Extracting Surface Patches from Complete Range Descriptions

R. B. Fisher, A. W. Fitzgibbon, D. Eggert and C. Robertson
Department of Artificial Intelligence
University of Edinburgh
Edinburgh, UK EH1 2QL

KEYWORDS: range data segmentation, quadric surface extraction

Abstract

Constructing a full CAD model of a part requires feature descriptions from all sides; in this case we consider analytic surfaces as the geometric primitives. Most previous research in surface extraction has concentrated on extracting surfaces from a single view. This leads to several problems with aligning and combining partial surface fragments in order to produce complete part models. We have avoided these problems by adapting our single view, range data segmentation program to extract surfaces, and thus models, directly from fully registered range datasets.

1. Introduction

Constructing a full CAD model of a part requires feature descriptions from all sides. This seems obvious, but how to achieve that description is the question. The two main approaches are to extract partial surface descriptions from individual views (*e.g.* of range data) and then combine the descriptions, or to register the range datasets and then extract the surfaces.

The former approach has the advantage of having many existing algorithms for surface extraction, even from many years ago [27, 3, 17, 15, 21]. A recent comparison in [18] concluded that in many ways our single image planar surface segmentation algorithm has the best performance among current algorithms. In fact, our algorithm actually extracts quadric surfaces but the comparison in [18] applied only to planar surfaces. Thus, following the first approach is both a theoretical and a practical possibility. The single view approach also has the advantage that orthographic or spherical coordinate views are

typically what range sensors produce and it is clear what the topology of the range image is (*i.e.* it is easy to determine what the neighboring surface points are, which makes surface reconstruction simple).

The disadvantages of stitching together surfaces from single views are serious, however. The first problem is how to register accurately the features extracted from single views into a single reference frame. It is easily possible to align the views provided three independent non-parallel planes can be found in each pair of views. Two planes provide orientation alignment but allow one translational degree of freedom, which is resolved by the third plane. Faugeras [12] gives standard algorithms for transformation estimation. Our experience has shown that this provides only limited accuracy, so our models are typically accurate only to about 1-2 mm when using a range sensor that has 0.15 mm measurement standard deviation. To improve registration, many people use variations of the Iterated Closest Point algorithm [4] to provide fine alignment using the raw range data after the feature-based alignment provides an initial coarse registration.

The problem of registering multiple range datasets has been well researched [2, 5, 9, 10, 14, 16, 24, 25, 28, 29, 33, 34, 35, 38, 39] in the past few years, so we will not review it here. However, the result of this research is that it is now possible to obtain quite good registration of clean range image datasets of smooth curved and developable surfaced objects.

However, even if the views can be accurately aligned there is still the problem of stitching together the features. This is a rather complex problem because individual views might only provide a fragment of a surface and even a pair of views might not provide a complete view. One might use a symbolic surface-stitching algorithm to produce new surfaces. Orr [26] investigated methods for producing a combined planar surface from multiple overlapping views. His algorithm required complex reasoning in order to construct the boundary of a combined surface from the boundaries of the individual surfaces, particularly because positioning errors require one to use statistical tests to determine if vertices observed in the two views are the same. The problems that have been encountered derive mainly from errors in registration: it is difficult to determine when features found on surface boundaries arise from segmentation limitations, statistical variations or nearby real features. Orr overcame some of the problems using a Mahalanobis distance test to assess the likelihood of statistical variations, but problems remained with the reliability of estimating surface statistics and the resolution of the remaining alternative merging hypotheses.

The research presented here avoids these problems by adapting a single view quadric surface range segmentation program to extract surfaces and thus complete object models directly from fully registered 3D range datasets. It can be applied to both industrial parts with simple developable surfaces and more complex surfaces.

There has been some previous research on extracting geometric surfaces from full 3D datasets, of which [13] is a good example, except that the reported work addressed mainly planar surfaces. Successful previous research on extracting descriptions from segmented range images has included [30, 31, 32] who explored combining both surface patches (refitting surfaces to regions that overlap in registered multiple views) and combining polygonal surfaces. Our work improves on this work by handling curved surfaces better. An alternative is to fit completely covering spline meshes to a polygonalized 3D dataset, once the views have been registered and combined [22, 20]. The work presented here explores similar ideas, except considers analytic surfaces occurring on many manufactured industrial parts.

