ELSEVIER

Image and Vision Computing 17 (1999) 113–123

image AND vision COMPUTING

Fast global registration of 3D sampled surfaces using a multi-z-buffer technique☆

Raouf Benjemaa*, Francis Schmitt

École Nationale Supérieure des Télécommunications, CNRS URA 820 — Images Department, 46 rue Barrault, 75634 Paris, Cedex 13, France

Received 21 July 1997; received in revised form 21 January 1998; accepted 4 February 1998

Abstract

We present a new method for the global registration of several overlapping three-dimensional (3D) surfaces sampled on an object. The method is based on the ICP algorithm and on a segmentation of the unstructured sampled points in an optimized set of z-buffers. This multi-z-buffer technique provides a 3D space partitioning which allows the registration process to detect quickly all the overlapping surfaces and to concentrate on them even when the surfaces overlap each other only slightly. It also greatly accelerates the search of the nearest neighbours in the establishment of the point-to-point correspondence between two overlapping surfaces. Then a randomized iterative registration is processed on the surface set. We have tested an implementation of this technique on real sampled surfaces. It appears to be rapid, accurate and robust, especially in the case of highly curved objects.

Keywords: Global registration; 3D data matching; ICP algorithm; 3D digital imaging; 3D surface partitioning

1. Introduction

A single scan of a complex three-dimensional (3D) object is, in general, not sufficient to fully describe its surface because of the presence of occluded parts. Additional scans from different viewpoints are thus required to recover the occluded parts and improve the surface description. To reach these different viewpoints, we often lose the exact position of a reference frame bound to the object because of imprecise or non-measured mechanical motion of the scanner or manual modification in the fixation of the object. The problem is then to integrate these multiple partially overlapping scans so that they all match together precisely.

The registration of overlapping sampled surfaces without feature extraction has been explored by many researchers [5,7,20,13,14]. We call *pairwise registration* the registration of only two surfaces at the same time. The widely used pairwise registration approach is based on the ICP (iterative closest point) algorithm [5]. It corresponds to the minimization of a mean square distance between two sets of matched points. It is an iterative optimization technique which proceeds in two steps: (1) the determination of a point-to-point correspondence between two overlapping surfaces; (2) the estimation of the best rigid transformation which puts the second surface in registration with the first one. This last optimization problem has been well resolved by Faugeras and Hebert [8] and Horn [10] by using an elegant quaternion technique. It is the determination of the point-to-point correspondence which requires the highest computation cost in the matching algorithms.

Most of the previous methods register only two surfaces at a time. We call *global registration* the simultaneous registration of more than two overlapping surfaces. Kamgar-Parsi et al. [11] have proposed a global registration approach using a dynamic system for the two-dimensional (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 an extension of this approach has been proposed by Stoddart and Hilton [19] for the 3D registration of multiple free-form surfaces. Their dynamic system is made of a set of springs of null length 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. For the global matching of several 3D surfaces, Bergevin et al. [4,18] have proposed another approach based on a

☆ Based on "Fast global registration of 3D sampled surfaces using a multi-z-buffer technique" by Raouf Benjemaa and Francis Schmitt which appeared in Proceedings of 3-D Digital Imaging and Modelling, Ottawa, May 1997; pp. 113–120. © 1999 IEEE.

* Corresponding author. E-mail: benjemaa@ima.enst.fr

0262-8856/99/$ - see front matter
PII: S0262-8856(98)00115-2

modified ICP algorithm. At each iteration, each surface 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 the rigid transformations. Such multi-surface matching techniques provide a better global registration than a set of successive pairwise registrations where a new surface is matched at each registration to the set of the previously registered surfaces and definitively fixed to this set after rigid transformation. They distribute the residual errors more homogeneously over all the surface overlaps.

In this paper we propose also a method based on the ICP algorithm for the global matching of several overlapping 3D digitized surfaces. Its main interest is in the first step of the algorithm where all the overlapping parts of the digitized surfaces are segmented in a set of optimized z-buffers by using the Gauss sphere. This multi-z-buffer technique is a space partitioning which is well adapted to 3D surfaces. It strongly accelerates the point-to-point correspondence between all pairs of overlapping surface parts. A first and shortened presentation of this technique has already been published in French [2]. In the second step of the ICP algorithm we use, for the global registration, an iterative optimization process which is a modified version of the one proposed by Bergevin et al. [4] where we move each surface part immediately after the estimation of its rigid transformation, the registration order being randomly chosen so as not to favour any surface part. As for all the methods based on the ICP algorithm, the initial positions of the surfaces need to be close enough for the registration to work properly. If not, a quick and rough interactive registration has to be performed first [1].

The authors have previously proposed a z-buffer technique which already greatly accelerates the determination of the point-to-point correspondence between two surfaces [1]. However, the problem of choosing the best direction of projection for the z-buffer is not obvious when the shape of the overlapping part of the two surfaces is strongly bent. In such a case the use of a single z-buffer already appears inappropriate. This problem becomes clearly critical when we have to register a greater number of partially overlapping surfaces. The multi-z-buffer technique presented in this paper provides an efficient solution to this problem. One of its principal properties is to quickly detect all the overlaps and to allow the registration process to concentrate on them even when the surfaces overlap each other only to a very small extent.

However, for the z-buffer or multi-z-buffer technique to work properly, we have to assume that the surfaces have been sampled with a sufficiently high and homogeneous density. This condition is not too strong a limitation as nowadays the available range finders provide high density data. This approach requires also an estimation of the normal direction at each sampled point which can be obtained in a preprocessing step on the initial data.

The remainder of this paper is organized as follows. Section 2 recalls how a single z-buffer can greatly accelerate an ICP algorithm. Then the registration of two surfaces based on their segmentation into an optimized set of z-buffers is presented in Section 3. We discuss also the estimation of the normal at each sampled point. In Section 4 we describe a generalization of this technique to solve the problem of the global registration of a large set of partially overlapping sampled surfaces. Finally, experimental results are shown in Section 5.

2. ICP algorithm acceleration by using a z-buffer

The ICP algorithm [5] estimates a rigid transformation for matching two sets S_1 and S_2 of 3D unstructured points. The first step of the algorithm consists of establishing the correspondence between the points of S_1 and S_2 by finding for each point of S_1 the closest point of S_2 to form a pair. Variants to this approach estimate a point to surface distance by using as additional information the normals at the points. The second step is the estimation of the rigid transformation that minimizes the mean square distance between the pairs of corresponding points. We use here the quaternion-based solution proposed by Faugeras [9] and Horn [10] for this purpose.

