

# Free-form 3D object reconstruction from range images

C. Schütz, T. Jost, H. Hügli

Institute for Microtechnology  
University of Neuchatel, rue Breguet 2  
CH-2000 Neuchatel, Switzerland  
email: christian.schutz@imt.unine.ch  
fax: +41 32 718 34 02

## Abstract

The increasing use of virtual object representations for various applications creates a need for fast and simple object digitizing systems. Range finders provide a convenient way to digitize solid objects and permit the accurate and fast scanning of an object shape without any probe contact. However, only one view of an object can be captured at once and therefore for most objects several views have to be combined in order to obtain a description of the complete surface. We consider a digitizing system which captures and triangulates views of a real world 3D object and finally registers and integrates them. Registration is based on geometric matching and uses an interactively entered pose estimate. Integration is performed by a new fusion algorithm proposed in this paper. This algorithm takes advantage of the previous view registration to remove the redundant overlap area of two views and to fuse together their respective meshes by a gap filling algorithm. The fusion algorithm integrates well in the whole reconstruction process and is simple and successful.

**Keywords:** 3D object modeling, surface registration, virtual reality, range imaging

## 1. Introduction

There exists a high demand for digitizing systems which allow to generate a virtual model from a real world object. Applications such as teleshopping, rapid prototyping and 3D object recognition [SCHb] need a virtual representation of the 3D geometry of an object. Therefore, 3D surface digitizers get used more and more since the model construction with a standard modeler is a quite tedious task especially for objects of arbitrary shape.

Range images provide a direct access to the 3D geometric information of object surfaces. Current range finders allow an accurate digitizing of an object surface at low cost and high speed. Since most objects self occlude, one acquisition captures only a subpart of the entire object surface. Therefore, there exists a need to combine several object views into one unique object representation.

The registration of such views becomes straightforward if the object is placed for instance on a turn table where the relative transformation between the views is well defined by the rotation angle of the turn table. However, especially for objects with concave surface regions, additional views are needed to scan the complete object surface. To access self occluded regions, a more sophisticated mechanical system or visual markers may help to determine the transformations between the different object poses.

To avoid such measuring systems, several authors [BES] [TUR] proposed to use the object surface geometry to register two surfaces. An iterative closest point matching algorithm establishes the correspondence between the common parts of the two surfaces and minimizes a distance error. This geometric matching algorithm needs a good initial estimate in order to converge towards the best solution [HUG]. Therefore, we consider an interactive interface based on stereo display and space mouse that allows the user to roughly align the two surfaces before the automatic precise registration is started.

Once the surfaces are matched, the next step is the surface fusion. Several systems have been proposed to integrate range data from different object acquisitions. Early approaches triangulate the measured point set in 3D space and iteratively sculpt the surface polyhedron [BOI]. More recent approaches (see [PIT] for an overview) avoid the expensive 3D space triangulation and integrate meshes obtained from triangulation in 2.5D using the range image neighborhood relationships. They mainly differ in how they treat the redundant overlapping zone of the registered surfaces from two different acquisitions. Different methods for the erosion of the overlapping area and the fusion of the remaining triangle meshes have been proposed [PIT] [TUR].

This paper proposes a new mesh fusion algorithm which differs from other approaches by that it uses the closest point information from the geometric matching to detect the frontiers of the overlapping surface areas. The redundant part of one surface is removed and the remaining mesh is attached to the second surface mesh by a gap filling algorithm. Compared to others, the algorithm is simpler and integrates well in the complete reconstruction process.

Section 2 presents the overall architecture of the 3D digitizing system and the user interface. Its different modules are explained in the following sections and finally some results of reconstructed free-form objects using range images are shown.

## 2. System architecture

This section describes the different elements the proposed digitizing system consists of. It consists of two blocks: view digitizing and view integration. The view digitizing block generates a virtual view of the observed object surface. The view integration block iteratively integrates each new virtual view in the virtual model under construction.

The view digitizing block measures the points of the visible object surface, completes zones with missing data and triangulates the surface points. The resulting output represents a virtual view of the real object. The view digitizing block includes the acquisition, the hole filling and the view triangulation module. Section 3 explains the detailed implementation of these modules.

