A Solution for the Registration of Multiple 3D Point Sets Using Unit Quaternions

Raouf Benjemaa and Francis Schmitt

cole Nationale Suprieure des Tlcommunications
CNRS URA 820 - TSI dept
46 rue Barrault, F-75634 Paris Cedex 13, France
benjemaa@ima.enst.fr, schmitt@ima.enst.fr
WWW home page: http://www-ima.enst.fr/~benjemaa

Abstract. Registering 3D point sets is a common problem in computer vision. The case of two point sets has been analytically well solved by several authors. In this paper we present an analytic solution for solving the problem of a simultaneous registration of M point sets, $M > 2$, by rigid motions. The solution is based on the use of unit quaternions for the representation of the rotations.
We show that the rotation optimization can be decoupled from the translation one. The optimal translations are given by the resolution of a linear equation system which depends on the rotated centroid of the overlaps. The unit quaternions representing the best rotations are optimized by applying an iterative process on symmetric 4×4 matrices. The matrices correspond to the mutual overlaps between the point sets.
We have applied this method to the registration of several overlapping 3D surfaces sampled on an object. Our results on simulated and real data show that the algorithm works efficiently.

1 Introduction

Registering 3D point sets with 3D rigid motions is a common problem in computer vision and robotics. This problem typically occurs when 3D data are acquired from different viewpoints by stereo, range sensing, tactile sensing, *etc.*

We call **pairwise registration** the registration of two points sets. However, the case of a larger number of point sets overlapping each other occurs often. We can sequentially apply a pairwise registration by matching two by two the different sets. This widely used scheme doesn't take into account the whole correspondences between the data sets during a registration step. It remains essentially a local approach and it may cause error-distribution problems as pointed out in [10, 4, 15, 3]. For example the residual error of each individual pairwise registration can be low but unfortunately we have frequently observed a propagation and a cumulation of the registration errors.

Thus it appears to be much more efficient to register simultaneously the multiple point sets in order to keep the residual registration errors homogeneously distributed. We call **global registration** the simultaneous registration of multiple point sets which partially overlap each other and for which we know a correspondence between their points into each overlap.

In this paper we propose an analytic quaternion-based solution to solve the problem of the global registration of M point sets, with know correspondences, $M > 2$. The correspondence establishment depends on the type of data to be registered. To determine the correspondence between several point sets sampled on an object surface, the authors have proposed an efficient method based on a space partitioning with a multi-z-buffer technique [1,3]. By combining this fast correspondence establishment and the quaternion based global registration, the ICP (*Iterative Closest Point*) algorithm originally proposed by Besl and McKay [5] to register two point sets can be conveniently generalized to globally register multiple point sets.

The next section describes previous works on the registration of multiple data sets. We then recall some properties of the unit quaternions in section 3 and the classic quaternion-based solution for registering two point sets [7,8] in section 4 . In section 5 we state the problem of the global registration. The optimal solutions for the translations and then for the rotations are detailed in sections 6 and 7, respectively. Finally, experimental results on both synthetic and real data are shown in section 8.

2 Literature Review

Different analytic methods –singular value decomposition, polar decomposition and quaternion representation– have been proposed for registering two point sets with known correspondences. Each of them computes the rigid transformation, as a solution to a least squares formulation of the problem. For an overview and a discussion of these techniques see Kanatani [11] and references therein. Recently a comparative analysis of the various method was given in [12]. It was concluded that no one algorithm was found to be superior in all cases to the other ones.

Only few authors have investigated the registration of multiple point sets as a global problem. We may distinguish three different categories in the literature: **(a)** dynamic-system-based global registration, **(b)** iterative global registration, and **(c)** analytic global registration.

Kamgar-Parsi *et al.* [10] have developed a global registration method using a dynamic system for the 2D registration of multiple overlapping range images. The position of each range image is then optimized according to a 2D rigid transformation with three degrees of freedom (one for the rotation and two for the translation).

Recently Stoddart and Hilton [15] have also proposed a method in category (a) for the 3D global registration of multiple free-form surfaces. Their dynamic system is made of a set of springs of length null and whose extremities are connected between pairs of corresponding points on two overlapping surfaces. The registration is then obtained by solving the equation of the Lagrangian mechanic with an iterative Euler resolution.

In category (b), Bergevin *et al.* [4] have proposed an iterative algorithm based on a modified ICP algorithm to register multiple range images. At each iteration, each range image is successively matched with all the others, and its rigid transformation is estimated. The surfaces are simultaneously moved only at the end of the iteration, after the estimation of the complete set of rigid transformations.

Inspired by this work [4], the authors have developed an iterative method to simultaneously register multiple 3D unstructured data scattered on the surface of an object from different viewpoints [3]. This method is dramatically accelerated by using a multi-z-buffer space partitioning. Unlike Bergevin *et al.*'s method, each surface is immediately transformed when its rigid motion has been estimated. This way, the convergence is accelerated. In order to not favor any surface, its registration order is randomly chosen at each iteration.

The authors have recently proposed an analytic global registration solution based on a linearization of the rotations [2]. The optimal rigid transformation values are given by the simple resolution of two linear-equation systems. It is assumed that the rotation angles are small. Thus the method performs only when data sets are not too far from each others.

The global registration method presented in this paper belongs also to category (c) but the assumption of small rotation angles is not necessary. It is a generalization of the quaternion-based solution of the pairwise registration which has been independently proposed by Faugeras and Hebert [6, 7] and Horn [8].

3 Representing rotations by unit quaternions

The reader not familiar with quaternions can refer to [9, 14, 8, 13]. Quaternions will be denoted here by using symbols with dots above them.

A quaternion $\dot{q}$ can be seen as either a four-dimensional vector $(q_0, q_x, q_y, q_z)^t$, or a scalar q_0 and a tri-dimensional vector $(q_x, q_y, q_z)^t$, or a complex number with three different imaginary parts ($\dot{q} = q_0 + iq_x + jq_y + kq_z$).