The two steps of the ICP algorithm are iterated until convergence of the registration process. With a brute force search, the complexity of the first step would be quadratic in the number of points. It is thus essential to reduce this complexity, especially when we have to process several millions of points as is our case.

The remainder of this section is organized as follows. We first show in Section 2.1 how the use of a z-buffer accelerates the establishment of the correspondence between two overlapping surfaces. Then in Section 2.2 we compare the search complexity of the z-buffer approach with the traditional k-d tree approach and we discuss in Section 2.3 the limitation of the mono z-buffer structure.

2.1. Rapid matching by using a z-buffer

In order to accelerate the first step of the ICP algorithm and by assuming that the density of the two sets of 3D points is homogeneous enough, we have proposed a method based on a twin z-buffer structure which provides an explicit space partitioning [1]. It is similar to two depth images, one per set of points, sharing a common referential and allowing us to apply 2D image processing techniques on the 3D data. The correspondence between the two sets of 3D points can then be efficiently obtained by using the 2D connectivity of the twin z-buffer cells.

To build the twin z-buffer structure, we first chose a direction of projection and two orthogonal directions defining the z-buffer referential, and the size of the square cells tessellating the z-buffer plane. We then compute the

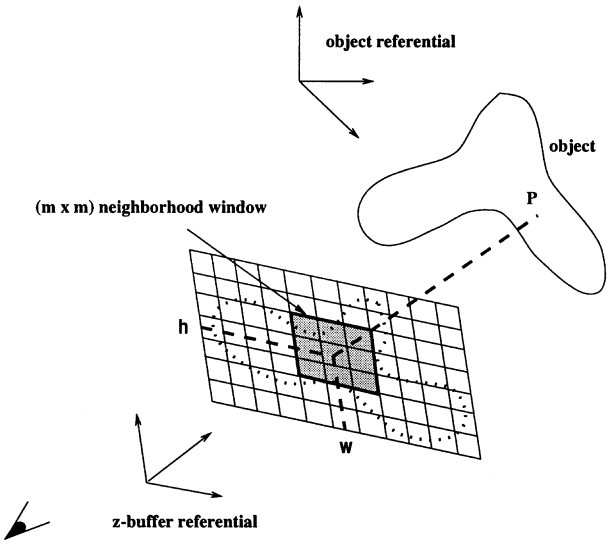

Fig. 1. Neighbourhood access in z-buffer structure.

minimal box aligned with these directions and which includes $S_1 \cup S_2$. The direction of projection and the width and height of the minimal enclosing box define the orientation and size of the twin z-buffers. For each set we orthogonally project its points in the cells of the associated z-buffer. We keep only one point per cell, the nearest one along the direction of projection, and we store all its geometrical information (position, normal). This selection allows us, by adjusting the size of the cells, to save memory space when data are oversampled and to further reduce CPU time for solving the surface correspondence problem.

Then, to match the point P_2 of the scanned surface stored in $\text{cell}_2(w,h)$ of the second z-buffer, we look for its closest point P_1 of the first surface. This is efficiently done by testing only the points of S_1 which have been stored in the first z-buffer and belonging to an $(m \times m)$ window centred on $\text{cell}_1(w,h)$ as shown in Fig. 1. This way the first step of the ICP algorithm is greatly accelerated. To reinforce the robustness of the point-to-point correspondence process, we have added two matching criteria: a minimal density of points to be fulfilled inside the $(m \times m)$ window and a minimal distance to be satisfied by $P_1 P_2$ [1].

The set of corresponding points restricted to such a $(m \times m)$ neighbourhood allows us to determine a first estimate of the rigid transformation. The choice of the size m of the window is not critical for a surface which has been digitized on a real object with a sufficiently continuous surface, even if the initial misregistration is greater than the width of the window. The iterative process of the ICP algorithm will progressively correct the residual misregistration between the two surfaces and improve the correspondence establishment by allowing at each step the $(m \times m)$ windows to progressively slip along S_1 until a minimum of the mean-square distance between the corresponding points has been reached. However, a global minimum can not be guaranteed as usual in the case of the ICP algorithm. In practice we first choose for m a large value in order to estimate the initial misregistration error. An adaptative hierarchical strategy can then be used by decreasing the value of m with the residual error during the iteration process. Typically m varies in our experiments from 11 to 5.

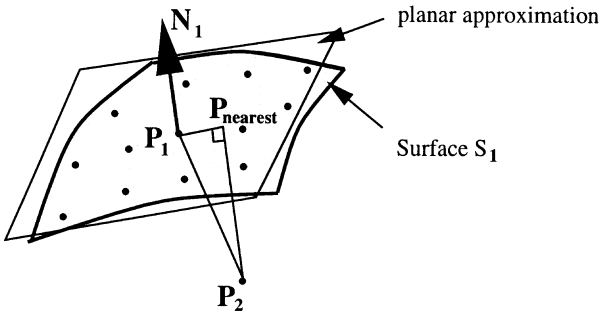

Fig. 2. Refinement of the point-to-point correspondence.

This simple point-to-point correspondence as used in our previous works [1–3] can be further improved by limiting aliasing effects caused by the surface point sampling. For that we consider the tangent plane of surface S_1 at P_1. We then associate to P_2 not P_1, but P_{nearest}, the point corresponding to the projection of P_2 onto the tangent plane to S_1 at P_1 (Fig. 2). The estimation of the normal method we used is explained in Section 3.1.1. This refinement of the point-to-point correspondence improves the registration accuracy and allows us to further increase the size of the z-buffer cells to reduce the set of corresponding points and then to save CPU time.

2.2. Complexity gain by the z-buffer approach

The brute force search for finding the shortest distances from M_2 points to a given set of M_1 points is of complexity $O(M_1 \times M_2)$. Several methods exist to speed up the rocess such as: (a) hashing and indexing; (b) static space partitioning; (c) dynamic space partitioning; and (d) randomized algorithm. Many works such as [17,21] use a k-d tree structure for accelerating registration of unstructured 3D points sampled on a surface. This spatial data structure belongs to category (b) and a detailed description can be found in [16]. A k-d tree can be constructed in $O(M_1 \log M_1)$ time, so the cost of finding the shortest distances from M_2 points to M_1 points is of complexity $O(M_2 \log M_1) + O(M_1 \log M_1)$ which is lower than $O(M_2 \times M_1)$ except for very small M_2.