2. Complete Patch Extraction

The algorithm for complete surface extraction consists of these five stages:

1. Multiple view surface registration.
2. Initial surface polygonalization.
3. Local curvedness estimation and shape classification.
4. Surface growing.
5. Patch edge adjustment.

These are discussed in more detail in this section.

The algorithm used for **multiple view surface registration** is described in [10]. The algorithm registers multiple range images using a modified Iterated Closest Point algorithm [4, 11] and produces a cloud of range (x, y, z) values registered in a common reference frame. The algorithm assumes that every point has been observed in at least two range images, and thus avoids the problem of identifying points without a correspondence (*i.e.* no distance threshold is needed). It uses a force model, with "springs" connecting corresponding points that iterates over time to pull the datasets into alignment, rather than a closed form solution, because not all points are treated identically. Correspondence is from a point to the nearest tangent plane, as in [7], except for points at depth discontinuities, which are mapped to the tangent plane of points "around the corner" in the second view. This adaptation allows the use of the algorithm on locally and globally developable surfaces, where the translation aligning the surfaces is not well constrained because the two surfaces can translate over each other, as well as generic curved surfaces.

The **initial surface polygonalization** is performed using the Hoppe *et al* [19] algorithm. The purpose of this polygonalization is to construct local topology between the range points that can then be used as part of the curvature classification and surface growing processes. The problem that we're trying to solve is how to find the local surface shape. With range data acquired from a single viewpoint there is usually a local topology in the form of a regular grid imposed by the mechanical structure of the range sensor. With multiple registered range images the range data usually forms a cloud of points about the true surface. This means that getting an initial approximation of the surface shape is difficult and it is hard to determine point adjacency on the surface during the surface growing phase. Our solution to the problem was to tessellate the surface, which gives a regular 3D mesh on the surface, then to perform surface shape extraction and surface growing using the centroids of the mesh polygons.

Triangulation produces a set of connected small polygonal patches. Associated with each polygonal patch are the range points from the patch and also the neighboring patches. Local topology is the essential ingredient needed by most other range data segmentation algorithms, in that it specifies which of the range points are nearby. While other graph-based connection algorithms are possible to use, we chose the Hoppe algorithm for convenience.

With the local topology, it is easy to **estimate the local surface curvatures** at each point through a local surface fitting algorithm. This provides an estimate of the principal curvatures which could be used in a (Mean, Gaussian) curvature classification process, as in [3]. However, this process does not work well because of the reduced surface shape information available from the mesh vertices, as compared to a dense surface available from single view range images. Taubin [36] and Chen and Schmitt [6] give linear and quadratic complexity (respectively) algorithms for estimating the local principal curvatures at the vertices of a polyhedral mesh. Unfortunately, polygonalized surfaces on merged datasets are locally rather noisy so the curvature estimates are not particularly accurate. Our experience has also shown that the surface fitting algorithm has a more significant impact than the initial curvature classification. So, because our surface fitting algorithm uses a general quadric form rather than different forms specialized for the different shape classes, we only classify polygons into the classes $\{$planar, curved, edge$\}$.

Shape classification estimates the local curvedness by finding the maximum angle θ_{max} that nearby connected surface polygons turn away from the current polygon. This angle is found by examining the angle between polygon normals in a neighborhood of distance N polygons (default is $N = 2$) about the current polygon. The maximum angle gives an indication of how curved the surface is and forms the basis for the initial labeling of polygons into different surface shape classes:

CONDITION	POLYGON CLASS.
If $\theta_{max} < \tau_{plane}$ (4.7 deg)	**Planar**
If $\theta_{max} > \tau_{edge}$ (10 deg)	**Fold Edge**
Else	**Quadric**

These parameters were selected by hand-tuning, but were neither very stable (always reliable classifications) nor unstable (quite varying classifications). Changing the parameters gradually increases the numbers of misclassified polygons, but this did not cause the surface fitting algorithm much problem, other than increased work. Using a surface growing algorithm such as that used here, it is better to oversegment but still produce patches that are part of a single underlying surface which can then be grown together. Extreme oversegmentation must be avoided otherwise the initial surface patches are too small to get a decent fit. A finer mesh can help resolve the problem that at cost of increased computation. On the other hand, undersegmentation will produce initial regions combining portions of several surfaces. The initial fit to these regions will be bad, and gives parameters that do not lead to viable surfaces during surface growing.

