

# Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique

Raouf Benjemaa and Francis Schmitt

École Nationale Supérieure des Télécommunications  
Images Department  
46 rue Barrault, 75634 Paris Cedex 13, France  
benjemaa@ima.enst.fr, schmitt@ima.enst.fr

## Abstract

*We present a new method for the global registration of several overlapping 3D surfaces sampled on an object. The method is based on the ICP algorithm and on a segmentation of the sampled points in an optimized set of z-buffers. This multi-z-buffer technique provides a 3D space partitioning which greatly accelerates the search of the nearest neighbours in the establishment of the point-to-point correspondence between 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.*

## 1. Introduction

A single scan of a complex 3D object is in general not sufficient to fully describe its surface because of the presence of occluded parts. Additional scans from different view points are thus required to recover the occluded parts and improve the surface description. To reach these different view points, 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 [4, 6, 15, 11, 12]. The widely used approach is based on the ICP (Iterative Closest Point) algorithm [4]. 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: i) the

determination of a point-to-point correspondence between two overlapping surfaces, ii) 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 Horn [8] and Faugeras [7] 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. To register together more than two overlapping surfaces, Kamgar-Parsi *et al.* [9] have proposed an approach using a dynamic system for the 2D registration of multiple overlapping range images. Recently an extension of this approach has been proposed by Stoddart and Hilton [14] for the 3D registration of multiple free-form surfaces. Two other recent papers have proposed an extension to the ICP algorithm for matching simultaneously several 3D surfaces [3, 13]. These multi-surface matching techniques provide a better global registration by distributing the residual errors more homogeneously.

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 Gauss spheres. This space partitioning strongly accelerates the point-to-point correspondence between all pairs of overlapping surface parts. In the second step, we use for the global registration, an iterative optimization process which is similar to the one proposed by Bergevin *et al.* [3]. 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 interactive registration has to be preliminary performed [1].

A z-buffer partitioning of the 3D space already greatly accelerates the determination of the point-to-point corre-

spondence 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 appears already inappropriate. This problem becomes clearly critical when we have to register a greater number of partially overlapping surfaces. Our multi-z-buffer technique 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.

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. 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 [4] 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. The two steps of the ICP algorithm are iterated until convergence of the registration process. Without any other knowledge on the data, the first step has a complexity which is quadratic in the number of points. It is then computationally very expensive especially when we have to process several tens of millions of points as we do.

In order to accelerate the ICP algorithm and by assuming that the density of the 3D points is homogeneous enough, we have proposed a method based on a twin z-buffer structure which provides us with an explicit space partitioning [1]. It is similar to two depth images, one per set of points, and allows us to apply 2D image processing tech-

niques 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.

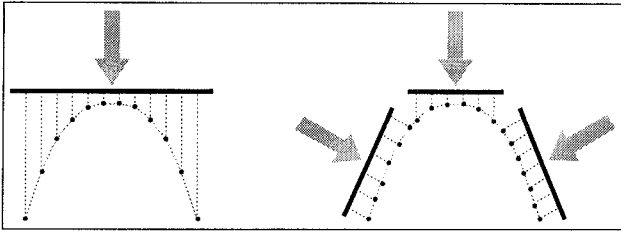


**Figure 1. Two renderings of an ancient coin before (left) and after (right) the mono-z-buffer registration.**

To build the twin z-buffer structure, we first chose a direction of projection and two orthogonal directions. We then compute the 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. We partition each set by projecting 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. This selection allows us, by adjusting the size of the cells where data are oversampled, to save memory space and to reduce CPU time for solving the surface correspondence problem.

Then, to match a point  $P_1$  in  $cell_1(w, h)$  of the first z-buffer, we look for the closest point  $P_2$  of  $P_1$  belonging to an  $(n \times n)$  window centered on  $cell_2(w, h)$  of the second z-buffer. This way the ICP algorithm is heavily accelerated. To reinforce its accuracy and robustness, we have added two matching criteria: a minimal density of points to test inside the  $(n \times n)$  window and a minimal distance to be satisfied by  $P_1 P_2$ .

This approach works well especially when the overlap between the two surfaces is sufficiently planar as illustrated in the example of figure 1. 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 Figure 2.



**Figure 2. The resampling of a curved surface is better in several adapted z-buffers (right) than in only one (left).**

### 3. Registration of two surfaces using multi-z-buffers

In the above technique with a single twin z-buffer structure, the direction of the projection is chosen interactively. 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 [2]. The registration is still processed in two stages. First, 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 to each region, a z-buffer perpendicular to its principal normal direction.

