Extracting Surface Patches from Complete Range Descriptions

R. B. Fisher
Department of Artificial Intelligence
University of Edinburgh
Edinburgh, UK EH1 2QL
rbf@dai.ed.ac.uk

A. W. Fitzgibbon
Department of Artificial Intelligence
University of Edinburgh
Edinburgh, UK EH1 2QL
andrewfg@dai.ed.ac.uk

D. Eggert
Department of Artificial Intelligence
University of Edinburgh
Edinburgh, UK EH1 2QL
eggertd@dai.ed.ac.uk

Abstract

Constructing a full CAD model of a part requires feature descriptions from all sides; in this case we consider surface patches as the geometric primitives. Most previous research in surface patch extraction has concentrated on extracting patches from a single view. This leads to several problems with aligning and combining partial patch fragments in order to produce complete part models. We have avoided these problems by adapting our single view, range data segmentation program to extract patches, and thus models, directly from fully merged 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 individual surface patches from individual views (e.g. of range data) and then merge them, or to merge the range data and then extract the surface patches.

The former approach has the advantage of having many existing algorithms for patch extraction, even from many years ago [23, 3, 14, 9, 17]. A recent comparison in [15] concluded that in many ways, our single image planar patch segmentation algorithm has the best performance among current algorithms. In fact, our algorithm actually extracts quadric surface patches, but the comparison in [15] applied only to planar patches. The single view approach also has the advantage that single orthographic or spherical coordinate views are typically what range sensors produce, and so it is clear what the topology of the range image is (i.e. it is easy to determine what are neighbouring surface points).

The disadvantages of merging patches from single views are serious, however. The first problem is how to register the features extracted from single views into a single reference frame. It is easily possible to align the views provided 3 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[11] gives standard algorithms for transformation estimation.

However, our experience has shown that this provides only limited accuracy, and so our models are typically accurate only to about 1-2 mm (using a range sensor that has 0.15 mm measurement standard deviation). Our belief is that the causes of this poor registration are: 1) the limited number of features that can be paired between two substantially different views, 2) the fact that we are working with industrial parts that have large developable surfaces (which do not locally provide the registration constraints obtainable from curved surfaces on more exotic shapes), and 3) the attempt by the least-square algorithm to spread the error across the whole of a surface patch (i.e. much of the patch will have very small errors while the edges might have substantially larger errors).

Improved registration is possible, as in our Imagine II object recognition system, which uses an adapted Iterated Closest Point algorithm [4] to provide fine alignment using the raw range data after the feature-based alignment provides an initial coarse registration. But, there is still the problem of merging the features. This is a rather complex problem, because individual views might only provide a fragment of a patch, and even a pair of views might not provide a complete view. So, one might use a patch-stitching algorithm [22] to produce new patches. These algorithms require complex reasoning in order to construct the boundary of a merged patch from the boundaries of the individual patches, particularly because positioning errors require one to use statistical tests to determine if vertices observed in the two views are the same.

We have avoided these problems by adapting our single view quadric patch range segmentation program to extract patches and thus complete object models directly from fully merged range datasets.

The problem of merging multiple range datasets has received considerable research[2, 5, 13, 7, 8, 12, 20, 21, 24, 25, 26, 27, 28, 29, 30] 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.

There has been some previous research on extracting geometric surface patches from full 3D datasets, of which [10] is a good example, except that the reported work addressed mainly planar patches.

This paper describes the algorithm used for surface patch extraction and presents results obtained from using the approach of surface merging first and then segmentation of full 3D range datasets. It can be applied to both industrial parts with simple developable surfaces and more complex surfaces.

2. Complete Patch Extraction

The algorithm for complete patch extraction consists of these stages:

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

These are discussed in more detail in this section.

