13.472J/1.128J/2.158J/16.940J
COMPUTATIONAL GEOMETRY

Lecture 21

Dr. K. H. Ko
Prof. N. M. Patrikalakis

Contents

Lecture 21

Object Matching

21.1 Various matching methods

- Moment method
- Principal component
- Contour and silhouette
- New representation
- Localization/registration
- Miscellaneous approaches

21.2 Methods through localization/registration

A basic goal of matching through localization/registration is to find *the best* rigid body transformation which aligns two objects as closely as possible. The correspondence search between two objects is a key issue in finding the best transformation for matching. Correspondence can be established by calculating distinct features of one object and locating the same ones on the other object. Therefore, the features have to be carefully chosen such that they are robustly extracted and invariant with respect to various transformations. Among various features, intrinsic differential properties are used for matching purposes. They are independent of parametrization and methods of representation, and only depend on the geometric shape of the object. Moreover, they are invariant under any rigid body transformations (rotation and translation).

21.3 Classification of matching methods

Two types of matching can be considered: *global* and *partial*. Simply, the global matching is regarded as the matching for whole objects, while the partial matching is considered as

the matching of part of objects. Matching problems can be further categorized based on the availability of correspondence or initial transformation information between two objects and the application of scaling. The classification of matching problems is summarized in Table 21.1. In the table, acronyms are used for simplification as follows:

- C : Correspondence information is provided.
- I : Initial information on correspondence is provided.
- N : No correspondence information is available.
- P : Partial matching.
- G : Global matching.
- WOS : Without scaling.
- WS : With scaling.

Criteria	Global matching		Partial matching	
	Without scaling	With scaling	Without scaling	With scaling
Correspondence information	CGWOS	CGWS	CPWOS	CPWS
Initial information	IGWOS	IGWS	IPWOS	IPWS
No information	NGWOS	NGWS	NPWOS	NPWS

Table 21.1: Classification of matching problems

When correspondence information is provided, which is one of the types CGWOS or CPWOS, then a matching problem is simply reduced to calculation of the rigid body transformation [3, 4]. If no correspondence is known, but a good initial approximation for the transformation is available (IGWOS or IPWOS), then popular iterative schemes such as the *Iterative Closest Point (ICP)* algorithm [1] can be employed. However, when no prior clue for correspondence or transformation is given (NGWOS or NPWOS), the matching problem becomes more complicated. In this case, the solution process must provide a means to establish such correspondence information such as in [2].

Scaling is another factor that needs to be considered separately. If a matching problem involves scaling effects, then direct comparison of quantitative measures cannot be used any longer. For the global matching case, a scaling factor can be estimated by the ratio of surface areas and applied to resolve the scaling transformation. However, when it comes to partial matching, such area information becomes useless for the scaling factor estimation. When the correspondence information between two objects is known (CGWS or CPWS), the scaling factor between the objects can be easily obtained by using the ratio of Euclidean distances between two sets of corresponding points or areas, or the ratio of the principal curvatures. If an initial scaling value as well as a good initial approximation is provided (IGWS or IPWS), the ICP algorithm by Besl [1] or other optimization schemes such as the *quasi-Newton method*

[12] can be employed. The problem of NGWS type can be solved by the moment method using the principal moment of inertia and ratio of areas or volumes. No attempt, however, has been made to solve the problems of NPWS type. For a more detailed overview, see Ko [6].

21.4 Problem statement

21.4.1 Distance metric

The Euclidean distance between two points $\mathbf{p}_1$ and $\mathbf{p}_2$ is defined as

$$d_e(\mathbf{p}_1, \mathbf{p}_2) = |\mathbf{p}_1 - \mathbf{p}_2|. \tag{21.1}$$

We also define the minimum distance between a surface $\mathbf{r}$ and a point $\mathbf{p}$ as follows:

$$d_{sp}(\mathbf{r}, \mathbf{p}) = min\{d_e(\mathbf{p}, \mathbf{p}_i), \forall \mathbf{p}_i \in \mathbf{r}\}. \tag{21.2}$$

21.4.2 Distance between a point and a parametric surface

Let us assume that we have a point $\mathbf{p}$ and a parametric surface $\mathbf{r} = \mathbf{r}(u, v)$, $0 \leq u, v \leq 1$. Then the squared distance function is defined as follows:

$$\begin{aligned} D(u, v) &= |\mathbf{p} - \mathbf{r}(u, v)|^2, \\ &= (\mathbf{p} - \mathbf{r}(u, v)) \cdot (\mathbf{p} - \mathbf{r}(u, v)). \end{aligned} \tag{21.3}$$

Finding the minimum distance between $\mathbf{p}$ and $\mathbf{r}$ is reduced to minimizing (21.3) within the square $0 \leq u, v \leq 1$. Therefore, the problem needs to be broken up into several sub-problems which consider the behavior of the distance function at the boundary and in the interior of the bound [10]. The sub-problems are summarized as follows: Find the minimum distances (1) in the interior domain, (2) along the boundary curves and (3) from the corner points. Among those minimum distances, the smallest one is chosen as the minimum distance between the point $\mathbf{p}$ and the surface $\mathbf{r}$. A robust calculation of the minima of the distance function (21.3) can be achieved by the *Interval Projected Polyhedron (IPP)* algorithm [13, 10, 14].

