Improving Segmentation Results by Studying Surface Continuity

Angel D. Sappa

Research and Development Department
Z + F UK Limited
Manchester - United Kingdom
angel.sappa@zf-uk.com

Abstract

This paper presents a process to improve the quality of range image segmentation by using geometrical relationships. The proposed technique consists of studying the surface continuity of an automatically generated surface model. Generally, surfaces are extracted independently (e.g., by means of a region growing algorithm) thus information about their connectivity is lost. Assuming that in most of the cases a surface cannot be disconnected with the others present in the given scene, occluded areas and crease edges can be recovered. Occluded regions are recovered by connecting surfaces that are represented by the same parameters. In addition, enforcing geometrical constraints, such as surface intersections, crease edges are recovered improving significantly the final model. Experimental results with automatically segmented real range images are presented.

1. Introduction

Range image segmentation algorithms are intended to extract the information from the surfaces contained in a given range image in order to reproduce the objects present in the scene. When complex scenes are considered an additional criterion, such as geometrical relationship, can improve significantly the obtained result. In this sense, algorithms for recovering occluded surfaces have been proposed recently in the literature (e.g., [1], [2], [3], [4]). In [1], the main objective has been to reconstruct surfaces behind occluding objects, producing visually acceptable reconstructions. That algorithm is carried out first by finding discontinuities and then detecting and studying the occluded areas. Others approaches to improve the segmentation result have been proposed in the reverse engineering field (e.g., [2],[3]). They consist of extracting a boundary representation from the segmented range images. Thus, edge points are grouped and merged generating new point positions used to define the sought edges. Although the obtained results are promising, the main problem of using points is that they are very sensitive to the noise. Finally, in [4] a technique for quality enhancement of reconstructed 3D models has been proposed. In that work orthogonality and parallelism of walls are exploited.

Different from these approaches, the current work focus the attention in the concept of *surface continuity* to improve the quality of the final model. It implicitly include two objectives, the occluded surface and crease edge recovering. The proposed algorithm explodes constraints or relationships between neighbour surfaces. It is implemented by using the surface parameters only, any consideration about edges—*jump or crease*—or edge point positions has to be done. This algorithm is devoted to handle with structured environments such as industrial facilities or buildings in general. In the first case mainly occluded surfaces are recovered while in the second case crease edges. In order to understand the problem, Fig. 1 shows an illustration in which occluded and unconnected surfaces are indicated.

The proposed algorithm consists of three stages. First a *region connectivity graph* is generated. This graph is used to represent the connectivity between regions and fill a *connectivity distance array.* The later allow us to obtain in a fast way the k nearest neighbours of a given region. The k nearest neighbours are taken into account during the second (occluding region recovering ($k > 1$)) and third (edge recovering ($k = 1$)) stages.

Section 2 describes the proposed technique. Experimental results by using automatically segmented real range images are presented in Section 3. Finally, conclusions and further improvements are given in section 4.

2. Surface continuity

The proposed algorithm assumes that a range image and its corresponding segmented model are given—in the

---

This work has been carried out as part of the *CAMERA* project (*CA*d *M*odelling of *B*uilt *E*nvironments from *R*ange *A*nalysis). *CAMERA* is an EC funded TMR network (ERB FMRX-CT97-0127).

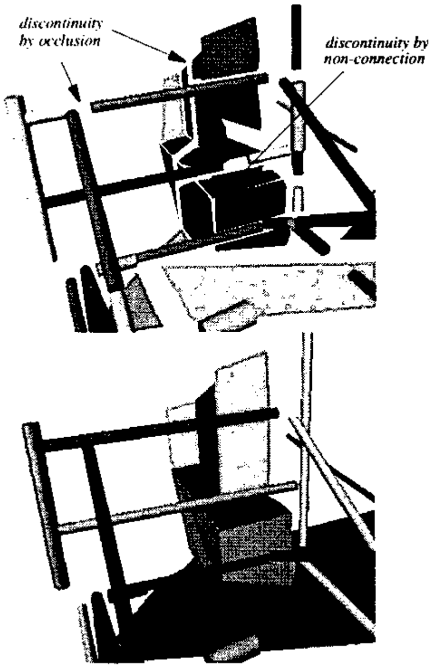


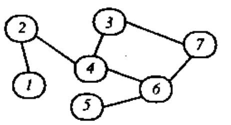
Figure 1. (*top*) Input 3D model (planes and cylinders) (*bottom*) Result after recovering occluded surfaces and crease edges.

current work the segmented models were generated by means of an hybrid technique [5], [6]. A model is represented by a set of planes and cylinders. A plane is defined by $N \cdot X = d$ where $(\cdot)$ is the dot product, N is the normal vector and d represents the distance to the reference frame. A cylinder is defined by (P_1, P_2, r), where (P_1, P_2) represent the points defining the axis and r represent the cylinder's radius.

