An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 40 (2000) 1171–1188
www.elsevier.nl/locate/camwa

Iterative Closest Geometric Objects Registration

QINGDE LI AND J. G. GRIFFITHS
Department of Computer Science
University of Hull, HU6 7RX, UK
q.li@dcs.hull.ac.uk

(Received December 1998; revised and accepted May 2000)

Abstract—In this paper, closed-form solutions are obtained for registering two sets of line segments, triangle patches, or even general simple geometric objects that are defined by a set of ordered points. Based on these new registration approaches, the iterative closest line segment registration (ICL) algorithm and the iterative closest triangle patch registration (ICT) algorithm are developed similar to the ICP algorithm. To simplify the mathematical representation, the concept of matrix scalar product is defined and some of its properties are given. The newly developed registration methods are tested. The test shows that the ICL algorithm and the ICT algorithm work much better than the conventional ICP algorithm considering that the ICL and the ICT algorithms are much less sensitive to the initial orientations of the object. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords—Rotation estimation, Matrix scalar product, Iterative line segment registration.

1. INTRODUCTION

In computer assisted surgery, one of the most important problems is to align the preoperative model with intraoperative data. Mathematically, this is a problem of estimating the coordinate transformations, usually involving rotation and translation, between the two coordinate systems in which the preoperative data and intraoperative data are presented. Generally speaking, what method is used to estimate the unknown transformation depends on whether point-to-point corresponding relations between the data sets are known or not. When the exact correspondence between the data sets is known, the data sets are called reference marks and the relevant registration approaches are called reference mark registrations, which are very quick and very accurate. However, to collect reference point data sets, external landmarks have to be implanted into the position where surgery will be carried out. This is invasive, and may bring about further harm to patients.

Another kind of registration data consists of very general data sets. The only information available is that the two data sets are collected from the same surface of a rigid object. Often one data set (called model data) has far more points than the other one (intraoperative data). As no point-to-point correspondence information is known about the data sets, nonlandmark registration techniques are required to match the two data sets, which are much more complicated

PII: S0898-1221(00)00230-3

than landmark registration techniques. This is because an iterative optimization procedure is inevitable in this case. For nonlandmark registration, two kinds of registration methodologies are very popular. One can be classified as the 'hat and head' registration approach. This treats one data set as the hat and another data set as the head, and thus, the data matching problem is just the matter of where to put the hat on the head. More precisely, let $\mathbb{P} = \{P_i\}_{i=1}^{N}$ and $\mathbb{Q} = \{Q_j\}_{j=1}^{M}$ be two data sets that are going to be matched. Let the unknown rotation and translation that link the two data sets be R and T. Then R and T can be estimated by minimizing the sum

$$\sum_{i=1}^{N} d(RP_i + T, \mathbb{Q}),$$

where $d(P, \mathbb{Q})$ is the distance from a point P to the data set $\mathbb{Q}$ defined by

$$d(P, \mathbb{Q}) = \min_{1 \le j \le M} \|P - Q_j\| .$$

This approach is direct in idea but is time consuming and computationally expensive. Though great improvement has been made, it will still take some time before it becomes useful in practice. Another kind of registration method can be classified as iterative closest point registration (ICP), developed from the method given in [1]. These approaches are realized by iteratively calculating the closest points in model data for each intraoperative data point. In this way, the nonlandmark registration problem is turned into an iterative landmark registration problem. The ICP algorithm is always convergent and a transformation can always be found. The problem is that in many cases, it does not converge to the expected transformation. More general discussion on medical data registration techniques can be found in [2,3]. In this paper, approaches for matching two sets of line segments or two sets of triangle patches are developed first. Based on these techniques, algorithms similar to ICP are proposed. Instead of searching for the closest point corresponding to each data point, a closest line segment (or triangle patch) in model data is calculated for each line segment (or triangle patch) in the intraoperative data. Our test shows that in most cases, it will converge to the expected position. This paper is organized in the following way. In Section 2, the concept of matrix scalar product is defined by which the relative mathematical representations will be comparatively simple and can reflect geometric intuition. In Sections 3 and 4, line segment and triangle patch registration techniques are provided. In Section 5, a nonlandmark registration technique is provided similar to the ICP. In Section 6, test results are presented.

2. THE MATRIX SCALAR PRODUCT AND ITS PROPERTIES

Definition 1. *Let $A = (a_{ij})$ and $B = (b_{ij})$ be two $n \times m$ matrices. The scalar product of the two matrices, denoted by $A \cdot B$, is defined as*

$$A \cdot B = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} b_{ij}. \tag{1}$$

It is evident that this definition is a natural generalization of the vector scalar product.

Let A,B be $n \times m$ real matrices. Let A_{1*}, $A_{2*}, \dots, A_{n*}$ and A_{*1}, $A_{*2}, \dots, A_{*m}$ be the row vectors and column vectors of A and let B_{1*}, $B_{2*}, \dots, B_{n*}$ and B_{*1}, $B_{*2}, \dots, B_{*m}$ the row vectors and column vectors of B.

Proposition 1.

$$\begin{aligned} A \cdot B &= A_{1*} \cdot B_{1*} + A_{2*} \cdot B_{2*} + \cdots + A_{n*} \cdot B_{n*} \\ &= A_{*1} \cdot B_{*1} + A_{*2} \cdot B_{*2} + \cdots + A_{*m} \cdot B_{*m}. \end{aligned}$$

PROPOSITION 2.

$$\mathbf{I} \cdot A = \mathrm{tr} A, \tag{2}$$

$$A \cdot B = B \cdot A, \tag{3}$$

$$(A + B) \cdot C = A \cdot C + B \cdot C, \tag{4}$$

$$A \cdot B = \mathrm{tr} A^{\top} B = \mathrm{tr} A B^{\top} = \mathrm{tr} B^{\top} A = \mathrm{tr} B A^{\top}, \tag{5}$$

$$A \cdot A = \|A\|^2, \tag{6}$$

where $\| \cdot \|$ denotes Frobenius norm and in (2) $\mathbf{I}$ is an $n \times n$ identity matrix.

PROPOSITION 3. *Let A be an $n \times m$ matrix, and let X be an m-dimensional vector, and Y an n-dimensional vector. Then*

$$Y \cdot (AX) = A \cdot \left(YX^{\top}\right). \tag{7}$$

PROOF.

$$Y \cdot (AX) = \sum_{i=1}^{n} y_i \sum_{j=1}^{m} r_{ij} x_j = \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} y_i x_j = A \cdot \left(YX^{\top}\right).$$ ∎

PROPOSITION 4. *Let A,B,C be $n \times m$, $n \times k$, and $k \times m$ matrices, respectively. Then*

$$A \cdot (BC) = B \cdot \left(AC^{\top}\right). \tag{8}$$

PROOF.

$$A \cdot (BC) = \sum_{j=1}^{m} A_{*j} \cdot (BC_{*j}) = \sum_{j=1}^{m} B \cdot \left(A_{*j} C_{*j}^{\top}\right) = B \cdot \left(\sum_{j=1}^{m} A_{*j} C_{*j}^{\top} \right)$$

and

$$\sum_{j=1}^{m} A_{*j} C_{*j}^{\top} = AC^{\top}$$

follows directly. ∎

PROPOSITION 5. *Let A,B be $n \times m$, $n \times m$ matrices*

$$\|A - B\|^2 = \|A\|^2 + \|B\|^2 - 2A \cdot B, \tag{9}$$

where $\| \cdot \|$ denotes the Frobenius Norm.