The algorithm used for **multiple view merging** is described in [7]. The algorithm merges multiple range images using a modified Iterated Closest Point algorithm [4] 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

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

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 [6], 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 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 tessellation** is performed using the Hoppe and DeRose [16] algorithm. The purpose of this tessellation is to construct local topology between the range points that can then be used as part of the curvature classification and region growing processes. The problem that we're trying to solve here 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 merged 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 second, 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, and then do surface shape extraction and surface growing using the centres-of-mass of the mesh polygons.

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

With a 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 can be used in a (mean, gaussian) curvature classification process, as in [3]. Surface patches are then formed by grouping polygons with a similar curvature classification to form the initial seed patches for a surface fitting algorithm. Here, because our experience has shown that the surface fitting algorithm has a more significant impact than the initial curvature classification, and because our surface fitting algorithm uses a general quadric form rather than different forms specialised for the different shape classes, we only classify polygons into the classes $\{$planar, curved, edge$\}$. Surface fitting is still applied to connected polygons with the same shape class.

Our shape classification process estimates the local curvedness by finding the maximum angle θ_{max} that adjacent surface polygons turn away from the current polygon. This angle is found by examining the angle between polygon normals in a neighbourhood of distance N polygons (default is 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)	**Edge**
Else	**Quadric**

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 then joined to form the seeds of extended surface patches.

Complete surface patches are found by an iterative **surface growing** process that starts from the initial seed patches. The surface model is the general quadric path 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. As discussed above, the surface growing uses the centres-of-mass of the polygons, rather than the cloud of raw range points. This is because the surface growing algorithm extends patches at the patch boundaries, and therefore it needs to know which points are adjacent to the boundary. The tessellated surface mesh provides the adjacency information, and so we use the centres-of-mass of the mesh polygons as the data for the surface shape estimation. The key ideas behind the algorithm are:

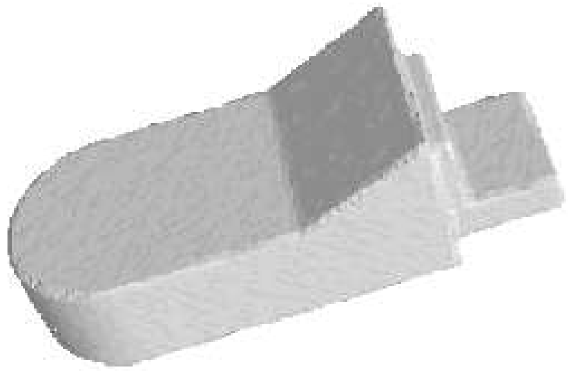

Figure 2. View of merged and tessellated range data for BAe part.

- There is competition between patches for polygons.
- The algorithm works iteratively between extending the patch shape and refitting the surface model to the grouped points.
- Small patches are deleted and their polygons are then available for merging into other patches.
- 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.
- Adjacent patches with similar shape parameters and at least 30% shared boundary pixels are merged.
- A specialised cylinder fitting is applied to the quadric surface patches on the final iteration.

Pseudo-code for the surface growing algorithm is given in Appendix A

Central to the surface patch 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 parameters), but has specialisations for some degenerate quadrics (i.e. planes and cylinders).

- The selection between the different model shape classes is based on minimising the surface fit error, subject to a bias factor related to the number of parameters needed for the fit. That is, if the fit error is comparable, the fit requiring the 3 plane parameters is preferred to the 9 parameter general quadric fit. More precisely, we use the Akaike information criterion[1] to bias the model selection in favour of simpler models:

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

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

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

  and the class i with the smallest e_i is chosen. ϵ_i is the geometric error distance between the model and the polygon centre-of-mass. Here, $\alpha = 32$ is a heuristically chosen weight factor, influenced by a MDL-style bias term[18].

The pseudo-code for the surface fitting algorithm is given in Appendix B.