The view integration block combines the virtual views into one virtual object becoming an entire model of the real object. The view integration block is composed of the interactive pose estimation, the automatic registration and the mesh fusion modules where the first two modules represent the view registration block. The view registration block aligns the different views and the mesh fusion module recombines the individual triangle meshes into a new global one covering the union surface of the single meshes. Section 4 presents the different methods used for these modules.

The proposed system allows an incremental construction of the virtual model. Fig. 2.1 gives an overview over the modules and the data flow during the digitizing process.

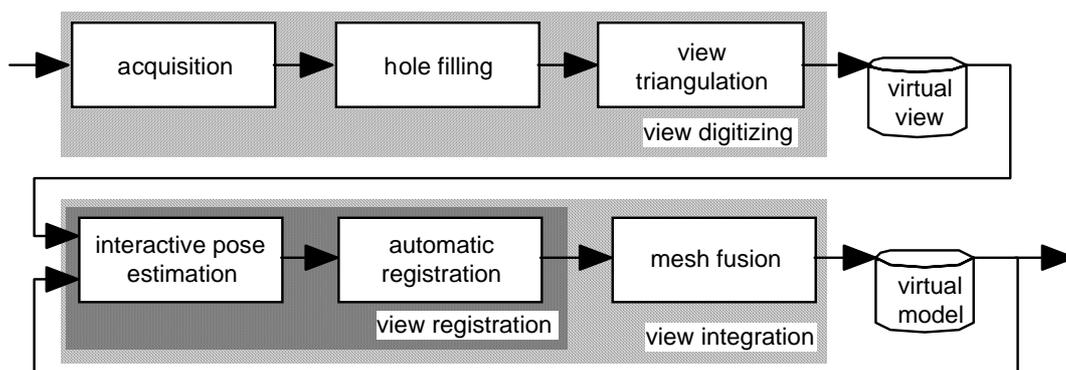


Fig. 2.1 Digitizing system architecture

All modules can be controlled by the same user interface shown in Fig. 2.2. A 3D cursor allows to select the different virtual objects. Their pose can be changed with a space mouse. The acquired data is rendered on a fast 3D graphics work station which updates the virtual scene at screen refresh rate.

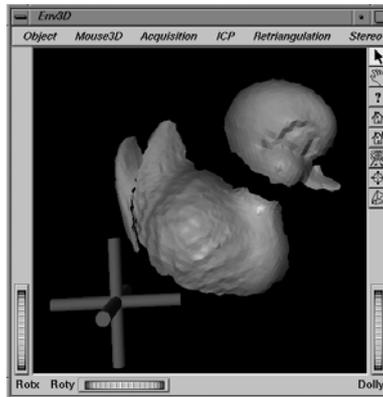


Fig. 2.2 Control interface of the digitizing system

The object data is organized in an indexed face set list which is used by the display routines and the data manipulation algorithms. The use of a common data structure reduces memory overhead and keeps data conversion at a minimum.

### 3. View digitizing

#### 3.1 Acquisition

There exist several methods to sense the three dimensional geometry of a real world object. They range from the classic contact probe over the visual marks used together with photogrammetry to new range finders which give access to accurate and fast scanning of an object shape without any need of contact or marks. Range finders use several principles to sense depth as for example stereo, triangulation, focus and others. In our laboratory, the geometry of the object surface is acquired by a range finder produced by ABW in Germany and working on the principle of space coding with projected stripe pattern and triangulation. This range finder measures scenes without moving parts. The measurement resolution is about 0.5mm. The coordinates measurements are arranged in a two dimensional array corresponding to the CCD camera image of the range finder and may be visualized as a range image where the pixel intensity corresponds to the camera-object distance as shown for a duck toy in Fig. 3.1.