Only surface orientation (fold edge) discontinuities are detected by this algorithm (depth discontinuities do not exist because the approach is viewpoint independent). Connected (using the local topology) polygons with the same shape class are joined to form the seeds of extended surfaces.

Complete surfaces are found by an iterative **surface growing** process that starts from the initial seed surfaces. The surface model is the general quadric surface form, unlike the $z = f()$ Monge form used by Besl [3] (which did go to fourth order, however). The algorithm adds and deletes members from the list of polygons associated with a given surface. The surface growing algorithm uses the centroids of the polygons, rather than the cloud of raw range points, because it extends surfaces at the patch boundaries and needs to know which points are adjacent to the boundary. The polygonal surface mesh provides the adjacency information, so we use the centroids of the mesh polygons as the data for the surface shape estimation. The key ideas behind the algorithm are:

- There is competition between surfaces for polygons.
- The algorithm works iteratively between extending the surface shape and refitting the surface model to the grouped points.
- Small surfaces are deleted and their polygons are then available for combining into other surfaces.
- Only planes are extracted on the first pass, as their curvatures are less stable.

- After a surface is fitted, polygons that do not lie within $\sigma_{noise}\sqrt{2}$ of the fitted surface are returned to the pool of unfitted polygons. σ_{noise} is the estimated standard deviation of the polygon patch centroid from the true surface.

- Adjacent surfaces with similar shape parameters and at least 30% shared boundary pixels are combined.

- A specialized cylinder fitting is applied to quadric surface surfaces on the final iteration.

A pseudo-code description of the reconstruction algorithm follows. At each of the N passes there is a set of surfaces R (initially created by connected component analysis on the surface) that are being grown by the algorithm. Each surface consists of a set of edge connected mesh polygons and its implicit surface parameters. There is also a pool of mesh polygons that are not yet associated with any surface.

```
For each of N passes
  For each current surface R
    Do {
      If current surface too small ( < 60 polygons),
        then move polygons into pool
      If first pass and N > 1, then only extract planes
      Fit quadric surface to R
      Fit plane surface to R
      If last pass, also do cylinder fit to surface to R
      If surface fit fails, then move polygons into pool
      If second or later pass, then
        If surface R is smaller than an adjacent surface,
          with whom it shares at least 30% of the
          total two-surface boundary and all of R's
          polygons are within 2σ_noise √2 of the
          other surface's surface, then move
          polygons into pool.
        Move R's polygons into the pool that are
          further than σ_noise √2 from R's surface.
```

Let there be A of these polygons.

Add the polygons from the pool that are adjacent to R and

that are closer than $\sigma_{noise}\sqrt{2}$ of R's surface.

Let there be B of these polygons.

Add polygons to R from adjacent

surfaces that are within $\sigma_{noise}\sqrt{2}$

of R's surface and are closer to R's

surface than to their original surface.

Let there be C of these polygons.

} while ($A > 3$ or $B + C > 3$)

Surface model selection and parameter estimation

Central to the surface fitting process is a least square shape parameter estimation process that determines both the surface type and the surface shape parameters. The key ideas of the shape parameter estimation are:

- A least-square error criterion is used.
- The fitting is for general quadrics (9 independent parameters: $ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0$), but has specializations for some degenerate quadrics (*i.e.* planes and cylinders). The parameters are estimated using Taubin's method [37].
- The cylinder is parameterized by 5 parameters: a line in space and the radius of the cylinder. The details are given in [37]. The parameters of the cylinder are found using the iterative least square algorithm found in [8].
- The plane fitting uses a standard least square method. If $(x_i, y_i, z_i)^T$ is the i^{th} data point, then let $\vec{p_i} = (x_i, y_i, z_i, 1)^T$. Then the parameter vector $(a, b, c, d)^T$ of the plane model $ax+by+cz+d = 0$ is the eigenvector with smallest eigenvalue from the scatter matrix $\sum_i \vec{p_i}\vec{p_i}^T$.
- The selection between the different model shape classes is based on minimizing the surface fit error, subject to a bias factor related to the number of parameters needed for the fit. If a bias factor were not used, the 9 parameter quadric case would generally have lower least square fitting error than a 3 parameter plane model, even on data coming from an underlying planar surface. We used

the Akaike information criterion [1] to bias the model selection in favor of simpler models, by providing a cost for the number of parameters used in the model.