The construction of the z-buffer structure is done in two linear passes on the data points. The spatial sorting time is then simply $O(M_1) + O(M_2)$. By using z-buffers we compute for the M_2 points only approximate shortest distances limited to a $m \times m$ neighbourhood window. Its complexity is $O[(m \times m)M_2]$. Thus, for each iteration the complexity remains in $O(M_1) + O(M_2)$. This shortest distance approximation is statistically correct when the projection of the local misregistration vector remains inside the $(m \times m)$ window. However, the iterative process of the ICP

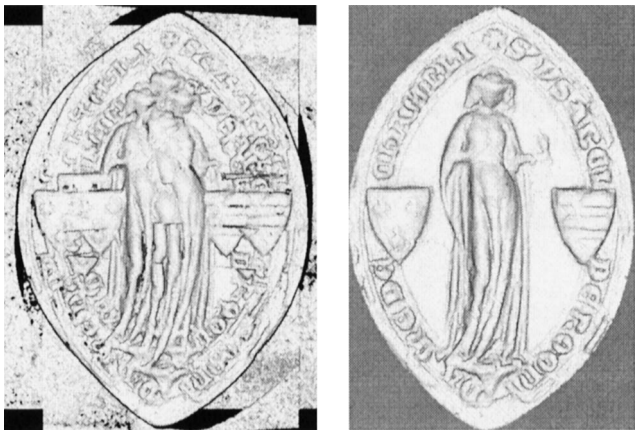

Fig. 3. Two renderings of an ancient seal before (left) and after (right) the mono-z-buffer pairwise registration.

algorithm allows a progressive correction of larger misregistration errors as described earlier. Furthermore, the effects of the approximation do not accumulate as the iteration proceeds because the second z-buffer is updated at each iteration after rigid motion of the corresponding sampled surface. So we don't lose accuracy and the ICP convergence is not really affected by the approximation. Then in the case of large misregistration errors we just trade off the speed up of each iteration of the ICP algorithm by a small increase in the number of iterations: the bigger the $m \times m$ window, the smaller the increase of this number.

2.3. Limitation of a mono z-buffer

The mono z-buffer approach works well especially when the overlap between the two surfaces is sufficiently planar. This is illustrated in the example of Fig. 3 where we can compare the rendering of a set of four digitized surfaces scanned on an ancient seal before (left) and after (right) the mono-z-buffer pairwise registration.

In the case of strongly curved overlaps, the experience shows that the matching algorithm does not converge properly because the digitized surfaces are not uniformly resampled in the twin z-buffer structure. To improved this approach we propose in the following section a multi-z-buffer technique which produces a more homogeneous resampling as illustrated in Fig. 4.

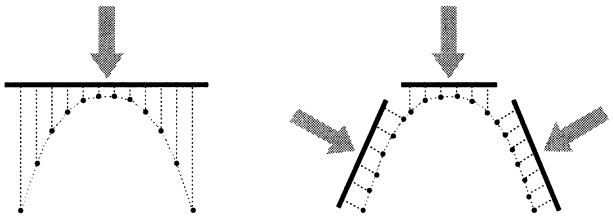

Fig. 4. The resampling of a curved surface is better in several adapted z-buffers (right) than in a single one (left).

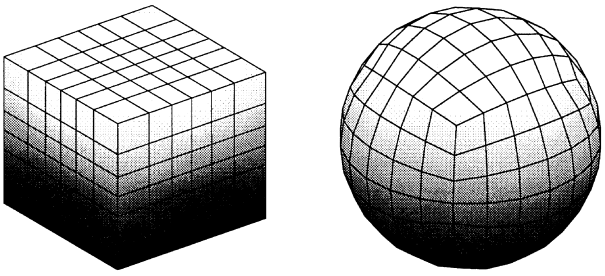

Fig. 5. Tessellation of the Gauss sphere (right) by the projection of a unit cube (left) subdivided into 216 patches.

3. Pairwise registration using a multi-z-buffer technique

In the above technique with a single twin z-buffer structure, the direction of the projection is chosen interactively in practice. In a multi-z-buffer approach the determination of these directions must be automated. The knowledge of the oriented normal of the object surface at each sampled point is then needed. The registration is still processed in two stages. First, the multi-twin z-buffers are optimized in order to efficiently define only the overlaps of the surfaces; this is obtained by clustering the normal directions through a Gauss sphere tessellation. Then the registration process is iterated while the correspondence problem is rapidly solved inside each individual twin z-buffer structure.

3.1. Surface segmentation by using the gauss sphere

The goal of this stage is to segment the overlaps of the sampled surfaces into flat enough regions, each of them having normal directions remaining approximately constant. We then associate with each region, a z-buffer perpendicular to its mean normal direction.

3.1.1. Normal estimation

In the case of range image data, an estimation of the normal direction at each sampled point can be easily

Procedure *Gauss-Sphere(S)*
Return $GS = \bigcup_p s_p$
/ Where* s^p *= { points* $\in S$ *associated to patch p}*/*
*/*and* $p \in 1$ *.. Gauss-sphere-patch-number*/*

Procedure *Multi-Zbuffer*(GS_1, GS_2)
FOR $p \in 1$ .. Gauss-sphere-patch-number
IF $(s_1^p \neq \emptyset$ **AND** $s_2^p \neq \emptyset)$ **{**
$\vec{D}^p = Average\text{-}Orientation(s_1^p \bigcup s_2^p)$;
$eb_1^p = Min\text{-}Enclosing\text{-}Box(s_1^p, \vec{D}^p)$;
$eb_2^p = Min\text{-}Enclosing\text{-}Box(s_2^p, \vec{D}^p)$;
overlap $= eb_1^p \cap eb_2^p$;
IF $(overlap \neq \emptyset)$ **{**
$zbf_{12}^p = Create\text{-}Zbuffer(overlap, \vec{D}^p, s_1^p)$;
$zbf_{21}^p = Create\text{-}Zbuffer(overlap, \vec{D}^p, s_2^p)$; **} }**

Fig. 6. Creation of multiple twin z-buffers on the overlaps of surfaces S_1 and S_2.

obtained. But when using a laser plane range finder which is swept along the surface of the object with a combination of translation and rotation, the data can not always be structured as a range image. However, the sequential acquisition of planar curves after small motions allows us to quickly define a neighbourhood around each data point and to estimate its normal vector onto this neighbourhood. The geometric discontinuities of the surface are first detected by adapting a previous algorithm developed for range images [6]. Then the normal direction estimation is done by avoiding crossing over these discontinuities. When the data are scattered but dense enough, their surface neighbourhood can be retrieved by 3D volumetric partitioning. The estimation of the normal direction can then be obtained by robust statistic methods with an underlying local surface model.

