.40	58.35	54.07	58.13	45.95
(d)	0.32	-0.42	61.17	53.97	63.55	47.52
(e)	0.34	-0.40	61.88	53.94	63.41	47.18

Table 2.

Recently, we have also developed another new algorithm which does not require an exhaustive search for determining the motion without point correspondences [8]. The algorithm is fast and accurate in the noise-free case. However, it is believed that problems of missing points and very noisy data may cause more inaccuracy in that algorithm. The frequency domain algorithm determines the motion parameters by the correlations in the Fourier transform domain. It may have the advantage of averaging the noise effect because every noise component in the coordinates has its contribution to the values of the Fourier transform. If the noise contained in the data is random, it becomes random phase factor in its Fourier transform domain. Intuitively, it may be canceled out. The problem of missing points may not be serious because the global maximum of cost functions can still exist with a smaller values. Nevertheless, the real effects of the noise and missing points require more careful theoretical study and computer simulation in the future research.

VI. CONCLUSION

In this paper, we present a new algorithm for the motion estimation from 3-D coordinates of the feature points on a rigid body. The algorithm does not require the knowledge of the point correspondences. Properties of the Fourier transformation is utilized to estimate the motion parameters. The algorithm determines the motion parameters by computing the correlations in the Fourier transforms of the functions defined on the coordinates of the feature points at two time instances. Efficient approaches for finding the rotation axis and rotation angle are described. The translation parameter can then be readily determined. Simulation results with artificial data are presented. Some practical aspects such as the finite numbers of levels in the motion parameters and the noise effect for implementing the algorithm are briefly addressed.

REFERENCES

[1] J.W. Roach and J.K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans. on PAMI*, Vol. 6, pp. 554-562, Nov. 1980.

[2] T.S. Huang and R.Y. Tsai, "Image sequence analysis: Motion estimation," in *Image Sequence Analysis*, Ed. T.S. Huang, Springer-Verlag, 1981.

[3] H.C. Longnet-Higgins, "A computer program for reconstructing a scene from two projections," *Nature*, Vol. 293, pp. 133-135, Sept. 1981.

[4] R.Y. Tsai and T.S. Huang, "Uniqueness and estimation of 3-D motion parameters of rigid bodies with curved surfaces," *IEEE Trans. on PAMI*, Vol. 6, pp. 13-27, Jan. 1984.

[5] S.D. Blostein and T.S. Huang, "Estimating motion from range data," First Conference on Artificial Intelligence Application, *IEEE Computer Society*, pp. 246-250, Dec. 1984.

[6] T.S. Huang and S.D. Blostein, "Robust algorithm for motion estimation based on two sequential stereo range pairs," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 518-523, 1985.

[7] O.D. Faugeras and M. Hebert, "A 3-D reconstruction and positioning algorithm using geometrical matching between primitive surfaces," *Proc. IJCAI-83*, the Eighth International Joint Conference on Artificial Intelligence, Karlsrhhe, West Germany, August, 1983.

[8] Z. Lin, H. Lee and T.S. Huang, "Finding 3-D point correspondences in motion estimation," submitted to *the Eighth International Conference on Pattern Recognition*, 1986.

PART II

LEAST-SQUARES ESTIMATION OF MOTION PARAMETERS FROM 3-D POINT CORRESPONDENCES

T.S. Huang, S.D. Blostein and E.A. Margerum

I. INTRODUCTION

In many computer vision applications, notably the estimation of motion parameters of a rigid object using 3-D point correspondences[1] and the determination of the relative attitude of a rigid object with respect to a reference[2], we encounter the following mathematical problem. We are given two 3-D point sets $p_i = [x_i, y_i, z_i]^t$, $p'_i = [x'_i, y'_i, z'_i]^t$, $i = 1, 2, \ldots, N$, (Here, p_i and p'_i are considered as 3×1 column matrices. The superscript t denotes transposition.) which are related by

$$p'_i = Rp_i + T + N_i \quad (1)$$

where R is a 3×3 rotation matrix, T is a translation vector (3×1 column matrix), and N_i a noise vector. We want to find R and T to minimize