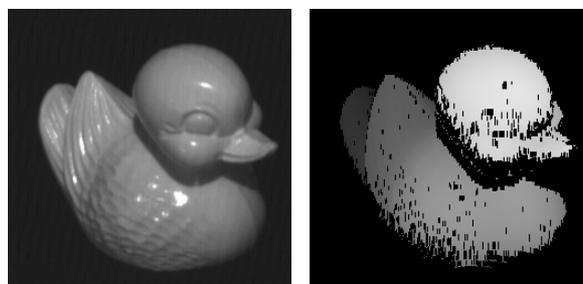


Fig. 3.1 Intensity and range images of a duck toy

#### 3.2 Hole filling

The black pixels in the range image represent missing data (holes). They are found where the measurement confidence is low and are often caused by shadow regions or failed stripe coding. A hole filling module removes them by performing an algorithm that fills small regions. A dilation

transforms the range image as follows: If a non-valid point has more than three valid points in its eight neighborhood the mean value of the valid neighbors is copied in the dilated range image; Otherwise the unchanged point value is copied in the dilated image. Typically the dilation is repeated for six iterations.

The results on a real object, shown in Fig. 3.2, present the successful filling of the small surface holes. Large regions with missing data remain open since their interpolation would not result in an accurate approximation of the real object surface.

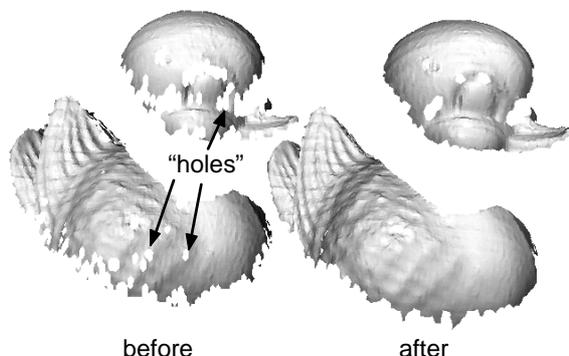


Fig. 3.2 Real object showing the results of the hole filling method

### 3.3 View triangulation

After the hole filling process, the measured surface points are triangulated. Since the measurements are ordered in a regular grid namely the range image the triangulation of the surface becomes straightforward as proposed by [RUT]. The range image is traced from the upper left to the lower right corner and a local triangulation is performed for every pixel. The local triangulation algorithm creates two triangles covering the square grid mesh formed by the points  $P_{i,j}$ ,  $P_{i+1,j}$ ,  $P_{i,j+1}$  and  $P_{i+1,j+1}$  where  $i$  and  $j$  are the row and column indices of the current position in the range image. If the four points represent valid data there exist two possible ways to triangulate the square depending on which diagonal is selected. Following the principle of the Delaunay triangulation, the shortest diagonal is selected which creates triangles with a maximal size of the smallest angle. This results in a smooth surface approximation since there are no triangles with long edges. If one of the four points is not valid then one triangle is constructed with the remaining three points and no triangle is build at all if more than one point is missing.

For several applications, the full range image resolution is not necessary and a subsampling of the rows and columns by a factor  $r$  allows to reduce the number of points and to process the data faster. This data reduction can be done easily by increasing the local triangulation mesh by a factor  $r$ , typically equal to 2 or 4.

The range finder senses also the intensity information which is assigned to the vertices of the triangles. This gives an approximate representation of the object texture if the mesh is fine enough. Fig 3.3 shows the triangulated surface of one view of a rabbit toy triangulated with reduction  $r = 2$ .

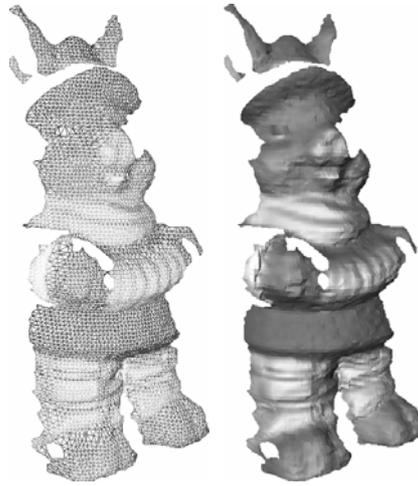


Fig. 3.3 View triangulation results for a rabbit toy in wire frame and rendered view

Checking the validity of the range points is not sufficient to avoid bad triangles. Other authors [TUR] [RUT] showed that additional checks are necessary to avoid the connection of range points separated by a discontinuity step in the range image. Points which are next to one another in the range image are not necessarily neighbors on the object surface as illustrated by Fig. 3.4.