3.1.2. Tessellation of the gauss sphere

The Gauss sphere is a unit sphere on which are mapped all the oriented normals of a 3D surface. It offers a good way to segment the surface into flat regions according to its normal orientations. This segmentation can be obtained by using a tessellation of the Gauss sphere into small homogeneous patches and by doing a classification of these patches regardless of the object complexity.

A method to tessellate homogeneously and symmetrically the Gauss sphere is to project onto it a unit cube subdivided in rectangular patches following a $\tan(\theta)$ law along its edges, $\theta \in [-(\pi/4), (\pi/4)]$ [12]. The projected patches have nearly the same area and provide a regular tessellation on the Gauss sphere. The main advantage of this tessellation is to allow a quick localization of a given normal. The unit cube face pointed by the normal is determined by simple logical tests on the three normal coordinates. Then the localization of the Gauss sphere patch containing the normal is obtained by using a look up table on quantified values of the two normal coordinates associated to this face.

Fig. 5 shows an example of the Gauss sphere tessellation into 216 patches, the cube edges being subdivided into six pieces.

3.1.3. Optimization of the twin z-buffer set covering the surface overlaps

The determination of the multiple twin z-buffers covering the overlaps of two surfaces S_1 and S_2 is based on the segmentation of each sampled surface in a Gauss sphere. We assume first that S_1 and S_2 are roughly in registration. If not, a quick interactive registration has to be performed first [1].

Let GS_1 and GS_2 denote the Gauss sphere associated to S_1 and S_2, respectively. If a patch p of GS_1 and its twin patch p in GS_2 are both non-empty, they contain normals belonging to two parts s_1^p and s_2^p of S_1 and S_2, respectively. s_1^p and s_2^p may overlap. An overlapping test is processed after projection of their 3D points in a plane perpendicular to the normal defined at the centre of patch p. If the minimal enclosing boxes of the projections of s_1^p and s_2^p overlap each other, a twin z-buffer $\{zbf_{12}^p, zbf_{21}^p\}$ is affected to patch p. Its dimensions are deduced from the intersection of the two minimal boxes while its orientation $\mathbf{D}^p$ is the average of the normals of s_1^p and s_2^p. A pseudo code of the algorithm is presented in Fig. 6. This multi-z-buffer approach allows an efficient selection of subsets of S_1 and S_2 which are likely to overlap. It reduces strongly the numbef of points to be-considered for the correspondence problem and it subdivides it into sub-problems, one per twin z-buffer.

The success of the determination of the twin z-buffer set is conditioned by the amount of the rotation error in the initial mismatch. If this error is greater than the size of a Gauss sphere patch, there might be no non-empty matching pairs of patches in GS_1 and GS_2, thus impeding the construction of any z-buffer. Such a situation may happen if we try to register two sufficiently rotated sampled surfaces of a simple polyhedron with sparse Gauss spheres (one normal per face). However, this limitation is not so crippling in practice because the objects we consider generally are not strictly polyhedral and because we limit in practice the subdivision of the Gauss sphere (up to a maximum factor of 6 as in Fig. 5. If the rotation error is high, a mono-z-buffer registration can first be performed to decrease this error value.

```
T^0 = T_init; k = 0;    /*initialization*/
GS_1=Gauss-sphere(S_1); GS_2=Gauss-sphere(T^0(S_2));
Multi-Zbuffer(GS_1, GS_2);
REPEAT UNTIL convergence {
    k = k + 1;
    L = EMPTY;          /*list initialization*/
    IF(zbf_21^p exists) /*and thus zbf_12^p */ {
        z-buffer-fast-matching(zbf_21^p, zbf_12^p);
        update-matching-list(L); }
    T = Optimal-rigid-motion(L);
    T^k = T × T^(k-1);
    Update-all-non-Empty-Zbuffers zbf_21^p with T^k(s_2^p); }
```

Fig. 7. Iterative registration of S_2 to S_1.

3.2. Iterative registration

After the determination of the optimal set of twin z-buffers which include the surface overlaps, the sampled points are projected into their respective z-buffers and eventually clipped. The registration is then processed as follows: for each twin z-buffer structure a point-to-point correspondence is quickly established as detailed in Section 2. The couples of matched points are stored in a list L common to the full set of twin z-buffers. Then the rigid transformation from surface S_2 to surface S_1 is optimized by minimizing the mean square distance between the matched points of list L. The quaternion technique [10] is used for this purpose.

The points of the surface S_2 are then transformed according to this rigid motion and all the z-buffers associated to S_2 are updated. The matching operation is then iterated until convergence. A pseudo code of the algorithm is presented in Fig. 7.

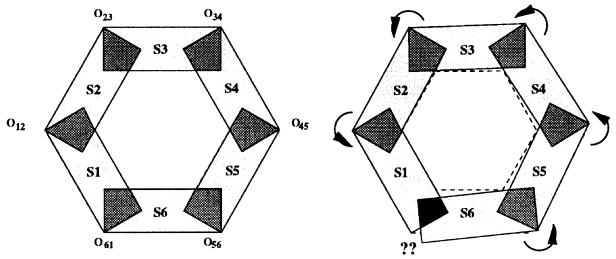

Fig. 8. Error propagation in closed string of surface.

The knowledge of the normal at each point being assumed, the complexity of the multi-z-buffer approach to compute the matching list L remains $O(M_1) + O(M_2)$: the construction of the Gauss spheres needs one linear pass on the data and the construction of the twin-z-buffers necessitates another pass; for each twin-z-buffer, the computation of the approximate shortest distance is identical as in the mono-z-buffer case. As the determination of the optimal rigid motion by using the quaternion technique is linear with the length of list L, the global complexity of an iteration of the ICP algorithm remains linear in the number of 3D data points.

4. Extension to a global registration of N surfaces

When we have N surfaces to register simultaneously, $N > 2$, the problem becomes more difficult. We first show the need and the usefulness of a global registration in Section 4.1. Then we propose in Section 4.2 an extension of the multi-z-buffer technique for a global registration of N surfaces. This global registration method is a modified version of the one proposed by Bergevin et al. [4].