21.4.3 Distance metric function

A function can be defined using the squared distance function (21.3) to formulate a matching problem. Suppose that we have a NURBS surface $\mathbf{r}_B$ and an object $\mathbf{r}_A$ represented in discrete points or surfaces. Then, a matching problem can be stated as finding the rigid body transformation (a translation vector $\mathbf{t}$ and a rotation matrix $\mathbf{R}$) so that a global distance metric function

$$\Phi = \sum_{\forall \mathbf{p} \in \mathbf{r}_A} d_{sp}(\mathbf{r}_B, (\sigma \mathbf{R}\mathbf{p} + \mathbf{t})) \tag{21.4}$$

becomes minimal, where σ is a scaling factor.

21.5 Matching problems : CGWOS, CPWOS, CGWS or CPWS

21.5.1 Resolving the scaling effects

Matching problems of CGWS or CPWS type involve the scaling effects. Therefore, a scaling factor has to be estimated so that the scaling transformation is performed, before calculating the rigid body transformation. Since correspondence information between two objects are available, a scaling factor can be estimated by using the ratio of the principal curvatures at the corresponding points. After scaling has been resolved, the matching problems of CGWS or CPWS type are treated as those of CGWOS or CPWOS type.

21.5.2 Rotation and translation

Suppose that we have two 3-tuples $\mathbf{m}_i$ and $\mathbf{n}_i (i = 1, 2, 3)$, and the correspondence information for each point. From these points, a translation vector and a rotation matrix can be calculated. The translation vector is easily obtained by using the centroids of each 3-tuple. The centroids $\mathbf{c}_m$ and $\mathbf{c}_n$ are given by

$$\mathbf{c}_m = \frac{1}{3}\sum_{i=1}^{3} \mathbf{m}_i, \quad \mathbf{c}_n = \frac{1}{3}\sum_{i=1}^{3} \mathbf{n}_i, \tag{21.5}$$

and the difference between $\mathbf{c}_m$ and $\mathbf{c}_n$ becomes the translation vector $\mathbf{t}_T = \mathbf{c}_n - \mathbf{c}_m$. A rotation matrix consists of three unknown components (the Euler angles). Since the two 3-tuples provide nine constraints, the rotation matrix may be constructed by using some of the constraints. But the results could be different if the remaining constraints are used for the rotation matrix calculation [3]. In order to use all the constraints equally, the least squares method may be employed [3]. The basic solution by Horn [3] is described below. Suppose that the translation has been performed. Then what is left is to find the rotation matrix $\mathbf{R}$ so that

$$\begin{aligned} \Phi' &= \sum_{i=1}^{3} |\mathbf{n}_i - (\mathbf{R}\mathbf{m}_i)|^2 \\ &= \sum_{i=1}^{3} |\mathbf{n}_i|^2 - 2\sum_{i=1}^{3} \mathbf{n}_i \cdot (\mathbf{R}\mathbf{m}_i) + \sum_{i=1}^{3} |\mathbf{R}\mathbf{m}_i|^2 \end{aligned} \tag{21.6}$$

is minimized. Here, $D = \sum_{i=1}^{3} \mathbf{n}_i \cdot (\mathbf{R}\mathbf{m}_i)$ has to be maximized to minimize Φ'. The problem can be solved in the quaternion framework. A quaternion can be considered as a vector with four components, i.e. a vector part in 3D and a scalar part. A rotation can be equivalently defined as a unit quaternion $\check{\mathbf{q}} = \left[\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2})a_x, \sin(\frac{\theta}{2})a_y, \sin(\frac{\theta}{2})a_z\right]$ which represents a rotation movement around (a_x, a_y, a_z) by θ degree. In the quaternion framework, the problem is reduced to the eigenvalue problem of the 4×4 matrix $\mathbf{H}$ obtained from the correlation matrix $\mathbf{M}$:

$$\mathbf{H} = \begin{bmatrix} s_{11}+s_{22}+s_{33} & s_{23}-s_{32} & s_{31}-s_{13} & s_{12}-s_{21} \\ s_{23}-s_{32} & s_{11}-s_{22}-s_{33} & s_{12}+s_{21} & s_{31}+s_{13} \\ s_{31}-s_{13} & s_{12}+s_{21} & s_{22}-s_{11}-s_{33} & s_{23}+s_{32} \\ s_{12}-s_{21} & s_{31}+s_{13} & s_{23}+s_{32} & s_{33}-s_{22}-s_{11} \end{bmatrix}, \tag{21.7}$$

where

$$\mathbf{M} = \sum_{i=1}^{3} \mathbf{n}_i \mathbf{m}_i^T = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}. \tag{21.8}$$

The eigenvector corresponding to the maximum positive eigenvalue is a quaternion which minimizes the equation (21.6). An orthonormal rotation matrix $\mathbf{R}$ can be recovered from a quaternion $\check{q} = [q_0, q_1, q_2, q_3]$ by

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}. \tag{21.9}$$

The procedure described above can also be applied to the case where more than three corresponding point pairs are provided.