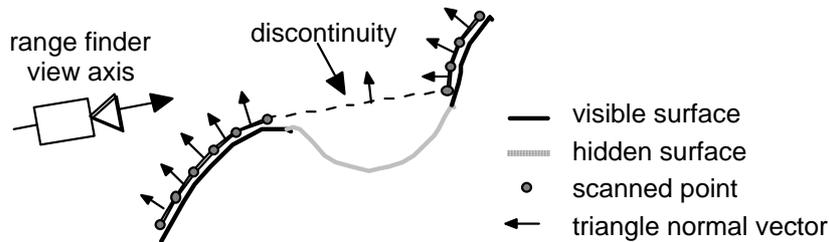


Fig. 3.4 Triangle size and orientation check

Therefore, we have to ensure that occluded parts are not covered by triangles. First, only triangles with edges smaller than  $4 \cdot s \cdot r$  are kept, where  $s$  is the sampling grid distance which is about 0.5mm for our range finder and  $r$  the reduction factor introduced before. Second, triangles which angle between the triangle normal vector and the sensor view axis exceeds  $75^\circ$  are rejected.

Finally, the triangulated virtual view is stored in a triangle mesh format common to data visualization and manipulation algorithms. It consists of the vertex coordinates, vertex intensities, face indices and face normals.

#### 4. View integration

Adding a new view to the virtual model under construction requires first the view to be registered to the model and then their meshes to be fused together.

##### 4.1 View registration

As stated in the introduction, the scanning process is kept as simple as possible and allows an operator to place the real world object in any stable pose on the acquisition field. Therefore, the transformation between the reference frames of a new acquisition and the virtual model is not known a priori and has to be determined in a first step. Since no external measurements of the object pose are available, we have to rely on the object surface characteristics in order to register it with the virtual model. This assumes that the virtual model and the new object view have at least some surface parts in common which allow to establish correspondences between them. Therefore, view registration is performed in a two step process: interactive rough pose estimation and then

automatic precise registration.

#### 4.1.1 Interactive rough pose estimation

Since human perception easily identifies corresponding surface parts for any object type and shape we use an interactive graphic interface that permits an operator to enter a pose estimate for the two objects to be aligned. Both the virtual model and the new view are rendered in 3D and can be manipulated in all six degree of freedoms using a space mouse as input device. However, even a sophisticated object rendering and pose manipulation hardware is not sufficient to align the objects precisely. In fact, there is no measure apart from the visual feedback indicating the quality of the surface matching. Therefore, the interactive interface only provides a rough pose estimate to be used as a hint for the automatic precise registration. Fig. 4.1 shows an example of two roughly aligned surfaces used as starting configuration for the automatic registration.



Fig. 4.1 Roughly aligned views of a cat toy

#### 4.1.2 Automatic precise registration

Besl proposed a surface registration algorithm called ICP [BES]. This geometric matching algorithm registers two surfaces starting from an initial pose estimate. The algorithm proceeds iteratively. First, it pairs every point of one surface called P with the closest point of an other surface called X. These pairs of closest points are used to calculate the rigid transformation ( $\mathbf{R}$ ,  $\mathbf{t}$ ), which minimizes their mean square coupling distance or error. The surface P is then translated and rotated by the resulting transformation and the algorithm starts again with the closest point coupling. This algorithm has been shown to converge fast but not necessarily towards the optimal solution. A good starting configuration is preliminary to a successful convergence. However, as the range of successful starting configurations is quite large (see [HUG] and Fig. 4.1) this does not impose to much constraints to the operator when entering a pose estimate.

In our implementation of the ICP algorithm, vertices of the surface P are coupled with points inside the triangles of the surface X. This results in a more accurate matching compared to a matching limited to the triangle vertices only.