4.1. Interest of a global registration

The method described in Section 3 allows the registration of digitized surfaces with complex shapes, but only two at a time. In practice when we have to digitize an object with a complex shape, we need to scan it from many points of view and the problem then is to register all together the resulting sampled surfaces. We could sequentially apply the above registration by matching two by two the different surfaces. However the pairwise registration of several tens of scans is not convenient. Pairing off the overlapped surfaces is an easy but very tedious task. Furthermore, a strategy should be defined by the operator to choose the surface registration order. The accuracy of the final registration depends on this choice and the residual errors after a sequence of pairwise registrations can be heterogeneously distributed. Despite the accuracy of the individual pairwise registration, the surface of the whole object may remain badly registered in some places. A typical illustration of this situation is the registration of a closed string of digitized surfaces as shown in Fig. 8. The schematic object illustrated in Fig. 8 (left) is composed of six surfaces (S_1,$\cdots$,S_6) shaded in light gray. The dark gray area represents the surface overlaps (O_{12}, O_{23},$\cdots$, O_{61}). In Fig. 8 (right) the arrows indicate that the successive pairs have been sequentially registered. The residual error of each individual pairwise registration can be low, but unfortunately we frequently observe a propagation and a cumulation of the registration error in such a way that when overlap O_{56} has been successfully registered, the closing overlap O_{61} may present large discrepancies between surfaces S_6 and S_1, sometimes much larger than at the initialization step.

Thus it would be much more efficient to register the digitized surfaces simultaneously in order to keep the residual errors of the registration homogeneously distributed.

```
FOR i ∈ 0..N {
    set T_i^0;          /*initial transformations; T_0^0 = I */
    GS_i = Gauss-Sphere(T_i^0(S_i)); }
FOR i ∈ 0..(N − 1)
FOR j ∈ i..N
    Multi-Zbuffer(GS_i, GS_j);
k = 0;          /*iteration number */
REPEAT UNTIL convergence {
    k = k + 1;
    FOR i ∈ 1..N, in random order {
        L = EMPTY;          /*list initialization*/
        FOR j ∈ 0..N, j ≠ i
        IF(zbf_ij^p exists) /*and thus zbf_ji^p */ {
            z-buffer-fast-matching(zbf_ij^p, zbf_ji^p);
            update-matching-list(L); }
    T = Optimal-rigid-motion(L);
    T_i^k = T × T_i^{k−1};
    Update-all-non-Empty-Zbuffers zbf_ij^p with T_i^k(s_i^p); } }
```

Fig. 9. Global registration algorithm.

4.2. Global registration algorithm

Inspired by the work of Bergevin et al., we have extended the previous pairwise registration based on the multi-z-buffer technique to a global registration.

Given a set of N surfaces scanned on a 3D object, we chose without loss of generality one of them as a *master* (S_0) and the others as slaves (S_1, S_2,$\cdots$, S_{N-1}). The reference frame of the *master surface* S_0 is then defined as the world reference frame. Thus the registration process must determine $(N - 1)$ rigid transformations corresponding to the motion of each slave into the master reference frame.

As in the pairwise registration, the first step in the global registration is the determination and the optimization of the twin z-buffer set covering all the mutual overlaps by using the Gauss sphere technique. We proceed as follow. First each sampled surface S_i, $i \in \{0,\cdots,N\}$, is segmented into a Gauss sphere GS_i as explained in Section 3.1.2 and the minimal enclosing box in each patch p of GS_i is computed. Then for each patch p we compare two by two the minimal enclosing boxes which are not empty. Let us assume that it is the case for patch p in GS_i and GS_j. Then the two parts s_i^p

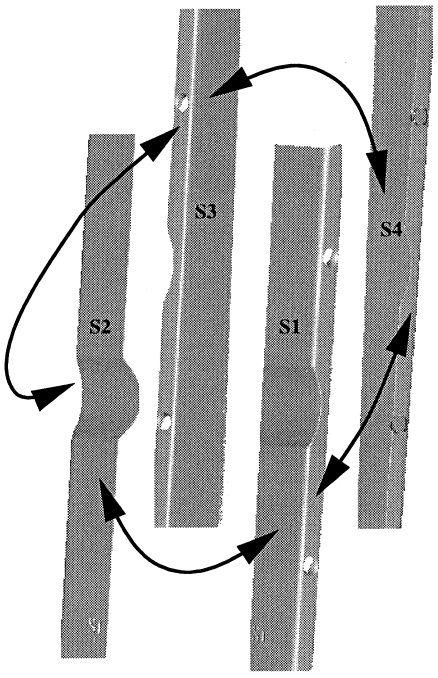

Fig. 10. Four different scans of a vibraphone bar and their overlapping relationship.

and s_j^p of S_i and S_j may overlap in patch p. We then define a twin z-buffer $\{zbf_{ij}^p,\ zbf_{ji}^p\}$ whose dimensions and direction are determined as in Section 3.1.3. At the end of this process only parts of the surfaces overlapping two by two are conserved. All the sampled points of the isolated surfaces are removed.

The second step is the iteration of the registration process itself. At each iteration k a rigid transformation T_i^k is optimized for each slave surface S_i, $i \in 1\ldots N$. The matching point list L of S_i is updated by quickly matching, in the appropriate twin z-buffers, S_i with its overlapping surfaces. The rigid transformation T_i^k is obtained by applying the quaternion technique [10] to list L. The position of S_i is then corrected and all the z-buffers associated to S_i updated.

Our global registration algorithm differs from that of Bergevin et al. [4] in the way the rigid transformations are applied. In Bergevin et al.'s algorithm, at each iteration k each slave surface is matched with all the others first, then the rigid transformations are simultaneously applied to the slave surfaces only at the end of the iteration. In our method each slave surface is immediately transformed when its rigid transformation has been estimated. In this way, the convergence is accelerated. In order not to favour any slave surface, its registration order is randomly chosen at each iteration. The pseudo-code of this algorithm is shown in Fig. 9.

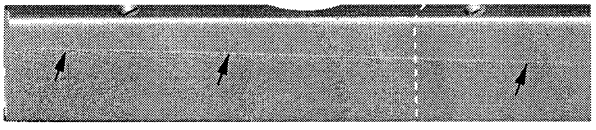

Fig. 11. Result of the pairwise registration.

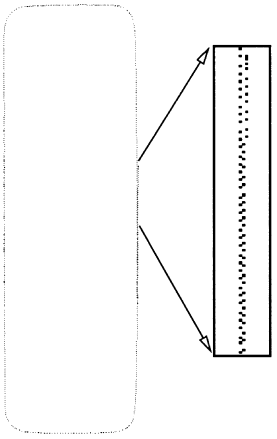

Fig. 12. Zoomed slice of the vibraphone bar after pairwise registration.