21.6 Matching problems : IGWOS, IPWOS, IGWS or IPWS

21.6.1 Iterative Closest Point (ICP) algorithm [1] for IGWOS or IPWOS

Algorithm

- *The point set P with N_p points $\{\vec{p}_i\}$ from the data shape and the model shape X (with N_x supporting geometric primitives: points, lines, or triangles) are given.*
- *The iteration is initialized by setting $P_0 = P$, $\vec{q}_0 = [1, 0, 0, 0, 0, 0, 0]^T$ and $k = 0$. The registration vectors are defined relative to the initial data set P_0 so that the final registration represents the complete transformation. Steps 1,2,3, and 4 are applied until convergence within a tolerance τ.*
  a. *Compute the closest points: $Y_k = C(P_k, X)$.*
  b. *Compute the registration: $(\vec{q}_k, d_k) = Q(P_0, Y_k)$.*
  c. *Apply the registration: $P_{k+1} = \vec{q}_k(P_0)$.*
  d. *Terminate the iteration when the change in mean-square error falls below a preset threshold $\tau > 0$ specifying the desired precision of the registration: $|d_k - d_{k+1}| < \tau$.*

21.6.2 ICP algorithm for scaling effects

When initial information on transformation is provided, the ICP method can be extended to resolve scaling effects in the matching problem. A scaling factor σ is included in the objective function (21.4). The scaling transformation is performed at step c in the ICP algorithm. In this case we have to provide seven initial values (three for translation, three for rotation and one for scaling).

21.7 Matching problems : NGWOS or NPWOS

21.7.1 Search method [2]

Chua and Jarvis [2] developed a method to align two objects through registration assuming no prior knowledge of correspondence between two range data sets. They use a bi-quadratic polynomial to fit data points in the local area in the least squares sense and calculate the principal curvatures and Darboux frames. Then they construct a list of sensed data points based on the fit error. Three seed points are selected to form the list such that the area of the triangle represented by the seed points becomes maximized to reduce any mismatch error. Three constraints (curvature, distance and direction) are imposed to sort out possible corresponding points out of the model data set. Then a list of transformations can be obtained from the candidate points and an optimum transformation is selected. Various searching algorithms are described and demonstrated in [2].

21.7.2 KH method

The overall diagram of the KH method [8] is shown in Figure 21.1. The input of the process includes two objects and three pairs of the Gaussian and the mean curvatures at three different non-collinear locations. The algorithm yields the minimum value of Φ in the equation (21.4), and the corresponding rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$. Since no scaling effect is involved, we assume that a scaling factor $\sigma = 1$.

Step 10

Step 10 is to select three non-collinear points $\mathbf{m}_1$, $\mathbf{m}_2$ and $\mathbf{m}_3$ on $\mathbf{r}_1$ away from the boundary of $\mathbf{r}_1$ where each point has different, Gaussian K and mean curvature H values. At $\mathbf{m}_i$, we have K_i and H_i, where $i = 1, 2, 3$. Next, subdivide $\mathbf{r}_2$ into rational Bézier surface patches $\mathbf{B}_j$ $(j = 1, \cdots, n)$ by inserting appropriate knots [5, 11]. Then for each rational Bézier surface patch $\mathbf{B}_j$, we express K_j and H_j in the bivariate rational Bernstein polynomial basis using rounded interval arithmetic to formulate the problem. This allows us to use the *Interval Projected Polyhedron (IPP) algorithm* [10, 13] for solving nonlinear polynomial systems. For each pair K_i and H_i, we solve the following system of equations by the IPP technique.

$$\begin{aligned} K_j(u, v) &= K_i \pm \delta_K, \\ H_j(u, v) &= H_i \pm \delta_H, \quad (j = 1, \cdots, n \ \ and \ \ i = 1, 2, 3), \end{aligned} \tag{21.10}$$

where δ_K and δ_H represent the uncertainty of estimated curvatures from data points. For each pair of K_i and H_i, a list of roots $L_i = (u_{ik}, v_{ik})$, $(k = 1, \cdots, d_i)$ is obtained.

Step 12 (selection process)

A simple pruning search based on the Euclidean distance can be applied to the selection process. We have three lists of candidate points, $L_i = (u_{ik}, v_{ik})$, $(k = 1, \cdots, d_i)$ from which one 3-tuple $(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3)$

$$\mathbf{n}_1 = \mathbf{r}_2(u_{1k}, v_{1k}), \ \ \mathbf{n}_2 = \mathbf{r}_2(u_{2k}, v_{2k}), \ \ \mathbf{n}_3 = \mathbf{r}_2(u_{3k}, v_{3k}), \tag{21.11}$$

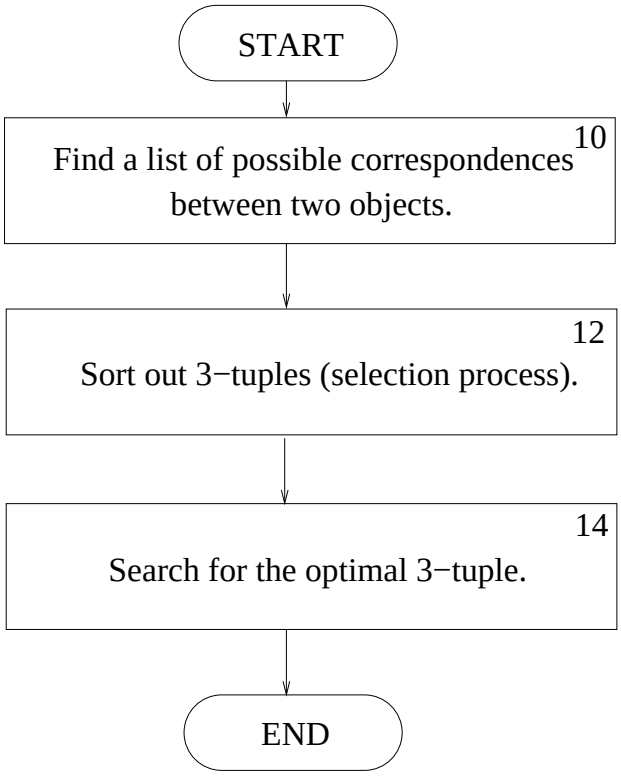

Figure 21.1: A diagram of the KH method

is selected to satisfy the following Euclidean distance constraints simultaneously