PROOF. According to the definition of the Frobenius norm, we have

$$\begin{aligned} \|A - B\|^2 &= \sum_{j=1}^{m} \|A_{*j} - B_{*j}\|^2 \\ &= \sum_{j=1}^{m} \|A_{*j}\|^2 + \|B_{*j}\|^2 - 2A_{*j} \cdot B_{*j} \\ &= \|A\|^2 + \|B\|^2 - 2A \cdot B. \end{aligned}$$ ∎

COROLLARY 1. *If R is a real orthogonal matrix, then*

$$\|A - RB\|^2 = \|A\|^2 + \|B\|^2 - 2R \cdot \left(AB^{\top}\right). \tag{10}$$

PROOF. The proof follows directly from Propositions 4 and 5. ∎

3. CLOSED-FORM LINE SEGMENT REGISTRATION

DEFINITION 2. *Let P_1, $P_2 \in \mathbb{R}^3$ be two points. The ordered pair $[P_1, P_2]$ is called a line segment in $\mathbb{R}^3$. The set of all line segments on $\mathbb{R}^3$ is denoted as $\mathcal{L}$.*

DEFINITION 3. *Let $\mathbb{L} \in \mathcal{L}$ be a line segment in $\mathbb{R}^3$, F is a transformation on space $\mathbb{R}^3$. Then this transformation can be extended to be a line segment transformation by defining*

$$F[P_1, P_2] = [FP_1, FP_2]. \tag{11}$$

$[FP_1, FP_2]$ is called the transformation of line segment $\mathbb{L}$. For translation, we will write $T[P_1, P_2] = [P_1 + T, P_2 + T]$ more naturally as $[P_1, P_2] + T$.

Similarly, the other operations on $\mathbb{R}^3$ can also be extended to line segments.

DEFINITION 4. *Let $\mathbb{L}_1 = [P_1, P_2]$, $\mathbb{L}_2 = [Q_1, Q_2] \in \mathcal{L}$ be two line segments, and a, b two real numbers. We define*

$$a\mathbb{L}_1 + b\mathbb{L}_2 = [aP_1 + bQ_1, aP_2 + bQ_2]. \tag{12}$$

It should be noted that this definition is different from the set operation obtained with the conventional extension principle.

Let $\mathbb{L}_1 = [P_1, P_2], \mathbb{L}_2 = [Q_1, Q_2] \in \mathcal{L}_2$ be two line segments. Geometrically, $\mathbb{L}_1$ and $\mathbb{L}_2$ can be represented as functions in the form: $f_1(\lambda) = P_1 + \lambda(P_2 - P_1)$ and $f_2(\lambda) = Q_1 + \lambda(Q_2 - Q_1)$, respectively, where $0 \le \lambda \le 1$. If λ is incremented by $d\lambda$, then f_1 and f_2 are incremented by $(P_2 - P_1)d\lambda$ and $(Q_2 - Q_1)d\lambda$, respectively. The distance between these two micro-line segments can be approximated by the area of the trapezium that has height $\|f_1(\lambda) - f_2(\lambda)\|^2$ with top-edge and bottom edge defined by $(P_2 - P_1)d\lambda$ and $(Q_2 - Q_1)d\lambda$ approximately, i.e.,

$$\frac{1}{2}(\|P_2 - P_1\| + \|Q_2 - Q_1\|)\|f_1(\lambda) - f_2(\lambda)\|^2 d\lambda.$$

We choose to use $\|f_1(\lambda) - f_2(\lambda)\|^2$ rather than $\|f_1(\lambda) - f_2(\lambda)\|$ to measure the distance between points $f_1(\lambda)$ and $f_2(\lambda)$ only for the convenience of computation. The distance between the two line segments can thus, be described by the following integration:

$$\begin{gathered}\frac{l_1 + l_2}{2} \int_0^1 \|f_1(\lambda) - f_2(\lambda)\|^2 \, d\lambda = \frac{l_1 + l_2}{6} \\ \cdot \left(\|P_1 - Q_1\|^2 + \|P_2 - Q_2\|^2 + (P_1 - Q_1) \cdot (P_2 - Q_2)\right),\end{gathered} \tag{13}$$

where $l_1 = \|P_2 - P_1\|$, $l_2 = \|Q_2 - Q_1\|$.

DEFINITION 5. *Let $\mathbb{L}_1 = [P_1, P_2], \mathbb{L}_2 = [Q_1, Q_2]$ be two line segments. The distance between the two line segments is defined as (13) and is denoted by $D(\mathbb{L}_1, \mathbb{L}_2)$.*

It should be noted that the value of the above integration depends on the corresponding relations between the ends of the two line segments. Thus, the distance between two line segments defined above is direction dependent.

In paper [4], the distance between line segments has been defined for the case where the lengths of the line segments are equal. We will show that our definition is more general.

PROPOSITION 6. *Let $\mathbb{L}_1 = [P_1, P_2]$, $\mathbb{L}_2 = [Q_1, Q_2] \in \mathcal{L}$ be two line segments. Then*

$$D(\mathbb{L}_1, \mathbb{L}_2) = \frac{l_1 + l_2}{2}\left(\|O_1 - O_2\|^2 + \frac{l_1 l_2}{12}\|V_1 - V_2\|^2 + \frac{1}{12}(l_1 - l_2)^2\right), \tag{14}$$

where $O_1 = (P_1 + P_2)/2$, $O_2 = (Q_1 + Q_2)/2$ are the centers of the two line segments, the unit vectors V_1, V_2 are their directions and l_1, l_2 their lengths.

The geometric meaning of the measure is clear. The first term of equation (14) measures the difference between the two line segments in position, the second term measures the difference in direction, and the third term measures the difference in length.

PROOF.

$$
\begin{aligned}
\|O_1 - O_2\|^2 &= \frac{1}{4}\left(\|P_1 - Q_1\|^2 + \|P_2 - Q_2\|^2 + 2(P_1 - Q_1)\cdot(P_2 - Q_2)\right),\\
l_1 l_2 \|V_1 - V_2\|^2 &= 2 l_1 l_2 - 2(P_2 - P_1)\cdot(Q_2 - Q_1),\\
l_1 l_2 \|V_1 - V_2\|^2 + (l_1 - l_2)^2 &= \|P_1 - Q_1\|^2 + \|P_2 - Q_2\|^2 - 2(P_1 - Q_1)\cdot(P_2 - Q_2),
\end{aligned}
$$

and so

$$
\begin{aligned}
&\frac{l_1 + l_2}{2}\left(\|O_1 - O_2\|^2 + \frac{l_1 l_2}{12}\|V_1 - V_2\|^2 + \frac{1}{12}(l_1 - l_2)^2\right)\\
&\qquad = \frac{l_1 + l_2}{6}\left(\|P_1 - Q_1\|^2 + \|P_2 - Q_2\|^2 + (P_1 - Q_1)\cdot(P_2 - Q_2)\right). \quad \blacksquare
\end{aligned}
$$

When $l_1 = l_2 = l$, we have

$$
\begin{aligned}
D(\mathbb{L}_1, \mathbb{L}_2) &= l\|O_1 - O_2\|^2 + \frac{l^3}{12}\|V_1 - V_2\|^2\\
&= l\|O_1 - O_2\|^2 + \frac{l^3}{6}(1 - V_1 \cdot V_2).
\end{aligned}
$$

This is the definition given in [4].

Though (14) is more direct than (13), we will mainly use equation (13) as it is easier to compute from points.

Next, we will establish a line segment registration algorithm. Let

$$
\{[P_n, P'_n]\}_{n=1}^{N}, \qquad \{[Q_n, Q'_n]\}_{n=1}^{N}
$$

be two sets of line segments. Suppose that there exists a rotation R and a translation T such that

$$
[Q_n, Q'_n] = R[P_n, P'_n] + T + [\varepsilon_n, \varepsilon'_n], \qquad n = 1, 2, \dots, N,
$$

where $[\varepsilon_n, \varepsilon'_n], n = 1, 2, \dots, N$, are experimental errors. We estimate R and T by minimizing

$$
\begin{aligned}
\Sigma = \sum_{n=1}^{N} \frac{l_{1_n} + l_{2_n}}{6}\Big(&\|Q_n - RP_n - T\|^2 + \|Q'_n - RP'_n - T\|^2\\
&+ (Q_n - RP_n - T)\cdot(Q'_n - RP'_n - T)\Big),
\end{aligned}
\tag{15}
$$

where l_{1_n}, l_{2_n} are the lengths of $[P_n, P'_n]$ and $[Q_n, Q'_n]$, respectively.