$$e_{plane} = \frac{1}{\sigma^2_{noise}} \Sigma \epsilon_i^2 + 3\alpha$$

$$e_{cylinder} = \frac{1}{\sigma^2_{noise}} \Sigma \epsilon_i^2 + 5\alpha$$

$$e_{quadric} = \frac{1}{\sigma^2_{noise}} \Sigma \epsilon_i^2 + 9\alpha$$

and the class i with the smallest e_i is chosen. ϵ_i is the geometric error distance between the model and the polygon center-of-mass. Here, $\alpha = 32$ is a heuristically chosen weight factor, influenced by a MDL-style bias term [23].

The pseudo-code for the surface fitting algorithm follows:

Get e_{plane} from plane fit
Get $e_{quadric}$ from quadric fit
If plane and quadric fit successful (ie. least-square
 scatter matrices are full rank)
  If cylinder fit is desired
    Get $e_{cylinder}$ from cylinder fit
  Choose surface type with best fit
Else if plane fit successful
  Then choose Plane type
Else if quadric fit successful
  Then choose Quadric type
Else return NoFit

Surface intersection and boundary refinement

Because our reconstruction process is intended for manufactured parts, a phase of **surface edge adjustment** occurs. A majority of surfaces on engineered parts are planes and cylinders and whenever these surfaces intersect at fold edges, geometric descriptions of the surface intersection are easily obtainable. When the surfaces do not intersect at fold edges, *i.e.* the intersection has tangent plane continuity, we do

not perform any model improvement at present. This can leave a ragged boundary joining the surfaces but the surface join itself is smooth. If there is no strong image evidence to place the boundary *i.e.* there is only a change in curvature (but accurately locating this is difficult) we speculate that the best approach to locating this boundary is to exploit some problem-specific knowledge (such as knowledge that such boundaries are usually straight), or ask for user assistance.

In addition, range data near orientation discontinuities on real parts is often affected by specular reflections or the range sensor "footprint" acquires data from two different surfaces. Thus, one cannot obtain very accurate surface edge descriptions from the data near an edge. On the other hand, the intersection of the fitted ideal surfaces provide much better edge descriptions. As the intersections are usually either infinite in extent or extend considerably beyond the real surface edge, we apply statistical tests to determine the extent of the ideal edge in the real data by examining proximity of range measurements to the ideal intersection edge.

We have only implemented the non-degenerate cases of a plane intersecting a quadric, but the other cases are similar (although require more special cases).

The main steps of the surface intersection algorithm are:

1. Form the unbounded intersection curve by intersecting the unbounded mathematical surfaces passing through the finite surfaces obtained by surface fitting.

2. Identify the portion of the unbounded curve that is adjacent to the two intersected surfaces. This requires examining the registered range data to ensure that the intersection curve has nearby range points (within $3\sigma_noise$). When the intersection curve has no nearby data, the boundary curve is terminated. Fill in small gaps. This might produce more than one segment if the surfaces join at several places.

3. Replace the ragged boundary that was found by surface fitting by the intersection curves, connecting the curve endpoints to the remaining surface boundary (a point list).

3. Results

Here we show the results of the segmentation process on several parts. In Figure 1, we see a view of the combined range data for a part with planar, cylindrical and conical surfaces. The boundary is a little ragged because the Hoppe algorithm [19] was used to tessellate the surface and is somewhat sensitive to local surface noise, resulting in small outlying polygonal patches. Figure 2 shows several views of

the segmented surfaces when represented by a CAD-like surface representation. Figure 3 shows the combined range data for another similar part, with a close-up in Figure 4. Notice that although there is a plane-cylinder join with tangent plane continuity, which leads to the ragged surface shown in Figure 5, the complete reconstructed object surface is smooth. (This is a degenerate case of when a plane intersects a quadric, and was not implemented.) Figure 6 shows several views of its segmented surfaces. One can see a little raggedness where the cylindrical surface has the smooth join with the planar surface and improving this is a topic of future research. Otherwise, there is no observable detail to indicate any flaws with the method, because the segmentation algorithm has worked well on these parts. Some numerical results are given below.