$$\begin{aligned} ||\mathbf{m}_1 - \mathbf{m}_2| - |\mathbf{n}_1 - \mathbf{n}_2|| &< \delta_{select}, \\ ||\mathbf{m}_2 - \mathbf{m}_3| - |\mathbf{n}_2 - \mathbf{n}_3|| &< \delta_{select}, \\ ||\mathbf{m}_3 - \mathbf{m}_1| - |\mathbf{n}_3 - \mathbf{n}_1|| &< \delta_{select}, \end{aligned} \tag{21.12}$$

where δ_{select} is a user-defined tolerance.

Step 14

The correspondence information between each point $\mathbf{m}_i$ on $\mathbf{r}_1$ and $\mathbf{n}_i$ on $\mathbf{r}_2$ is established, from which a list of translation vectors and a rotation matrices can be obtained. We choose a translation vector and a rotation matrix which produces a minimum value of equation (21.4) with $s = 1$, see Ko *et al.* [8].

21.8 Matching problems : NGWS

Since global matching is considered, scaling effects in the matching problem of this type can be easily resolved by using integral properties such as surface areas or volumes. After scaling transformation, the center of mass and the principal moments are used to find the rigid body transformation.

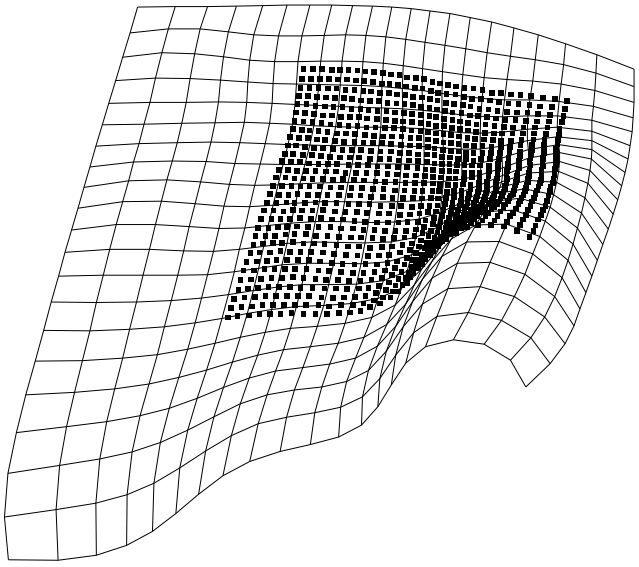

Figure 21.2: Localized surfaces

Examples are presented to demonstrate the proposed algorithms. Solids bounded by bicubic integral B-spline surface patches A and B are used. Solid A is enclosed in a rectangular box of $25mm \times 23.48mm \times 11mm$. Here, the height of solid A is $25mm$. Figure 21.3 shows a sequence of operations for matching of the two surfaces using the principal moments of inertia of input solids. In this example, for clarity, only part of the boundary surfaces of the solids are displayed. The smaller solid has been translated, rotated, uniformly scaled and reparameterized. In Figure 21.3-(A), two boundary surfaces of the input solids are shown with their control points. Those two surfaces have similar shape but different numbers of control points and parametrization. Matching the centroids of the two solids is performed by translating the small solid by the position difference between the centroids, which is demonstrated in Figure 21.3-(B). The orientation of the largest principal moment of inertia of the solid A is aligned to that of the largest one of the solid B. Similarly, the remaining two orientations are aligned based on the values of the principal moments of inertia. After matching the orientations of the principal moment of inertia, the two solids are aligned in their orientations as shown in Figure 21.3-(C). Figure 21.3-(D) shows that the two solids match after uniform scaling obtained from the ratio between the volumes of the two solids, 4.651, is applied to the small solid.

21.9 Matching problems : NPWS

21.9.1 Umbilical point method [7]

The correspondence search for matching problems of NGWOS or NPWOS type only deals with qualitative aspects. Since the ω-plane is not affected by scaling, only qualitative correspondence can be established in the process. This implies that without a scaling factor applied, a rigid body transformation cannot be obtained for aligning two surfaces. Therefore, a scaling factor has to be estimated before any transformation is considered.

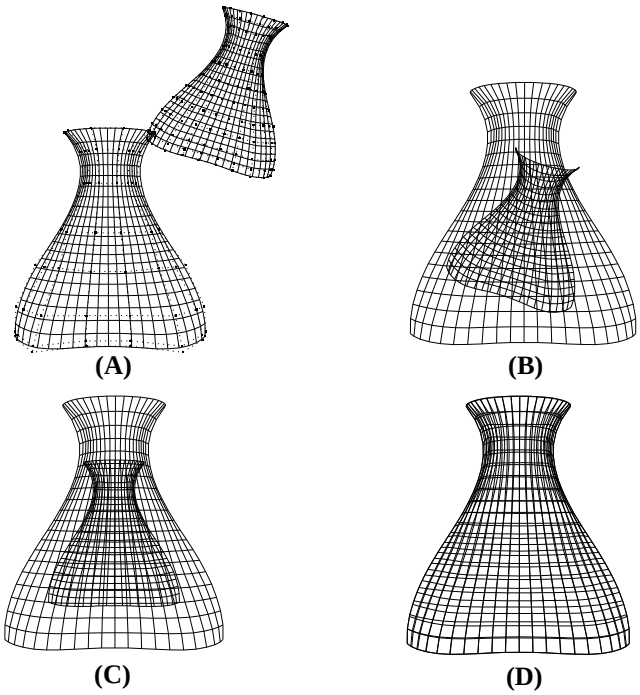

Figure 21.3: Matching via integral properties

Method 1

Let us assume that we have two surfaces $\mathbf{r}_1$ and $\mathbf{r}_2$, where $\mathbf{r}_1$ is an approximated surface of input data points. The overall procedure is shown in Figure 21.4.