Let $\frac{\partial \Sigma}{\partial T} = 0$. It follows immediately that the optimal translation should be chosen as

$$
T = \bar{Q} - R\bar{P}, \tag{16}
$$

where

$$
\begin{aligned}
\bar{P} &= \sum_{n=1}^{N} w_n \frac{P_n + P'_n}{2}, &\qquad \bar{Q} &= \sum_{n=1}^{N} w_n \frac{Q_n + Q'_n}{2},\\
w &= \sum_{n=1}^{N} (l_{1_n} + l_{2_n}), & w_n &= \frac{l_{1_n} + l_{2_n}}{w}.
\end{aligned}
\tag{17}
$$

Let

$$
[\bar{P}_n, \bar{P}'_n] = [P_n, P'_n] - \bar{P}, \qquad [\bar{Q}_n, \bar{Q}'_n] = [Q_n, Q'_n] - \bar{Q},
$$

$n = 1, 2, \ldots, N$. Substituting (16) into (15), we have

$$\Sigma = \sum_{n=1}^{N} \frac{l_{1_n} + l_{2_n}}{6} \left(\left\| \bar{Q}_n - R\bar{P}_n \right\|^2 + \left\| \bar{Q}'_n - R\bar{P}'_n \right\|^2 + (\bar{Q}_n - R\bar{P}_n) \cdot (\bar{Q}'_n - R\bar{P}'_n) \right). \tag{18}$$

This sum can be further written as

$$\Sigma = \Delta - R \cdot \sum_{n=1}^{N} \frac{l_{1_n} + l_{2_n}}{6} \left(2\bar{Q}_n \bar{P}_n^{\top} + 2\bar{Q}'_n \bar{P}'^{\top}_n + \bar{Q}_n \bar{P}'^{\top}_n + \bar{Q}'_n \bar{P}_n^{\top} \right), \tag{19}$$

where

$$\Delta = \sum_{n=1}^{N} \frac{l_{1_n} + l_{2_n}}{6} \left(\left\| \bar{Q}_n \right\|^2 + \left\| \bar{P}_n \right\|^2 + \left\| \bar{Q}'_n \right\|^2 + \left\| \bar{P}'_n \right\|^2 + \bar{Q}_n \cdot \bar{Q}'_n + \bar{P}_n \cdot \bar{P}'_n \right)$$

is a constant independent of rotation R. Therefore, minimizing (18) is equivalent to maximizing

$$\Gamma = R \cdot A, \tag{20}$$

where matrix

$$A = \sum_{n=1}^{N} \frac{l_{1_n} + l_{2_n}}{6} \left(2\bar{Q}_n \bar{P}_n^{\top} + 2\bar{Q}'_n \bar{P}'^{\top}_n + \bar{Q}_n \bar{P}'^{\top}_n + \bar{Q}'_n \bar{P}_n^{\top} \right). \tag{21}$$

Now, there are several ways to find the optimal R from (20). This paper only considers two methods. One way is to use Eigen-techniques. Another way is to use the singular value decomposition technique. We first discuss how the eigen-technique can be used to find the rotation R.

When the rotation matrix R is represented with Euler-Rodrigus parameters in form [5]

$$R = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \tag{22}$$

with $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$, $R \cdot A$ in (20) can be directly written as a quadratic form

$$\Gamma = \mathbf{q}^{\top} \mathbf{Q} \mathbf{q}, \tag{23}$$

where $\mathbf{q} = (q_0, q_1, q_2, q_3)^{\top}$ and $\mathbf{Q}$ is a 4×4 matrix constructed from A by [1]

$$\mathbf{Q} = \begin{pmatrix} \mathrm{tr}(A) & \mathbf{b}^{\top} \\ \mathbf{b} & A + A^{\top} - \mathrm{tr}(A)\mathbf{I}_3 \end{pmatrix}, \tag{24}$$

where $\mathbf{b} = (a_{32} - a_{23}, a_{13} - a_{31}, a_{21} - a_{12})^{\top}$ is a column vector and $\mathbf{I}_3$ is the 3×3 identity matrix. The maximum value of quadratic form (23) subject to $\|\mathbf{q}\| = 1$ is attained with the unit eigenvector of $\mathbf{Q}$ corresponding to its greatest eigenvalue [6]. This gives the eigen-algorithm (also known as the quaternion algorithm). A similar technique has been used by Faugeras in [7] and a more comprehensive discussion in using quaternions to solve reference point matching problems can be found in [8].

Another way to compute the optimal R from (20) is the SVD algorithm which uses the singular value decomposition technique to calculate the rotation matrix. Instead of writing $R \cdot A$ as a quadratic form, it can be written in the form

$$\Gamma = R_{*1} \cdot A_{*1} + R_{*2} \cdot A_{*2} + R_{*3} \cdot A_{*3}. \tag{25}$$

This representation immediately shows that to maximize Γ, the three mutually orthogonal column vectors of rotation matrix R should be chosen to be as close as possible to the column vectors of matrix A. This is equivalent to finding the closest rotation matrix to A in the sense of the Frobenius norm. Let the singular value decomposition of matrix A in (21) be $UWV^{\top}$, with $UU^{\top} = VV^{\top} = I$, $W = \mathrm{dia}(w_1, w_2, w_3)$, and $w_1 \geq w_2 \geq w_3 \geq 0$. It is known [9–11] that the closest rotation matrix to A is $UV^{\top}$ when $\det A \geq 0$, and is $UJV^{\top}$ when $\det A < 0$, where $J = \mathrm{dia}(1, 1, -1)$. This gives the SVD algorithm.

4. CLOSED-FORM TRIANGULAR PATCH REGISTRATION

In this section, the concept of a triangle patch is introduced. The contents of this section can then be treated like that of the previous section. Triangle patches can be seen as a generalization of the line segments introduced in the previous section. The main results of this section can be further generalized to an ordered point set of any size without any difficulty.

DEFINITION 6. *Let $P_1,P_2,P_3 \in \mathbb{R}^3$ be three points. The ordered triple $[P_1,P_2,P_3]$ is called a triangle patch in $\mathbb{R}^3$. The set of all triangle patches on $\mathbb{R}^3$ is denoted as $\mathcal{T}$.*

DEFINITION 7. *Let $\mathbb{T} \in \mathcal{T}$ be a triangle patch in $\mathbb{R}^3$. F is a transformation on space R. Then this transformation can be extended to apply to a triangle patch by means of the definition*

$$F[P_1,P_2,P_3] = [FP_1,FP_2,FP_3]. \tag{26}$$

$[FP_1,FP_2,FP_3]$ is called the transformation of triangle patch $\mathbb{T}$. For translation, we also write

$$T[P_1,P_2,P_3] = [P_1+T,P_2+T,P_3+T] = [P_1,P_2,P_3]+T.$$

Plus and scale operations can also be defined on triangle patches as they are on line segments.