The product on the left or on the right of a quaternion $\dot{q}$ by another quaternion $\dot{r} = (r_0, r_x, r_y, r_z)^t$, can conveniently be expressed in terms of the following products of the vector $\dot{q}$ by a 4×4 orthogonal matrix $\mathbb{R}$ or $\bar{\mathbb{R}}$ respectively:

$$\dot{r}\dot{q} = \begin{bmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & -r_z & r_y \\ r_y & r_z & r_0 & -r_x \\ r_z & -r_y & r_x & r_0 \end{bmatrix} \dot{q} = \mathbb{R}\dot{q}, \quad (1)$$

or

$$\dot{q}\dot{r} = \begin{bmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & r_z & -r_y \\ r_y & -r_z & r_0 & r_x \\ r_z & r_y & -r_x & r_0 \end{bmatrix} \dot{q} = \bar{\mathbb{R}}\dot{q}. \quad (2)$$

The unit quaternions are an elegant representation of rotation. Let recall some of their fundamental properties. A rotation R of angle θ around axe $\boldsymbol{u}$ can be represented by the unit quaternion $\dot{q} = (\cos\theta/2, \sin\theta/2\boldsymbol{u})$. This rotation applied to a vector $\boldsymbol{r}$ is then expressed as a multiplication of quaternions : $R(\boldsymbol{r}) = \dot{q}\dot{r}\dot{q}^*$, where $\dot{r}$ is a pure imaginary quaternion ($r_0 = 0$) and $\dot{q}^*$ denotes the conjugates of $\dot{q}$ ($\dot{q}^* = q_0 - iq_x - jq_y - kq_z$).

The scalar product h of two vectors $\boldsymbol{r}^i = (x^i, y^i, z^i)^t$ and $\boldsymbol{r}^j = (x^j, y^j, z^j)^t$ which are transformed by the rotations R^i and R^j respectively:

$$h = R^i(\boldsymbol{r}^i) \cdot R^j(\boldsymbol{r}^j),$$

can be written as:

$$h = (\dot{q}^i\dot{r}^i\dot{q}^{*i}) \cdot (\dot{q}^j\dot{r}^j\dot{q}^{*j}),$$

where $\dot{q}^i$ and $\dot{q}^j$ are the unit quaternions corresponding to R^i and R^j and $\dot{r}^i$ and $\dot{r}^j$ are the pure imaginary quaternions corresponding to $\boldsymbol{r}^i$ et $\boldsymbol{r}^j$, respectively.

Given that the dot product between two quaternions is preserved when multiplied by an unit quaternion, we can rewrite h in the following form

$$h = (\dot{q}^{*j}\dot{q}^i\dot{r}^i) \cdot (\dot{r}^j\dot{q}^{*j}\dot{q}^i).$$

Considering the matrix forms (1) and (2) of the quaternion product, it follows that

$$h = (\bar{\mathbb{R}}^i\dot{q}^{*j}\dot{q}^i) \cdot (\mathbb{R}^j\dot{q}^{*j}\dot{q}^i),$$

which also can be written in the form

$$h = (\dot{q}^{*j}\dot{q}^i)^t \mathbb{N}^{ij} (\dot{q}^{*j}\dot{q}^i), \tag{3}$$

where $\mathbb{N}^{ij} = \bar{\mathbb{R}}^{i^t}\mathbb{R}^j$ is a symmetric matrix having the form

$$\mathbb{N}^{ij} = \begin{bmatrix} a & e & f & g \\ e & b & h & k \\ f & h & c & l \\ g & k & l & d \end{bmatrix},$$

where

$$a = x^ix^j + y^iy^j + z^iz^j, \quad b = x^ix^j - y^iy^j - z^iz^j,$$
$$c = -x^ix^j + y^iy^j - z^iz^j, \; d = -x^ix^j - y^iy^j + z^iz^j,$$

and

$$e = y^iz^j - z^iy^j, \; f = z^ix^j - x^iz^j, \; g = x^iy^j - y^ix^j,$$
$$h = x^iy^j + y^ix^j, \; k = z^ix^j + x^iz^j, \; l = y^iz^j + z^iy^j.$$

By permuting the indexes i and j, the matrix $\mathbb{N}^{ji} = \bar{\mathbb{R}}^{j^t}\mathbb{R}^i$ can be written as follow:

$$\mathbb{N}^{ji} = \begin{bmatrix} a & -e & -f & -g \\ -e & b & h & k \\ -f & h & c & l \\ -g & k & l & d \end{bmatrix}.$$

This allows us to simply verify the useful following property:

$$\dot{q}^t \mathbb{N}^{ij} \dot{q} = \dot{q}^{*t} \mathbb{N}^{ji} \dot{q}^*, \quad \forall \dot{q}. \tag{4}$$

4 Pairwise registration using quaternions

Faugeras and Hebert [7], and Horn [8], have proposed a quaternion-based solution to register a set of n points $S^2 = \{P_i^2\}$ with a set of n points $S^1 = \{P_i^1\}$ where each point P_i^2 is in correspondence with the point P_i^1 with the same index. The rigid transformation T^2 to be applied to S^2, defined by the rotation R^2 and the translation t^2, is optimized by minimizing the following cost function:

$$E = \sum_{i=1}^{n} \| P_i^1 - R^2(P_i^2) - t^2 \|^2. \tag{5}$$

The optimal translation is given by the difference between the centroid of S^1 and the transformed centroid of S^2

$$t^2 = \bar{P^1} - R^2(\bar{P^2}). \tag{6}$$

The unit quaternion representing the best rotation is the unit eigenvector corresponding to the maximum eigenvalue of the following 4×4 matrix:

$$\begin{bmatrix} S_{xx}+S_{yy}+S_{zz} & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & -S_{xx}+S_{yy}-S_{zz} & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & -S_{xx}-S_{yy}+S_{zz} \end{bmatrix}$$

where $S_{xx} = \sum_{i=1}^{n} x'^2_i x'^1_i$, $S_{xy} = \sum_{i=1}^{n} x'^2_i y'^1_i$, $\ldots$, x'^j_i, y'^j_i and z'^j_i being the centered coordinates of the points P_i^j, $i = 1..n; j = 1..2$.

5 Specification of the global registration problem

We assume that there are M overlapping sets of points $S^1, S^2, ..., S^M$. The global registration process must find the best rigid transformations $T^1, T^2, ..., T^M$ to be applied to each point set.

We denote $O^{\alpha\beta} \subset S^\alpha$ the overlap of S^α with S^β, $\alpha,\beta \in [1..M]$. $O^{\alpha\beta}$ is composed of $N^{\alpha\beta}$ points $P_i^{\alpha\beta} \in S^\alpha$, $i = 1..N^{\alpha\beta}$. Each point $P_i^{\alpha\beta}$ is matched with a point $P_i^{\beta\alpha}$ belonging to $O^{\beta\alpha} \subset S^\beta$. Mutually, the point $P_i^{\beta\alpha}$ is matched with $P_i^{\alpha\beta}$. Then we have $N^{\alpha\beta} = N^{\beta\alpha}$.