In step 100, all generic umbilical points are located on both surfaces $\mathbf{r}_1$ and $\mathbf{r}_2$ using the IPP algorithm [9, 10]. Non-generic umbilical points are not used in this process. If a generic umbilical point does not exist, this procedure cannot be applied.

In step 102, the correspondence search is performed. The value ω in the complex plane is scale-independent so that qualitative correspondences can be built from this step. Suppose that matched pairs are denoted as m_k, $(k = 1, \cdots, n_k)$, where n_k is the number of matched pairs. Then when at least one pair is found, the next step 104 is performed. If no correspondence is established, then the algorithm stops, implying that the umbilical point method cannot be used in this case.

Step 104 resolves the scaling transformation. To calculate a scaling factor, the normal curvatures are evaluated at the corresponding umbilical points on both surfaces $\mathbf{r}_1$ and $\mathbf{r}_2$. Then the ratio between them is obtained as a scaling factor. Suppose that a surface $\mathbf{r}$ is scaled with a scaling factor σ, denoted as $\mathbf{r}_\sigma$. Then the normal curvature κ on $\mathbf{r}$ is scaled to be $\frac{\kappa}{\sigma}$ on $\mathbf{r}_\sigma$. Therefore using this relation, the scaling factor can be recovered.

In step 106, after sorting out candidate points, a rigid body transformation is estimated by using the unit quaternion method [3]. Since the number of matched pairs is more than three and if at least three pairs survive the selection process, the problem reduces to finding a rigid body transformation with three known corresponding pairs. Using the methods in [3] a rotation matrix and a translation vector can be calculated. If the matched pairs fail in the

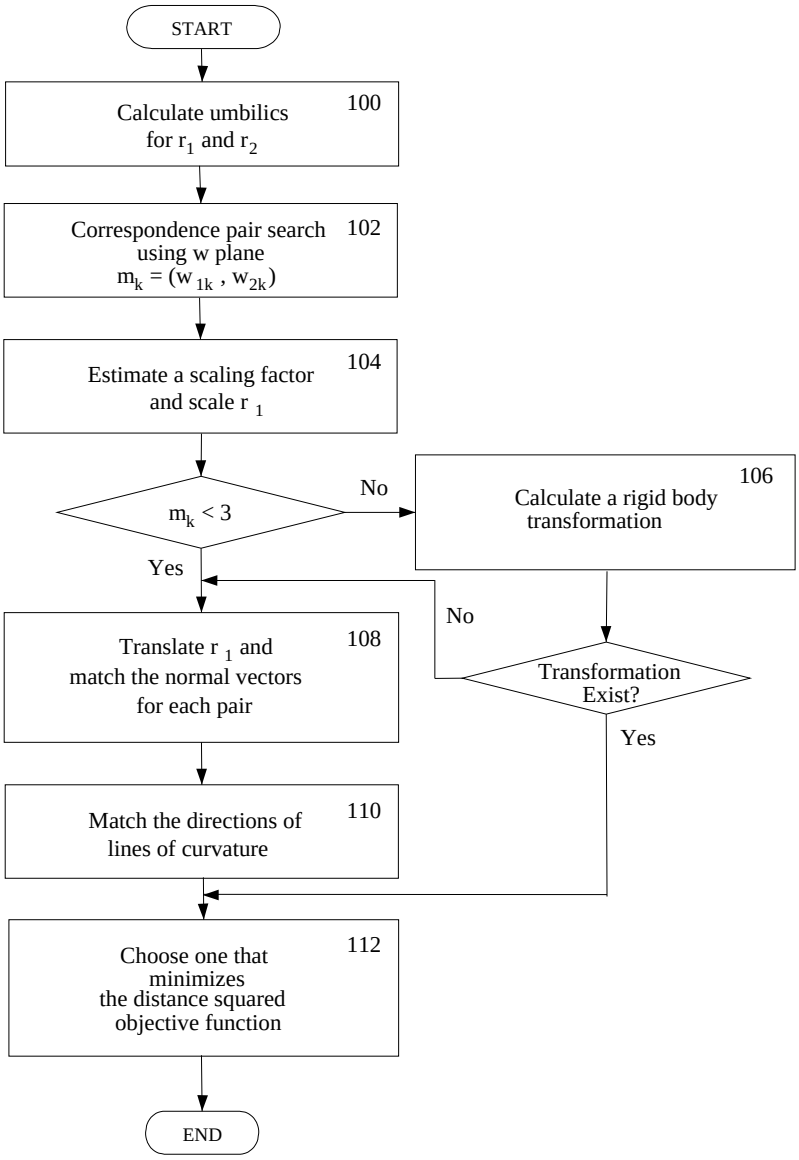

Figure 21.4: A diagram for matching using umbilics.

selection process, then the algorithm goes to step 108 which deals with the matching process with less than three matched pairs.