In the original algorithm, one surface is a subpart of the other which is not the case in our application where both surfaces contain data not present in the other. The ICP algorithm needs therefore to be modified as proposed by Turk [TUR]. Closest points which are too far apart are not considered to be corresponding points and are therefore not coupled. The calculated closest points couplings are therefore weighted as follows:

$$w_k = \begin{cases} 1 & d_k < (c \cdot s \cdot r)^2 \\ 0 & \text{else} \end{cases} \quad \text{with } d_k = \|\mathbf{p}_k - \mathbf{x}_k\|^2 \text{ and } k \in [1, \dots, N_p] \quad \text{Eq. 4.1}$$

which results in the modified error minimization

$$e(\mathbf{R}, \mathbf{t}) = \frac{1}{W} \sum_{N_p} w_k \|\mathbf{R}\mathbf{p}_k + \mathbf{t} - \mathbf{x}_k\|^2 \quad \forall \text{ pair } (\mathbf{p}_k, \mathbf{x}_k) \text{ and } W = \sum_{N_p} w_k \quad \text{Eq. 4.2}$$

This modification assigns the weight zero to invalid couplings as shown in Fig. 4.2. The decision threshold for a valid coupling square distance is set to the product  $(c \cdot s \cdot r)^2$  where  $s$  equals the sampling distance and  $r$  equals the reduction rate both introduced in section 3.3. The constant  $c$  allows to control the convergence and the precision of the matching.

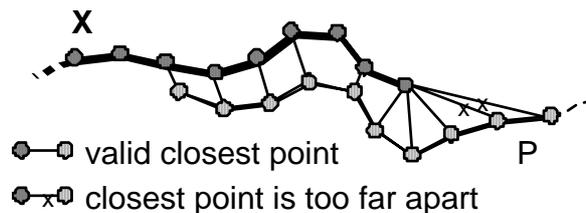


Fig. 4.2 Closest point couplings for two surfaces

Experiments on several objects showed that the modified ICP algorithm converges in about 50 iterations. As mentioned before, the two surfaces should have enough common data points. 30 to 50 percent of common surface has been observed to be a good amount. Fig. 4.3 shows the same surfaces as in Fig. 4.1 after the execution of the automatic matching.



Fig. 4.3 Registered surfaces from a cat toy

In order to verify the registration quality and to stop the iteration a pertinent measure is needed. The minimization error  $e$  corresponding to the mean of the square distances is a measure generally used to qualify the matching. Another statistical measure which has been used successfully to qualify matched surfaces [SCHa] is the deviation of the square distances indicating the matching regularity. Both values should be as low as possible. But, this may lead to a solution where only very few points are coupled. In order to avoid such cases, matchings with a high number of coupled points on the surface P are selected, as proposed by Krebs [KRE].

## 4.2 Mesh fusion

There exist several methods to integrate registered surfaces acquired from different views [TUR] [RUT] [PIT] [SOU] [HIL]. They differ mainly in how they treat the redundant overlapping zone of the two registered surfaces and can be separated into two groups presented in Fig. 4.4: partial erosion and complete retriangulation of the surface points.

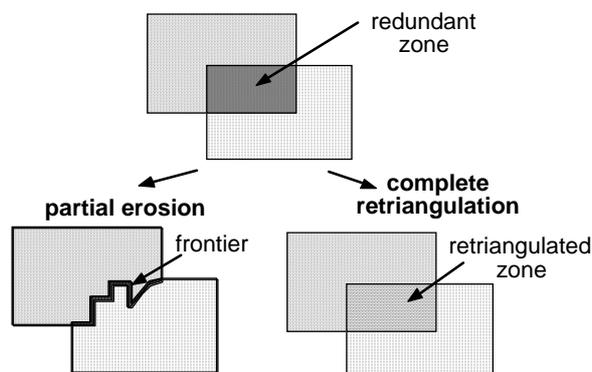


Fig. 4.4 Different approaches to integrate object views

Methods using a partial erosion approach [PIT] [TUR] erode the overlapping surfaces until the overlap disappears. The two triangle meshes are then recombined at their frontiers in order to have one unique mesh for the union of the two surfaces. Other authors [HIL] [SOU] [RUT] discard the mesh information from the triangulated views if calculated at all and retriangulate the overlapping zone or even the complete point set.

Since the object views can be easily triangulated using the range image structure as shown in section 3.3, we opt for the partial erosion approach which keeps intact as much as possible of the triangle mesh structure. We propose a new mesh fusion algorithm that benefits from the closest point relationships established during the geometric matching. There is no need to run an extra routine to erode overlapping surfaces and to detect the surface frontiers as done in other work [PIT] [TUR].