The optimal rigid transformations are usually specified as the minimum of an objective function which can be chosen as the sum of the squared Euclidean distances between the matched points of all the overlaps:

$$E = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} \|T^{\alpha}(P_i^{\alpha\beta}) - T^{\beta}(P_i^{\beta\alpha})\|^2. \tag{7}$$

In order to simplify the notations, we introduce here $O^{\alpha\alpha}$ the overlap of the point set S^{α} with itself. Its contribution to error E having to remain always null, we just impose for each $\alpha \in [1..M]$:

$$N^{\alpha\alpha} = 0. \tag{8}$$

This cost function takes simultaneously into account the motions of the M surfaces. The residual errors will be homogeneously distributed in the whole mutual overlaps. One can notice that by taking $M = 2$ and by setting the transformation T^1 to the identity transformation, we retrieve equation (5).

6 Optimization of translations

We are looking for the translations which minimize the cost function (7). We show that this set of optimal translations is given by solving a linear equation system. We also show that the optimization of the rotations can be decoupled from the values of the translations.

6.1 Solution of optimal translations

Suppose that each rigid transformation T^{α} ($\alpha = 1, \ldots, M$) is composed of the rotation R^{α} and the translation t^{α}, the cost function E, defined in equation (7) can be written:

$$E = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} \|R^{\alpha}(P_i^{\alpha\beta}) - R^{\beta}(P_i^{\beta\alpha}) + t^{\alpha} - t^{\beta}\|^2,$$

or,

$$E = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \Big(\sum_{i=1}^{N^{\alpha\beta}} \|R^{\alpha}(P_i^{\alpha\beta}) - R^{\beta}(P_i^{\beta\alpha})\|^2 + 2[t^{\alpha} - t^{\beta}] \cdot \sum_{i=1}^{N^{\alpha\beta}} [R^{\alpha}(P_i^{\alpha\beta}) - R^{\beta}(P_i^{\beta\alpha})]$$
$$+ N^{\alpha\beta} \|t^{\alpha} - t^{\beta})\|^2 \Big).$$

However the right term of the scalar product can be expressed by $N^{\alpha\beta}[R^{\alpha}(\bar{P}^{\alpha\beta}) - R^{\beta}(\bar{P}^{\beta\alpha})]$ where $\bar{P}^{\alpha\beta}$ and $\bar{P}^{\beta\alpha}$ are the centroid of $O^{\alpha\beta}$ and $O^{\beta\alpha}$ respectively;

$$\bar{P}^{\alpha\beta} = \frac{1}{N^{\alpha\beta}} \sum_{i=1}^{N^{\alpha\beta}} P_i^{\alpha\beta}.$$

The cost function E can be written in the form:

$$E = E_R + E_{t,R},$$

where

$$E_R = \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} \|R^\alpha(P_i^{\alpha\beta}) - R^\beta(P_i^{\beta\alpha})\|^2,$$

and

$$E_{t,R} = \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} N^{\alpha\beta}(2[t^\alpha - t^\beta]\cdot[R^\alpha(\bar{P}^{\alpha\beta}) - R^\beta(\bar{P}^{\beta\alpha})] + \|t^\alpha - t^\beta\|^2),$$

We notice that E_R does not depend on translations. So the values of translations which minimize the cost function E will be given by the minimization of $E_{t,R}$. Let rewrite $E_{t,R}$ as follows:

$$E_{t,R} = C_1 + C_2,$$

where

$$\begin{aligned} C_1 &= 2\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} N^{\alpha\beta}[t^\alpha - t^\beta]\cdot[R^\alpha(\bar{P}^{\alpha\beta}) - R^\beta(\bar{P}^{\beta\alpha})], \\ &= 4\sum_{\alpha=1}^{M}[t^\alpha \cdot \sum_{\beta=1}^{M} N^{\alpha\beta}(R^\alpha(\bar{P}^{\alpha\beta}) - R^\beta(\bar{P}^{\beta\alpha}))], \end{aligned}$$

and

$$\begin{aligned} C_2 &= \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} N^{\alpha\beta}\|t^\alpha - t^\beta\|^2, \\ &= 2\sum_{\alpha=1}^{M}[t^{\alpha 2}(\sum_{\beta=1}^{M} N^{\alpha\beta})] - 2\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} N^{\alpha\beta} t^\alpha \cdot t^\beta. \end{aligned}$$

We have used here the data property $N^{\alpha\beta} = N^{\beta\alpha}$. $E_{t,R}$ can then be written in matrix form

$$E_{t,R} = 2(X^t\mathbf{A}X + 2X^tB), \tag{9}$$

where, $X = (t^1, t^2, \ldots, t^M)^t$,

$$\mathbf{A} = \begin{bmatrix} N^1 & -N^{12} & \ldots & -N^{1M} \\ -N^{21} & N^2 & \ldots & -N^{2M} \\ \vdots & \vdots & \ddots & \vdots \\ -N^{M1} & -N^{M2} & \ldots & N^M \end{bmatrix},$$

(where $N^\alpha = \sum_{\beta=1}^{M} N^{\alpha\beta}$) and

$$B = \begin{pmatrix} \sum_{\beta=1}^{M} N^{1\beta}[R^1(\bar{P}^{1\beta}) - R^\beta(\bar{P}^{\beta 1})] \\ \sum_{\beta=1}^{M} N^{2\beta}[R^2(\bar{P}^{2\beta}) - R^\beta(\bar{P}^{\beta 2})] \\ \vdots \\ \sum_{\beta=1}^{M} N^{M\beta}[R^M(\bar{P}^{M\beta}) - R^\beta(\bar{P}^{\beta M})] \end{pmatrix}.$$