DEFINITION 8. *Let $\mathbb{T}_1 = [P_1,P_2,P_3], \mathbb{T}_2 = [Q_1,Q_2,Q_3] \in \mathcal{T}$ be two triangle patches, and a,b two real numbers. We define*

$$a\mathbb{T}_1 + b\mathbb{T}_2 = [aP_1+bQ_1, aP_2+bQ_2, aP_3+bQ_3]. \tag{27}$$

We now discuss how to introduce the concept of distance between triangle patches to measure their closeness. Let $\mathbb{T}_1 = [P_1,P_2,P_3], \mathbb{T}_2 = [Q_1,Q_2,Q_3] \in \mathcal{T}$ be two triangle patches. Geometrically, they can be represented as functions

$$f_1(u,v) = P_1 + u(P_2-P_1) + v(P_3-P_1)$$

and

$$f_2(u,v) = Q_1 + u(Q_2-Q_1) + v(Q_3-Q_1),$$

respectively, where $0 \le u,\ v \le 1$, and $u+v \le 1$. As with line segments, we define a cylinder with height $\|f_1(u,v)-f_2(u,v)\|^2$. The top surface is defined with triangle $\mathbb{T}_1$ and bottom is defined with triangle $\mathbb{T}_2$, and the volume of the cylinder is used to measure closeness of the two triangles. Given increments du and dv to variable u and v, the cylinder gets an increment

$$C\|f_1(u,v)-f_2(u,v)\|^2\,du\,dv,$$

where

$$\begin{aligned} C = {} & \|(P_2-P_1)\times(P_3-P_1)\| + \|(Q_2-Q_1)\times(Q_3-Q_1)\| \\ & + \|((P_2-P_1)+(Q_2-Q_1))\times((P_3-P_1)+(Q_3-Q_1))\|, \end{aligned} \tag{28}$$

and × denotes vector product of vectors. Thus, the volume is

$$\begin{aligned} \iint_{\mathcal{D}} C\|f_1(u,v)-f_2(u,v)\|^2\,du\,dv = {} & \frac{C}{12}\left(\|P_1-Q_1\|^2 + \|P_2-Q_2\|^2 + \|P_3-Q_3\|^2\right. \\ & \left. +(P_1-Q_1)\cdot(P_2-Q_2) + (P_1-Q_1)\cdot(P_3-Q_3) + (P_2-Q_2)\cdot(P_3-Q_3)\right), \end{aligned} \tag{29}$$

where $\mathcal{D} = \{(u,v) \mid 0 \le u+v \le 1,\ 0 \le u,\ v \le 1\}$.

DEFINITION 9. *Let $\mathbb{T}_1 = [P_1,P_2,P_3]$, $\mathbb{T}_2 = [Q_1,Q_2,Q_3] \in \mathcal{T}$ be two triangle patches. The distance between the two triangle patches is defined as (29) and is denoted by $D(\mathbb{T}_1,\mathbb{T}_2)$.*

It should be noted that the distance between two triangle patches defined above depends on the ordering of their vertices. Following Proposition 6, we have the following.

PROPOSITION 7. *Let $\mathbb{T}_1 = [P_1, P_2, P_3]$, $\mathbb{T}_2 = [Q_1, Q_2, Q_3] \in \mathcal{T}$ be two triangle patches. Then*

$$\begin{aligned} D(\mathbb{T}_1, \mathbb{T}_2) = \frac{C}{2} \Bigg(& \|O_1 - O_2\|^2 \\ & + \frac{1}{36} \left(l_1 l'_1 \|V_1 - V'_1\|^2 + l_2 l'_2 \|V_2 - V'_2\|^2 + l_3 l'_3 \|V_3 - V'_3\|^2 \right) \\ & + \frac{1}{36} \left((l_1 - l'_1)^2 + (l_2 - l'_2)^2 + (l_3 - l'_3)^2 \right) \Bigg), \end{aligned} \tag{30}$$

where $O_1 = (P_1 + P_2 + P_3)/3, O_2 = (Q_1 + Q_2 + Q_3)/3$ denote the centers of the two triangles, unit vector $V_1, V_2, V_3, V'_1, V'_2, V'_3$ are the corresponding directions of edges, and $l_1, l_2, l_3, l'_1, l'_2, l'_3$ are corresponding lengths of the three edges.

The proof of this proposition is similar to that of Proposition 6.

The first term of equation (30) measures the difference between the two triangle patches in position, the following three terms measure the differences of the two triangle patches in orientation, and the last three terms measure the differences in size.

Now, we discuss a technique for triangle patch registration. Let

$$\{[P_n, P'_n, P''_n]\}_{n=1}^{N}, \qquad \{[Q_n, Q'_n, Q''_n]\}_{n=1}^{N}$$

be two sets of triangle patches. Suppose that there exist a rotation R and a translation T such that

$$[Q_n, Q'_n, Q''_n] = R[P_n, P'_n, P''_n] + T + [\varepsilon_n, \varepsilon'_n, \varepsilon''_n], \qquad n = 1, 2, \ldots, N,$$

where $[\varepsilon_n, \varepsilon'_n, \varepsilon''_n]$, $n = 1, 2, \ldots, N$ are experimental errors. We estimate R and T by minimizing

$$\begin{aligned} \Sigma &= \sum_{n=1}^{N} D(R[P_n, P'_n, P''_n] + T,\ [Q_n, Q'_n, Q''_n]) \\ &= \sum_{n=1}^{N} \frac{C_n}{12} \left(\|Q - RP_n - T\|^2 + \|Q' - RP'_n - T\|^2 \right. \\ &\quad + \|Q'' - RP''_n - T\|^2 + (Q - RP_n - T) \cdot (Q' - RP'_n - T) \\ &\quad + (Q - RP_n - T) \cdot (Q'' - RP''_n - T) \\ &\quad \left. + (Q' - RP'_n - T) \cdot (Q'' - RP''_n - T) \right), \end{aligned} \tag{31}$$

where

$$\begin{aligned} C_n = {} & \|(P'_n - P_n) \times (P''_n - P_n)\| + \|(Q'_n - Q_n) \times (Q''_n - Q_n)\| \\ & + \|((P'_n - P_n) + (Q'_n - Q_n)) \times ((P''_n - P_n) + (Q''_n - Q_n))\|. \end{aligned}$$

Let $\frac{\partial \Sigma}{\partial T} = 0$. It follows immediately that the optimal translation is chosen as

$$T = \bar{Q} - R\bar{P}, \tag{32}$$

where

$$\begin{aligned} \bar{P} &= \sum_{n=1}^{N} w_n \frac{P_n + P'_n + P''_n}{3}, & \bar{Q} &= \sum_{n=1}^{N} w_n \frac{Q_n + Q'_n + Q''_n}{3}, \\ w &= \sum_{n=1}^{N} C_n, & w_n &= \frac{C_n}{w}. \end{aligned} \tag{33}$$

Let

$$\begin{aligned} [\bar{P}_n, \bar{P}'_n, \bar{P}''_n] &= [P_n, P'_n, P''_n] - \bar{P}, \\ [\bar{Q}_n, \bar{Q}'_n, \bar{Q}''_n] &= [Q_n, Q'_n, Q''_n] - \bar{Q}, \end{aligned}$$

$n = 1, 2, \ldots, N$. Substituting (32) into (31), we have

$$\begin{aligned}\Sigma = \sum_{n=1}^{N} \frac{C_n}{12} \Big(& \left\|\bar{Q}_n - R\bar{P}_n\right\|^2 + \left\|\bar{Q}'_n - R\bar{P}'_n\right\|^2 + \left\|\bar{Q}''_n - R\bar{P}''_n\right\|^2 \\ & + (\bar{Q}_n - R\bar{P}_n) \cdot (\bar{Q}'_n - R\bar{P}'_n) + (\bar{Q}_n - R\bar{P}_n) \cdot (\bar{Q}''_n - R\bar{P}''_n) \\ & + (\bar{Q}'_n - R\bar{P}'_n) \cdot (\bar{Q}''_n - R\bar{P}''_n) \Big) .\end{aligned} \tag{34}$$

This sum can be further written as

$$\Sigma = \Delta - R \cdot B, \tag{35}$$

where Δ is a constant independent of rotation R and

$$\begin{aligned}B = \sum_{n=1}^{N} \frac{C_n}{12} \Big(& 2\bar{Q}''_n \bar{P''}_n^{\top} + 2\bar{Q'}_n \bar{P'}_n^{\top} + 2\bar{Q}_n \bar{P}_n^{\top} \\ & + \bar{Q}_n \bar{P'}_n^{\top} + \bar{Q}_n \bar{P''}_n^{\top} + \bar{Q'}_n \bar{P}_n^{\top} + \bar{Q'}_n \bar{P''}_n^{\top} + \bar{Q''}_n \bar{P}_n^{\top} + \bar{Q''}_n \bar{P'}_n^{\top} \Big) .\end{aligned}$$

Therefore, minimizing (34) is equivalent to maximizing

$$R \cdot B .$$

The way to estimate R from B is the same as that discussed in the previous section.