$$\Sigma^2 = \sum_{i=1}^{N} \| p'_i - (Rp_i + T) \|^2 \quad (2)$$

A brute force way of finding the solution is to express Σ^2 in terms of the six motion parameters (3 from T and 3 from R) and carry out the minimization by standard iterative methods. (We note that the elements of the rotation matrix R can be expressed in terms of, e.g., the 3 Euler angles, or two components of the direction cosines of the rotation axis plus the angle of rotation.) However, Σ^2 is a highly nonlinear function of the 6 unknowns so that standard iterative methods may or may not converge to the correct solution depending on the initial guess. In fact, we have tried a Newton-type algorithm and found that it often converges to a point where the first derivative of Σ^2 with respect to the 6 unknowns are zero but which is not the global minimum.

In this paper, we describe a two-part algorithm which involves iteration. However, the iteration procedure always converges to the correct solution.

II. A CENTROID COINCIDENCE THEOREM

We show that the determination of R and T can be decoupled because of the following

Theorem 1. Let $\hat{R}$ and $\hat{T}$ be the least-squares solution to Eq. (1), i.e., they minimize Σ^2 defined in Eq. (2). Then the centroids of $\{p'_i\}$ and $\{\hat{R}p_i+\hat{T}\}$ coincide.

Proof: Let

$$p''_i = [x''_i, y''_i, z''_i]^t \triangleq \hat{R}p_i+\hat{T} \tag{3}$$

and

$$p \triangleq \frac{1}{N}\sum_{i=1}^{N} p_i \text{ (Centroid of } \{p_i\})$$

$$p' \triangleq \frac{1}{N}\sum_{i=1}^{N} p'_i \text{ (Centroid of } \{p'_i\})$$

$$p'' \triangleq \frac{1}{N}\sum_{i=1}^{N} p''_i \text{ (Centroid of } \{p''_i\}).$$

We restate the least-squares problem of Section I as follows:

$$\text{Given} - \{p_i\},\{p'_i\}; i = 1,2,\ldots,N$$

$$\text{Find} - \{p''_i\}$$

to minimize

$$\Sigma^2 = \sum_{i=1}^{N} \| p'_i - p''_i \|^2$$

subject to the rigidity constraints

$$\| p''_i - p''_j \|^2 = \| p_i - p_j \|^2, \text{for all } i \text{ and } j.$$

We can attempt to solve this problem using Lagrangian multipliers. Let

$$F = \sum_{i=1}^{N} \| p'_i - p''_i \|^2 + \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_{ij}\{ \| p''_i - p''_j \|^2 - \| p_i - p_j \|^2\}$$

where $\lambda_{ij} = \lambda_{ji}$. Differentiating F with respect to x''_i,

$$\frac{\partial F}{\partial x''_i} = 0 = 2(x''_i - x'_i) + \sum_{j=1}^{N} \lambda_{ij} 2(x''_i - x''_j)$$

Summing over i=1 to N, we get

$$\sum_{i=1}^{N}(x''_i - x'_i) = 0$$

Similarly for the y and the z components. Therefore

$$p' = p'' = \hat{R}p+T \tag{4}$$

QED

This centroid coincidence theorem implies that we can simplify our original least-squares problem by first translating the point set $\{p_i\}$ by

$$T_C \triangleq p' - p$$

so that at the new position this point set will have a centroid which is coincident with that of $\{p'_i\}$. Then we have only $\hat{R}$ to solve for.

Formally, let

$$q_i \triangleq p_i - p$$

$$q'_i \triangleq p'_i - p'$$

then

$$p'_i - (Rp_i+T) = q'_i + p' - Rq_i - RP - T = q'_i - Rq_i$$

by virtue of Eq. (4).

