# Method for registering overlapping range images of arbitrarily shaped surfaces for 3D object reconstruction

Eric Bittar, Stephane Lavallee, Richard Szeliski

Eric Bittar, Stephane Lavallee, Richard Szeliski, "Method for registering overlapping range images of arbitrarily shaped surfaces for 3D object reconstruction," Proc. SPIE 2059, Sensor Fusion VI, (20 August 1993); doi: 10.1117/12.150242

SPIE.

Event: Optical Tools for Manufacturing and Advanced Automation, 1993, Boston, MA, United States

# A method for registering overlapping range images of arbitrarily shaped surfaces for 3-D object reconstruction

Eric Bittar †, Stéphane Lavallée†, Richard Szeliski‡,

†TIMB - TIMC - IMAG

Faculté de Médecine de Grenoble, 38 700 La Tronche, France

‡Digital Equipment Corporation, Cambridge Research Lab

One Kendall Square, Bldg. 700, Cambridge, MA 02139

## Abstract

This paper presents a method to register overlapping 3-D surfaces which we use to reconstruct entire three-dimensional objects from sets of views. We use a range imaging sensor to digitize the object in several positions. Each pair of overlapping images is then registered using the algorithm developed in this paper. Rather than extracting and matching features, we match the complete surface, which we represent using a collection of points. This enables us to reconstruct smooth free-form objects which may lack sufficient features. Our algorithm is an extension of an algorithm we previously developed to register 3-D surfaces. This algorithm first creates an octree-spline from one set of points to quickly compute point to surface distances. It then uses an iterative nonlinear least squares minimization technique to minimize the sum of squared distances from the data point set to the octree point set. In this paper, we replace the squared distance with a function of the distance, which allows the elimination of points that are not in the shared region between the two sets. Once the object has been reconstructed by merging all the views, a continuous surface model is created from the set of points. This method has been successfully used on the limbs of a dummy and on a human head.

## 1. Introduction

The problem of 3-D shape reconstruction can be defined and solved in many ways. We are particularly interested in the application of this problem to the reconstruction of anatomical organs of the body and to other applications in the medical field. The method we propose in this paper uses a range imaging sensor (RIS) to digitize the object in several positions. We get around the limited visibility of the RIS by viewing the object from several positions, which allows us to digitize all parts of it, even certain concavities. The selection of object positions must be done carefully in order to obtain a sufficient number of overlapping views to cover the whole part of the surface that we want to digitize. This method is an alternative to the use of several range imaging systems linked together, which are more expensive and difficult to calibrate. For instance, mounting a RIS on the end-effector of a robot can be used to digitize complex shapes, but this raises some difficult problems of intrinsic and extrinsic calibration: due to the distance between the object and the robot, any orientation errors of the end-effector are magnified, which can lead to erroneous reconstruction when views are merged.

The main application of our technique is the reconstruction of an external surface of the body from a number of overlapping views. For instance, in reconstructive surgery, it is necessary to quantify the result of an intervention (during or after the operation), and a single range image is usually not enough to reconstruct the whole surface of interest (face, breast, etc...). For example, to reconstruct a face, the patient has to move his head in 3 positions relative to the sensor. This technique could also be used to register two range images of a patient taken at different times, in order to register the patient position relative to the operating table.

Our RIS consists of a fixed video camera which observes the projection onto the object of a moving laser plane. The laser translates along one axis. This sensor is carefully calibrated, using to a method specially developed for this system which uses both a mathematical camera model (N-planes B-Spline) and an accurate mechanical calibration



0-8194-1324-0/93/$6.00

setup [7]. For each view, we obtain a set of curves that lie on parallel planes. These curves consist of a succession of 3-D points of the surface of the object. Each pair of overlapping images is then processed by our registration algorithm.

Many techniques for registering two 3-D surfaces are based on feature extraction and matching. This consists of detecting surface singularities such as significant points, edges, curves, or regions in both views, and matching them. Our approach is based on using lower-level information in order to reconstruct smooth free-form objects which may lack sufficient features. It is based on an existing matching algorithm [17] which registers two surfaces when the first one is included in the second one. We extended this algorithm to allow for partial overlap in the surfaces. The technique is an iterative process (a nonlinear least squares minimization technique) which minimizes the sum of a function of distances from one set to the other. The choice of the function allows the elimination of the points that are not in the shared region between the two sets.

We begin the paper in section 2 with a review of related work in the field of 3D-shape reconstruction, medical image registration, and computer vision. In section 3, we summarize the previous algorithm that forms the basis of this work, including the fast computation of point to surfaces distances using *octree spline* 3-D distance maps. Section 4 presents a first approach to solve the problem of overlapping views, using user interaction. Section 5 details the solution we propose to obtain an automatic registration. In section 6, we show that our solution must be embedded in a higher-level process. Section 7 presents some experimental results with real data. Finally, the algorithm, extensions of the method, and future improvements are discussed in section 8.

# 2. Previous work

In general, registration methods are divided into two categories, i.e., those with and those without features. We will give here a brief summary of the techniques in both categories. For more complete information, see [8] and [3].

The first main issue involved in the feature correspondence methods is the choice of a set of features. Typically, significant surface points [8, 21], surface line segments [11], surface curves [21, 15, 12], or surface regions [11, 18, 10] are used. The correspondence between the features must then be established, which is done by subgraph isomorphism [10], hypothesis-and-verification [11], Hough transform [16], or geometric hashing [12]. Finally, the transformation using this feature correspondence is computed.

Registration without correspondence methods come in two varieties: methods based on the visibility of the same set of points [25], and methods based on minimizing a distance measure [5, 17, 3]. An automated algorithm for matching 3-D surfaces with other 3-D surfaces (such as the head skin surface) has been developed by Pelizzari [19]. Our previous algorithm solved this problem by minimizing the sum of squared distances between the transformed points on one surface and a stationary description of the other surface [17, 7]. It also solved the more difficult problem of registering a 3-D surface with its 2-D projections [17, 5].