Let us estimate the complexity of a global registration iteration (REPEAT loop in Fig. 9) by considering N surfaces with M points in each. The total amount of data is $\mathcal{M} = MN$. The Gauss spheres construction requires $N\mathrm{O}(M) = \mathrm{O}(\mathcal{M})$. A multi-z-buffer (GS_i, GS_j) requires $\mathrm{O}(M) + \mathrm{O}(M)$, then the complete set of $N(N-1)/2$ multi-z-buffers requires $N(N-1)\mathrm{O}(M)$, i.e. $N\mathrm{O}(M)$.

For each surface S_i (see loop FOR i of the global iteration in Fig. 9), matching the z-buffers requires $(N-1)(m \times m)O(M)$ and the length of the matching list L is $(N-1)\mathrm{O}(M)$. With the computation of the optimal rigid motion of a surface $S_i[(N-1)\mathrm{O}(M)]$ and the update of the corresponding z-buffers $zbf_{ij}^p[(N-1)\mathrm{O}(M)]$, the complexity of a global registration iteration is then $N(N-1)[(m \times m)\mathrm{O}(M) + \mathrm{O}(M) + \mathrm{O}(M) + \mathrm{O}(M)]$ that is $N\mathrm{O}(\mathcal{M})$.

5. Experimental results

The above registration technique has been applied to data obtained from different objects scanned with a Kréon Industrie laser range finder. The different scans were coarsely matched using an interactive tool developed for this purpose.

(a) A vibraphone bar is used to illustrate the problem of the registration of a closed string of scanned surfaces. Fig. 10 shows four different scans (S_1, S_2, S_3, S_4) of the vibraphone bar. The arrows indicate the overlapping relationship

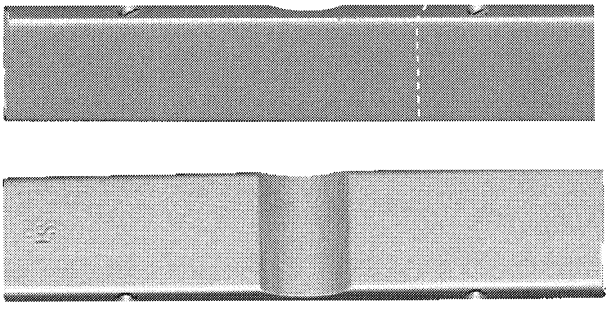

Fig. 13. Two different views after global registration.

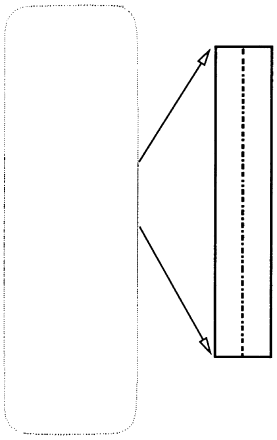

Fig. 14. Zoomed slice of the vibraphone bar after global registration.

between scans. We have applied to this set of scans both the pairwise registration and the global registration.

The pairwise registration is processed to match S_2 to S_1, S_3 to S_2 and S_4 to S_1. The result of the integration of S_4 and S_3 which are not directly matched together is rendered in Fig. 11. An artefact ig clearly perceptible in this rendering. This mismatching is more visible when we extract the slice corresponding to the white dashed line. This slice is shown in Fig. 12. The shift present in the zoomed part is about 0.2 mm.

Such an artefact is eliminated by applying the global registration technique as shown in Fig. 13. A slice at the same localization is shown in Fig. 14. We notice that all scans are well matched.

The evolution of the residual errors of the global registration is shown in Fig. 15. After 10 iterations the residual errors are already well distributed and the convergence of the process is nearly reached.

The time needed to optimize the rigid transformations depends on the number of scans, the number of surface points and the distribution of the overlaps. For the vibraphone bar, the four scans contain about 620 000 points each. Each iteration took about 160 s on a Sun Sparc Station running at 40 MHz. The root mean square error for the four scans is 0.049 mm after 25 iterations.

(b) The Greek bust shown in Fig. 17 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 Fig. 16. A quick and rough interactive registration was first performed (Fig. 17-left). Then all the scans were registered simultaneously with the global technique (Fig. 17-right). In order to accelerate the registration process we have previously resampled the 3D data at a resolution of 0.2 mm by using a technique of 3D partitioning and applying a 3D low pass filter. With this reduced number of points (750 000) the CPU time needed for each iteration is only 100 s. The registration process staxted with an initial RMS error of

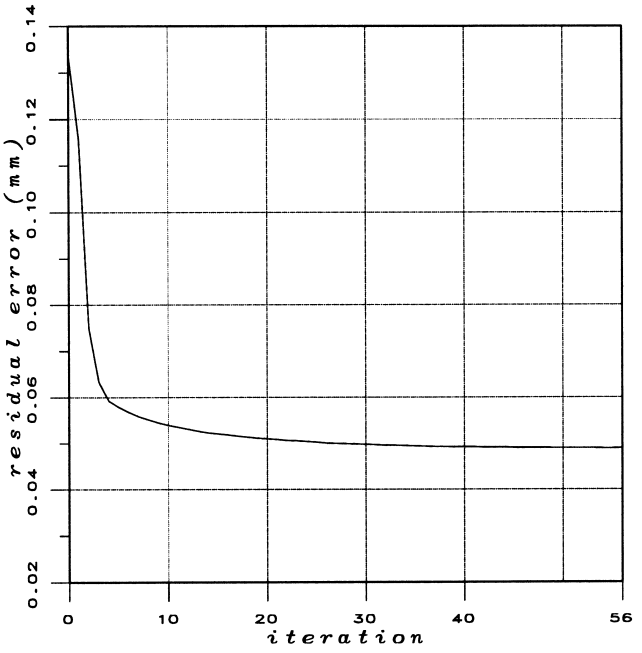

Fig. 15. Global registration convergence of the four scans of the vibraphone bar.

0.66 mm. After 28 iterations the RMS error became lower than 0.12 mm.