The remainder of this section is organized as follows. First, the estimation of an oriented normal at each sampled surface point is addressed in section 3.1.1. Then we explain in section 3.1.2 how the surfaces are segmented by a Gauss sphere. Finally, the optimization of the twin z-buffer set which entirely covers the overlaps of the two surfaces is described in section 3.1.3.

##### 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 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

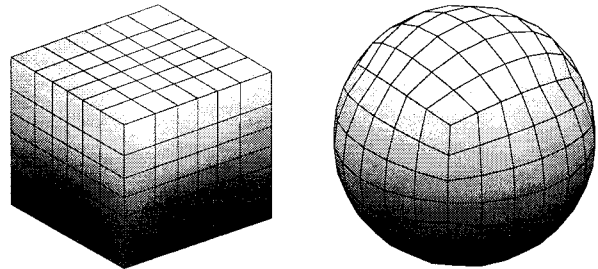
planar curves after small motions allows us to quickly define a neighborhood around each data point and to estimate its normal vector onto this neighborhood. The geometric discontinuities of the surface are first detected by adapting to our data a previous algorithm developed for range images [5]. Then the normal direction estimation is done by avoiding crossing over these discontinuities.

##### 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 whatever the object complexity.

A method to tessellate homogeneously and symmetrically the Gauss sphere is to project on it a unit cube subdivided in rectangular patches following a  $\tan(\theta)$  law along its edges,  $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$  [10]. 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.

Figure 3 shows an example of the Gauss sphere tessellation into 216 patches, the cube edges being subdivided into 6 pieces.



**Figure 3. Tessellation of the Gauss sphere (right) by the projection of a unit cube (left) subdivided into 216 patches**

##### 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 preliminary performed [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 center 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  $\bar{D}^p$  is the average of the normals of  $s_1^p$  and  $s_2^p$ . A pseudo code of the algorithm is presented in figure 4. 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 number of points to be considered for the correspondence problem and it subdivides it into sub-problems, one per twin z-buffer.

```

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

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

```

Figure 4. Creation of multiple twin z-buffers on the overlaps of surfaces  $S_1$  and  $S_2$

### 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 [8] is used for this purpose.

The points of the surface  $S_2$  are then transformed according to this rigid transformation 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 figure 5.

```

 $T^0 = T_{init}$ ;  $k = 0$ ; /*initialization*/
 $GS_1 = \text{Gauss-sphere}(S_1)$ ;  $GS_2 = \text{Gauss-sphere}(T^0(S_1))$ ;
Multi-Zbuffer( $GS_1, GS_2$ );
REPEAT UNTIL convergence {
   $k = k + 1$ ;
   $L = \text{EMPTY}$ ; /*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( $L$ ); }
   $T = \text{Optimal-rigid-motion}(L)$ ;
   $T^k = T \times T^{k-1}$ ;
  Update-all-non-Empty-Zbuffers  $zbf_{21}^p$  with  $T^k(s_2^p)$ ; }

```

Figure 5. Iterative registration of  $S_2$  to  $S_1$

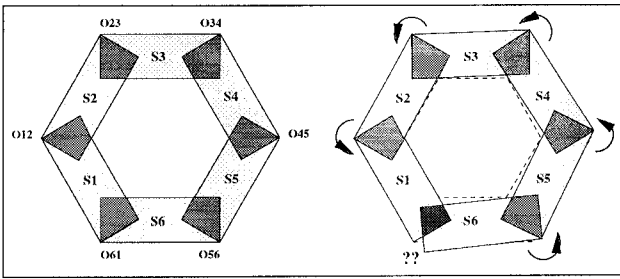
## 4. Extension to a global registration of N surfaces

When we have  $N$  surfaces to register simultaneously,  $N > 2$ , the problem becomes more global. 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.

### 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 can sequentially run 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 same places. A typical illustration of this situation is the registration of a closed string of digitized surfaces as shown in figure 6. The schematic object illustrated in figure 6(left) is composed of six surfaces ( $S_1, \dots, S_6$ ) shaded in light gray. The dark gray area represents the surface overlaps ( $O_{12}, O_{23}, \dots, O_{61}$ ). In figure 6(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 such 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.



**Figure 6. Error propagation in closed string of surface.**

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.