Figure 1. View of combined and tessellated range data for UFO part.

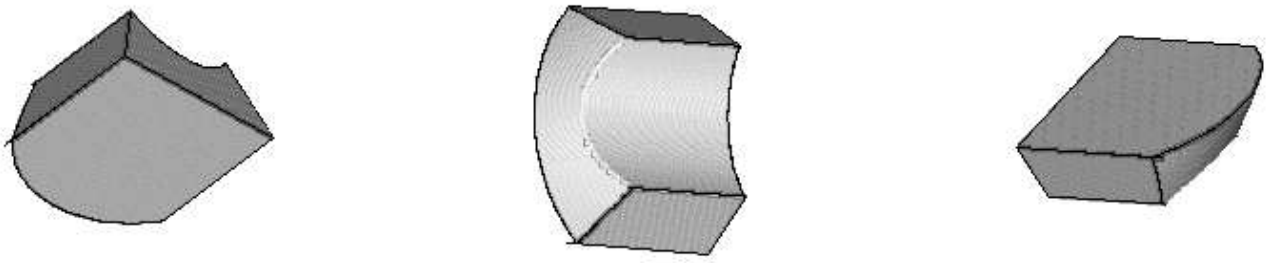


Figure 2. Three views of fully segmented UFO part with plane-cylinder and plane-plane intersections highlighted in black.

Figure 3. View of combined and tessellated range data for BAe part.

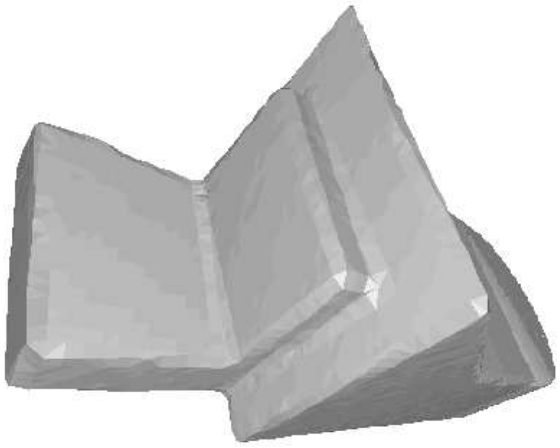


Figure 4. Close-up of coarser tessellated range data for BAe part that is used for surface growing.

Figures 7 and 8 show the combined range data and extracted raw surfaces for two simpler parts. Part (a) of the figures shows a view of the raw range data and part (b) shows the raw segmentation output before the infinite geometric surface shapes are extracted and boundaries intersected. What can be seen is the large surfaces with ragged boundaries that are related to the original surface polygonalization. The segmentation shown in Figure 8 used a coarser polygonalization and hence the patches are bigger.

Based on the geometric features extracted from the objects shown above, some accuracy statistics are (N is the number of instances of this relationship encountered in the experiments):

- mean angular error between planes is 0.28° (min = 0.01°, max = 0.70°, $N = 24$).
- mean cylindrical radius error is 0.24 mm (min = 0.03 mm, max = 0.50 mm, $N = 3$).

Figure 5. Close-up of ragged cylindrical surface boundary for BAe part at the point where the cylinder meets the plane with tangent plane continuity. The resulting complete surface is effectively smooth, but the boundary is not the ideal straight line expected.

- mean parallel plane separation is 0.39 mm (min = 0.10 mm, max = 1.29 mm, $N = 5$).
- mean angle error between cylinder axes and an orthogonal plane is 0.23° (min = 0.07°, max = 0.29°, $N = 5$).

It should be noted that our range sensor produces range data with standard deviation about $\sigma = 0.2$ mm nearly uniformly over the whole field of view. The geometric accuracy is also greatly affected by the success of the surface registration algorithm [10]. Hence, these accuracy figures are really a function of both processes.

4. Conclusions

This paper has presented a segmentation algorithm that has the advantage of avoiding the problem of fusing symbolic surface descriptions when producing a description of all sides of a part in a common reference frame. It depends primarily on having registered 3D dataset to provide the accurate relative positioning of the surface points, from which our quadric surface extraction algorithm can then accurately extract surfaces. Estimated feature position is on the order of 0.30 degrees and 0.4 mm and cylindrical radius is accurate to about 0.25 mm for parts with dimensions on the order of $10 \times 10 \times 10$ cm^3.