The registration of overlapping views has been explored with the feature extraction method by Kamgar, Jones and Rosenfeld [15] and Chen [8]. In [15], elevation contours are extracted from range images of the floor of the ocean to register two overlapping views, after which a global scheme is used to improve the local registration by imposing some compatibility constraints. However, their method only deals with 2-D rotations and translations. In [8], Chen proposes an interesting hierarchical method which uses features extraction (differential properties of surfaces) to provide an initial position to a featureless registration. However, the limitation of the feature-based methods still persists, i.e., whenever features cannot be detected, the algorithm cannot be initialized.

In medical imaging, the problem of image registration has been usually solved by using external fiducial markers placed on the body of the patient [14] or by interactively selecting pairs of matching points [23].

# 3. Previous matching algorithm

This section summarizes the previous algorithm intended to register two surfaces when one is a subset of the other [17].

## 3.1. Transformation and parameters

Our task is to compute the rigid body transformation between two positions 1 and 2 of an object in the range imaging coordinate system (for an extension of this approach to local non-rigid matching, see [24]). We formulate this problem as the estimation of the transformation $\mathbf{T}$ between $\mathrm{Ref}_1$ and $\mathrm{Ref}_2$, where $\mathrm{Ref}_1$ ($\mathrm{Ref}_2$, resp.) defines the reference system of the range imaging sensor relative to the object when the object is in position 1 (position 2, resp.). $\mathbf{T}$ is also called the attitude of the surface $S_1$ with respect to $S_2$. $\mathbf{T}$ can be defined by a translation vector $\mathbf{t} = (T_x, T_y, T_z)^t$ and a $3 \times 3$ rotation matrix $\mathbf{R}$. Several representations can be used for $\mathbf{R}$ (e.g., Euler angles,

quaternions, or rotation vectors [1]). We have chosen to use Euler angles $(\phi, \theta, \psi)^t$, where $\mathbf{R}$ is constructed from 3 rotations around $x, y, z$ with angles $\phi, \theta, \psi$,

$$\mathbf{R} = \begin{pmatrix} \cos\psi\cos\theta & -\sin\psi\cos\theta & \sin\theta \\ \cos\psi\sin\theta\sin\phi + \sin\psi\cos\phi & -\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & -\cos\theta\sin\phi \\ -\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi + \cos\psi\sin\phi & \cos\theta\cos\phi \end{pmatrix}. \quad (1)$$

If we gather the 6 parameters of the transformation $\mathbf{T}$ into a 6-component vector

$$\mathbf{p} = (T_x\ T_y\ T_z\ \phi\ \theta\ \psi)^t, \quad (2)$$

and use homogeneous coordinates, $\mathbf{T}(\mathbf{p})$ can be represented by a single $4 \times 4$ matrix

$$\mathbf{T}(\mathbf{p}) = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}. \quad (3)$$

The relation between a point $\mathbf{q} = (x\ y\ z\ 1)^t$ in $\mathrm{Ref}_1$ and the corresponding point $\mathbf{r} = (X\ Y\ Z\ 1)^t$ in $\mathrm{Ref}_2$ can be written as

$$\mathbf{r} = \mathbf{T}(\mathbf{p})\,\mathbf{q} \quad (4)$$

## 3.2. Minimization of a cost function

The low-level approach we use to register the two surfaces $S_1$ and $S_2$ is to estimate the parameters $\mathbf{p}$ which minimize a distance between $S_2$ and the transformation of $S_1$ by $\mathbf{T}(\mathbf{p})$. More specifically, we assume for now that the surface $S_1$ is represented as a collection of points $\{\mathbf{q}_i, i = 1 \ldots N\}$, and that the other surface $S = S_2$ is represented in some arbitrary way. The estimation task is then to find a geometric transformation $\mathbf{T}$ such that the transformed coordinates $\mathbf{r}_i = \mathbf{T}(\mathbf{p})\mathbf{q}_i$ all lie on the surface $S$. In practice, due to sensor noise and the inability to perfectly register two surfaces, this condition will never be satisfied. Instead, we pose the problem as a minimization of the cost function

$$\mathcal{C}(\mathbf{p}) = \sum_{i=1}^{N} \frac{1}{\sigma_i^2}[d(\mathbf{r}_i, S)]^2 = \sum_{i=1}^{N} \frac{1}{\sigma_i^2}[d(\mathbf{T}(\mathbf{p})\mathbf{q}_i, S)]^2, \quad (5)$$

where $d(\mathbf{r}_i, S) = \min_{\mathbf{s}\in S} \|\mathbf{r}_i - \mathbf{s}\|$ is the minimum Euclidean distance from the point $\mathbf{r}_i$ to $S$ and $\sigma_i^2$ is the variance associated with point $i$ [17, 3].

To perform the nonlinear least squares minimization, we use the Levenberg-Marquardt algorithm because of its good convergence properties [20]. See the details of our implementation in [17, 24].

Obviously, the choice of the initial transformation is important since such a method cannot pretend to avoid local minima. However, in our problem we always have a rough idea of the transformation we are looking for. For instance, if we acquire 3 range images of the patient who moves his head with respect to the sensor, we can initialize the method with zero-translations and rotations around a known axis of 30° or 45°. Similarly, if a RIS is mounted on a robot, even poor robot calibration will still produce a good set of initial transformations. Such knowledge was found to be sufficient for all our applications using this algorithm when one surface is a subset of the second one.

## 3.3. Fast distance computation using octree splines

The method described in the previous section relies on the fast computation of the distance $d(\mathbf{r}, S)$ and its gradient. To speed up this computation, we precompute a 3-D *distance map*, which is a function that gives the minimum distance to a surface $S$ from any point $\mathbf{r}$ inside a bounding volume $V$ that encloses $S$ [9, 4]. In looking for an improved trade-off between memory space, accuracy, speed of computation, and speed of construction, we developed a new kind of distance map which we call the *octree spline* [17, 5, 7]. The intuitive idea behind this geometrical representation is to have more detailed information (i.e., more accuracy) near the surface than far away from it. We start with the classical octree representation associated with the surface $S$ [22] and then extend it to represent a continuous 3-D function that approximates the Euclidean distance to the surface. This representation combines advantages of adaptive spline functions and hierarchical data structures.