In step 108, the orientations of the normal vectors at the corresponding umbilical points are aligned. First, translate the scaled surface $\mathbf{r}_1$ by the difference between the positions of the matched umbilical points. Then, align the normal vectors at the umbilical points. The alignment of the normal vectors can be achieved by using the unit quaternion method [3]. Let us assume that we have two normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ at the corresponding umbilical points for $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively. The problem of matching the normal vectors can be stated as: rotate $\mathbf{n}_1$ around the vector $\mathbf{N}_n = \frac{\mathbf{n}_1 \times \mathbf{n}_2}{\|\mathbf{n}_1 \times \mathbf{n}_2\|}$ by an angle θ formed by $\mathbf{n}_1$ and $\mathbf{n}_2$. The angle θ can be calculated by $\theta = \arccos(\mathbf{n}_1, \mathbf{n}_2)$, see [3] for details of rotation in the quaternion frame.

In step 110, matching of lines of curvature emanating from an umbilical point is performed. Depending on the type of the umbilical point, one (*lemon*) or three (*star and monstar*) lines of curvature pass through the umbilical point, and each direction can be determined by the structure of the cubic terms $C(x, y)$ as summarized in Lecture 20. The directions can be obtained by calculating angles of the lines of curvature with respect to the local coordinate system at the umbilical point [9, 10]. Using the angles, vectors which indicate the directions of lines of curvature at the umbilical point can be obtained. These vectors are calculated at the matched umbilical points on $\mathbf{r}_1$ and $\mathbf{r}_2$. Suppose that the number of the direction vectors on $\mathbf{r}_1$ is n_{v1} and the number of the direction vectors on $\mathbf{r}_2$ is n_{v2}. Choose one vector from $\mathbf{r}_2$ and align all of the vectors on $\mathbf{r}_1$. This process produces n_{v1} matched cases among which one match is chosen that minimizes equation (21.4). This alignment is achieved by rotation around the normal vector in the tangent plane at the matched umbilical point. Therefore, the rotation method using the unit quaternion can be used in this process [3].

Method 2

The rigid body transformation can also be obtained by using the KH method described in Section 4.2 after the scaling transformation is resolved. The algorithm is the same as in Figure 21.4 from step 100 through step 104. After the scaling transformation is resolved, the KH method can be used to find the rigid body transformation between two objects.

21.9.2 Optimization method [7]

A problem with scaling effects can be solved with an optimization technique. Since there is no quantitative measure that can be used to estimate a scaling value, the solution scheme has to resort to an optimum search method which can narrow down the best estimate from the possible set of candidate solutions.

The KH method can be treated as a function of the scaling factor which yields a value of Φ in equation (21.4) when a scaling factor is given. Namely, steps 10, 12 and 14 in the diagram of Figure 21.1 are grouped as a function f such that

$$f = \Phi(\sigma, \mathbf{R}, \mathbf{t}), \tag{21.13}$$

where Φ is the expression given in the equation (21.4), σ the scaling factor, $\mathbf{R}$ the rotation matrix and $\mathbf{t}$ the translation vector. Since the rotation matrix and the translation vector are

obtained from the KH method, we can reduce equation (21.13) to a function of one variable, or $f = \Phi(\sigma)$. Therefore, when σ is given as input, then f produces the best rigid body transformation (a translation vector and a rotation matrix) as well as the value of the object function defined in equation (21.4). When the selection process fails, the tolerance δ_{select} is iteratively increased so that any triplet can be obtained. When no triplet is found, then the function f is penalized to yield a very large value.

Using the function $f(\sigma)$, the problem can be formulated as a one dimensional optimization problem to find a scaling value which yields the minimum of f. A popular one dimension optimization scheme, the golden section search [12] can be employed to solve it.

An initial bracket $[a, b, c]$ of the scaling factor is provided which contains an optimum value, and satisfies $f(a) > f(b)$ and $f(c) > f(b)$. Suppose this bracket is I_0. The golden section search starts with I_0 and continues while the size of an interval containing the optimum value σ is larger than a user defined tolerance which determines the size of the interval.

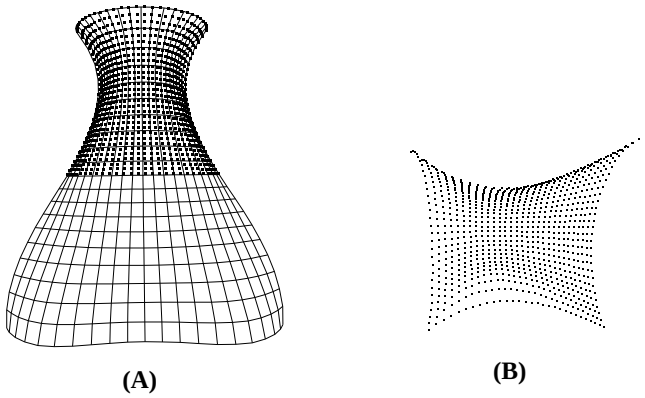

Figure 21.5: A matching example of NPWS type

21.10 Matching problems : offset method

21.10.1 Distance function

The minimum distance d from each measured point to an offset of the design surface is defined as the distance between the points $\mathbf{R}_i$ and $\mathbf{Q}_i$ minus the offset distance h, see Figure 21.6

$$d(\mathbf{R}_i, \mathbf{Q}_i, h) = |\mathbf{R}_i - \mathbf{Q}_i - h\mathbf{n}_i| \tag{21.14}$$

where $\mathbf{n}_i$ is the unit normal vector of $\mathbf{P}(u, v)$ at $\mathbf{Q}_i$ given by

$$\mathbf{n}_i = \frac{\mathbf{P}_u(u, v) \times \mathbf{P}_v(u, v)}{||\mathbf{P}_u(u, v) \times \mathbf{P}_v(u, v)||}. \tag{21.15}$$