Thus, Eq. (2) becomes

$$\Sigma^2 = \sum_{i=1}^{N} \| q'_i - Rq_i \|^2 \tag{5}$$

We note that in Eq. (3) (and in Eq. (1)) we have assumed implicitly that the rotation $\hat{R}$ is around an axis passing through the origin of our coordinate system. The translation $\hat{T}$ there is not the same as T_C. In fact, it is ready shown that

$$\hat{T} = p' - \hat{R}p \tag{6}$$

III. FINDING $\hat{R}$: PLANAR CASE

The original least-squares problem has been simplified to finding R to minimize Σ^2 in Eq. (5). A closed form solution exists for the case where $\{p_i\}$ and $\{p'_i\}$ lie on a plane. Without loss of generality, let us assume that they lie on the x-y plane. Then so do $\{q_i\}$ and $\{q'_i\}$. To simplify notations, we write

$$q_i = [x_i, y_i]^t = [r_i\cos\alpha_i, r_i\sin\alpha_i]^t$$

$$q'_i = [x'_i, y'_i]^t = [\rho_i\cos\beta_i, \rho_i\sin\beta_i]^t$$

Then Eq. (5) becomes

$$\Sigma^2 = \sum_{j=1}^{N} \| \rho_j e^{i\beta_j} - r_j e^{i(\alpha_j+\sigma)} \|^2 \tag{7}$$

where σ is the angle of rotation.

Theorem 2. Over $0 \leqslant \sigma < 2\pi$, Σ^2 attains exactly one local minimum (which is the global minimum) and one local maximum (which is the global maximum). The angles σ at which Σ^2 attains the minimum and the maximum satisfy

$$\tan\sigma = \frac{\sum_{i=1}^{N} \rho_i r_i \sin(\beta_i - \alpha_i)}{\sum_{i=1}^{N} \rho_i r_i \cos(\beta_i \alpha_i)} = \frac{\sum_{i=1}^{N} (x_i y'_i - x'_i y_i)}{\sum_{i=1}^{N} (x_i x'_i + y_i y'_i)} . \qquad (8)$$

Proof: The result is readly seen, if one examines

$$\frac{\partial \Sigma^2}{\partial \sigma} \text{ and } \frac{\partial^2 \Sigma^2}{\partial \sigma^2}.$$

QED

The procedure of finding the least-squares solution $\hat{\sigma}$ is to first use Eq. (8) to find two values of σ, and then substitute these into Eq. (7) to determine which values give the minimum.

IV. FINDING $\hat{R}$: GENERAL CASE

For the general case, where $\{q_i\}$,$\{q'_i\}$ does not lie on a plane, the result of Section III can be used to formulate an iterative procedure for finding the minimum of Σ^2 in Eq. (5).

We decompose the rotation matrix R into 3 successive rotations around the z-, x-, and y-axis, respectively, and by angles σ,ϕ, and x respectively[3]:

$$R = R_y(\psi) R_x(\phi) R_z(\theta) \qquad (9)$$

Eq. (5) now becomes

$$\Sigma^2(\psi,\phi,\theta) = \sum_{i=1}^{N} \| q'_i - R_y(\psi) R_x(\phi) R_z(\theta) q_i \|^2 \qquad (10)$$

The approach is to minimize Σ^2 with respect to one variable at a time with the remaining two fixed. This minimization is a 2-D problem and can be solved by using theorem 2. We proceed as follows:

(i) Let ϕ_o,θ_0 be the initial guesses for ϕ,θ, respectively. Then we find ψ to minimize $\Sigma^2(\psi,\phi_o,\theta_0)$. This is a 2-D problem where the two sets of 2-D points are projections of $\{q'_i\}$ and $\{R_x(\phi_o)R_z(\theta_o)q_i\}$ on the x-z plane.

Call the solution ψ_1.

(ii) Find ϕ to minimize

$$\Sigma^2(\psi_1,\phi,\theta_0) = \sum_{i=1}^{N} \| R_y^{-1}(\psi_1) q'_i - R_x(\phi) R_z(\theta_0) q_i \|^2 .$$

This is again a 2-D problem where the two sets of 2-D points are projections of $\{R_y^{-1}(\psi_1)q'_i\}$ and $\{R_z(\theta_0)q_i\}$ on the y-z plane.