As with the previously described distance map, the input to the octree spline construction algorithm is a set of $n$ points $\mathbf{s}_j$ regularly spread on the surface $S$. The algorithm performs the following steps (see [17] for details):

1. **Surface point octree construction:** First, the octree associated with the set of points $\mathbf{s}_i$ is built according to classic octree subdivision [22]. Starting from the initial cube $V$, each node that contains points (grey node) is recursively split into 8 sub-cubes until it contains no points (white node) or it has the maximal selected resolution (black node).
2. **Subdivision (refinement):** The octree previously computed may have large empty nodes near the surface, because no rules about subdivision near the surface have been introduced. To overcome this problem, we perform a further subdivision of the octree to ensure that two nodes which are neighbors along a face, edge, or corner differ in size by at most a factor of $k_S$ (in practice, we choose $k_S = 2$).
3. **Corner distance computation:** For each corner $\mathbf{c}$ of each terminal node (white or black), the distance $d(\mathbf{c}, S)$ is computed. The spatial organization of surface points created by the octree makes this process much faster since we can use best-first search to find this minimum distance quickly [17].
4. **Crack elimination (continuity enforcement):** Because the distance is computed at any point $\mathbf{r}$ by an interpolation based on the 8 corner values of the terminal node that contains $\mathbf{r}$ (see below), discontinuities or *cracks* can appear if we simply interpolate the true corner distance values. To avoid this, if a corner $\mathbf{c}$ of a node $N_1$ lies on a face or an edge of another node $N_2$ of greater size, then the distance value of the corner $\mathbf{c}$ is simply replaced by the distance computed at $\mathbf{c}$ by interpolation inside $N_2$.

After the previous steps have been performed, $d(\mathbf{r}, S)$ can be computed for any point $\mathbf{r}$ by first finding the terminal node $N$ that contains the point $\mathbf{r}$ (using classical binary search) and then using a trilinear interpolation of the 8 corner values $d_{ijk}$. We can compute the gradient $\mathbf{g} = \nabla d(\mathbf{r})$ of the distance function by simply differentiating the trilinear interpolant with respect to $u$, $v$, and $w$. Because $d$ is only $C^0$, $\mathbf{g}(\mathbf{r})$ is discontinuous on cube faces. However, these gradient discontinuities are relatively small and do not seem to affect the convergence of our iterative minimization algorithm.

# 4. First approach : the use of interactivity

To deal with a low percentage of erroneous data, the method described in the previous paper [17] had to suppress, in a final step, the statistical outliers. We applied this idea to the registration of overlapping range data sets. The data that are not in the overlap region are considered to be erroneous points and are not considered in the minimization process. However, the points that seem erroneous in a current pose estimate may be different from those in the correct registration attitude.

Based on this observation, we developed software to interactively move in translation and in rotation the two objects with respect to each other. This allows us to find an initial transformation quite close to the correct one. In that initial attitude, we suppress the points for which the current distance is greater than a given threshold, so that most of the points outside the overlapping region are eliminated, and then we start the standard least-squares minimization. The obvious disadvantage of this technique is that it requires human intervention. In practical cases, one often has to iterate many times the interactive move/point suppression/minimization process to reach an acceptable result.

# 5. Heuristics toward an automatic registration

A preferable approach is to develop an iterative process that selects outlier points automatically. However, a point suppressed at an iteration $k$ may become significant in a later iteration. What we need is a criterion to determine if a particular point will be or will not be used in a particular iteration without actually suppressing the point from the data set.

Instead of using (5), we now minimize a function C defined by the following equation:

$$C(\mathbf{p}) = \sum_{i=1}^{N} \alpha_i [d(\mathbf{r}_i, S)]^2 \tag{6}$$

where $\alpha_i$ is a weight associated with each point that we adjust at each minimization step. As a point may or may not be significant for a given attitude, we will choose $\alpha_i = 0$ for points where $d(\mathbf{r}_i, S)$ is large and $\alpha_i = 1$ for the others, so that only these contribute to the minimization step (see figure 1). Another possibility would be to choose to set the weight $\alpha_i$ as a function $f(d_i)$ which approaches zero as $d_i$ increases in order to penalize points far away from the reference surface $S_2$.

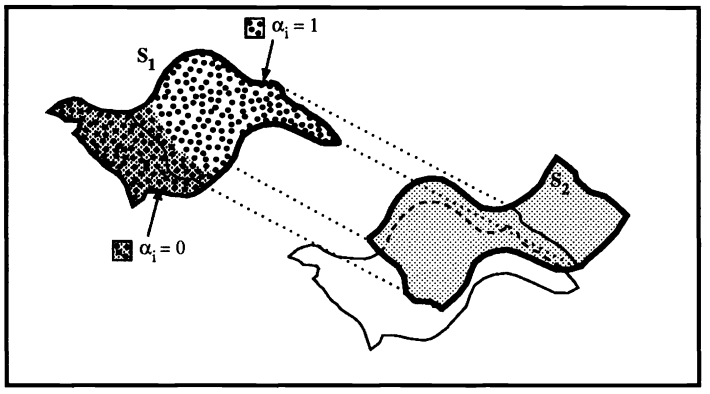



Figure 1: Visualization of the regions of weight $\alpha_i$ equal to 0 and 1

In our current implementation, we sort the points according to their distance to the model set, and we only keep the $p$% best, i.e., we minimize the mean error of the $p$% best points (this is a simple form of *robust statistics* [13]). However, this choice generates a greater number of local minima than the initial function, so that the convergence domain is slightly reduced. One has to start from an attitude of the data set that is closer to the solution attitude to perform a valid registration.

Furthermore, the percentage $p$ has to be determinated. A percentage $p$ equal to 100% draws us back to the usual algorithm, with all the data points. In this case, we search to minimize the mean error between the two sets, but this criterion doesn't allow the registration of overlapping sets. On the other end, a choice of a small $p$ may lead to a local minimum that can be quite far from the ideal solution. An ideal overlapping percentage of the two sets exists: it is the real overlapping percentage of the two sets $p_{\text{real}}$. We have a rough idea of its value but it is impossible to know it *a priori* with a good accuracy. Moreover, the described minimization process started with $p_{\text{real}}$ does not systematically converge to the ideal attitude, because of possible local minima. This procedure must be integrated with a higher-level layer to avoid this local minima problem.