(c) The statuette in Fig. 18 was digitized along six different views with the automated system developed by Orfanos-Papadopoulos et al. [15]. This system avoids the collision problems by taking into account the shadows and the occultation events during the scanning process. Each view corresponds to a sweeping process made of a large set of translation motions. Let us call a scan a set of digitized points obtained during a single translation motion. About 150 scans were then required for the six views with 55 scans containing exploitable data. In one view the scans are

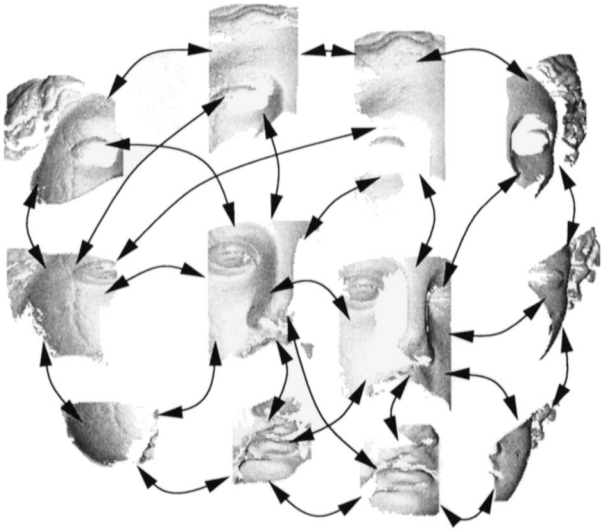

Fig. 16. Some pieces of a Greek bust mosaic (Hygia, Dion Museum, Greece).

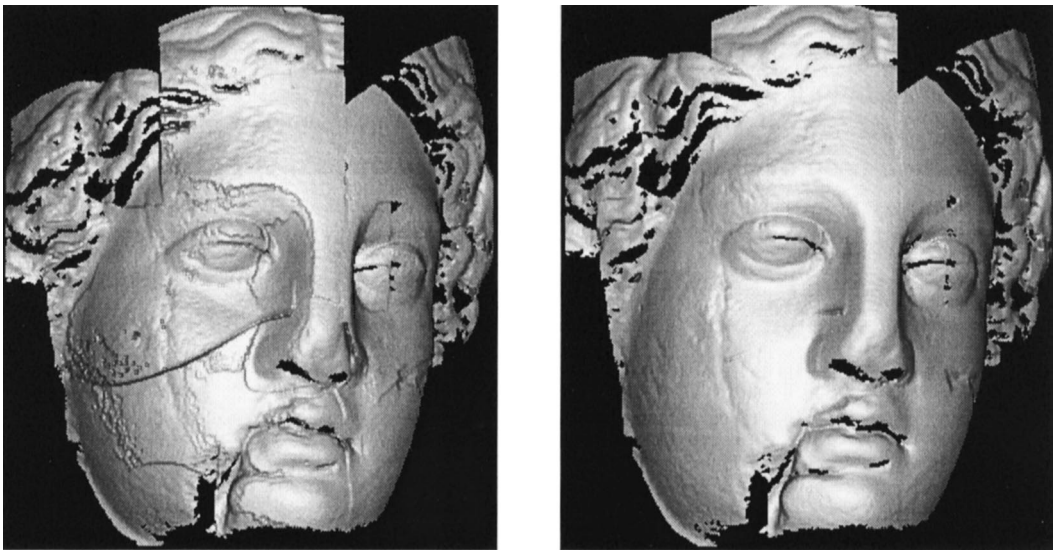

Fig. 17. The Greek bust after interactive registration (left) and automatic global registration (right) of 12 scans.

slightly misregistered because of small shiftings of the mechanical support of the statuette arising during the fast speed displacement between two translation sweeps. In order to reconstruct the surface of the statuette, we first globally register the scans of each view (intra-registration). This does not require an initial interactive registration, the inaccuracies in the positioning of each individual scan remaining small enough. Then the six views are registered together using the global method described in this paper (inter-registration).

Fig. 19 shows a rendering of the six views after their global intra-registration. The results of the six intra-registrations are summarized in Table 1. In order to accelerate the process, we have decreased the number of the data points by first resampling each scan at a resolution of 0.2 mm.

After the intra-registrations the scans are merged together in each view and again resampled at the resolution of 0.2 mm. to remove the oversampling inside the area corresponding to the surface overlaps. Then the global registration method is applied to the set of the six views. The final model contains 3 079 000 points but due to the resampling, only 808 000 points have been used for the global registration process. Fig. 20 shows three renderings from different viewpoints of the registered statuette data. The detailed results of the global inter-registration on the six views are shown in Table 2. The rotation angle and the translation length correspond to the optimized rigid motion applied to each view.

6. Conclusion

We have proposed a global registration method which can be used for a large set of digitized surfaces scanned on an object. It is based on a multi-z-buffer technique which concentrates on the mutual overlaps of the digitized surface. It allows strong reduction of the computation for the point-to-point correspondence problem. Experimental results demonstrate that the proposed registration method can match different parts of a wide variety of complex 3D objects in a reasonable time. The registration technique presented in this paper may find applications in computer graphics, electronic storage of 3D objects, rapid prototyping or reverse engineering.

Acknowledgements

The authors warmly thank the anonymous reviewers for their very helpful and constructive comments.This work has been partly supported by the European ESPRIT project AR-CHATOUR (EP III 9213).

Fig. 18. Front view of the statuette on its stand before digitization (courtesy of the artist, Roland Coignard).

Table 1
Results for each view of the global registration algorithm applied to the individual scans (intra-registration).

Views	View1	View2	View3	View4	View5	View6
Number of scans	10	15	11	8	7	4
Number of points (in thousands)	391	767	914	497	245	265
Number of matched points (in thousands)	71	208	198	84	40	70
RMS initial error (mm)	0.2	0.11	0.11	0.06	0.06	0.08
RMS optimized error (mm)	0.07	0.06	0.06	0.04	0.03	0.03
Number of iterations	16	19	6	7	7	7
CPU time of an iteration (s)	18	61	77	28	9	8

Table 2
Results of the global registration algorithm applied to the full set of six views (inter-registration).

Views	View1	View2	View3	View4	View5	View6
Number of resampled data points (in thousands)	118	167	224	142	80	77
Number of matched points (in thousands)	35	73	82	53	29	25
RMS initial error (mm)	0.58	0.50	0.55	0.69	0.71	0.75
RMS convergence error (mm)	0.27	0.14	0.22	0.30	0.28	0.46
Rotation angle (degree)	—	0.37	0.70	0.71	1.91	2.13
Translation length (mm)	—	1.04	0.53	0.84	1.06	1.38