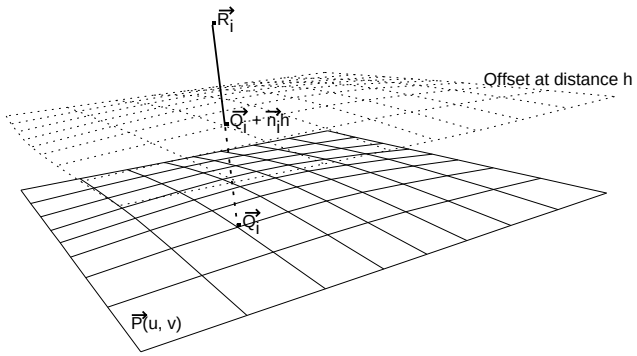

Figure 21.6: Definition of distance function.

21.10.2 Objective function

- The localization procedure minimizes the Root Mean Square Distance (RMS) between the points $\mathbf{r}_i$ and an offset to $\mathbf{P}(u, v)$ at a distance h. The RMS distance is thus given by

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^{m} d^2(\mathbf{r}_i, \mathbf{q}_i, h)}{m}} \tag{21.16}$$

  with

$$d(\mathbf{r}_i, \mathbf{q}_i, h) = |\mathbf{r}_i - \mathbf{q}_i - h\mathbf{n}_i| \tag{21.17}$$

  where $\mathbf{q}_i$ is the point nearest to the transformed points $\mathbf{r}_i$ on the offset surface with distance h from the design surface $\mathbf{P}(u, v)$. The position of the point $\mathbf{q}_i$ varies together with the motion of the point $\mathbf{r}_i$ depending on the six motion parameters. Therefore $\mathbf{q}_i$ can be viewed as a function depending on the variables $\psi, \theta, \phi, t_x, t_y, t_z, h$.

- It is sufficient to minimize the sum of the squared distances instead of RMS distance. Thus the objective function for the unconstrained minimization is given by

$$F(\psi, \theta, \phi, t_x, t_y, t_z, h) = \sum_{i=1}^{m} d^2(\mathbf{r}_i, \mathbf{q}_i, h). \tag{21.18}$$

- If we denote $\mathbf{x} = (\psi, \theta, \phi, t_x, t_y, t_z, h)^T$ then the problem can be reduced to searching for zeros of the gradient vector field $\nabla F(\mathbf{x})$, i.e.

$$\nabla F(\mathbf{x}) = \mathbf{0}. \tag{21.19}$$

  The Newton-Raphson method can be used to solve the nonlinear system (21.19), namely

$$\begin{aligned} [\mathbf{H}(\mathbf{x})]\delta\mathbf{x} &= -\mathbf{g}(\mathbf{x}) \\ \mathbf{x}^{(i+1)} &= \mathbf{x}^{(i)} + \delta\mathbf{x}. \end{aligned} \tag{21.20}$$

where $\mathbf{g}(\mathbf{x})$ and $[\mathbf{H}(\mathbf{x})]$ are the gradient vector and the Hessian matrix of the objective function $F(\mathbf{x})$ given by

$$\mathbf{g}(\mathbf{x}) = \nabla F \tag{21.21}$$

$$[\mathbf{H}(\mathbf{x})] = \nabla^2 F = \frac{\partial^2 F}{\partial x_i \partial x_j} \quad (i,j = 1,\ldots,7) \tag{21.22}$$

where the notation $\mathbf{x} = \{x_i\} = \{\psi, \theta, \phi, t_x, t_y, t_z, h\}$ is used.

21.10.3 Gradient vector

The minimization of the objective function can be speeded up by supplying the gradient vector of F at each iteration. Considering that

$$[C]_{t_x} = [C]_{t_y} = [C]_{t_z} = [C]_h = 0 \tag{21.23}$$

$$\mathbf{t}_\psi = \mathbf{t}_\theta = \mathbf{t}_\phi = \mathbf{t}_h = 0 \tag{21.24}$$