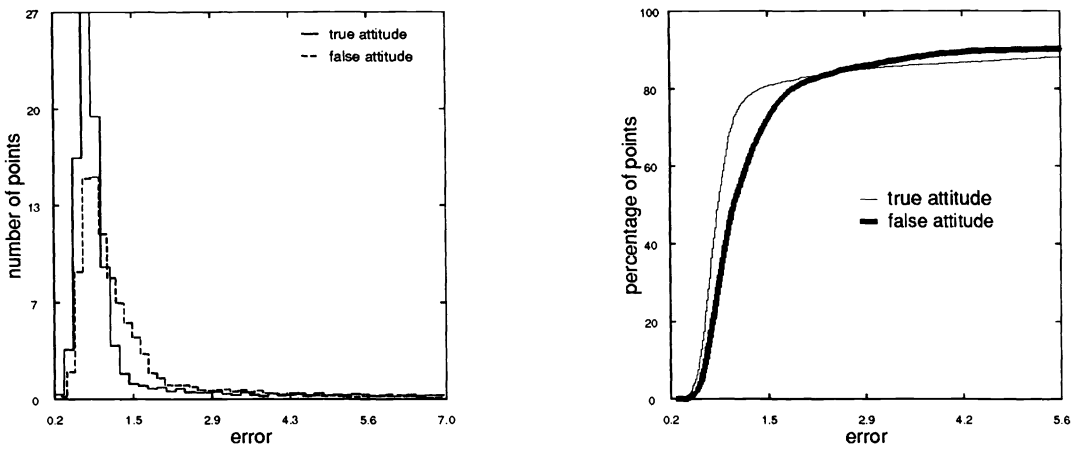
# 6. Higher-level process

## 6.1. Global method

The only way to ensure reaching the global minimum is to find the minimum of all the local minima. By partitioning the registration state space into local minima attraction regions, we can then start the algorithm from every created region, and we are certain to find all the local minima. But it is as difficult to characterize these regions as to find the local minima, since we must discretize the parameter space, and explore it systematically. Unfortunately, the state-space is of dimension 6, and it is too large to be extensively covered. When facing this problem, we could try to reduce the dimension of this space with an approximation criterion. Besl [3] does this by considering in certain cases only the 3 rotation parameters, after having made the centers of mass of the two sets coincident. But when the objects only overlap partially, the two centers of mass are not equal in the registration attitude. We could also experiment simulated annealing or genetic algorithms but we preferred to look for heuristics that are adapted to our specific problem.

## 6.2. A criterion to evaluate the transformation between the two sets

To be able to choose between two minima, we need an evaluation criterion that is different from the residual mean error after convergence. Indeed, we could exhibit some false attitudes where the mean error was better than the mean error for the correct attitude. To discriminate false and true attitudes, we must look at the distribution of residual errors. The histogram that gives the number of points as a function of the error shows a peak for the points of this overlapping set. This peak is higher for the correct attitude than for a false attitude, which is justified by the following remark. For the correct attitude, the $p$% best points are fitting the surface accurately, with an error that corresponds only to the range imaging sensor accuracy. For a false attitude, the $p$% best points fit more or less the surface with some errors ranging from zero (where the points in the wrong attitude are exactly on the reference surface) to a value that take sensor noise and surface misfit into account. For both cases, the mean residual error is the same, but the error distribution is different (see figure 2). We therefore decided to compute the percentage of




(a) (b)

Figure 2: Histograms of errors computed for two different final attitudes
For correct and false attitudes, the mean residual error is similar, but the distributions of errors are different. On the left (a), the histogram shows that for the correct attitude, most of errors are below the sensor accuracy (1.5 mm). On the right (b), the cumulative histograms show that the percentage of points for which the error is lower than the sensor accuracy (1.5 mm) is greater for the correct attitude than for the false attitude.

points for which the error is lower than the sensor accuracy $d_{sensor}$ ($d_{sensor}$ is estimated through reference objects [6]). This percentage defines a comparison criterion $C'(\mathbf{p})$ :

$$C'(\mathbf{p}) = 1/N \sum_{i=1}^{N} \delta_i \quad \text{with} \quad \delta_i = \begin{cases} 1 & \text{if } d(\mathbf{T}(\mathbf{p})\mathbf{q}_i, S) < d_{sensor} \\ 0 & \text{else} \end{cases} \tag{7}$$

### 6.3. Decreasing the percentage of best distances

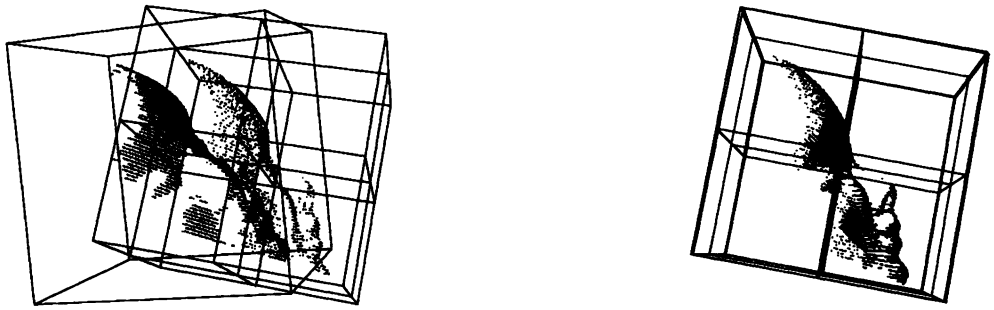
As we do not know how to determine the ideal percentage, we operate successive iterations with a decreasing value of the percentage. Starting with $p = 100\%$, we minimize our cost function to roughly position the object. This result is then used as the initial transformation of a new convergence using $p = 90\%$, etc., with values of the percentage decreasing by 10% at each loop, until a given lower bound $p_{\min}$ is reached. We chose $p_{\min} = 30\%$ because this is slightly inferior to the lowest value of the real overlapping percentage possible for our applications.