The following features characterize the proposed mesh fusion algorithm. We refer to the same surfaces P and X as introduced for the geometric matching, where the vertices on P are coupled with points on the triangles of X. The following steps give an overview of the different steps during

the mesh fusion:

- 1) **overlap detection:** The registration algorithm calculates for the vertices on surface P the closest points on the triangles of the surface X. Closest points with an euclidian distance below a defined threshold are coupled. During the geometric matching iterations, the coupled points on P converge to the overlapping area of the two surfaces P and X.
- 2) **overlap remove:** The redundant part of the surface P is deleted by removing the triangles with one or more coupled vertex. The remaining meshes are separated by a gap defined by a frontier on P and X.
- 3) **frontier detection:** Triangles where the geometric matching coupled only one vertex are connected to the frontier on P. Actually, the two non-coupled vertices build an edge of the frontier on P. The frontier on X is detected by a closest point search.
- 4) **gap filling:** The gap enclosed by the two frontiers is filled iteratively with triangles. Vertices on the two frontiers are used as candidates to build a filling triangle. The triangulation does not need projection into tangential planes which allows a correct triangulation of sharp edges. Triangles with a maximal opening angle are constructed in order to have the best surface approximation.

The implementation details are discussed below and illustrated by examples shown in the figures Fig. 4.5 and Fig. 4.6.

### Overlap detection

The closest point routine of the automatic registration module marks the vertices on surface P which overlap surface X. To ensure that only close points are marked the constant  $c$  of the coupling distance threshold in Eq. 4.1 is set to a small value during the last matching iterations. This results in a set  $C_V$  of coupled vertices with  $C_V = \{p_k \in P | w_k = 1\}$  and therefore  $C_V \subseteq P$ . The points member of  $C_V$  are marked in the left drawing in Fig. 4.5.

### Overlap remove

The remove process eliminates all the vertices member of  $C_V$  from the surface P. The clipped surface  $P_C = \{p_k \in P | p_k \notin C_V\}$  is separated from the surface X by a gap of about the size of the distance threshold used in Eq. 4.1 since this is the maximum distance for a vertex to be considered for the overlap. The resulting surfaces are shown in the right graph in Fig. 4.5.

### Frontier detection

A frontier is defined by the frontier list F which contains the ordered vertices of the surface  $P_C$  which limit the gap created by the above overlap remove process. The list F is build as follows:

During the geometric matching the list  $T_F = \left\{ \mathbf{t}_l = \{v_{l,0}, v_{l,1}, v_{l,2}\} \mid v_{l,i} \in P \text{ and } \sum_{i=0}^2 w_{l,i} = 1 \right\}$  containing the triangles with only one coupled vertex is established. For every triangle in  $T_F$ , the vertices which are not coupled are inserted to  $F = \left\{ v_{l,i} \in P \mid v_{l,i} \in \mathbf{t}_l \text{ with } \mathbf{t}_l \in T_F \text{ and } w_{l,i} = 0 \right\}$ . The order of the vertices is clockwise and defined by the normal vector of the triangle which an edge is part of. Such a frontier list is established for every frontier. The points member of F are marked in the right drawing in Fig. 4.5.