5. ITERATIVE LINE SEGMENT AND TRIANGLE PATCH REGISTRATION

In this section, the iterative closest line segment registration algorithm (ICL) and the iterative closest triangle patch registration algorithm (ICT) are presented. As the ICT algorithm is similar to the ICL algorithm in principle, it will just be discussed briefly.

Let $\mathcal{P} = \{P_n\}_{n=1}^{N}$, $\mathcal{Q} = \{Q_m\}_{m=1}^{M}$ be two data sets from the patient on the operating table and preoperative model, respectively. Usually, no point-to-point correspondence relations are known between the two data sets and the number M is much larger than N. In this case, the data set $\mathcal{Q}$ should be large enough to depict the surface of the object realistically, otherwise, the estimated rotation and translation might not be what we expect. The ICL algorithm works by searching for the closest line segments in $\mathcal{Q}$ for each line segment in $\mathcal{P}$.

In searching for the closest line segments, a dynamically weighted distance is used to measure the closeness of two line segments. Note that the length of a line segment is left unchanged by rotation and translation. For a given line segment in $\mathcal{P}$, its closest line segment in $\mathcal{Q}$ should have a similar length. If two line segments are significantly different in length, then one cannot be the transformation of the other. With this fact in mind, we could modify the distance between line segments by increasing the weight on the length difference to filter out the unlikely pairs of line segments. Therefore, at the beginning, we can set the weight on the length difference between two line segments to be very large such that whether two line segments are close mainly depends on whether they have similar lengths. Then the weight on length difference is gradually minimized according to the sum of distances between line segments in $\mathcal{P}$ and their corresponding line segments in $\mathcal{Q}$.

In this paper, the matching error in the k^{th} step of the iteration is used as the weight of differences in length, i.e., in k^{th} step, the following definition of distance between line segments is used in searching for the closest line segments in our ICL algorithm:

$$D_k(\mathbb{L}_1, \mathbb{L}_2) = D(\mathbb{L}_1, \mathbb{L}_2) + e_{k-1}(l_1 - l_2)^2, \tag{36}$$

where distance D is defined by (13) and e_{k-1} is the error sum in the $(k-1)^{\text{th}}$ step, and l_1, l_2 are the lengths of $\mathbb{L}_1$ and $\mathbb{L}_2$, respectively. Since the lengths of line segments are invariant under rotation and translation, the estimated rotation and translation based on distance D_k and distance D are the same. To enhance the efficiency of the ICL algorithm, it is proposed to sort the line segments of the intraoperative data according to their lengths so that longer line segments are compared first. The reason for doing this is that the orientation of the object is mainly determined by longer edges.

Let $\mathbb{L} = [P, P'] \in \mathcal{P}$ be a line segment, and $[\mathcal{Q}(P), \mathcal{Q}(P')]$ denote its closest line segment. Once the closest line segments for all elements in $\mathcal{P}$ have been found, the line segment registration method given in Section 3 is used on the data sets $\{[P_i, P_j] \mid i < j;\ i, j = 1, 2, \dots, N\}$ and $\{[\mathcal{Q}(P_i), \mathcal{Q}(P_j)] \mid i < j;\ i, j = 1, 2, \dots, N\}$. The estimated rotation and translation are then applied on $\mathcal{P}$. Then the updated $\mathcal{P}$ are used as the intraoperative data. This procedure is repeated until the difference between two consecutive error sums is smaller than the given tolerance. This procedure is always convergent under appropriate scaling. Let $\mathcal{P}^{(k)}$ be the point set after updating $\mathcal{P}$ k times. For each line segment $[P_i^{(k)}, P_j^{(k)}]$ in $\mathcal{P}^{(k)}$, the closest line segment $[\mathcal{Q}(P_i^{(k)}), \mathcal{Q}(P_j^{(k)})] \in \mathcal{Q}$ is defined as the line segment that satisfies

$$D_k\left(\left[P_i^{(k)}, P_j^{(k)}\right], \left[\mathcal{Q}\left(P_i^{(k)}\right), \mathcal{Q}\left(P_j^{(k)}\right)\right]\right) = \min_{Q, Q' \in \mathcal{Q}} D_k\left(\left[P_i^{(k)}, P_j^{(k)}\right], [Q, Q']\right).$$

ICL ALGORITHM.

1. Sort the line segments in $\mathcal{P}$ in such a way that the longer line segments are considered first in computing the closest line segments. In practice, if the number of points in $\mathcal{P}$ is n, we can just use the first n longest line segments to establish the registration.
2. Initialize rotation, translation, and e_0: R=I, T=0; e_0 should be initialized as large as possible.
3. For each pair of points $P^{(k)}, P'^{(k)} \in \mathcal{P}^{(k)}$, find a pair of points $Q, Q' \in \mathcal{Q}$ in a preoperative model such that the distance between line segments $[P, P']$ and $[Q, Q']$ are minimum in the sense of (36).
4. Using the line segment registration approach developed in Section 3 to compute rotation R' and translation T', set $R = R'R$ and $T = R'T + T'$.
5. Calculate the error sum

   $$e_k = \sum D_k\left(\left[P_i^{(k)}, P_j^{(k)}\right], \left[\mathcal{Q}\left(P_i^{(k)}\right), \mathcal{Q}\left(P_i^{(k)}\right)\right]\right).$$

   If $|e_k - e_{k+1}|$ is less than given tolerance, stop; otherwise, set

   $$P_n^{(k+1)} = R' P_n^{(k)} + T', \qquad n = 1, 2, \dots, N,$$

   and repeat from Step 3.

Now we show that this algorithm is always convergent under appropriate scaling. Let

$$e_k = \sum_{i,j=1;\ i<j}^{N} D_k\left(\left[P_i^{(k)}, P_j^{(k)}\right], \left[\mathcal{Q}\left(P_i^{(k)}\right), \mathcal{Q}\left(P_j^{(k)}\right)\right]\right), \tag{37}$$

and let $R^{(k)}, T^{(k)}$ be the estimated rotation and translation from data sets $\{[P_i^{(k)}, P_j^{(k)}]\}$ and $\{[\mathcal{Q}(P_i^{(k)}), \mathcal{Q}(P_j^{(k)})]\}$. Define

$$d_k = \sum_{i,j=1,\ i<j}^{N} D_k\left(\left[P_i^{(k+1)}, P_j^{(k+1)}\right],\ \left[\mathcal{Q}\left(P_i^{(k)}\right), \mathcal{Q}\left(P_j^{(k)}\right)\right]\right). \tag{38}$$

Let Rot and Tr represent any rotation and any translation, respectively. According to line segment registration algorithm,

$$\begin{aligned} d_k &= \sum_{i,j=1,\ i<j}^{N} D_k\left(\left[P_i^{(k+1)}, P_j^{(k+1)}\right],\ \left[Q\left(P_i^{(k)}\right), Q\left(P_j^{(k)}\right)\right]\right) \\ &= \sum_{i,j=1,\ i<j}^{N} D_k\left(R^{(k)}\left[P_i^{(k)}, P_j^{(k)}\right] + T^{(k)},\ \left[Q\left(P_i^{(k)}\right), Q\left(P_j^{(k)}\right)\right]\right) \\ &= \min_{\mathrm{Rot},\mathrm{Tr}} \sum_{i,j=1,\ i<j}^{N} D_k\left(\mathrm{Rot}\left[P_i^{(k)}, P_j^{(k)}\right] + \mathrm{Tr},\ \left[Q\left(P_i^{(k)}\right), Q\left(P_j^{(k)}\right)\right]\right) \\ &\le \sum_{i,j=1,\ i<j}^{N} D_k\left(\left[P_i^{(k)}, P_j^{(k)}\right],\ \left[Q\left(P_i^{(k)}\right), Q\left(P_j^{(k)}\right)\right]\right) = e_k. \end{aligned}$$