Call the solution ϕ_1.

(iii) Find θ to minimize

$$\Sigma^2(\psi_1,\phi_1,\theta) = \sum_{i=1}^{N} \| R_x^{-1}(\phi_1) R_y^{-1}(\psi_1) q'_i - R_z(\theta) q_i \|^2 .$$

This is again a 2-D problem where the two set of 2-D points are projections of $\{R_x^{-1}(\phi_1)R_y^{-1}(\psi_1)q'_i\}$ and $\{q_i\}$ on the x-y plane.

Call the solution θ_1.

At this stage, we have finished one cycle of iteration, with the resulting solution (ψ_1,ϕ_1,θ_1).

We repeat steps (i), (ii), and (iii) until some stopping criterion is met, e.g., when $\Sigma^2(\psi_k,\phi_k,\theta_k)$ is less than some preset threshold. Then our solution is $(\hat{\psi},\hat{\phi},\hat{\theta}) = (\psi_k,\phi_k,\theta_k)$.

Conditions for which $\Sigma^2(\psi,\phi,\theta)$ has one and only one local minimum, the global minimum, will be briefly discussed in Section VII of this paper. On trying a Newton-type algorithm in which minimization took place with respect to the 6 unknown motion parameters, it was discovered that $\Sigma^2(\psi,\phi,\theta)$ does have several saddle points. Since at each step our algorithm attains the global minimum in 2-D, it will not be trapped at such a saddle point. In fact, in hundreds of examples we have tried the algorithm on, it always converges to the current solution.

V. SUMMARY OF ALGORITHM

To find $\hat{R}$ and $\hat{T}$ to minimize Σ^2 in Eq. (2), we proceed as follows.

1) calculate the centroids p andp' of $\{p_i\}$ and $\{P'_i\}$ respectively.

2) calculate

$$q_i = p_i - p \qquad q'_i = p'_i - p' \qquad (i=1,2,\ldots,N)$$

3) use the iterative algorithm of Section IV to find $(\hat{\psi},\hat{\phi},\hat{\theta})$. Then

$$\hat{R} = R_y(\hat{\psi}) R_x(\hat{\phi}) R_z(\hat{\theta}) .$$

4) and

$$\hat{T} = p' - \hat{R}p .$$

VI. COMPUTER SIMULATION RESULTS

Computer simulations have been carried out on a VAX 11/780 to test our algorithm. Two measures of performance were tested. First, the running time of the algorithm as a function of the number of point correspondences was determined. Second, the sensitivity of the algorithm's rotation parameter estimates to noise in the point positions was investigated.

In order to measure the computer CPU time used by the algorithm, six sets of 3-D points $\{p_i\}$ were generated of sizes 3, 7, 11, 16, 20 and 30 respectively. Each point was located arbitrarily inside a 6×6×6 cube centered at the origin. Then $\{p'_i\}$ were calculated for each of the six sets by rotating $\{p_i\}$ by an angle of 75° around an axis through the origin with direction cosines (0.6 , 0.7 , 0.39). No translation was added to this motion since it can be trivially removed at the first step by calculating centroids. Instead, the iterative determination of the three rotation angles was explored. For all experiments, the initial guessses for the rotation angles θ, ϕ, and ψ were taken to be zero. Table I shows the running times for each of the six point sets. Beside each time is the number of iterations taken for the change in two successive Σ^2 values to be less than 0.001. This corresponds to an accuracy of about 0.5% in the rotation angles. In a second experiment, Gaussian random noise of mean zero and standard deviation 0.5 was added to each coordinate of $\{p_i\}$ and $\{p'_i\}$. The resulting CPU times shown in Table II are similar to those reported in the first experiment. For the sake of accuracy, the simulation times appearing in the two tables were calculated from an average value of 500 runs. For the noiseless case and the set of 16 points, the succesive values of θ, ϕ, ψ, and Σ^2 for each iteration are reported in Table III.