Minimizing $E_{t,R}$ is equivalent to the minimization of the function $Q(X)$:

$$Q(X) = X^t \mathbf{A} X + 2X^t B. \tag{10}$$

It should be noted that the matrix $\mathbf{A}$ is not invertible; its determinant is null, the sum of each line or each column being null. The cost function E is unchanged if the same rigid transformation is applied simultaneously to all M point sets. The reference frame of one point set should be chosen as an absolute one and only the $(M-1)$ other sets should be moved. The choice of this set is arbitrary and does not affect the registration solution. Let fix the first set by setting $R^1 = I$ and $t^1 = 0$. Equation (10) becomes,

$$Q(\bar{X}) = \bar{X}^t \bar{\mathbf{A}} \bar{X} + 2\bar{X}^t \bar{B}, \tag{11}$$

where $\bar{X}$ and $\bar{B}$ are the vectors X and B deprived from their first element, and where

$$\bar{\mathbf{A}} = \begin{bmatrix} N^2 & \ldots & -N^{2M} \\ \vdots & \ddots & \vdots \\ -N^{M2} & \ldots & N^M \end{bmatrix}.$$

It may be of interest to note that despite the suppression of the first line and the first column of $\mathbf{A}$, the matrix $\bar{\mathbf{A}}$ still contains the terms $N^{\alpha 1}$ in the diagonal elements. Therefore the overlaps with the set S^1 are still considered.

$Q(\bar{X})$ is a quadratic form which is minimal when $\bar{\mathbf{A}}\bar{X} = -\bar{B}$. So, the value of the translations are simply obtained by the inversion of the matrix $\bar{\mathbf{A}}$:

$$\bar{X}_{min} = -\bar{\mathbf{A}}^{-1}\bar{B}. \tag{12}$$

The translations are given by a linear combination of the differences between the rotated centroids of the overlaps $O^{\alpha\beta}$ and $O^{\beta\alpha}$. Again if $M = 2$, we retrieve equation (6)) for a pairwise registration.

6.2 Decoupling between rotations and translations

Using the result of optimal translations, we show now that the optimization of the rotations can be decoupled from the translations.

For $\bar{X} = \bar{X}_{min}$, equation (11) becomes

$$Q(\bar{X}_{min}) = -\bar{B}^t \bar{\mathbf{A}}^{-1} \bar{B}. \tag{13}$$

We then obtain for the global cost function E: $E(\bar{X}_{min}) = E_R + E_{t,R} = E_R + 2Q(\bar{X}_{min})$; *i.e.* finally:

$$E(X_{min}) = \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} \|R^\alpha(P_i^{\alpha\beta}) - R^\beta(P_i^{\beta\alpha})\|^2 - 2\bar{B}^t\bar{\mathbf{A}}^{-1}\bar{B}. \qquad (14)$$

The function E depends no more on translations as shown in equation (14). The rotations can then be optimized independently from translations .

7 Optimization of rotations using unit quaternions

In this section we solve the problem of the optimization of rotations by minimizing the cost function E defined in equation (14).

We start by rewriting expressing E as a function of quaternions in section 7.1. We show that minimizing E is equivalent to maximizing another cost function. Then a sequential algorithm is proposed in section 7.2 in order to maximize this new cost function. Finally, in section 7.3 we prove that this algorithm usually converges to a local minimum.

7.1 Rewriting expressing E with unit quaternions

Let $e_i^{\alpha\beta} = \|R^\alpha(P_i^{\alpha\beta}) - R^\beta(P_i^{\beta\alpha})\|^2$.

Using the preservation of norm by the rotations, we have:

$$e_i^{\alpha\beta} = \|P_i^{\alpha\beta}\|^2 + \|P_i^{\beta\alpha}\|^2 - 2R^\alpha(P_i^{\alpha\beta}) \cdot R^\beta(P_i^{\beta\alpha}).$$

The first term of E becomes,

$$\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} e_i^{\alpha\beta} = -2\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} R^\alpha(P_i^{\alpha\beta}) \cdot R^\beta(P_i^{\beta\alpha}) + K \qquad (15)$$

where $K = 2\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} \|P_i^{\alpha\beta}\|^2$ is a constant which does not depend on rotations. By combining equations (14) and (15) and by ignoring the constant term K, minimizing E is equivalent to maximizing

$$H = \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} R^\alpha(P_i^{\alpha\beta}) \cdot R^\beta(P_i^{\beta\alpha}) + \bar{B}^t\bar{\mathbf{A}}^{-1}\bar{B}. \qquad (16)$$

Considering relation (3), the first term of H,

$$H_1 = \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} R^\alpha(P_i^{\alpha\beta}) \cdot R^\beta(P_i^{\beta\alpha}),$$

can be written by using unit quaternions:

$$H_1 = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} (\dot{q}^{*\beta}\dot{q}^{\alpha})^t \mathbb{Q}_i^{\alpha\beta} (\dot{q}^{*\beta}\dot{q}^{\alpha}),$$

or also, by integrating the sum over i,

$$H_1 = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} (\dot{q}^{*\beta}\dot{q}^{\alpha})^t \mathbb{Q}_R^{\alpha\beta} (\dot{q}^{*\beta}\dot{q}^{\alpha}), \tag{17}$$

where $\mathbb{Q}_R^{\alpha\beta} = \sum_{i=1}^{N^{\alpha\beta}} \mathbb{Q}_i^{\alpha\beta} =$

$$\begin{bmatrix} S_{xx}^{\alpha\beta} + S_{yy}^{\alpha\beta} + S_{zz}^{\alpha\beta} & S_{yz}^{\alpha\beta} - S_{zy}^{\alpha\beta} & S_{zx}^{\alpha\beta} - S_{xz}^{\alpha\beta} & S_{xy}^{\alpha\beta} - S_{yx}^{\alpha\beta} \\ S_{yz}^{\alpha\beta} - S_{zy}^{\alpha\beta} & S_{xx}^{\alpha\beta} - S_{yy}^{\alpha\beta} - S_{zz}^{\alpha\beta} & S_{xy}^{\alpha\beta} + S_{yx}^{\alpha\beta} & S_{zx}^{\alpha\beta} + S_{xz}^{\alpha\beta} \\ S_{zx}^{\alpha\beta} - S_{xz}^{\alpha\beta} & S_{xy}^{\alpha\beta} + S_{yx}^{\alpha\beta} & -S_{xx}^{\alpha\beta} + S_{yy}^{\alpha\beta} - S_{zz}^{\alpha\beta} & S_{yz}^{\alpha\beta} + S_{zy}^{\alpha\beta} \\ S_{xy}^{\alpha\beta} - S_{yx}^{\alpha\beta} & S_{zx}^{\alpha\beta} + S_{xz}^{\alpha\beta} & S_{yz}^{\alpha\beta} + S_{zy}^{\alpha\beta} & -S_{xx}^{\alpha\beta} - S_{yy}^{\alpha\beta} + S_{zz}^{\alpha\beta} \end{bmatrix},$$