On the other hand, with the definition of closest line segments, we have

$$\begin{aligned} e_{k+1} &= \sum_{i,j=1,\ i<j}^{N} D_{k+1}\left(\left[P_i^{(k+1)}, P_j^{(k+1)}\right],\ \left[Q\left(P_i^{(k+1)}\right), Q\left(P_j^{(k+1)}\right)\right]\right) \\ &\le \sum_{i,j=1,\ i<j}^{N} D_{k+1}\left(\left[P_i^{(k+1)}, P_j^{(k+1)}\right],\ \left[Q\left(P_i^{(k)}\right), Q\left(P_j^{(k)}\right)\right]\right) \\ &= \sum_{i,j=1,\ i<j}^{N} D\left(\left[P_i^{(k+1)}, P_j^{(k+1)}\right],\ \left[Q\left(P_i^{(k)}\right), Q\left(P_j^{(k)}\right)\right]\right) + e_k\delta_k \\ &= \sum_{i,j=1,\ i<j}^{N} D_k\left(\left[P_i^{(k+1)}, P_j^{(k+1)}\right],\ \left[Q\left(P_i^{(k)}\right), Q\left(P_j^{(k)}\right)\right]\right) \\ &\quad + e_k\delta_k - e_{k-1}\delta_k \\ &= d_k + (e_k - e_{k-1})\delta_k \le e_k + (e_k - e_{k-1})\delta_k, \end{aligned}$$

where

$$\begin{aligned} \delta_k &= \sum_{i,j=1,\ i<j}^{N} \left(\left\|P_i^{(k+1)} - P_j^{(k+1)}\right\| - \left\|Q\left(P_i^{(k)}\right) - Q\left(P_j^{(k)}\right)\right\|\right)^2 \\ &= \sum_{i,j=1,\ i<j}^{N} \left(\|P_i - P_j\| - \left\|Q\left(P_i^{(k)}\right) - Q\left(P_j^{(k)}\right)\right\|\right)^2. \end{aligned}$$

From the relation

$$e_{k+1} \le e_k + (e_k - e_{k-1})\delta_k,$$

we see that if $e_k \le e_{k-1}$, then $e_{k+1} \le e_k$. Thus, if we could choose e_0 such that it is bigger than e_1, then the nonnegative real number sequence $\{e_k\}$ will be nonincreasing and bounded below, and it must be convergent.

It can be seen directly that a necessary condition that $e_{k+1} \le e_k$ is $\delta_k \le 1.0$. On the other hand, when $\delta_1 < 1.0$ is uniformly valid with the choice of e_0, then e_0 can always be chosen to be big enough such that it is larger than e_1. Since δ_k is just a sum of differences of lengths of corresponding line segments, its upper bound should be very small if the searching line segments are actually included in the model data sets. Even when the intraoperative data are not good enough, an upper bound 1.0 can always be set to δ_1 uniformly by rescaling the data sets or scale the sum of squared length differences directly such that it is less than one, say, the average sum of squared length differences can be used. Therefore, under appropriate scaling, the real number sequence $\{e_k\}$ will always be decreasing and thus convergent.

The iterative triangle patch algorithm (ICT) proceeds much the same way. The ICT algorithm can be obtained by replacing a line segment with a triangle patch in the above algorithm. As in the ICL algorithm, the following dynamic weighted distance is used instead of using (9).

$$D_k(\mathbb{T}_1, \mathbb{T}_2) = D(\mathbb{T}_1, \mathbb{T}_2) + e_k \left(d_1^2 + d_2^2 + d_3^2\right), \tag{39}$$

where e_k is the error sum in the k^{th} step defined in a similar way as (37) and d_1, d_2, d_3 are differences in lengths of three corresponding edges of the two triangle patches.

For each triangle patch in $\mathcal{P}$, the ICT algorithm will search for the closest triangle patches in the preoperative model data and use the triangle patch registration method given in Section 4 to compute the rotation and translation. The ICT algorithm works more robustly than the ICP and the ICL, but it spends more time in computing the closest triangle patches.

6. TESTING RESULT

The iterative line segment registration algorithm and triangle patch registration algorithm have been tested and compared with the ICP algorithm. Three different geometric objects have been considered in our experiments: a set of space line segments, a space curve, and a surface, see Figures 1–3. Having sampled the first data set (corresponding to preoperative data), the object is randomly transformed by a rotation and a translation and then the second data set (corresponding to intraoperative data) is sampled. The algorithms ICP, ICL, and ICT are applied to the two data sets to estimate the transformation.

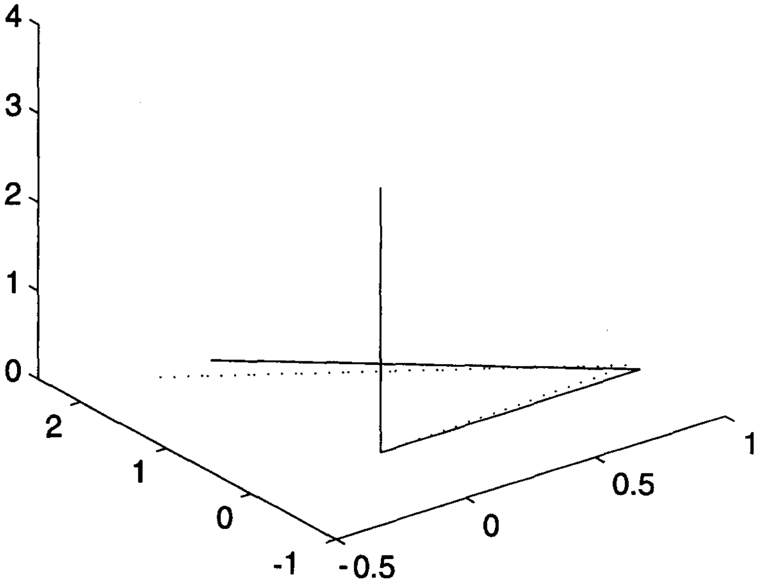

Figure 1. The line segments matching.

It is shown that when the second data set (intraoperative data) is just the transformation of a subset of the first data set, the estimated transformation obtained with the ICL and the ICT algorithm will be exactly the true transformation performed and the number of iterations is just two or three in most cases. When the second data set is not the transformation of a subset of the first one, it takes more iterations to converge and the transformation estimated may not necessarily be very close to the true transformation, though it is close in most of the cases. To test the stability of the ICL and the ICT algorithms, different numbers of data sets are sampled

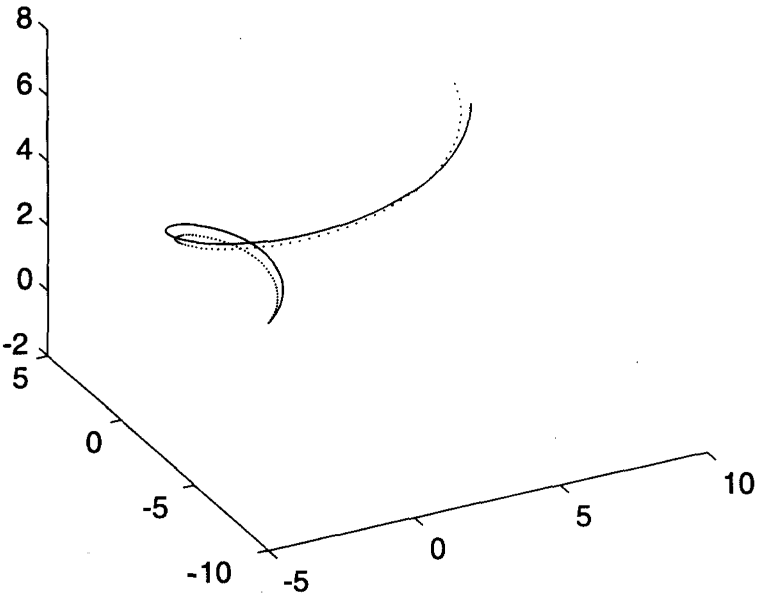

Figure 2. The curve matching.