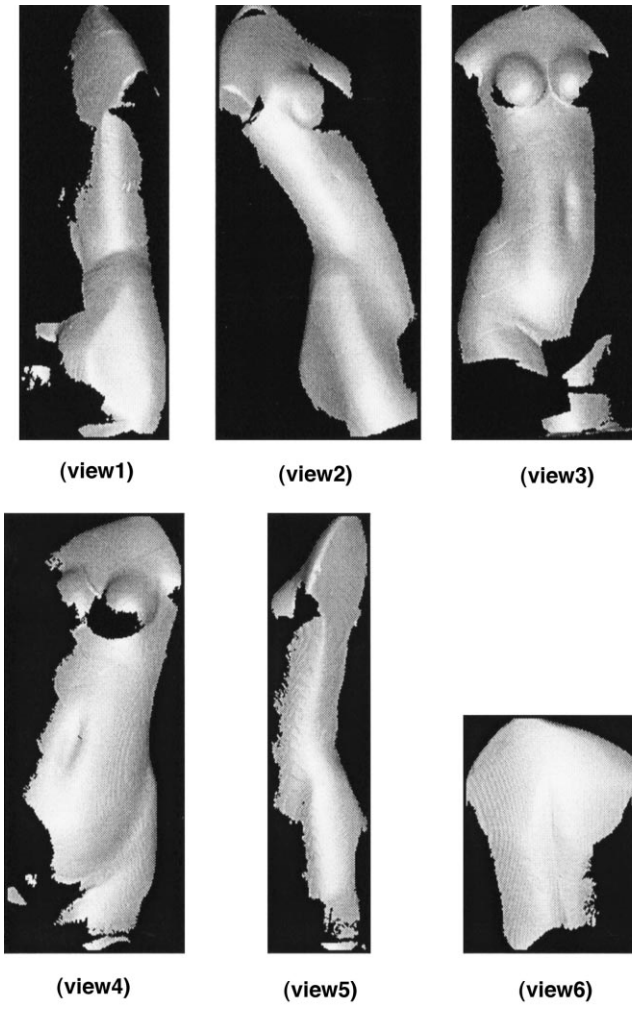

Fig. 19. Rendering of the six views of the statuette after the global registration of their individual scans (intra-registration).

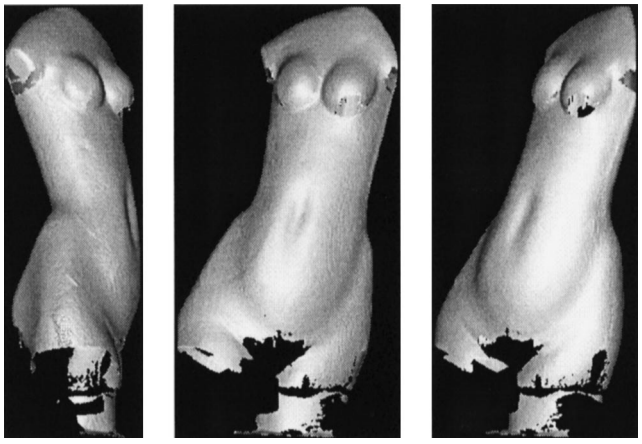

Fig. 20. Three different Renderings of the fusion of the six views of the statuette after their global registration (inter-registration).

References

[1] R. Benjemaa, F. Schmitt, Registering range views of complex objects. 4th European Conferences on Rapid Prototyping — Paris, 12 p., Oct. 1995.
[2] R. Benjemaa, F. Schmitt, Recalage rapide de surfaces 3D après projection dans des multi-z-buffers. 5th European Conferences on Rapid Prototyping — Paris, 11 p., Oct. 1996.
[3] R. Benjemaa, 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, May 1997, pp. 113–120, IEEE Computer Society Press, Los Alamitos, CA.
[4] R. Bergevin, M. Soucy, H. Gagnon, D. Laurendeau, Towards a general multi-view registration technique, IEEE Trans. PAMI 18 (5) (1996) 540–547.
[5] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, IEEE Trans. PAMI 14 (2) (1992) 239–256.
[6] X. Chen, F. Schmitt, Surface modeling of range data by constrained triangulation, Comput. Aided Des. 26 (8) (1994) 632–645.

[7] Y. Chen, G. Medioni, Object modeling by registration of multiple range views. Proceedings of the IEEE International Conference on Robotics and Automation, Apr. 1991, pp. 2724–2729.
[8] O.D. Faugeras, M. Hebert, A 3D recognition and positioning algorithm using geometrical matching between primitive surfaces. International Joint Conference on Artificial Intelligence, Karlsruhe, Germany, 1983, pp. 996–1002.
[9] O.D. Faugeras, M. Hebert, The representation and recognition and locating of 3-D objects, Int. J. Robotic Res. 5 (3) (1986) 27–52.
[10] B.K.P. Horn, Closed-form solution of absolute orientation using unit quaternions, J. Opt. Soc. Am. A 4 (4) (1987) 629–642.
[11] B. Kamgar-Parsi, J.L. Jones, A. Rosenfeld, Registration of multiple overlapping range images: Scenes without distinctive features, IEEE Trans. PAMI 13 (g) (1991) 857–871.
[12] J. Maillot, Trois Approches du Plaquage de Texture sur un Objet Tridimensionnel. Ph.D. thesis, Universitat de Paris VI, Oct. 1991.
[13] T. Masuda, A robust method for registration and segmentation of multiple range images, Comput. Vis. Image Understanding 61 (3) (1995) 295–307.
[14] V. Moron, P. Boulanger, H. Redarce, A. Jutard, Mise en correspondance du modèle CAO d'un objet avec son image 3D: Application à l'inspection, RFIA '96 Proc. Rennes 2 (1996) 913–922.
[15] D. Papadopoulos-Orfanos, F. Schmitt, Automatic 3D digitization using a laser rangefinder with a small field of view. Proceedings of the IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling, Ottawa, Canada, May 1997, pp. 60–67, IEEE Computer Society Press, Los Alamitos, CA.
[16] F. Preparata, M. Shamos, Computational Geometry, An Introduction. Springer-Verlag, New York, 1986.
[17] D. Simon, M. Hebert, T. Kanade, Real-time 3D pose estimation using a high-speed range sensor. International Conference on Robotics and Automation, 1994, pp. 2235, 2241.
[18] M. Soucy, G. Godin, R. Baribeau, F. Blais, R. Rioux, Sensors and algorithms for the construction of digital 3D colour of real objects, ICIP '96 Proc. 2 (1996) 409–412.
[19] A.J. Stoddart, A. Hilton, Registration of multiple point sets. XPR '96, Vienna, Austria, Aug. 1996.
[20] G. Turk, M. Levoy, Zippered polygon meshes from range image. SIGGRAPH '94 Proceedings, 1994; ACN SIGGRAPH, pp. 311–318.
[21] Z. Zhang, Iterative point matching for registration of free-form curves. Proceedings of the IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling, vol. 2(2), Ottawa, Canada, 1994, pp. 119–152, IEEE Computer Society Press, Los Alamitos, CA.