$$S_{xx}^{\alpha\beta} = \sum_{i=1}^{N^{\alpha\beta}} x_i^{\alpha\beta} x_i^{\beta\alpha}, \qquad S_{xy}^{\alpha\beta} = \sum_{i=1}^{N^{\alpha\beta}} x_i^{\alpha\beta} y_i^{\beta\alpha}, \qquad S_{xz}^{\alpha\beta} = \sum_{i=1}^{N^{\alpha\beta}} x_i^{\alpha\beta} z_i^{\beta\alpha}, \ldots,$$

with $(x_i^{\alpha\beta}, y_i^{\alpha\beta}, z_i^{\alpha\beta})^t = P_i^{\alpha\beta}$ and $(x_i^{\beta\alpha}, y_i^{\beta\alpha}, z_i^{\beta\alpha})^t = P_i^{\beta\alpha}$. In another side, as shown in appendix A, the quantity $\bar{B}^t \bar{\mathbf{A}}^{-1} \bar{B}$ can be expressed in the form:

$$\bar{B}^t \bar{\mathbf{A}}^{-1} \bar{B} = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} (\dot{q}^{*\beta}\dot{q}^{\alpha})^t \mathbb{Q}_t^{\alpha\beta} (\dot{q}^{*\beta}\dot{q}^{\alpha}), \tag{18}$$

(*cf.* appendix A for the expression of $\mathbb{Q}_t^{\alpha\beta}$). According to equations (17) and (18), equation (16) becomes :

$$\boxed{H = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} (\dot{q}^{*\beta}\dot{q}^{\alpha})^t \mathbb{Q}^{\alpha\beta} (\dot{q}^{*\beta}\dot{q}^{\alpha})} \tag{19}$$

where $\mathbb{Q}^{\alpha\beta} = \mathbb{Q}_R^{\alpha\beta} + \mathbb{Q}_t^{\alpha\beta}$.

$\mathbb{Q}_R^{\alpha\beta}$ and $\mathbb{Q}_t^{\alpha\beta}$ are 4×4 symmetrical matrices which have the same form than $\mathbb{N}^{ij}$ (*cf.* section 3). So $\mathbb{Q}^{\alpha\beta}$ verify the property (4).

7.2 Sequential algorithm

Let consider now the problem of maximizing the function H of equation (19) subject to the $4(M-1)$ vector $\mathbf{q} = (\dot{q}^2, \dot{q}^3, \ldots, \dot{q}^M)^t$. The quaternion $\dot{q}^1$ is set to the identity quaternion, the reference frame of S_1 being chosen as the absolute one. We propose a sequential method to optimize the $(M-1)$ quaternions defined by the vector $\mathbf{q}$. The approach is the following one: at each iteration all the quaternions $\dot{q}^j$ are fixed excepted one of them. We determine this last one so that H is maximized. To be more precise, we start with an initial vector $\mathbf{q}_0$ which is arbitrarily defined or provided by a pre-computing step. We then construct a sequence of vectors $\mathbf{q}_m$, $m = 1, 2, \ldots$ where the transition from $\mathbf{q}_m$ to $\mathbf{q}_{m+1}$ is done in $(M-1)$ steps: $\dot{q}^2_{m+1}$ is determined in the first step, then $\dot{q}^3_{m+1}$, ..., and finally $\dot{q}^M_{m+1}$. The quaternion $\dot{q}^j_{m+1}$ is the unique solution of the following maximization problem: determine the quaternion $\dot{q}^j_{m+1}$ belonging to the unit quaternion set $\mathcal{Q}$ such as:

$$\begin{cases} H(\dot{q}^2_{m+1}, \ldots, \dot{q}^j_{m+1}, \dot{q}^{j+1}_m, \ldots, \dot{q}^M_m) \geq H(\dot{q}^2_{m+1}, \ldots, \dot{q}^{j-1}_{m+1}, \dot{q}^j, \dot{q}^{j+1}_m, \ldots, \dot{q}^M_m), \\ \forall \dot{q}^j \in \mathcal{Q} \end{cases} \tag{20}$$

When all the quaternions are fixed except $\dot{q}^j$, maximizing H (equation (19)) according to $\dot{q}^j$ becomes a simple problem. By ignoring the constant terms, this maximization is equivalent to the maximization of the following function $H(\dot{q}^j)$:

$$\begin{aligned} H(\dot{q}^j) &= \sum_{\beta=1,\beta\neq j}^{M} (\dot{q}^{*\beta}\dot{q}^j)^t \mathbb{Q}^{j\beta} (\dot{q}^{*\beta}\dot{q}^j) + \sum_{\alpha=1,\alpha\neq j}^{M} (\dot{q}^{*j}\dot{q}^{\alpha})^t \mathbb{Q}^{\alpha j} (\dot{q}^{*j}\dot{q}^{\alpha}), \\ &= 2 \sum_{\beta=1,\beta\neq j}^{M} (\dot{q}^{*\beta}\dot{q}^j)^t \mathbb{Q}^{j\beta} (\dot{q}^{*\beta}\dot{q}^j) \qquad \text{(according property (4)).} \end{aligned}$$

Then by using the matrix form of the quaternion product (1) :

$$\begin{aligned} H(\dot{q}^j) &= 2 \sum_{\beta=1,\beta\neq j}^{M} (Q^{*\beta}\dot{q}^j)^t \mathbb{Q}^{j\beta} (Q^{*\beta}\dot{q}^j), \\ &= 2 \sum_{\beta=1,\beta\neq j}^{M} \dot{q}^{j^t} (Q^{*\beta^t} \mathbb{Q}^{j\beta} Q^{*\beta}) \dot{q}^j, \end{aligned}$$

which can be written into the form,

$$H(\dot{q}^j) = 2\dot{q}^{j^t} \mathbf{N}^j \dot{q}^j, \tag{21}$$

where $\mathbf{N}^j = \sum_{\beta=1,\beta\neq j}^{M} Q^{*\beta^t} \mathbb{Q}^{j\beta} Q^{*\beta}$.

$H(\dot{q}^j)$ is a quadratic form. The optimal unit quaternion which maximizes this function is then just the eigenvector corresponding to the highest eigenvalue of the matrix $\mathbf{N}^j$.