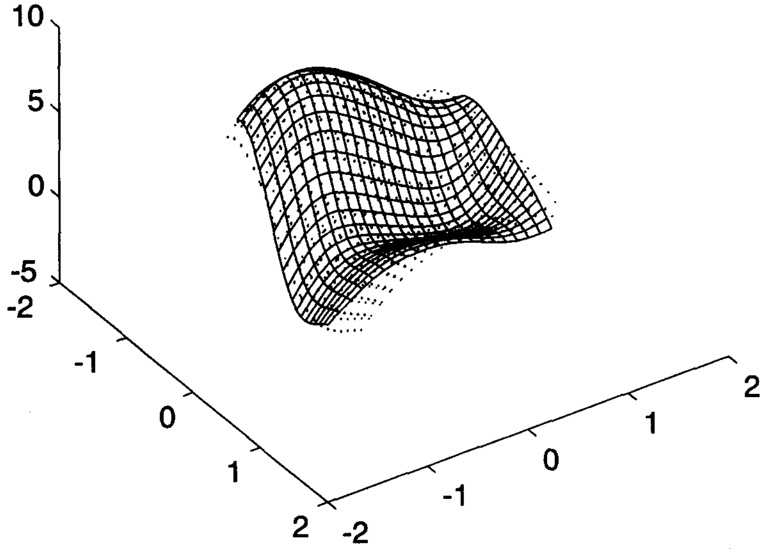

Figure 3. The surface matching.

for each object, with various orientations. The experiment results show that both the ICL and the ICT algorithms are much more stable than the ICP algorithm.

Obviously, the convergence speed depends largely on the size and the quality of the data used, but it is also affected by the initial value assigned to e_0. Too large e_0 tends to result in the convergence speed slower and too small e_0 may result in improper convergence. In our test, the ideal e_0 is chosen between 10^{10}–10^{30}.

Much of the experiment has been done to compare the ICL and the ICT algorithms with the ICP algorithm. As we know, the ICP algorithm is very sensitive to the initial orientation of the object. Thus, it is natural to ask whether the newly developed methods are less sensitive to initial orientations. The first way to show such a stability for these algorithms is to compute the probability of success for a series of experiments. In the experiment, we say that a matching process is successful if both the error between the estimated rotation and the true rotation and the error between estimated translation and the true translation are less than the given threshold.

Tables 1–3 show the percentage of successes over 500 runs of these three algorithms with different data sets in using the ICP, the ICL, and the ICT algorithms. The figures are obtained by setting the threshold to be 0.2 for line segments object and curve object and 0.4 for the surface object. The closeness between two rotation matrices is defined as the Frobenius norm of their difference and the closeness of two translations is measured by the norm of the difference between the two vectors which define the translations. The number of points in the model data is in the range of 60–1000 and the number of points in the second data set is just between 4–6.

Table 1. Percentage of success for line segments.

Points in Second Data	4	5	6
ICP	37.4	35.1	14.3
ICL	78.8	91.8	99.6
ICT	100.0	100.0	100.0

Table 2. Percentage of success for curve.

Points in Second Data	4	5	6
ICP	10.6	15.1	16.7
ICL	74.8	93.8	98.7
ICT	83.3	95.2	98.2

Table 3. Percentage of success for surface.

Points in Second Data	5	7	9
ICP	3.1	2.1	3.5
ICL	43.5	64.8	92
ICT	52.6	75.1	93.6

It can be see from the figures that the ICL and the ICT algorithm are much less sensitive to the initial orientations of the object. In all cases, the percentage of success is more than 70 for line segments and curve with the ICL and the ICT algorithms compared with less than 20 percent of success with the ICL algorithm. The figures in the table also show that with the increase of number of points in the second data sets, the ICL and the ICT become more and more robust, while there is no such sign for the ICP algorithm.

Table 4. Difference in rotation, translation, and distance for line segments with 4 points in the test data set.

	Mean			Std		
	Rotation	Translation	Distance	Rotation	Translation	Distance
ICP	1.6626	0.8648	1.6755	1.2116	0.7507	1.3280
ICL	0.1253	0.1177	0.2219	0.4234	0.2752	0.2630
ICT	0.1271	0.1411	0.4071	0.0739	0.0536	0.1282

Table 5. Difference in rotation, translation, and distance for line segments with 5 points in the test data set.

	Mean			Dev		
	Rotation	Translation	Distance	Rotation	Translation	Distance
ICP	1.6995	0.8022	1.9817	1.2215	0.7215	1.4866
ICL	0.2091	0.1675	0.3550	0.6382	0.4082	0.6115
ICT	0.1391	0.1497	0.5039	0.0544	0.0563	0.1650

Table 6. Difference in rotation, translation, and distance for line segments with 6 points in the test data set.

	Mean			Dev		
	Rotation	Translation	Distance	Rotation	Translation	Distance
ICP	2.0121	1.0921	3.0404	1.0776	0.7007	1.1353
ICL	0.1861	0.1484	1.0994	0.6016	0.3727	0.5039
ICT	0.1036	0.1388	1.3265	0.0639	0.0445	0.1370

Table 7. Difference in rotation, translation, and distance for curve with 4 points in the test data set.

	Mean			Dev		
	Rotation	Translation	Distance	Rotation	Translation	Distance
ICP	2.0372	5.1269	3.3219	1.1156	3.3631	2.0127
ICL	0.6733	1.6610	1.4082	1.0407	2.4570	1.5658
ICT	0.3409	0.7728	0.7065	0.8503	1.9574	1.5621

Table 8. Difference in rotation, translation, and distance for curve with 5 points in the test data set.

	Mean			Dev		
	Rotation	Translation	Distance	Rotation	Translation	Distance
ICP	1.9769	4.9529	5.0027	1.1702	3.3979	3.0511
ICL	0.4656	1.0212	1.3700	0.9080	2.3405	1.7400
ICT	0.0816	0.1876	0.7006	0.2118	0.4901	1.0160

Table 9. Difference in rotation, translation, and distance for curve with 6 points in the test data set.

	Mean			Dev		
	Rotation	Translation	Distance	Rotation	Translation	Distance
ICP	1.2347	3.7532	4.1150	1.1352	3.4012	2.5323
ICL	0.1366	0.2655	0.8782	0.3423	0.9151	1.1480
ICT	0.0751	0.1666	0.8626	0.0814	0.2128	0.9346

Table 10. Difference in rotation, translation, and distance for surface with 5 points in the test data set.

	Mean			Dev		
	Rotation	Translation	Distance	Rotation	Translation	Distance
ICP	2.1157	2.5736	4.0252	0.8368	1.5330	1.3005
ICL	0.6150	1.3876	2.9965	0.2207	0.3765	0.6095
ICT	0.6751	0.9171	0.9749	0.8347	1.3943	0.1733

The robustness of the ICL and ICT algorithms can also be shown by computing the average errors between true transformation and estimated transformation as well as errors in object matching. Tables 4–12 show that the average errors and the relevant standard deviations for both ICL and ICT algorithms are much smaller than those obtained with ICP algorithm.

The ICL and the ICT algorithms are designed for problems where only very few points in the intraoperative data set are used, normally between 4–10, while the points in preoperative

Table 11. Difference in rotation, translation, and distance for surface with 7 points in the test data set.