Acknowledgements

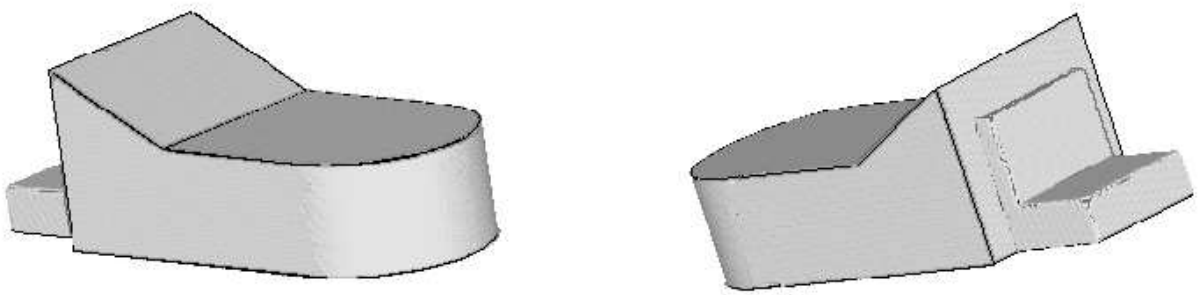


Figure 6. Two views of fully segmented BAe part with plane-cylinder and plane-plane intersections highlighted in black.

The work presented in this paper was funded by a UK EPSRC grant GR/H86905. Thanks to Anthony Ashbrook for technical assistance.

References

[1] H. Akaike. “A New Look at the Statistical Model Identification”, IEEE Trans. on Automatic Control, Vol. 19, pp716-723, 1974.

[2] R. Bergevin, D. Laurendeau, and D. Poussart. “Registering range views of multipart objects”, Computer Vision and Image Understanding 61(1), pp 1-16, 1995.

[3] P. J. Besl. Surfaces in Early Range Image Understanding. PhD Dissertation, Electrical Engineering and Computer Science Department (RSD-TR-10-86), University of Michigan, 1986.

[4] P. J. Besl and N. D. McKay. “A method for registration of 3-D shapes”, IEEE Trans Pat. Anal. and Mach. Intel., 14(2), pp 239-256, 1992.

[5] G. Blais and M. D. Levine. “Registering multiview range data to create 3D computer objects”, Technical Report TR-CIM-93-16 Center for Intelligent Machines, McGill University, Montreal, Canada, Nov 1993.

[6] X. Chen and F. Schmitt. “Intrinsic surface properties from surface triangulation”, Proc. 2nd European Conf. on Computer Vision, (ed) G. Sandini, pp 739–743 St. Margherita Ligure, Italy, May 1992.

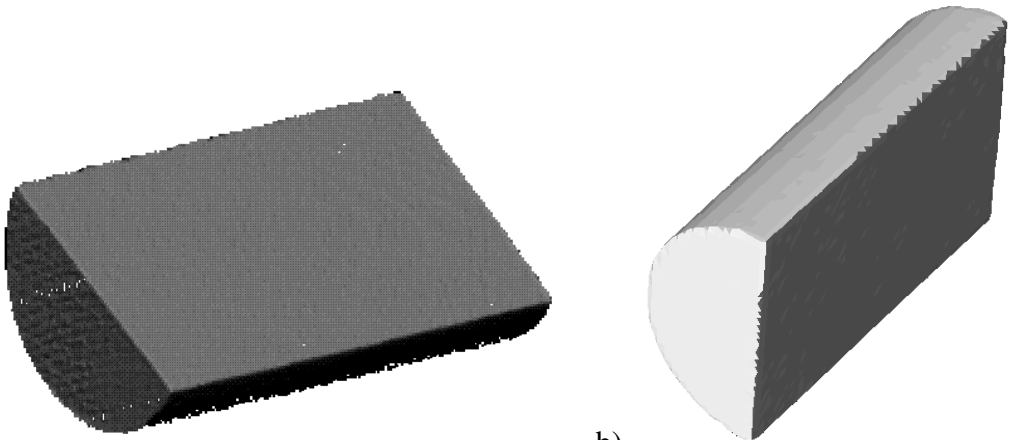


Figure 7. Raw (a) polygonalization and (b) a view of extracted unintersected surfaces.