Taking into account the previous assumption, the proposed algorithm consists of tree stages. First, the surfaces contained in the given model are considered as nodes of a graph. Edges in that graph link neighbours regions in the model. From that region connectivity graph a connectivity distance array is filled. In the second and third stages occluded surfaces and crease edges are recovered by testing candidate surfaces. These stages are described below.

2.1. Region connectivity graph generation

This first stage consists of generating a *region connectivity graph*. Regions contained in the given model will be represented as nodes in that graph. Each node has associated the surface parameters defining that region. Neighbour regions in the model—regions connected along one of its boundary in the range image—generate connected node in the graph. The neighbourhood relationship is easily obtained by considering the point positions into the 2D array of the given range image and the label associated with each one of those points. Thus, by comparing the label associated with each point with the labels associated with each one of its eight neighbour in the range image the connectivity between regions is obtained. Edges linking nodes in the graph have associated a unitary cost connection. These values are further used to fill a 2D array that represents the shortest distance—*path*—between two regions, *connectivity distance array.*



	1	2	3	4	5	6	7
1	0	1	3	2	4	3	4
2	1	0	2	1	3	2	3
3	3	2	0	1	3	2	1
4	2	1	1	0	2	1	2
5	4	3	3	2	0	1	2
6	3	2	2	1	1	0	1
7	4	3	1	2	2	1	0

Figure 2. (*left*) Illustration of a connectivity graph. (*right*) Connectivity distance array.

The aforementioned array is used to find the k nearest neighbour of a given region. It is filled in as follow. First, all the region relationship from the region connectivity graph is obtained. It means, cells corresponding to connected nodes in the graph are filled with a unitary value (if region i and j are connected in the graph $d_{(i,j)} = d_{(j,i)} = 1$, zero is assigned to symmetric cells $d_{(i,i)} = 0$). Next, by using this information the remaining cells are filled by means of an iterative process. It consists of going through all the rows of the array and propagating the cost connection to those regions already connected with the mentioned row. In order to explain it let us suppose the case of node number four in the illustration presented in Fig. 2(*left*). There, node number four is connected with nodes number two, three and six, thus a one is placed in row four columns two, three and six. The iterative propagation algorithm involves taken row number two and connecting it with regions three and six, in case of they are not already connected. The distance assigned to the cells (2,3) and (2,6) will be the value indicated in cells (4,3) and (4,6) increased in $d_{(4,2)}$ units (one unit in the current example). Thus, now in that illustration $d_{(2,3)} = 2$ and $d_{(2,6)} = 2$, in addition the corresponding symmetric cells are updated $d_{(3,2)} = 2$ and $d_{(6,2)} = 2$. After propagating the connectivity of region four to region two, the same process is carried out over rows number three and six. Next, the following row in the array is processed (row number five). After processing all the rows, this iterative process starts again with row number one, until all the cells are filled.

2.2. Occluded surface recovering

The first improvement step consists in recovering occluded surfaces. Given a region i, all the cells in the row number i, from the connectivity distance array, with a value higher that one and lower than some user defined

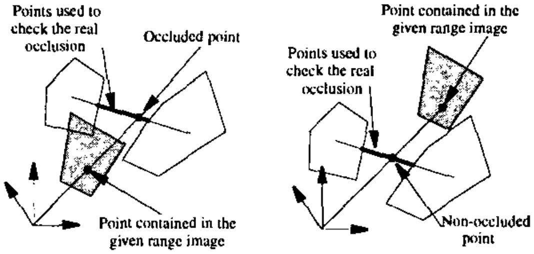


Figure 3. (*left*) Illustration of an occluded region. (*right*) Illustration of a niche.

threshold are considered (in the current implementation six was the maximum considered distance). Thus, going through all those cells that contains a value between the indicated interval, and defined by the same kind of surface than region *i*, the candidates to be joined with region *i* are obtained. As it was indicated before, only planar and cylindrical surfaces are contained into the segmentation result. Hence, two different cases have to be considered, region *i* is approximated by a plane or region *i* is approximated by a cylinder.