7.3 Algorithm convergence

We show that the sequential algorithm proposed in the previous section does always converge monotonically to a local maximum.

Let $\mathbf{q}_{m,j} = (\dot{q}^2_{m+1}, \ldots, \dot{q}^j_{m+1}, \dot{q}^{j+1}_m, \ldots, \dot{q}^M_m)$. We start to prove that the cost function H of equation (19) is upper bounded. Using the fact that the product of two unit quaternions is still a unit quaternion, we see that each quadratic term of the double sum of H is always smaller than the highest eigenvalue $\lambda^{\alpha\beta}_{max}$ of the 4×4 symmetric matrix $\mathbb{Q}^{\alpha\beta}$. By doing the summation over α and β we verify that for any set of quaternions $\dot{q}^1, \dot{q}^2, \cdots, \dot{q}^M$:

$$H(\dot{q}^1, \dot{q}^2, \cdots, \dot{q}^M) \leq \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \lambda^{\alpha\beta}_{max}. \tag{22}$$

Thus the cost function H is upper bounded by the sum of the highest eigenvalue of the matrices $\mathbb{Q}^{\alpha\beta}$. Since, the series $H(\mathbf{q}_{m,j})$ is increasing by construction, this proves that our algorithm converges to a local maximum. If it converges to a global maximum is still an open question.

8 Experimental results

We have performed a global registration of M = 4 simulated sets containing each nine 3D points. Each set partially overlaps all the other ones. There are then $C^2_4 = 6$ different overlaps described by the subsets $O^{\alpha\beta}$. Each subset contains only 3 points in this simulation. The point coordinates vary between -100 and +100 units. The first point set is chosen as reference frame and is fixed. Each one of the three other sets is rotated by a random unit quaternion and translated by a random distance. In this example we do not add noise, thus the misregistration error value is expected to be very low.

We found that only 50 iterations were necessary to reduce the residual error from an initial value of 843 units to $0.2\ 10^{-3}$ unit. The CPU time needed is very low, it is lower than one millisecond for each iteration on a Sun Sparc Station running at 300 MHz. It does not depend on the number of points taken into account in the overlaps, the matrices $\mathbb{Q}^{\alpha\beta}$ being precomputed.

In this problem ground truth is available and so we can compute the residual error of the obtained solution from the true values for the unit quaternions and the translations. The relative residual error for the angle of the quaternion is about 10^{-5} and the one for the translations is about 10^{-6}. These results show that the intrinsic precision of the algorithm is very high. Its accuracy allows its use for many registration applications in computer vision.

We have for example applied this quaternion-based global registration on 3D points sampled on surface of real objects with a laser range finder. The Greek bust shown in Figure 2 is a difficult object to scan due to the presence of deep concavities. In this example 12 different scans have been recorded by using translational motions with a step of 0.1 mm. They contain more than 2 million points. The resulting partially overlapping parts of the object are illustrated in Figure 1. A quick and rough interactive registration was first performed (Figure 2-left). Then all the scans were registered simultaneously (Figure 2-right) by using an

ICP approach. This global registration is an extension of the classical ICP algorithm proposed by Besl and McKay [5] to register two sets. Two steps are iterated until convergence. In the first step, a point-to-point correspondence is efficiently established between the mutual overlaps of the sampled surfaces by using the multi-z-buffer space partitioning described in [3]. In the second step, the quaternion-based registration proposed in this paper is applied to simultaneously optimize the rigid motions. Figure 3 shows the convergence of the ICP algorithm by displaying a curve of pseudo-time against RMS residual global error. The initial RMS error was 0.68 mm and after one iteration falls down to 0.19 mm. The RMS error for the 12 scans is 0.11 mm after 20 iterations. The CPU time required for each iteration of the global ICP registration is 15 seconds on a Sun Sparc Station running at 300 MHz .

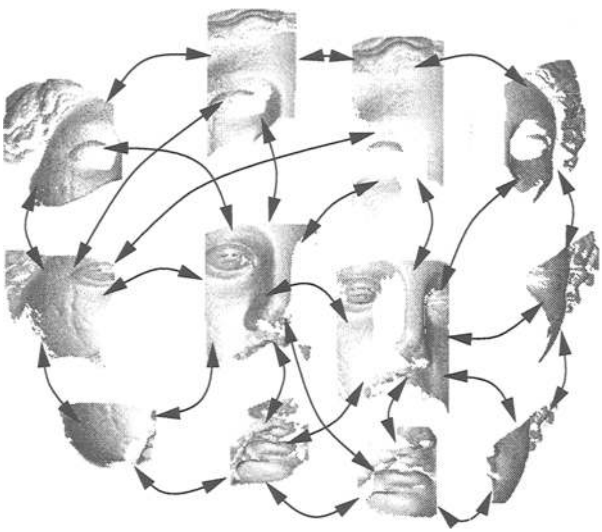

Fig. 1. Some pieces of a Greek bust mosaic (Hygia, Dion Museum, Greece).

A statuette was also digitized along 6 different views shown in Figure 4. Only four ICP iterations were needed to globally register these 6 views, the global RMS error decreasing from 0.61 mm to 0.21 mm. Figure 5 shows three renderings from different viewpoints of the registered data of the statuette.

9 Conclusion

We have proposed in this paper a complete solution to the problem of the global registration of multiple 3D point sets with known correspondences. It is based on the representation of the rotations by unit quaternions. Compared to our previous work based on a linearization of the rotations [2], this new approach

Fig. 2. The Greek bust after interactive (left) and global (right) registration of 12 scans.

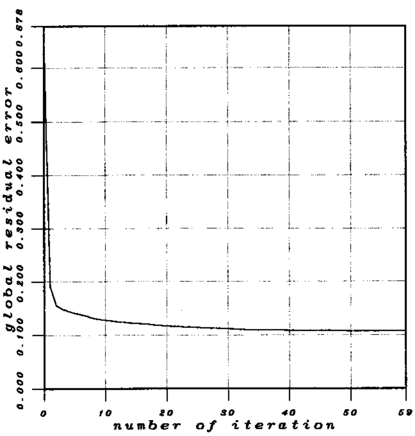

Fig. 3. ICP algorithm convergence of the 12 scans of the Greek bust.

does not require any assumption on the initial position of the point sets. It works well even when the data are initially faraway.