The algorithm's sensitivity to noise was experimentally determined for the set of 16 points considered in the previous experiments. Zero mean Gaussian random noise ($\sigma = 0.5$), was added to the point set in 1000 separate trials of the algorithm. Statistics based on the resulting rotation angles are listed in Table IV. As shown the average % error in the motion parameters was about 3.0. For trials with errors three standard deviations away from this average error, the average % error was about 10.0. Since the noise was Gaussian, this corresponds to 99.7% of the trials having relative errors of less than 10%. Considering that the average perturbation of a point coordinate by the added noise was 11.5%, the algorithm can be said to be quite robust.

Table I
Time per run (millisecs). Noiseless Case.

# pts	msec	# iterations
3	226.6	47
7	124.2	14
11	104.8	8
16	95.2	5
20	113.8	5
30	207.2	6

Table II
Time per run (ms) with added Zero mean, (σ=0.5) Gaussian Noise added

#pts	msec	#iterations
3	126.8	25
7	108.2	12
11	105.2	8
16	94.2	5
20	135.0	6
30	111.0	6

Table III
Rotation Angles at each Iteration for 16 point, Noiseless Case.

Iteration	θ	ϕ	ψ	Σ^2
0	0.0000	0.0000	0.0000	479.3
1	-0.8080	-0.6569	-0.7497	30.46
2	-1.0681	-0.5412	-0.8891	0.9215
3	-1.1183	-0.5299	-0.9122	0.0241
4	-1.1265	-0.5284	-0.9159	0.0006
5	-1.1279	-0.5282	-0.9164	0.0000

Table IV
Performance Statistics* for Estimated Parameters (in added N (0,0.5) noise).

1000 trials	θ	ϕ	ψ
bias	-0.00416	0.00073	-0.00001
standard dev	0.0330	0.0245	0.0344
avg % error	2.348	3.629	2.989
% error, 99.7% of trials within 8.808	13.89	11.16	

* for size 16 point set, avg number of iterations = 5

VII. DISCUSSIONS

After the completion of the work reported in this paper, it was brought to our attention that an algorithm developed by Faugeras and Hebert[4] in a different context (that of matching piecewise - planar surfaces in 3-D object recognition) can be used as a noniterative method of finding $\hat{R}$ to minimize Σ^2 of Eq. (5). Their elegant approach uses a quaternion formulation. A 3-D rotation can be represented by a unit quaternion. In terms of this quaternion, it can be shown that the function Σ^2 can be expressed in a quadratic form. The quaternion that achieves the minimum value of Σ^2 is then the eigenvector of the 4×4 matrix of the quadratic form associated with the smallest eigenvalue. The minimum value of Σ^2 is simply equal to this smallest eigenvalue. Furthermore, Σ^2 will have a unique, global minimum if the eigenvalues of the 4×4 matrix are distinct. Our experiments have shown that the 4×4 matrix has repeated eigenvalues where each of the two point sets are collinear. In such cases, it is clear geometrically that $\hat{R}$ and $\hat{T}$ are not unique.

In the meantime, Professor K.S. Arun[5] of the University of Illinois also independently developed a noniterative method of determining $\hat{R}$ based on the singular value decomposition of a 3×3 matrix.

Work is currently underway to compare the three methods (one iterative, two noniterative) with respect to computer time requirements. We hope to be able to report on the results in the near future.

REFERENCES

1. S.D. Blostein and T.S. Huang, Estimating 3-D Motion from range data, Proc. 1st Conf. Artificial Intelligence Applications, pp. 246-250, Denver, CO, Dec. 1984.
2. D. Cyganski and J.A. Orr, Applications of Tensor Theory to object recognition and Orientation determination, IEEE Trans. on PAMI, Vol. 7, pp. 663-673, Nov. 1985.
3. R. Paul, *Robot Manipulators*, MIT Press, 1981.
4. O.D. Faugeras and M. Hebert, A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces, Proc. Int. Joint Conf. Artificial Intelligence, pp. 996-1002, Karlshure, West Germany, August 1983.
5. K.S. Arun, Least-Squares Fitting of Two 3-D Point Sets, unpublished Memo, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, October 1985.