Finally, because this process is intended for manufactured parts, a phase of **patch edge adjustment** occurs. A majority of surfaces on engineered parts are planes and cylinders, and whenever these surfaces intersect at fold edges, then geometric descriptions of the patch intersection are easily obtainable. When the surfaces do not intersect at fold edges, that is, the intersection has tangent plane continuity, we do not perform any model improvement at present. This can leave a ragged boundary joining the surface patches, but the surface join itself is smooth. As there is no strong image evidence to place the boundary (ie. there is 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 (i.e. knowledge that such boundaries are straight), or ask for user assistance.

In addition, range data near orientation discontinuities on real parts is often not very good, because of specular reflections and also because the range sensor "footprint" will acquire data from two different surfaces. Thus, one cannot obtain very accurate patch edge descriptions from the data near an edge, whereas 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 patch 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.

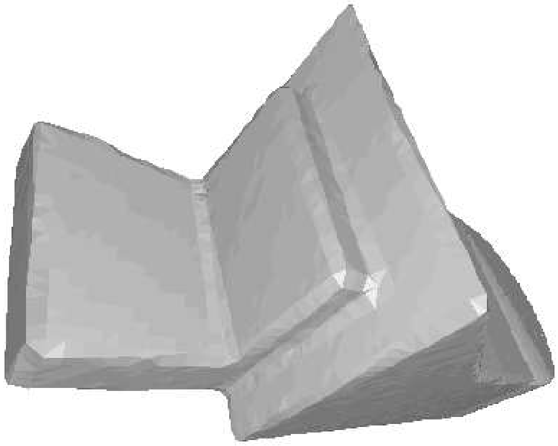

Figure 3. Close-up of coarser tessellated range data for BAe part that is used for surface growing.

3. Results

Here we show the results of the segmentation process on 2 parts. In Figure 1, we see a view of the merged range data for a part with planar, cylindrical and conical surface patches. The boundary is a bit ragged because we used the Hoppe algorithm [16] to tessellate the surface and this algorithm is somewhat sensitive to local surface noise, resulting in small outlying polygonal patches. Figure 5 shows several views of the segmented surface patches, when represented by a CAD-like surface representation. Figure 2 shows the merged range data for another similar part, with a close-up in Figure 3. Notice that, although there is a plane-cylinder join with tangent plane continuity, which leads to the ragged patch shown in Figure 4, the complete reconstructed object surface is smooth. Figure 6 shows several views of its segmented surface patches. One can see a little raggedness where the cylindrical surface has the smooth join with the planar surface. Improving this is a topic of future research. In a sense, there is not much to see because the segmentation algorithm has worked well on these parts.

4. Conclusions

This paper has presented a segmentation algorithm that has the advantage of avoiding the problem of patch merging when producing a description of all sides of a part in a common reference frame. It depends primarily on having merged 3D dataset to provide the accurate relative positioning of the surface points, from which our quadric surface patch extraction algorithm can then accurately

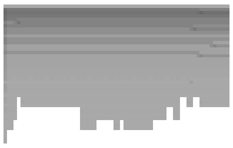

Figure 4. Close-up of ragged cylindrical patch boundary for BAe part at the point where the cylinder meets the plane with tangent plane continuity. The resulting complete surface is effectively smooth, but the boundary is not the ideal straight line expected.

extract surface patches. Segmented patch position is on the order of the range error (0.25 mm), even with engineering parts that have hard-to-handle developable surfaces.

Acknowledgements

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* Vol 61 No 1 (1995) pp 1-16

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

[7] 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

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

[9] 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.

[10] 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.

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

[12] 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 (June 1994) pp 581-586

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

[14] 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.

[15] 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.*, Vol 18(7), pp 673–689, July 1996.

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

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

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

[19] A. Lorusso, D. Eggert, R. B. Fisher. "A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations". *Proc. British Machine Vision Conf. BMVC95*, Birmingham, pp 237–246, September 1995.

[20] 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 (June 1994) pp 935-938

[21] T. Masuda and N Yokoya. "A robust method for registration and segmentation of multiple range images", *Computer Vision and Image Understanding*, Vol 61 No 3 (1995) pp 295-307

[22] 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.

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

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

[25] 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

[26] 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.