During our experiments, we used this heuristic and checked the comparison criterion as well. The best transformation is chosen at the end of the whole iterative process: among the transformations obtained at the end of convergence for each one of the tested percentages, we select the transformation that minimizes the criterion $C'(\mathbf{p})$ defined in (7). In some cases, we observed that the attitude obtained at the end of the 50% minimization loop is better than the one later obtained with 30% of the data.

## 7. Experimental results

### 7.1. Accuracy test using a known transformation

For our first experiments, we attached a polystyrene head to an articulated plate coded with verniers. This allowed us to calculate the true rigid body transformation between the head surface in two different locations [7]. We then applied our algorithm to the two range images. The first image was used to build an octree-spline that constitutes the model. Starting from an initial transformation, we ran our iterative registration algorithm between the range data of the second image and the model. At each iteration $k$, we compute the error transformation $\Delta\mathbf{T}^{(k)}$ and we extract the norm of translation error $\Delta\mathbf{t}^{(k)}$ and $\Delta\alpha^{(k)}$, the angle of the rotation component of $\Delta\mathbf{T}^{(k)}$. The values of $||\Delta\mathbf{t}^{(k)}||$ and $|\Delta\alpha^{(k)}|$ are displayed to monitor the convergence of the algorithm towards the optimal solution. These values constitute a test on the real accuracy of the method. The results are shown in figures 3 and 4. Our experiments have produced good results. Compared to the previous algorithm described in [7], our technique has similar rotation and translation errors, but no longer requires a complete overlap of the two surfaces.



(a) (b)

Figure 3: Convergence of the registration algorithm for the phantom face.
(a) initial configuration, $||\Delta \mathbf{t}^{(0)}|| = 58.5mm$, $|\Delta\alpha^{(0)}| = 21.5°$,
(b) after 6 times 10 iterations, ie p%=100, 90, 80, 70, 60, 50, 40 $||\Delta \mathbf{t}^{(60)}|| = 1.5mm$, $|\Delta\alpha^{(60)}| = 0.01°$.

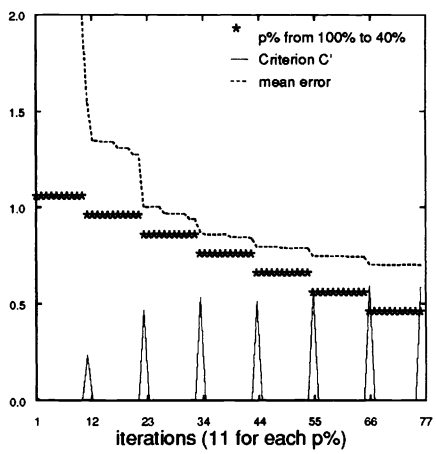
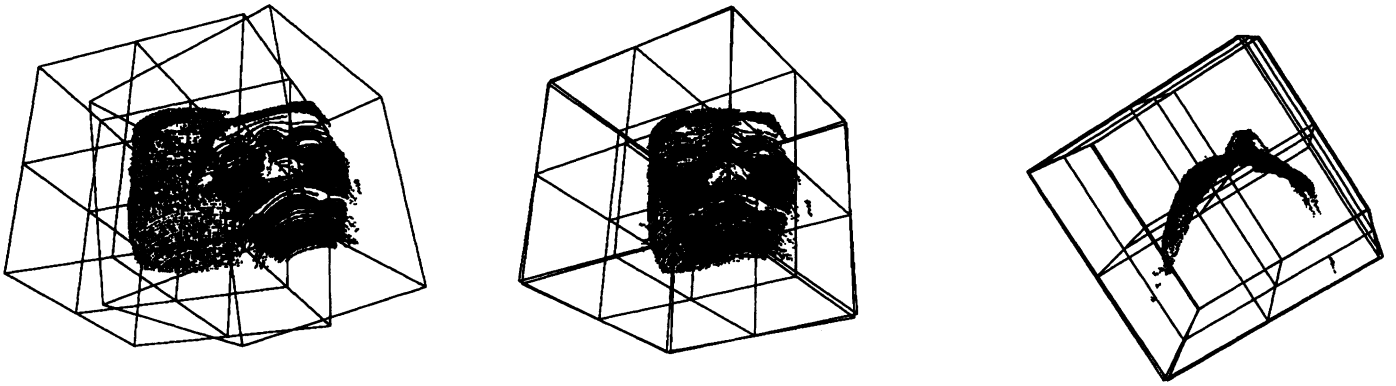



Figure 4: Curves describing a typical convergence for the phantom face.
The first curve shows the evolution of the mean error as a function of the number of iterations. Note that this error curve has discontinuities each time the percentage retained is descreased. A second curve shows that the percentage of best values is decreased from 100% to 40% by steps of 10%. The third curve shows the evolution of the criterion $C'$ computed at the end of a convergence obtained for a given percentage. The best value of this criterion is obtained at the end of the convergence obtained with a percentage of 50%.



(a) (b) (c)

Figure 5: Example of a real human head (Gilles).
About 70% of points overlap between both surfaces at the correct attitude. (a) initial configuration, $||\Delta \mathbf{t}^{(0)}|| = 62mm$, $|\Delta\alpha^{(0)}| = 14°$. (b) after 7 × 10 iterations, i.e., p=100%, 90%, 80 %, 70%, 60%, 50%, and 40%, $||\Delta \mathbf{t}^{(70)}|| = 0.8mm$, $|\Delta\alpha^{(70)}| = 1.28°$. (c) Another view of the final configuration

## 7.2. Other experiments

We made numerous experiments on the limbs and the torso of a dummy, and on a living head. We succeeded in reconstructing these objects as shown in figures 6–9.