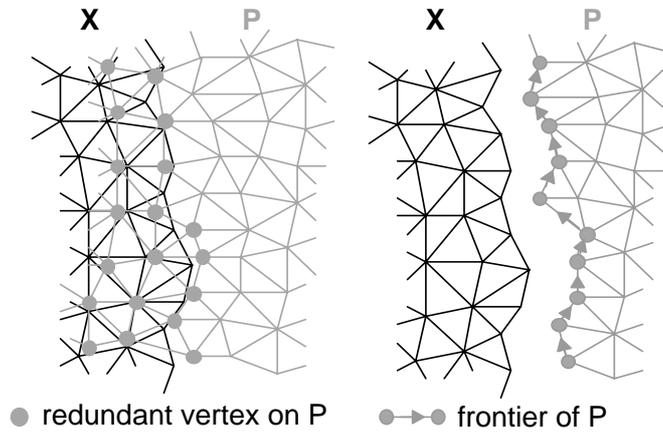


Fig. 4.5 Remove of the redundant overlap area and frontier detection

### Gap filling

The gap between the two surfaces  $X$  and  $P$  is filled with triangles in order to join the two meshes. The different frontiers on  $P$  delimiting these gaps are processed sequentially. The filling process is initialized for a frontier on  $P$  with the search of the first vertex  $\mathbf{x}_N$  on the frontier of  $X$ . To do so, the first two vertices  $\mathbf{f}_0$  and  $\mathbf{f}_1$  of the frontier list  $F$  are selected and the nearest point  $\mathbf{x}_N$  to  $\mathbf{f}_0$  and  $\mathbf{f}_1$  on the frontier of  $X$  is calculated. The nearest point to  $\mathbf{f}_0$  and  $\mathbf{f}_1$  is the point with the minimal sum of the distances  $\|\mathbf{x}_N - \mathbf{f}_0\|$  and  $\|\mathbf{x}_N - \mathbf{f}_1\|$ . The point  $\mathbf{x}_N$  is on the surface frontier of  $X$  if the list of the vertices member of the triangles connected to  $\mathbf{x}_N$  is not empty after eliminating duplicates. Once the first vertex  $\mathbf{x}_N$  of the frontier on  $X$  is determined, the first bridge triangle joining the two frontiers is constructed with the vertices  $\mathbf{x}_N$ ,  $\mathbf{f}_0$  and  $\mathbf{f}_1$  as shown in the left graph in Fig. 4.6. The frontier list  $F$  is updated by setting its first vertex  $\mathbf{f}_0$  equal to  $\mathbf{x}_N$ .

The following algorithm fills the gap iteratively starting with the above initialization. Two candidate vertices are selected to build the next bridge triangle. One is  $\mathbf{f}_2$ , the third vertex in the frontier list  $F$  and the other one is  $\mathbf{x}_c$ , the next vertex in counter-clockwise order on the frontier of  $X$ . These two candidates form together with the vertices  $\mathbf{f}_0$  and  $\mathbf{f}_1$  of  $F$  the next potential bridge triangles as shown in the right graph in Fig. 4.6. The candidate which encloses the maximal angle is selected in order to obtain a regular triangulation. The frontier list  $F$  is updated with the new vertices as follows:  $\mathbf{f}_0$  is set equal to  $\mathbf{x}_c$  if  $\mathbf{x}_c$  is chosen or  $\mathbf{f}_1$  is removed from  $F$  if the candidate  $\mathbf{f}_2$  is selected. The candidate selection starts again with the modified frontier list and the above procedure is applied until  $F$  contains only two vertices.

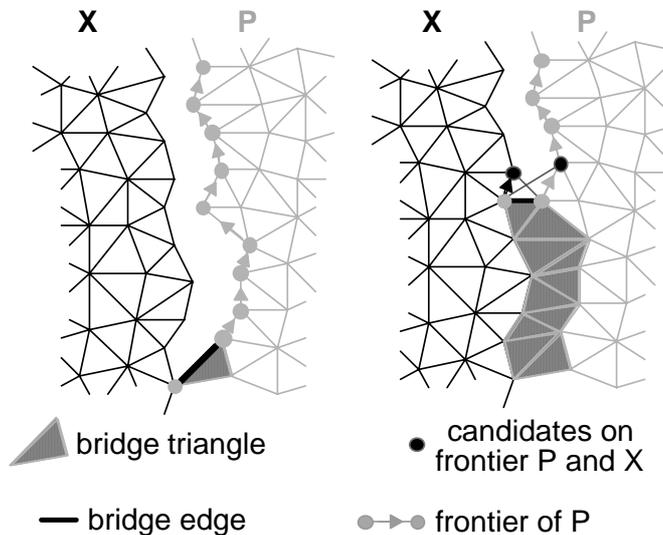


Fig 4.6 Gap filling initialization and iteration procedure

The following deteriorations of the gap filling algorithm have to be avoided. First, if the frontiers of the two surfaces diverge which results in a large gap or second if the bridge triangle normal is negative indicating a filling in the wrong direction. In these cases new candidates are calculated, the filling process is initialized with the next edge from the edge list or the filling is stopped.