In the first case, region *i* will be joined with region *j* (which is also approximated by a planar surface) if: *1)* the angle between their normal vectors $\{N_i, N_j\}$ is below than ten degrees and *2)* $|d_i - d_j| < 0.1(d_i + d_j)/2$. In the second case, when both surfaces are cylinders, they will be merged together if: *1)* they have a similar orientation (ten degrees between their axes's orientation), *2)* $|r_i - r_j| < 0.1(r_i + r_j)/2$ and *3)* the distances between the points defining the axis of cylinder *i* and the line containing the axis of cylinder *j* is below to $l_i(r_i + r_j)/(l_i + l_j)$ where l_i represents the length of segment $\overline{P_{1_i}P_{2_i}}$ and l_j the length of segment $\overline{P_{1_j}P_{2_j}}$, this comparison is also examined in the other sense.

In both cases, when two surfaces (planes or cylinders) satisfy their corresponding criterion the points defining them are joined and refitted updating the surface parameters. However, before joining two surfaces into a single one, it is necessary to verify that the data points contained into the gap between them are nearest to the sensor than the current obtained new surface. It is, verify that the gap between them is a real occluded area and not a niche [1]. In order to proof that, a process carried out in the 2D array of the given range image is proposed. It consists of computing a straight line linking the centres of both surfaces. Points defining that straight line (obtained by applying Bresenham's algorithm [7]) that does not belong to neither region *i* nor *j* are used to test the occlusion. The occlusion involves computing the 3D distance between these points to the reference frame. These straight lines, linking the reference frame with the Bresenham's points, also will intersect the new computed surface (surface resulting from merging region *i* with region *j*). Thus, if the distances between the points of the given range image (selected by the Bresenham's algorithm) to the reference frame is smaller than the distance from the corresponding point obtained from the intersection with the new surface, the obtained surface is a really occluded one. On the contrary the computed surface is discarded and the regions are not joined because there is a niche between them (see illustration in Fig. 3).

In case of the previous criterion is satisfied, that is there is an occluded area between the considered surfaces, the gap between them has to be filled. This is easy in the case of the merged surfaces are cylinders. A single cylinder covering both merged regions and the present gap is obtained directly from the fitting function (parameters (P_1, P_2, r), axis defined by starting and ending points and radius). On the contrary, in case of the merged surfaces are planes, it is not trivial to define the boundary of the plane contained in the occluded region. In the current implementation, the straight line obtained before by using the Bresenham's algorithm is used to define two new lines connecting the regions. These two lines are placed upper and below the previously computed line. As the same as before, the points defining these lines are used to test if they are really occluded areas or not. In case that all the points defining one of these lines satisfy that criteria the points over the plane are considered as members of the occluded surface, and a new line (upper to the accepted line in case that the last one was the upper line or below in the other case) is generated. The checking algorithm now is carried out over this new line. This algorithm is applied until no new line is included.

Fig. 1 (*top*) gives an example of the 3D model used as input (segmentation result from an hybrids segmentation algorithm). In that image are shown different cylinders belonging to the same pipeline and planar patches belonging to the same surface unconnected due to occlusions. Fig. 1 (*bottom*) presents the result obtained after recovering occluded surfaces. In addition, in that illustration are shown some discontinuities originated by unconnected neighbour surfaces (surfaces obtained independently). They were recovered by using the technique explained below.

2.3. Crease edges recovering

Surfaces obtained by a segmentation algorithm generally are not connected. Then, simple assumptions—such as surfaces merging together along an intersecting line—produce significant improvements in the final result. The edges are obtained by studying connectivity between neighbour regions, as the same as before, the connectivity distance array is employed but now only cells with a unitary value are considered. At the present only planar regions were studied. This stage consists of recovering the crease edges by computing the intersection between

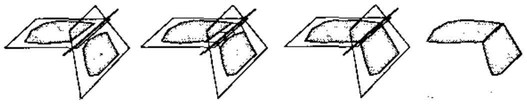


Figure 4. Illustration of the steps to recover a crease edge.

consecutive planes. Assuming that two planes are not parallels ($N_1 \times N_2 \neq 0$) and are defined as:

$$\begin{aligned} N_1 \cdot X &= d_1 \\ N_2 \cdot X &= d_2 \end{aligned} \qquad (1)$$

the intersection is a line which can be written as:

$$X = aN_1 + bN_2 + u(N_1 \times N_2) \qquad (2)$$

where ($\times$) represent the cross product and u is the parameter of the line. Taking the dot product of (2) with each one of (1) the a and b parameters can be obtained as:

$$\begin{aligned} a &= (d_1 N_2 \cdot N_2 - d_2 N_1 \cdot N_2)/\mathrm{Det} \\ b &= (d_2 N_1 \cdot N_1 - d_1 N_1 \cdot N_2)/\mathrm{Det} \\ \mathrm{Det} &= (N_1 \cdot N_1)(N_2 \cdot N_2) - (N_1 \cdot N_2)^2 \end{aligned}$$