$$(\mathbf{q}_i)_h = 0 \tag{21.25}$$

$$(\mathbf{n}_i)_h = 0 \tag{21.26}$$

$$h_\psi = h_\theta = h_\phi = h_{t_x} = h_{t_y} = h_{t_z} = 0 \tag{21.27}$$

$$h_h = 1 \tag{21.28}$$

the gradient vector of the objective function reduces to

$$\begin{aligned}
\nabla F &= \begin{bmatrix} F_\psi \\ F_\theta \\ F_\phi \\ F_{t_x} \\ F_{t_y} \\ F_{t_z} \\ F_h \end{bmatrix} = \sum_{i=1}^{m} \begin{bmatrix} d_\psi^2(\mathbf{r}_i, \mathbf{q}_i, h) \\ d_\theta^2(\mathbf{r}_i, \mathbf{q}_i, h) \\ d_\phi^2(\mathbf{r}_i, \mathbf{q}_i, h) \\ d_{t_x}^2(\mathbf{r}_i, \mathbf{q}_i, h) \\ d_{t_y}^2(\mathbf{r}_i, \mathbf{q}_i, h) \\ d_{t_z}^2(\mathbf{r}_i, \mathbf{q}_i, h) \\ d_h^2(\mathbf{r}_i, \mathbf{q}_i, h) \end{bmatrix} = \sum_{i=1}^{m} \begin{bmatrix} 2d(\mathbf{r}_i, \mathbf{q}_i, h) \cdot d_\psi(\mathbf{r}_i, \mathbf{q}_i, h) \\ 2d(\mathbf{r}_i, \mathbf{q}_i, h) \cdot d_\theta(\mathbf{r}_i, \mathbf{q}_i, h) \\ 2d(\mathbf{r}_i, \mathbf{q}_i, h) \cdot d_\phi(\mathbf{r}_i, \mathbf{q}_i, h) \\ 2d(\mathbf{r}_i, \mathbf{q}_i, h) \cdot d_{t_x}(\mathbf{r}_i, \mathbf{q}_i, h) \\ 2d(\mathbf{r}_i, \mathbf{q}_i, h) \cdot d_{t_y}(\mathbf{r}_i, \mathbf{q}_i, h) \\ 2d(\mathbf{r}_i, \mathbf{q}_i, h) \cdot d_{t_z}(\mathbf{r}_i, \mathbf{q}_i, h) \\ 2d(\mathbf{r}_i, \mathbf{q}_i, h) \cdot d_h(\mathbf{r}_i, \mathbf{q}_i, h) \end{bmatrix} \\
&= \sum_{i=1}^{m} \begin{bmatrix} 2(\mathbf{r}_i - \mathbf{q}_i - \mathbf{n}_i h) \cdot ([C]_\psi \mathbf{R}_i - (\mathbf{q}_i)_\psi - (\mathbf{n}_i)_\psi h) \\ 2(\mathbf{r}_i - \mathbf{q}_i - \mathbf{n}_i h) \cdot ([C]_\theta \mathbf{R}_i - (\mathbf{q}_i)_\theta - (\mathbf{n}_i)_\theta h) \\ 2(\mathbf{r}_i - \mathbf{q}_i - \mathbf{n}_i h) \cdot ([C]_\phi \mathbf{R}_i - (\mathbf{q}_i)_\phi - (\mathbf{n}_i)_\phi h) \\ 2(\mathbf{r}_i - \mathbf{q}_i - \mathbf{n}_i h) \cdot (\mathbf{t}_{tx} - (\mathbf{q}_i)_{tx} - (\mathbf{n}_i)_{tx} h) \\ 2(\mathbf{r}_i - \mathbf{q}_i - \mathbf{n}_i h) \cdot (\mathbf{t}_{ty} - (\mathbf{q}_i)_{ty} - (\mathbf{n}_i)_{ty} h) \\ 2(\mathbf{r}_i - \mathbf{q}_i - \mathbf{n}_i h) \cdot (\mathbf{t}_{tz} - (\mathbf{q}_i)_{tz} + (\mathbf{n}_i)_{tz} h) \\ 2(\mathbf{r}_i - \mathbf{q}_i - \mathbf{n}_i h) \cdot \mathbf{n}_i \end{bmatrix} = \sum_{i=1}^{m} \begin{bmatrix} 2\mathbf{d}_i \cdot [C]_\psi \mathbf{R}_i \\ 2\mathbf{d}_i \cdot [C]_\theta \mathbf{R}_i \\ 2\mathbf{d}_i \cdot [C]_\phi \mathbf{R}_i \\ 2d_i^x \\ 2d_i^y \\ 2d_i^z \\ -2\mathbf{d}_i \cdot \mathbf{n}_i \end{bmatrix}
\end{aligned} \tag{21.29}$$

Unfortunately, the Hessian matrix can not be expressed explicitly. Therefore we use the quasi-Newton's method.

Bibliography

[1] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[2] C. S. Chua and R. Jarvis. 3D free-form surface registration and object recognition. *International Journal of Computer Vision*, 17(1):77–99, 1996.

[3] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America, Series A*, 4(4):629–642, 1987.

[4] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America, Series A*, 5(7):1127–1135, 1988.

[5] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. A. K. Peters, Wellesley, MA, 1993. Translated by L. L. Schumaker.

[6] K. H. Ko. *Algorithms for Three-Dimensional Free-Form Object Matching*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, March 2003.

[7] K. H. Ko, T. Maekawa, and N. M. Patrikalakis. Algorithms for optimal partial matching of free-form objects with scaling effects. Submitted for publication, December 2002, MIT, Department of Ocean Engineering, Cambridge, MA, 2002.

[8] K. H. Ko, T. Maekawa, and N. M. Patrikalakis. An algorithm for optimal free-form object matching. *Computer Aided Design*, 35(10):913–923, September 2003.

[9] T. Maekawa, F.-E. Wolter, and N. M. Patrikalakis. Umbilics and lines of curvature for shape interrogation. *Computer Aided Geometric Design*, 13(2):133–161, March 1996.

[10] N. M. Patrikalakis and T. Maekawa. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag, Heidelberg, February 2002.

[11] L. A. Piegl and W. Tiller. *The NURBS Book*. Springer, New York, 1995.

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1988.

[13] E. C. Sherbrooke and N. M. Patrikalakis. Computation of the solutions of nonlinear polynomial systems. *Computer Aided Geometric Design*, 10(5):379–405, October 1993.

[14] J. Zhou, E. C. Sherbrooke, and N. M. Patrikalakis. Computation of stationary points of distance functions. *Engineering with Computers*, 9(4):231–246, Winter 1993.