[27] 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).

[28] A. J. Stoddart, A. Hilton, "Registration of multiple point sets", *Proc. 13th Int. Conf. on Pattern Recognition*, pp B40-44 Vienna, Austria, (1996).

[29] 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.

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

A. Pseudo-code of Surface Growing Algorithm

```
For each of N passes
    For each current region R
        Do {
            If region too small ( < 60 polygons),
                then move polygons into pool
            Fit surface (R), doing cylinder case if last pass
            If surface fit fails, then move polygons into pool
            If first pass of many, then only extract
                initial planes
            If second or later pass, then
                If region is smaller than an adjacent region,
                    with whom it shares at least 30% of the
                    total two-region boundary and all of its
                    polygons are within 2σ_noise√2 of the
                    other region's surface, then move
                    polygons into pool.
                Move the A polygons into the pool that are
                    further than σ_noise√2 from the surface.
                Add the adjacent B polygons from the pool
                    that are within σ_noise√2 of the surface.
                Add the C polygons from adjacent
                    regions that are within σ_noise√2
                    of this surface and are closer to this
                    surface than to their original owner.
        } while (A > 3 or B+C > 3)
```

B. Pseudo-code of Surface Fitting Algorithm

```
    Get e_plane from plane fit
    Get e_quadric from quadric fit
    If plane and quadric fit successful
        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
```

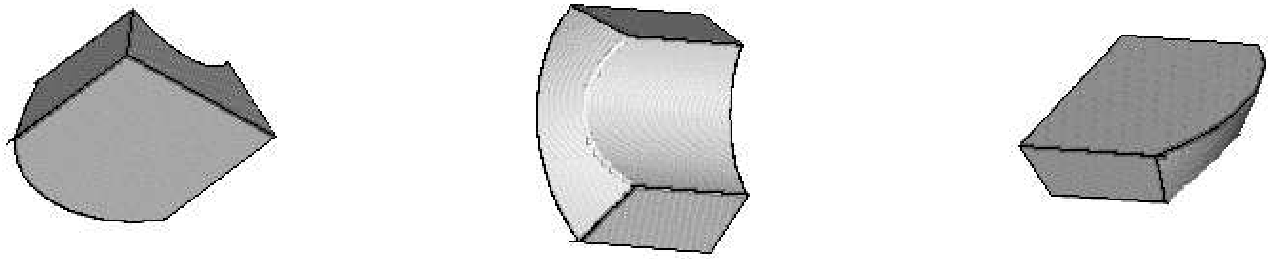

Figure 5: Three views of fully segmented UFO part with plane-cylinder and plane-plane intersections highlighted in black.

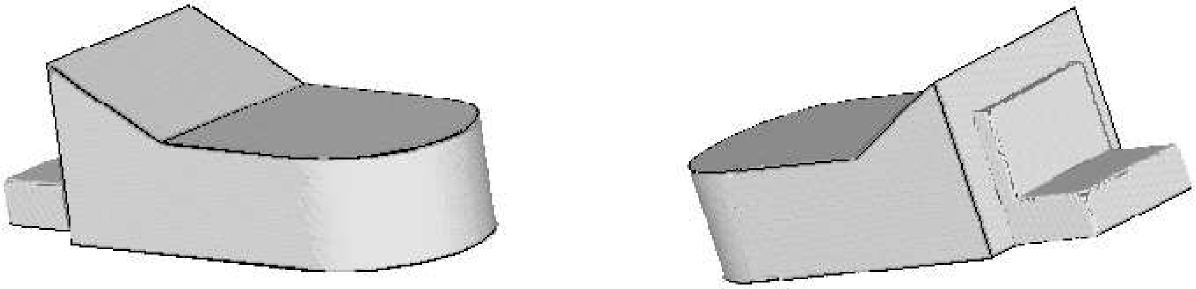

Figure 6: Two views of fully segmented BAe part with plane-cylinder and plane-plane intersections highlighted in black.