	Mean			Dev		
	Rotation	Translation	Distance	Rotation	Translation	Distance
ICP	2.2706	2.9904	3.4057	0.8760	1.9573	1.3736
ICL	0.5996	0.6705	1.7966	0.8221	0.2151	0.6559
ICT	0.3100	0.5402	0.9695	0.6925	1.1445	0.3149

Table 12. Difference in rotation, translation, and distance for surface with 9 points in the test data set.

	Mean			Dev		
	Rotation	Translation	Distance	Rotation	Translation	Distance
ICP	2.3007	2.9269	4.5968	1.8298	3.0929	1.5027
ICL	0.1116	0.1506	2.0263	0.1184	0.0640	0.5962
ICT	0.2065	0.2761	2.8030	0.1009	0.0822	1.0202

data can be huge so that the object surface can be fully described by the data. Thus, it is natural to ask whether the ICL and the ICT algorithms are feasible for this problem in practice as its computation complexity will be $O(NM^2)$ in searching the closest line segments for the ICL algorithm and $O(NM^3)$ in searching the closest triangle patches for the ICT algorithm. Generally speaking, the ICL and the ICT algorithms should not be directly applied to the data. Some preprocessing for preoperative data is needed. For example, the geometric invariants under rotation and translation can be considered to remove those unlikely pairs of line segments and triangle patches. Before applying the ICL algorithm, we could first select possible line segments in the preoperative data by considering whether a line segment has similar length to some line segment in the intraoperative data. Let P_1, P_2 be two points in the intraoperative data, and let their corresponding position in preoperative space be Q_1, Q_2. If the maximum distance between the neighboring elements in the preoperative data is δ, then the difference between $\|P_2 - P_1\|$ and $\|Q_2 - Q_1\|$ cannot be larger than 2δ. In this way, the number of line segments considered in the ICL will be greatly reduced. As the triangle patch algorithm provides more geometric invariants, more information can be used to select the possible triangle patches used in the ICT algorithm. This not only solves the problem of the feasibility in using the ICL algorithm and the ICT algorithm, but also increases the robustness of these two algorithms.

As far as the computing time is concerned, it depends not only on the size of the data sets, but also on their quality. The total computation time consists of the time used for selecting the possible line segments (or triangle patches) and the time used to estimate the transformation based on the selected line segments (or triangle patches). Increasing the number of points in matching data sets will only increase the time used for selecting the possible line segments (or triangle patches), but not necessarily the iteration times. When the number of points in the model data set is not too large, the time used mainly depends on the convergence speed. But for large model data set, the computation speed will be determined mainly by the time used for selecting possible line segments (or triangle patches). As for the ICT algorithm, the computation time really depends on the value of the threshold for selecting the possible triangle patches. A big threshold may result in thousands of triangle patches being selected and it may take hours to finish the matching process. The number of triangle patches selected for an appropriate threshold should be around 512 times the number of triangle patches selected from the second data sets. Figures 4 and 5 show the computation times in using the ICL and the ICT algorithms to match two data sets from a curve, where N represents the number of points in the second data set. The code is written in C++ and is run under Microsoft Windows NT with a CELERON

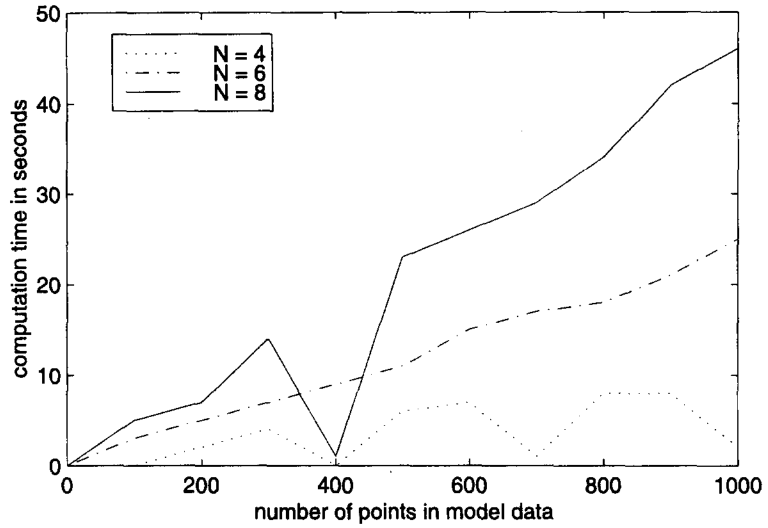

Figure 4. Computation time vs. the number of points in model data for the ICL algorithm.

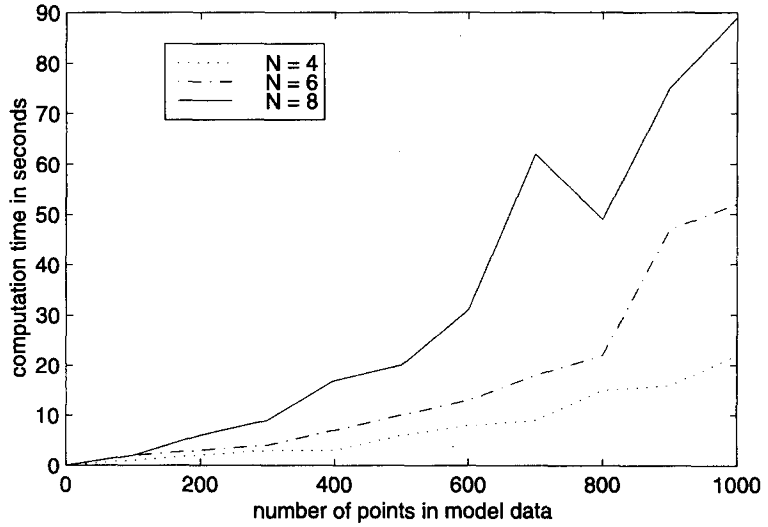

Figure 5. Computation time vs. the number of points in model data for the ICT algorithm.

400 MHz processor. As can be seen, it uses less than a minute to match two data sets for the ICL algorithm with the number of points in model data ranging from 100–1000. When applying the ICT algorithm to the same data sets, its speed is not as slow as expected, as we can see from the figure. However, if we ignore the selecting procedure, it does take a few hours to establish the match when the number of points in the model data is larger than 700. The good thing about the ICT and the ICL algorithms is that they can still give good estimates even if the model data set is a bit sparse, while the ICP cannot. Therefore, we can first use the ICT or the ICL algorithm to find a good initial solution (one or two iterations) with a subset of model data. This initial can then be further tuned by the ICP algorithm.

REFERENCES

1. P.J. Besl and N.D. McKay, Least-squares fitting of two 3-D point sets, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **14** (2), 239–256, (1992).
2. L.G. Brown, A survey of image registration techniques, *ACM Computing Surveys* **24** (4), 326–376, (1992).
3. J.B.A. Maintz and M.A. Viergever, A survey of medical image registration, *Medical Image Analysis* **2** (1), 1–36, (1998).

4. B.K. Parsi and B.K. Parsi, Matching sets of 3D line segments with application to polygonal arc matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19** (10), 1090–1099, (1997).
5. S.L. Altmann, *Rotation, Quaternion, and Double Groups*, Oxford University Press, New York, (1986).
6. G.W. Steward, *Introduction to Matrix Computing*, Academic Press, Orlando, FL, (1973).
7. O.D. Faugeras and M. Hebert, The representation, recognition, and locating of 3-D objects, *The International Robotics Research* **5** (3), 27–52, (1986).
8. B.K.P. Horn, Closed-form solution of absolute orientation using unit quaternions, *J. Opt. Soc. Am. A* **4** (4), 629–642, (1987).
9. K.S. Arun, T.S. Huang and S.D. Blostein, Least-squares fitting of two 3-D point sets, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **9** (5), 698–700, (1987).
10. S. Uneyama, Least-squares estimation of transformation parameters between two point patterns, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **13** (4), 376–380, (199).
11. Q. Li and J.G. Griffiths, *Some Algorithms for Coordinate Systems Alignment*, Technical Report, Department of Computer Science, University of Hull, (1998).