[7] Y. Chen and G Medioni. “Object modeling by registration of multiple range images”, Image and Vision Computing, 10(3) pp 145–155, 1992.

[8] J. E. Dennis and D. M. Gay and R. E. Welsch. An adaptive nonlinear least-squares algorithm. ACM Trans on Math. Software, 7(3), pp 348–368, 1981.

[9] C. Dorai, J. Weng, and A K Jain. “Optimal registration of multiple range views”, Proc. 12th IAPR Int. Conf. on Pattern Recognition, Jerusalem, Israel, pp 569-571, Oct 1994.

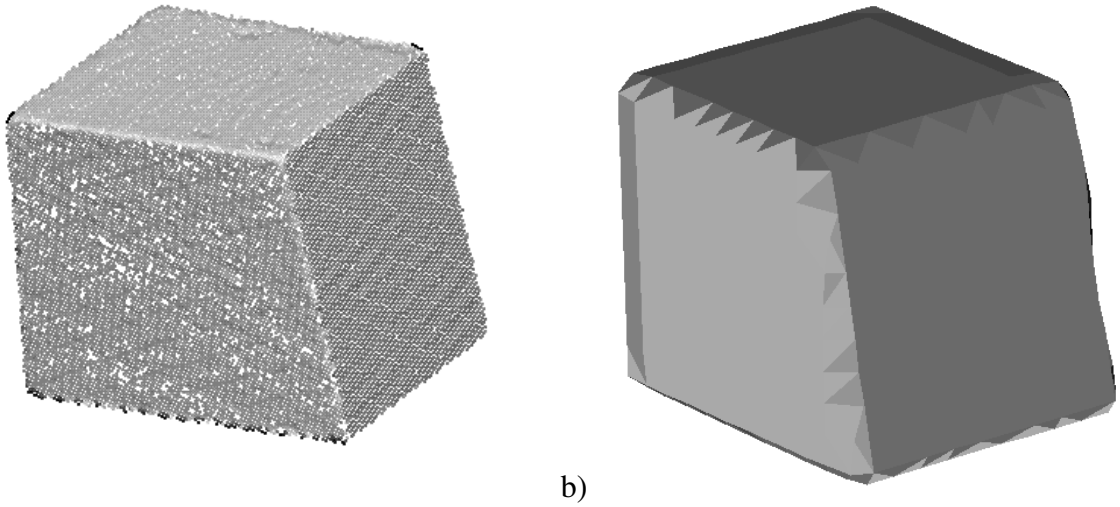
[10] D. Eggert, A. W. Fitzgibbon, R. B. Fisher. “Simultaneous registration of multiple range views for use in reverse engineering”, Proc. Int. Conf. on Pat. Recog., pp 243–247, Vienna, Aug. 1996.

[11] D. W. Eggert, A. Lorusso, R. B. Fisher, ”Estimating 3-D rigid body transformations: a comparison of four major algorithms”, Mach. Vis. and Applic., Vol 9, pp 272–290, 1997.

[12] O. D. Faugeras. Three-dimensional computer vision : a geometric viewpoint, Cambridge, Mass. MIT Press, 1993.

[13] O. D. Faugeras, M. Hebert. “A 3-D Recognition and Positioning Algorithm Using Geometric Matching Between Primitive Surfaces”, Proceedings 8th Int. Joint Conf. on Artificial Intelligence, pp 996-1002, 1983.

[14] J. Feldmar and N. Ayache. “Rigid and affine registration of smooth surfaces using differential properties”, Proc. 3rd European Conf. on Computer Vision Stockholm, Sweden, pp 397-406, May 1994.



a) b)

Figure 8. Raw (a) polygonalization and (b) a view of extracted unintersected surfaces.

[15] P. J. Flynn and A. K. Jain. "BONSAI: 3D Object Recognition Using Constrained Search", IEEE Trans Pat. Anal. and Mach. Intel., 13(10), pp 1066-1075, Oct 1991.

[16] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau. "Registration of multiple range views for automatic 3-D model building", Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, Seattle, WA, pp 581-586, June 1994.