## 4.2. Global registration algorithm

Inspired by the work of Bergevin *et al.* [3], we have extended the previous pairwise registration based on multi-z-buffer 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* ( $S_1, S_2, \dots, S_{N-1}$ ). 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  $S_i, i \in \{0, \dots, N\}$ , 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  $s_i^p$  and  $s_j^p$  of  $S_i$  and  $S_j$  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  $S_i, i \in 1..N$ . 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 [8] 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 Bergevin's by the way the rigid transformations are applied. In Bergevin'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. This way, the convergence is accelerated. In order to not favour any slave surface, their registration order is randomly defined at each iteration. The pseudo-code of this algorithm is shown in figure 7.

```

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

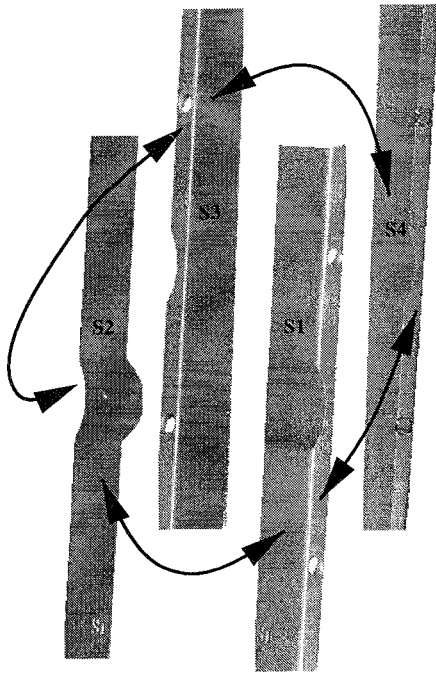
```

**Figure 7. Global registration algorithm**

## 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.



**Figure 8. Four different scans of a vibraphone bar and their overlapping relationship.**

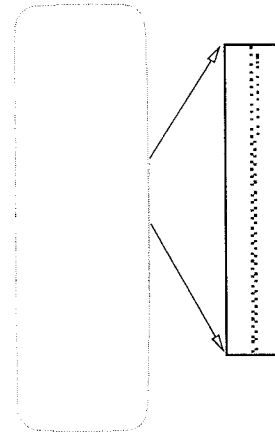
A vibraphone bar is used to illustrate the problem of the registration of a closed string of scanned surfaces. Figure 8 shows four different scans ( $S_1, S_2, S_3, S_4$ ) of the vibraphone bar. The arrows indicate the overlapping relationship between scans. We have applied to this set of scans both 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 figure 9. An artifact is clearly perceptible in this rendering. This mismatching is more visible when we extract the slide corresponding to the white dashed line. This slide is shown in figure 10. The shift present in the zoomed part is about 0.2 mm.



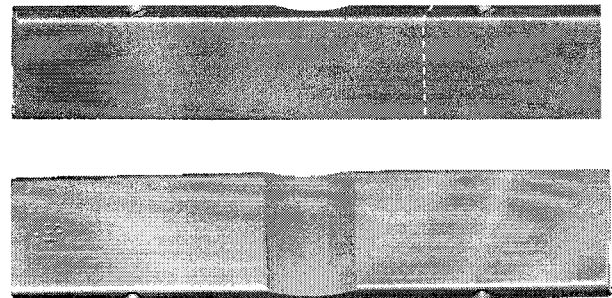
**Figure 9. Result of the pairwise registration.**

Such artifact is eliminated by applying the global registration technique as shown in figure 11. A slide at the same



**Figure 10. Zoomed slide after pairwise registration.**

localization is shown in figure 12. We notice that all scans are well matched.

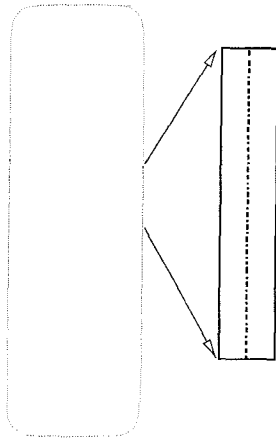


**Figure 11. Two different views after global registration.**

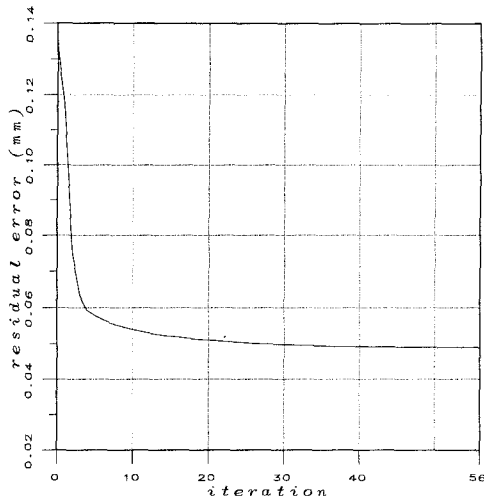
The evolution of the residual errors of the global registration is shown in figure 13. After 10 iterations the residual errors are already well distributed and the convergence of the process is nearly reached.

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 vibraphone bar, the 4 scans contain about 620,000 points each. Each iteration took about 160 seconds on a Sun SPARC-2000 machine. The root mean square error for the 4 scans is 0.049 mm after 25 iterations.