The experimental results have shown the excellent behaviour of the proposed algorithm to reach the optimum. With this solution the classic ICP algorithm which register only two point sets can be easily generalized into a **k-ICP** algorithm to register simultaneously k point sets ($k > 2$).

This method which has been successfully applied to the registration of several overlapping 3D sampled surfaces could be also very useful for other applications in computer vision.

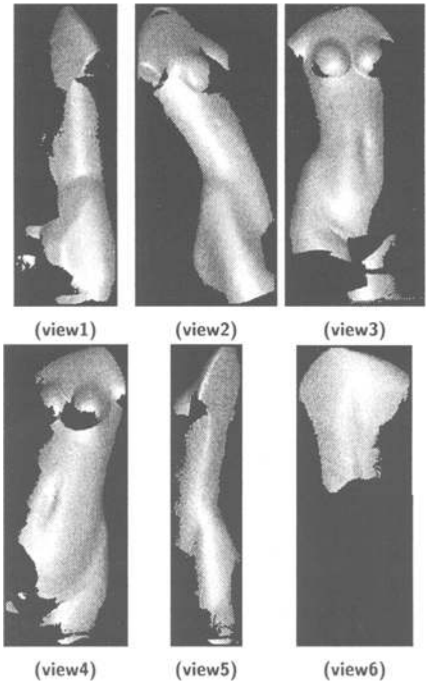

Fig. 4. Rendering of six views of a statuette (courtesy of the artist, Roland Coignard).

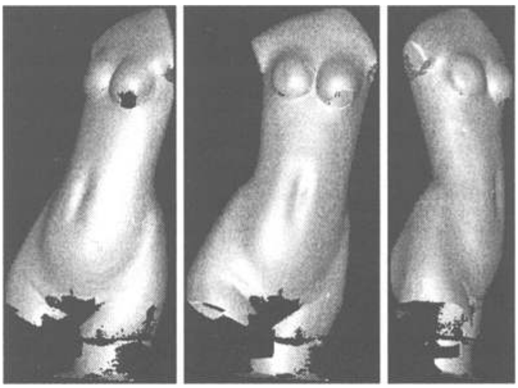

Fig. 5. Three different Renderings of the fusion of the six views of the statuette after their global registration.

Acknowledgment

This work was partly supported by the European ESPRIT project ARCHATOUR (EP III 9213).

Appendix A : development of $\bar{B}^t\bar{\mathbf{A}}^{-1}\bar{B}$

We look here for the development of the form $\bar{B}^t\bar{\mathbf{A}}^{-1}\bar{B}$ defined in section 6. Let denote $c^{\alpha\beta} = R^\alpha(N^{\alpha\beta}\bar{P}^{\alpha\beta})$ the transformed centroid of the overlap $O^{\alpha\beta}$ by the rotation R^α, multiplied by the number of points of $O^{\alpha\beta}$. We associate to each rotation R^α its unit quaternion $\dot{q}^\alpha$. Let denote a_{ij} the element (i,j) of the matrix $\bar{\mathbf{A}}^{-1}$ where i and $j \in [2..M]$. In order to homogenize the indices of the sums, we introduce the following null terms with an index 1, $a_{1j} = a_{i1} = 0$. Then:

$$\begin{aligned}
\bar{B}^t\bar{\mathbf{A}}^{-1}\bar{B} &= \sum_{k=1}^{M}\sum_{l=1}^{M} a_{kl}[\sum_{\alpha=1}^{M}(c^{k\alpha} - c^{\alpha k}) \cdot \sum_{\beta=1}^{M}(c^{l\beta} - c^{\beta l})],\\
&= \sum_{k=1}^{M}\sum_{l=1}^{M} a_{kl}\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}(c^{\alpha k}\cdot c^{\beta l} - c^{\alpha k}\cdot c^{l\beta} - c^{k\alpha}\cdot c^{\beta l} + c^{k\alpha}\cdot c^{l\beta}),\\
&= \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}(\sum_{k=1}^{M}\sum_{l=1}^{M} a_{kl}c^{\alpha k}\cdot c^{\beta l}) - \sum_{\alpha=1}^{M}\sum_{l=1}^{M}(\sum_{k=1}^{M}\sum_{\beta=1}^{M} a_{kl}c^{\alpha k}\cdot c^{l\beta})\\
&\quad - \sum_{k=1}^{M}\sum_{\beta=1}^{M}(\sum_{\alpha=1}^{M}\sum_{l=1}^{M} a_{kl}c^{k\alpha}\cdot c^{\beta l}) + \sum_{k=1}^{M}\sum_{l=1}^{M}(\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} a_{kl}c^{k\alpha}\cdot c^{l\beta}).
\end{aligned}$$

By appropriately changing some indices, this last quantity can be transformed into:

$$\bar{B}^t\bar{\mathbf{A}}^{-1}\bar{B} = \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{k=1}^{M}\sum_{l=1}^{M}\mu_{kl}^{\alpha\beta}c^{\alpha k}\cdot c^{\beta l},$$

where $\mu_{kl}^{\alpha\beta} = a_{kl} - a_{k\beta} - a_{\alpha l} + a_{\alpha\beta}$. Since $\bar{\mathbf{A}}^{-1}$ is a symmetric matrix ($a_{ij} = a_{ji}$), we have $\mu_{kl}^{\alpha\beta} = \mu_{lk}^{\beta\alpha}$. Let use now equation (3) to develop $\bar{B}^t\bar{\mathbf{A}}^{-1}\bar{B}$:

$$\begin{aligned}
\bar{B}^t\bar{\mathbf{A}}^{-1}\bar{B} &= \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{k=1}^{M}\sum_{l=1}^{M}\mu_{kl}^{\alpha\beta}R^\alpha(N^{\alpha k}\bar{P}^{\alpha k})\cdot R^\beta(N^{\beta l}\bar{P}^{\beta l}),\\
&= \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{k=1}^{M}\sum_{l=1}^{M}\mu_{kl}^{\alpha\beta}(\dot{q}^{*\beta}\dot{q}^\alpha)^t\mathbb{N}_{kl}^{\alpha\beta}(N^{\alpha k}\bar{P}^{\alpha k}, N^{\beta l}\bar{P}^{\beta l})(\dot{q}^{*\beta}\dot{q}^\alpha)\\
&= \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}(\dot{q}^{*\beta}\dot{q}^\alpha)^t\mathbb{Q}_t^{\alpha\beta}(\dot{q}^{*\beta}\dot{q}^\alpha),
\end{aligned}$$