[17] R. L. Hoffman and A. K. Jain. "Segmentation and Classification of Range Images", IEEE Trans Pat. Anal. and Mach. Intel., 9(5) pp 608–620, Sept 1987.

[18] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, R. Fisher. "An Experimental Comparison of Range Segmentation Algorithms", IEEE Trans. Pat. Anal. and Mach. Intel., 18(7), pp 673–689, July 1996.

[19] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. "Surface Reconstruction from Unorganized Points", Computer Graphics, 26(2), pp 71–78, 1992.

[20] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction, Proc. SIGGRAPH, pp 295-302, 1994.

[21] X. Y. Jiang and H. Bunke. "Fast Segmentation of Range Images into Planar Regions by Scan Line Grouping", Mach. Vis. and Applic., 7(2), pp 115–122, 1994.

[22] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. Proc. SIGGRAPH: computer vision, pp 313-324, 1996.

[23] Y.G. Leclerc. “Constructing Simple Stable Descriptions for Image Partitioning”, Int. J. Computer Vision, Vol 3, 73–102, 1989.

[24] F. Lu and E E Milios. “Robot pose estimation in unknown environments by matching 2D range scans”, Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, Seattle, WA, pp 935-938, June 1994.

[25] T. Masuda and N Yokoya. “A robust method for registration and segmentation of multiple range images”, Computer Vision and Image Understanding, 61(3), pp 295-307, 1995.

[26] M. J. L. Orr, J. Hallam, R. B. Fisher. “Fusion through Interpretation”, Proc. 2nd European Conf. on Computer Vision, (ed) G. Sandini, pp 801–805, St. Margherita Ligure, Italy, 1992.

[27] M. Oshima, Y. Shirai. “Object Recognition Using Three-Dimensional Information”, Proc. 7th Int. Joint Conf. on Artificial Intelligence, pp 601-606, 1981.

[28] B. Parvin and G. Medioni. “B-rep from unregistered multiple range images”, Proc., Int. Conf. on Robotics and Automation, pages 1602–1608, 1992.

[29] M. Potmesil. “Generating models of solid objects by matching 3D surface segments”, Proceedings of the 8th Int. Joint Conf. on Artificial Intelligence, Karlsruhe, West Germany (Aug 1983) pp 1089-1093.

[30] V. Sequeira, J.G.M Gonçalves, M. I. Ribeiro. 3D Environment Modelling Using Laser Range Sensing. J. of Robotics and Autonomous Systems, 16(1), pp 114-127, 1995.

[31] V. Sequeira, J. Gonçalves, M. I. Ribeiro. 3D Reconstruction of Indoor Environments. Proc. 3rd Int. Conf. Image Proc., Lausanne, Switzerland, pp. 17P2, September 16-19, 1996.

[32] V. Sequeira. Active Range Sensing for Three-Dimensional Environment Reconstruction. Eur. Commission Joint Research Centre Tech Note I.97.26, Feb 1997.

[33] M. Soucy and D. Laurendeau. “Surface modeling from dynamic integration of multiple range views”, Proc. 11th Int. Conf. on Pattern Recognition, pages 449–452, 1992.

[34] A. J. Stoddart, S. Lemke, A. Hilton, T. Renn. “Estimating pose uncertainty for surface registration”, British Machine Vision Conference, pp 23-32, Edinburgh, UK, 1996.

[35] A. J. Stoddart, A. Hilton. “Registration of multiple point sets”, Proc. 13th Int. Conf. on Pattern Recognition, pp B40-44 Vienna, Austria, 1996.

[36] G. Taubin. “Estimating the tensor of curvature of a surface from a polyhedral approximation”, Proc. 5th Int. Conf. on Computer Vision, pp 902-907, 1995.

[37] G. Taubin. “Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation”, Proc. IEEE PAMI, 13(11), pp1115–1138, 1991.

[38] B. C. Vemuri and J. K. Aggarwal. “3-d model construction from multiple views using range and intensity data”, IEEE Conf. on Computer Vision and Pattern Recognition, pages 435–437. Miami Beach, Florida, 1988.

[39] Z. Zhang. “Iterative point matching for registration of free-form curves and surfaces”, International Journal of Computer Vision, 13(2), pp 119-152. 1994.