The bust shown in figure 15 is a difficult object to scan because of the presence of strong concavities. It needed as many as 60 different scans containing more than 11 million points. Some of the parts of the object are illustrated in figure 14. The scans were registered with the pairwise technique, so some artifacts are still present in figure 15. We



**Figure 12. Zoomed slide after global registration.**



**Figure 13. Global registration convergence.**

are now experimenting global registration on such kinds of data.

## 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 us a 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 and reverse engineering.

## Acknowledgment

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

## References

- [1] R. Benjemaa and F. Schmitt. Registering range views of complex objects. *4th European Conferences on Rapid Prototyping - Paris*, 12 pages, Oct. 1995.
- [2] R. Benjemaa and F. Schmitt. Recalage rapide de surfaces 3D après projection dans des multi-zbuffers. *5th European Conferences on Rapid Prototyping - Paris*, 11 pages, Oct. 1996.
- [3] 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.
- [4] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. on PAMI*, 14(2):239–256, Feb. 1992.
- [5] X. Chen and F. Schmitt. Surface modeling of range data by constrained triangulation. *Computer Aided Design*, 26(8):632–645, Aug. 1994.
- [6] Y. Chen and G. Medioni. Object modeling by registration of multiple range views. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2724–2729, Apr. 1991.
- [7] O. D. Faugeras and M. Hebert. The representation and recognition and locating of 3-D objects. *Int. J. Robotics Res.*, 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, Apr. 1987.
- [9] 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, Sept. 1991.
- [10] J. Maillot. *Trois Approches du Plaquage de Texture sur un Objet Tridimensionnel*. PhD thesis, Univ. de Paris VI, Oct. 1991.
- [11] T. Masuda. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding*, 61(3):295–307, May 1995.
- [12] V. Moron, P. Boulanger, H. Redarce, and A. Jutard. Mise en correspondance du modèle CAO d'un objet avec son image 3D : Application à l'inspection. *RFIA'96 proceedings, Rennes*, vol. 2:913–684, 1996.
- [13] M. Soucy, G. Godin, R. Baribeau, F. Blais, and R. Rioux. Sensors and algorithms for the construction of digital 3D colour of real objects. *ICIP'96 Proceedings*, 2:409–412, Sept. 1996.
- [14] A. J. Stoddart and A. Hilton. Registration of multiple point sets. *ICPR'96, Vienna, Austria*, Aug. 1996.
- [15] G. Turk and M. Levoy. Zippered polygon meshes from range image. *SIGGRAPH'94 Proceedings*, pages 311–318, 1994.

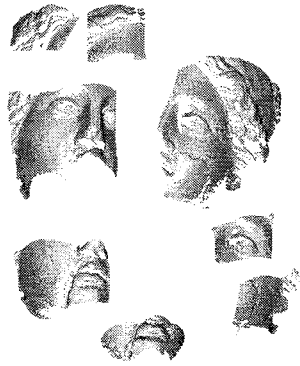


Figure 14. Some pieces of a Greek bust mosaic (Hygia, Dion Museum, Greece).

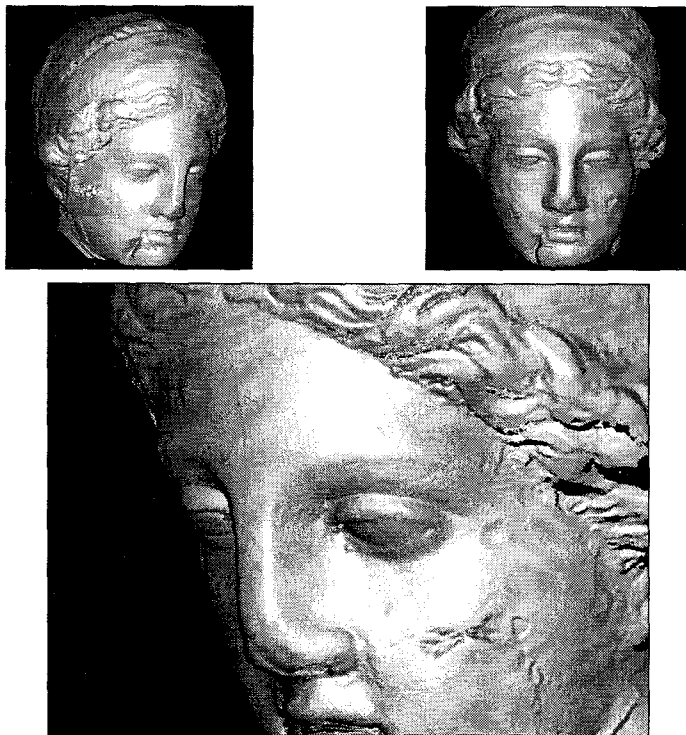


Figure 15. Three different views of the Greek bust after the registration of 60 scans.