where,

$$\begin{aligned}
\mathbb{Q}_t^{\alpha\beta} &= \sum_{k=1}^{M}\sum_{l=1}^{M}\mu_{kl}^{\alpha\beta}\mathbb{N}_{kl}^{\alpha\beta}(N^{\alpha k}\bar{P}^{\alpha k}, N^{\beta l}\bar{P}^{\beta l}),\\
&= \begin{bmatrix}
\bar{S}_{xx}^{\alpha\beta} + \bar{S}_{yy}^{\alpha\beta} + \bar{S}_{zz}^{\alpha\beta} & \bar{S}_{yz}^{\alpha\beta} - \bar{S}_{zy}^{\alpha\beta} & \bar{S}_{zx}^{\alpha\beta} - \bar{S}_{xz}^{\alpha\beta} & \bar{S}_{xy}^{\alpha\beta} - \bar{S}_{yx}^{\alpha\beta}\\
\bar{S}_{yz}^{\alpha\beta} - \bar{S}_{zy}^{\alpha\beta} & \bar{S}_{xx}^{\alpha\beta} - \bar{S}_{yy}^{\alpha\beta} - \bar{S}_{zz}^{\alpha\beta} & \bar{S}_{xy}^{\alpha\beta} + \bar{S}_{yx}^{\alpha\beta} & \bar{S}_{zx}^{\alpha\beta} + \bar{S}_{xz}^{\alpha\beta}\\
\bar{S}_{zx}^{\alpha\beta} - \bar{S}_{xz}^{\alpha\beta} & \bar{S}_{xy}^{\alpha\beta} + \bar{S}_{yx}^{\alpha\beta} & -\bar{S}_{xx}^{\alpha\beta} + \bar{S}_{yy}^{\alpha\beta} - \bar{S}_{zz}^{\alpha\beta} & \bar{S}_{yz}^{\alpha\beta} + \bar{S}_{zy}^{\alpha\beta}\\
\bar{S}_{xy}^{\alpha\beta} - \bar{S}_{yx}^{\alpha\beta} & \bar{S}_{zx}^{\alpha\beta} + \bar{S}_{xz}^{\alpha\beta} & \bar{S}_{yz}^{\alpha\beta} + \bar{S}_{zy}^{\alpha\beta} & -\bar{S}_{xx}^{\alpha\beta} - \bar{S}_{yy}^{\alpha\beta} + \bar{S}_{zz}^{\alpha\beta}
\end{bmatrix},
\end{aligned}$$

where, $\bar{S}_{xx}^{\alpha\beta} = \sum_{k=1}^{M} \sum_{l=1}^{M} \mu_{kl}^{\alpha\beta} \bar{x}^{\alpha k} \bar{x}^{\beta l}$, $\bar{S}_{xy}^{\alpha\beta} = \sum_{k=1}^{M} \sum_{l=1}^{M} \mu_{kl}^{\alpha\beta} \bar{x}^{\alpha k} \bar{y}^{\beta l}, \ldots$ with $(\bar{x}^{\alpha k}, \bar{y}^{\alpha k}, \bar{z}^{\alpha k})^t = N^{\alpha k} \bar{P}^{\alpha k}$ and $(\bar{x}^{\beta l}, \bar{y}^{\beta l}, \bar{z}^{\beta l})^t = N^{\beta l} \bar{P}^{\beta l}$.

$\mathbb{Q}_t^{\alpha\beta}$ has the same form as $\mathbb{N}^{ij}$ (*cf.* section 3) so it verifies the property (4).

References

1. R. Benjemaa and F. Schmitt. Recalage rapide de surfaces 3D aprs projection dans des multi-zbuffers. *5th European Conferences on Rapid Prototyping - Paris, 11 pages*, October 1996.
2. R. Benjemaa and F. Schmitt. Recalage global de plusieurs surfaces par une approche algbrique (in frensh). *RFIA'98, 11me Congrs AFCET de Reconnaissance des Formes et Intelligence Artificielle, Clermont-Ferrand, France*, January 1997, "accepted".
3. R. Benjemaa and F. Schmitt. Fast global registration of 3D sampled surfaces using a multi-z-buffer technique. *Proceedings of the IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling, Ottawa-Canada*, pages 113–120, May 1997, Extended version to be published in the Journal Image and Vision Computing.
4. R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Trans. on PAMI*, 18(5):540–547, May 1996.
5. P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. on PAMI*, 14(2):239–256, February 1992.
6. O. D. Faugeras and M. Hebert. A 3D recognition and positioning algorithm using geometrical matching between primitive surfaces. *Int. Joint. Conf. Artificial Intelligence, Karlsruhe, Germany*, pages 996–1002, 1983.
7. O. D. Faugeras and M. Hebert. The representation and recognition and locating of 3-D objects. *Int. Jour. Robotic Research*, vol. 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, April 1987.
9. T. D. Howell and J-C Lafon. The complexity of the quaternion product. Technical Report TR 75-245, Departement of Computer Science, Cornell University, Ithaca, N. Y., June 1975. ftp://ftp.netcom.com/pub/hb/hbaker/quaternion/cornellcstr75-245.ps.gz.
10. B. Kamgar-Parsi, J. L. Jones, and A. Rosenfeld. Registration of multiple overlapping range images: Scenes without distinctive features. *IEEE Trans. on PAMI*, 13(9):857–871, September 1991.
11. Kenichi Kanatani. Analysis of 3D rotation fitting. *IEEE Trans. on PAMI*, 16(5):543–549, May 1994.
12. A. Lorusso, D. W. Eggert, and R. B. Fisher. A comparison of four algorithms for estimating 3D rigid transformations. *British Machine Vision Conference, Birmingham, England*, pages 237–246, 1995.
13. L. Reyes-Avila. Quaternions : une reprsentation paramtrique systmatique des rotations finies. Technical Report 1303, INRIA-Rocquencourt, October 1990.
14. E. Salamin. Application of quaternions to computation with rotations. Technical report, Stanford AI Lab, 1979. ftp://ftp.netcom.com/pub/hb/hbaker/quaternion/stanfordaiwp79-salamin.ps.gz.
15. A. J. Stoddart and A. Hilton. Registration of multiple point sets. *ICPR'96, Vienna, Austria*, August 1996.