## 5. Results

The described 3D object reconstruction system has been implemented and used to create virtual models from several real world objects. Results are presented here for two toy objects. Six views have been used to create the cat toy model where ten views have been merged to obtain the rabbit model. The cat object consists of 9700 points and about 19000 triangles whereas the rabbit object contains 15000 points and 30000 triangles. The successful reconstructed objects are shown in Fig. 5.1.



Fig. 5.1 Digitized cat toy and rabbit toy with texture

The range finder assigns an object color to every vertex. If the triangle mesh is fine enough the object texture is maintained, as shown for a rabbit toy in Fig. 5.1.

As discussed above, the object views to be assembled need common surface parts with enough geometric structure in order to allow the automatic matching to converge to a stable solution. For example for a box, a view should contain at least three faces in order to find a stable alignment of two views. The proposed digitizing system is especially suited for objects of complex free-form shape.

## 6. Conclusions

The presented digitizing system constructs models of free-form 3D objects. It successfully registers and integrates object views acquired by a range finder. It combines interactive and automatic modules in order to provide maximal flexibility. The proposed system has been designed to have a minimal number of parameters to tune and a maximal flexibility in order to digitize objects of any shape. Only the closest point thresholds in the automatic matching and mesh fusion module have to be tuned.

The object views are acquired with a range finder and triangulated using the range image data structure. Geometric matching registers the roughly aligned surfaces at high precision since it establishes point to triangle correspondences. The integration uses a newly proposed algorithm which is easy to implement and which integrates well in the context of the whole digitizing system. It consists of a new overlap remove and a gap fill module. Several objects have been successfully digitized with the presented system.

## 7. Acknowledgments

This research has been funded by the Swiss national science foundation under project number 2100-43530.

## 8. References

- [BES] P.J. Besl and N.D. McKay, "A Method for Registration of 3-D Shapes," Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 14, no. 2, pp. 239-256, 1992.
- [BOI] J.-D. Boissonnat, "Geometric Structures for Three-Dimensional Shape Representation," ACM Transactions on Graphics, vol. 3, no. 4, pp. 266-286, 1984.
- [HIL] A. Hilton, A.J. Stoddart, J. Illingworth and T. Windeatt, "Marching triangles: range image fusion for complex object modelling," IEEE International Conference on Image Processing, Lausanne, vol. 2, pp. 381-384, 1996.
- [HUG] H. Hugli, Ch. Schutz, "How well performs free-form 3d object recognition from range images?," Intelligent Robots and Computer Vision XV, Algorithms, Techniques, Active Vision and Materials Handling, SPIE, Boston, Vol. 2904, 1996, pp. 66-74.
- [KRE] B. Krebs, P. Sieverding and B. Korn, "Correct 3D Matching via a Fuzzy ICP Algorithm for Arbitrary Shaped Objects," DAGM96 - Mustererkennung, Heidelberg, 1996.
- [PIT] R. Pito, "Mesh Integration Based on Co-Measurements," IEEE International Conference on Image Processing, Lausanne, vol. 2, 1996, pp. 397-400.
- [RUT] M. Rutishauser, M. Stricker, M. Trobina, "Merging Range Images of Arbitrarily Shaped Objects," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, 1994, pp. 573-580.
- [SCHa] Ch. Schutz, H. Hugli, "Towards the recognition of 3D free-form objects," Intelligent Robots and Computer Vision XIV, Algorithms, Techniques, Active Vision and Materials Handling, SPIE, Philadelphia, Vol. 2588, pp. 476-484, 1995.
- [SCHb] Ch. Schutz, N. Natonek, Ch. Baur, H. Hugli, "3D Vision in a Virtual Reality Robotics Environment," Proceedings of SPIE Telemanipulator and Telepresence Technologies, Boston, vol. 2901, 1996.
- [SOU] M. Soucy, D. Laurendeau, "A General Surface Approach to the Integration of a Set of Range Views," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 4, 1995, pp. 344-358.
- [TUR] G. Turk, M. Levoy, "Zippered Polygon Meshes from Range Images," Proceedings ACM Siggraph, Orlando, 1994, pp. 311-318.