In this way the intersection line is obtained, this line will contain the sought crease edge. Now it is necessary to project the boundary of the surfaces over that line in order to reach the surface continuity. First, in order to speed up the process, from the points used to fit each planar surface, their boundaries are extracted. From these two sets of points the distances to the intersection line is computed. Those points that have a distance lower than a given threshold are projected over the intersection line (minimum distance from the point to the line). The extremity points contained in the intersection line will define the crease edge segment (starting and ending points). This process is applied to every pair of neighbour planar regions. Fig. 4 gives an illustration of the different steps of the proposed technique.

3. Experimental results

Several structured environments have been considered to test the proposed technique. In all the cases a range image and its automatically segmented representation (3D model obtained from the real range data) are given as inputs. The CPU time to compute the different stages have been measured on a Pentium III, 1 GHz processor. Fig. 1 (*top*) presents a 3D model of a scene containing 52 regions. Fig. 1 (*bottom*) shows the result obtained with the proposed technique. It contains 43 regions and was computed in 12.33 sec. Fig. 5 (*top*) shows in wireframe a 3D model of a indoor environment, it contains 36 regions. Fig. 5 (*bottom*) presents the resulting model obtained with the proposed technique, it also contains 36 regions and was computed in 7.37 sec. A difference with the previous example, here only crease edges are recovered because there is not occluded surfaces present in the given input model. Fig. 1 and Fig. 5 correspond to real scenes.

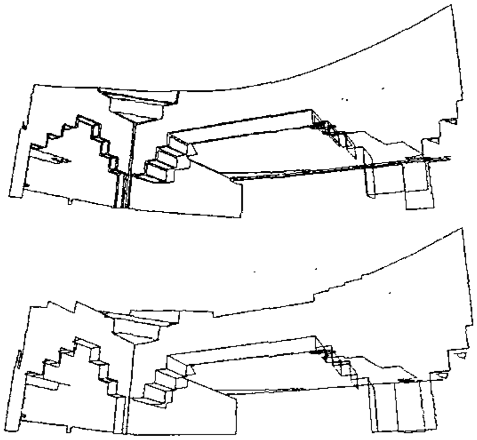


Figure 5. (*top*) Wireframe representation of the input model. (*bottom*) Result obtained with the proposed technique.

4. Conclusions and further improvements

A technique to improve the quality of 3D models has been presented. A difference with previous approaches it is based in the surface continuity assumption. Thus, occluded areas and crease edges are recovered generating an enhanced 3D model. At the present only plane intersections are considered during the crease edge recovering stage. An immediate extension to the current version will include the study of relationships between other kinds of surfaces (cylinder-cylinder, cylinder-plane).

5. References

[1] F. Stulp, F. Dell'Acqua and R. Fisher, "Reconstruction of surfaces behind occlusions in range images", *IEEE 3rd. Int. Conf. on 3-D Digital Imaging and Modeling*, Quebec, City, Canada, May-June 2001.

[2] B. Mills, F. Langbein, A. Marshall and R. Martin, "Approximate symmetry detection for reverse engineering", *6th. ACM Symp. Solid Modelling and Applications*, Ann Arbor, MI, USA, June 2001.

[3] F. Langbein, B. Mills, A. Marshall and R. Martin, "Finding approximate shape regularities in reverse engineered solid models bounded by simple surfaces", *6th. ACM Symp. Solid Modelling and Applications*, Ann Arbor, USA, June 2001.

[4] H. Cantzler, R. Fisher and M. Devy, "Quality enhancement of reconstructed 3D models using coplanarity and constraints", Technical Report, LAAS-CNRS.

[5] A. Sappa, "Towards an automatic CAD model generation from range images of industrial environments", Technical Report, Z+F UK Ltd.

[6] A. Sappa and M. Devy, "Fast range image segmentation by an edge detection strategy", *IEEE 3rd. Int. Conf. on 3-D Digital Imaging and Modeling*, Quebec, City, Canada, May-June 2001.

[7] T. Pablidis, *Algorithms for Graphics and Image Processing*. MD Computer Science Press, 1982