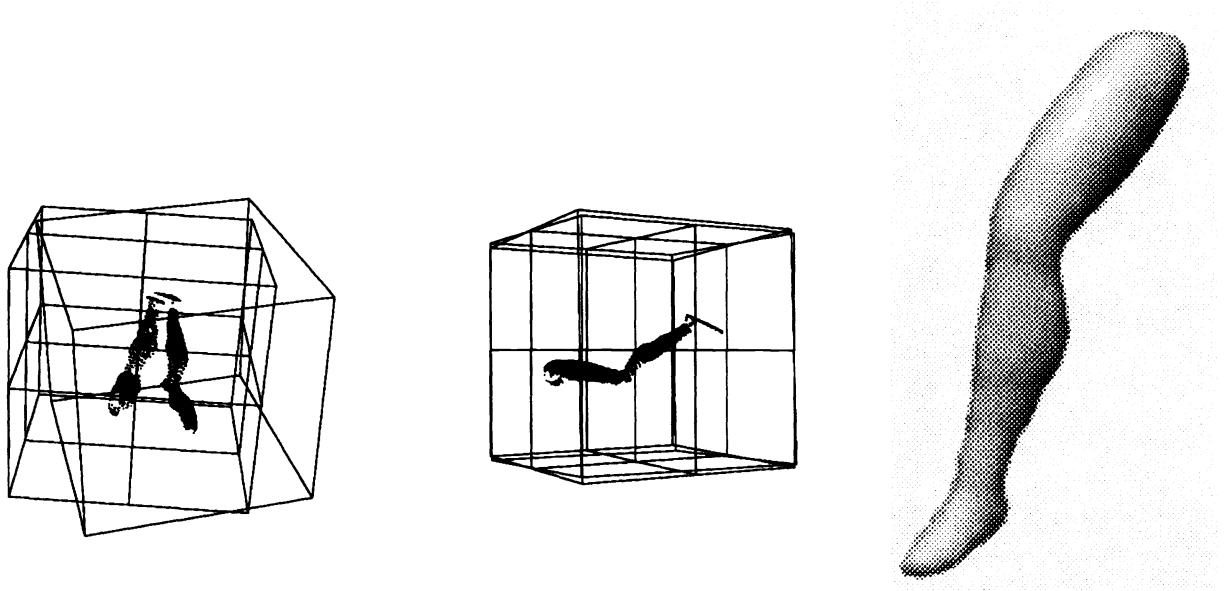
For these experiments, the real transformations between the views were unknown. So we applied our interactive process described in section 4 to compute reference transformations, considered as "correct" transformations, with which the result of our automatic algorithm can be compared. This enables us to compute errors $||\Delta \mathbf{t}^{(k)}||$ and $|\Delta\alpha^{(k)}|$ as in previous experiments. Since these values are only used to roughly discriminate between true and false attitudes, this must not be considered as a real accuracy test.

For each part of the disassembled dummy, we acquired about 10 range images and we applied our algorithm to each pair of views close to each other. At the end of this process, all views were merged into one set of points that represents the whole object. We did not considered registering the first and last views, although they were related, so the final 3D reconstruction cumulates errors obtained for each registration of pairs of views (see section 8 for the possibility of taking such global information into account). Then, to obtain a complete 3D representation suitable for 3D visualization, we passed the final set of points through a novel 2-step triangulation process developed in our laboratory. In the first step, the set of points is used to build an octree-spline. In the second step, a deformable model named *delta-snakes* is launched in the octree-spline to find an equilibrium in regions where the distance map function is minimum [2]. Since the representation of delta-snakes is always a set of connected triangles, the result is a triangulated surface that can be visualized with graphics software such as AVS.

Figure 5 shows the registration of two parts of a human face. *Gilles* was asked to rotate his head with respect to the sensor, and then the automatic registration was started from a rough estimate of the possible motion. A good convergence is reached because the shape has no near-invariant properties under rigid transformations.

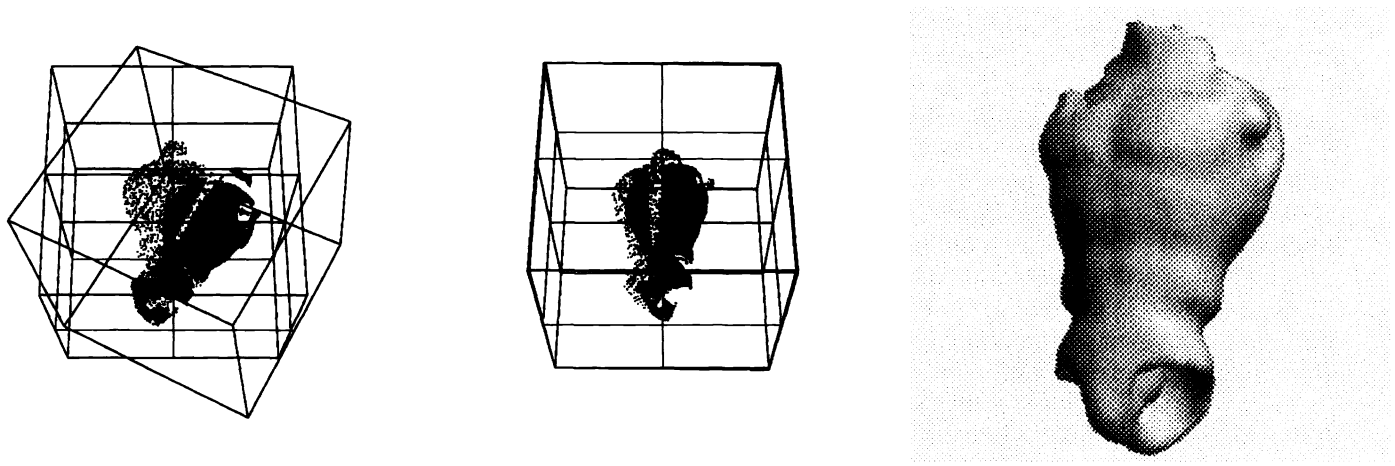
Figures 6 and 7 show the good results obtained for some parts of the dummy (leg and torso). About 10 views per part enabled us to obtain 3D reconstructions that are visually correct. Again, these surfaces are not near-invariant under rigid transformations. Note that the same sensor was used for both the human face and the dummy parts which are less that 10cm long. Thus, still better results could be obtained by reducing the field of view of the sensor (improving its accuracy).

Figure 8 shows a case where the algorithm is trapped in a local minimum. Since the arm of the dummy is roughly a cylinder, the orientation parameter around this cylinder axis is very unstable in the minimization process (the cost function is very flat along the direction corresponding to that parameter, and small local minima naturally appear). To escape from such a local minimum, we simply re-started the whole process (from 100% to 40%) from the final attitude previously reached. The result of the whole reconstruction for this difficult case is shown in figure 9.



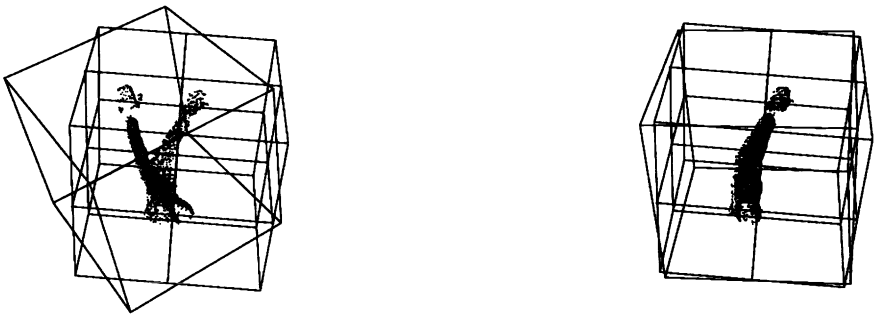
(a) (b) (c)

Figure 6: Example of the left leg.
About 80% of points overlap between both surfaces at the correct attitude. (a) initial configuration, $||\Delta \mathbf{t}^{(0)}|| = 25mm$, $|\Delta\alpha^{(0)}| = 18°$, (b) after $7 \times 10$ iterations $||\Delta \mathbf{t}^{(70)}|| = 0.25mm$, $|\Delta\alpha^{(70)}| = 2.8°$. (c) the reconstructed leg



(a) (b) (c)

Figure 7: Example of the torso.
About 75% of points overlap between both surfaces at the correct attitude. (a) initial configuration, $||\Delta \mathbf{t}^{(0)}|| = 13mm$, $|\Delta\alpha^{(0)}| = 81°$. (b) after $7 \times 10$ iterations $||\Delta \mathbf{t}^{(70)}|| = 0.44mm$, $|\Delta\alpha^{(70)}| = 0.56°$. (c) the reconstructed torso



(a) (b)

Figure 8: Example of the arm.
About 65% of points overlap between both surfaces. (a) initial configuration, $||\Delta \mathbf{t}^{(0)}|| = 17mm$, $|\Delta\alpha^{(0)}| = 36°$, (b) after $7 \times 10$ iterations, stuck in a local minimum $||\Delta \mathbf{t}^{(70)}|| = 1.01mm$, $|\Delta\alpha^{(70)}| = 6.36°$.

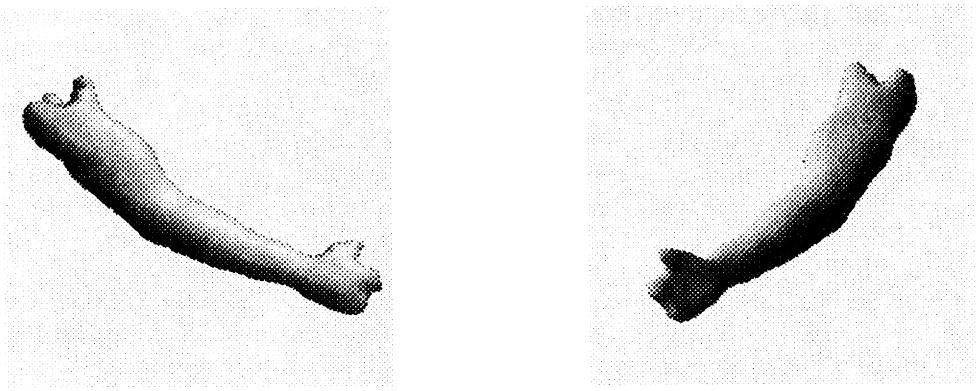


Figure 9: Views of the reconstructed arm

# 8. Discussion and perspectives

Our experiments show that our technique gives good results for an initial attitude close to the ideal attitude, i.e., within 10° to 70° of rotation, depending on the object. These results are not shape-independent, e.g., they are worse for cylindrically or spherically shaped objects than for objects without such properties of near-invariance under rigid transformations.

It may be possible to use a topological criterion in our algorithm. Indeed, for each individual connected component of the surface $S_1$, the matched points (which overlap with $S_2$) and the unmatched points (which do not overlap with $S_2$) are in two distinct connected regions. Since matched points and unmatched points are not mixed in space, we could take this effect into account in our cost function.

The main weakness of our algorithm is that for some surfaces that have cylindrical or spherical shapes, or for cases where a rough initial transformation can not be estimated a priori, it still falls into local minima. We intend to experiment adding a global layer to our method. It will be a global minimization on all the partial views at one time, to improve the number of constraints and converge more efficiently. The inter-view relations will be specified in a graph determined a priori (figure 10). Instead of the 6 parameters we now have, we will have $6 \times (N-1)$ independent parameters where $N$ is the number of views. For example, we can use the $6 \times (N-1)$ parameters that define the attitudes of view $2 \ldots N$ with respect to view 1. Any transformation from view $i$ to view $j$ can be computed from combinations of this minimal set of parameters. Octree-spline distance maps will be computed for each set of surface points. For each point in each view, the distance function will be replaced by the minimum distance computed in all adjacent views.

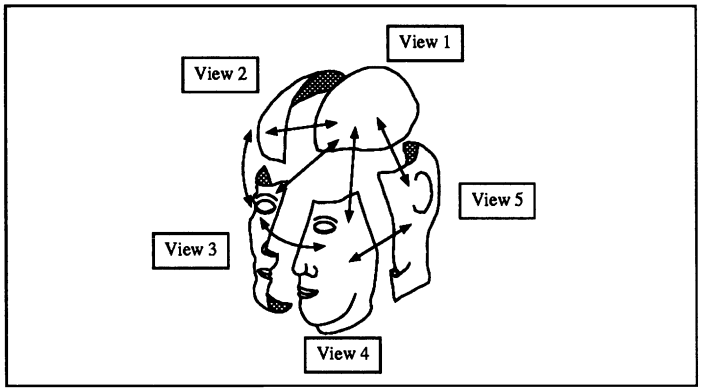



Figure 10: Graph of corresponding views

# 9. Conclusion

In this paper, we have presented an automatic method to register two overlapping sets of data that supports the global reconstruction of the shape of an object. This allows us to use only one range imaging sensor instead of expensive multiple range imaging sensors. We use a carefully calibrated range imaging sensor and a registration algorithm that does not need the extraction of features in the objects, which enables us to reconstruct smooth free-form objects. We use an octree spline to quickly compute the point-to-surface distance from anywhere is our space. We then perform a few loops of non-linear least squares minimization using the Levenberg-Marquardt algorithm with only a percentage of the best data (selected at each iteration) to register the two overlapping sets of data. This basic algorithm is run for several percentages of best data decreasing from 100% to 40%, the result obtained with one percentage being used as a starting point for the convergence with the next percentage. We found that such heuristics work well for our cases, except if the shape of the object is near invariant under rigid transformations, or if the initial attitude is too far from the correct attitude. Due to our digitizing method, we worked on the surface of the objects, but our matching algorithm can also deal with sets of points spread throughout volumes. In the medical field, this method has a direct application in digitizing external surfaces with a simple and accurate sensor.

# 10. Acknowledgments

The authors wish to thank Gilles Bayon for his helpful comments and Guillaume Çhampleboux for his help on calibration.

# References

[1] N. Ayache. *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception.* MIT Press, Cambridge, Massachusetts, 1991.

[2] E. Bainville. Reconstruction d'objets tridimensionnels a partir de silhouettes (in french). Technical report, TIMB, University of Grenoble, 1992.

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

[4] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[5] L. Brunie, S. Lavallée, and Szeliski R. Using forces fields derived from 3D distance maps for inferring the attitude of a 3D rigid object. In G. Sandini, editor, *Second European Conference on Computer Vision (ECCV'92)*, pages 670–675, Santa Margherita, Italy, May 1992. Springer Verlag, LNCS Series Vol. 588.

[6] G. Champleboux, S. Lavallée, P. Sautot, and P. Cinquin. Accurate calibration of cameras and range imaging sensors, the NPBS method. In *IEEE Int. Conf. on Robotics and Automation*, pages 1552–1558, Nice France, May 1992.

[7] G. Champleboux, S. Lavallée, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3-D object localization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, Champaign, Illinois, June 1992.

[8] X. Chen. *Vision-Based Geometric Modeling.* PhD thesis, Telecom Paris, December 1992.

[9] P.-E. Danielson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[10] T.J. Fan, G. Medioni, and R. Neviata. Matching 3D objects using surface descriptions. In *IEEE International Conference on Robotics and Automation*, pages 1400–1406, Philadelphia, January 1988.

[11] O. D. Faugeras and Hebert M. The representation, recognition, and positioning of 3-D shapes from range data. In T. Kanade, editor, *Three-Dimensional Machine Vision*, pages 301–353. Kluwer Academic, 1987.

[12] A. Guéziec and N. Ayache. Smoothing and matching of 3-d space curves. In G. Sandini, editor, *Second European Conference on Computer Vision (ECCV'92)*, pages 620–629, Santa Margherita, Italy, May 1992. Springer Verlag, LNCS Series Vol. 588.

[13] P. J. Huber. *Robust Statistics*. John Wiley & Sons, New York, New York, 1981.

[14] B.A. Kall, P.J. Kelly, and S.J. Goerss. Comprehensive computer-assisted data collection treatment planning and interactive surgery. In *SPIE, Medical Imaging, Vol. 767*, pages 27–35, 1987.

[15] B. Kamgar-Parsi, Jones J.L., and Rosenfeld A. Registration of multiple overlapping range images : scenes without distincive features. *IEEE Trans. on Pattern Anal. Machine Intell.*, 13:857–871, 1991.

[16] R. Krishnapuram and D. Casasent. Determination of three-dimensional object location and orientation from range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1158–1167, November 1989.

[17] S. Lavallée, R. Szeliski, and L. Brunie. Matching 3-D smooth surfaces with their 2-D projections using 3-D distance maps. In *SPIE Vol. 1570 Geometric Methods in Computer Vision*, pages 322–336, San Diego, CA, July 1991.

[18] J.C. Lee and E. Milios. Matching range images of human faces. In *Third Int.Conf. on Computer Vision*, pages 722–726, Japan, 1990.

[19] C.A. Pelizzari, G.T.Y. Chen, D.R. Spelbring, R.R. Weichselbaum, and C-T. Chen. Accurate 3-D registration of CT, PET, and-or MR images of the brain. *J. Computer Assisted Tomography*, 13(1):20–26, 1989.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, second edition, 1992.

[21] G. M. Radack and N. I. Badler. Local matching of surfaces using a boundary-centered radial decomposition. *Computer Graphics and Image Processing*, 45:380–396, 1989.

[22] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, Massachusetts, 1989.

[23] C. Schiers, U. Tiede, and K.H. Hohne. Interactive 3D registration of image volumes from different sources. In H.U. Lemke, editor, *Computer Assisted Radiology, CAR 89*, pages 667–669, Berlin, June 1989. Springer-Verlag.

[24] R. Szeliski and S. Lavallée. Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. In *SPIE Vol. 2031 Geometric Methods in Computer Vision II*, San Diego, CA, July 1993.

[25] K.D. Toennies, J.K. Udupa, G.R. Herman, Wornom I.L III, and Buchman S.R. Registration of 3-d objects and surfaces. *IEEE Computer Graphics and Applications*, 10(3):52–62, 1990.