Feature-Based Reverse Engineering of Mechanical Parts

William B. Thompson, Jonathan C. Owen, and H. James de St. Germain

Department of Computer Science
University of Utah
Salt Lake City, UT 84112

University of Utah Department of Computer Science
Techical Report UUCS-95-010 (revised), November 6, 1995.

Abstract

Reverse engineering of mechanical parts requires extraction of information about an instance of a particular part sufficient to replicate the part using appropriate manufacturing techniques. This is important in a wide variety of situations, since functional CAD models are often unavailable or unusable for parts which must be duplicated or modified. Computer vision techniques applied to 3–D data acquired using non-contact, three-dimensional position digitizers have the potential for significantly aiding the process. Serious challenges must be overcome, however, if sufficient accuracy is to be obtained and if models produced from sensed data are truly useful for manufacturing operations. This paper describes a prototype of a reverse engineering system which uses manufacturing features as geometric primitives. This approach has two advantages over current practice. The resulting models can be directly imported into feature-based CAD systems without loss of the semantics and topological information inherent in feature-based representations. In addition, the feature-based approach facilitates methods capable of producing highly accurate models, even when the original 3–D sensor data has substantial errors.

Keywords: reverse engineering, surface fitting, feature-based CAD

CAD models are often unavailable or unusable for parts which must be duplicated or modified. This is a particular problem for long life cycle systems for which spare part inventories have been exhausted and original suppliers are unable or unwilling to provide custom manufacturing runs of spare parts at affordable prices and in a timely manner. For many parts, either CAD systems were not used in the original design or the documentation on the original design is otherwise inadequate or unavailable. For a variety of reasons, CAD models, even when they exist, may not be sufficient to support modification or manufacturing using modern methods. Finally, shop floor changes to the original design may mean that the original CAD model no longer accurately reflects the geometry of the part. *Reverse engineering* techniques can be used to create CAD models of a part based on sensed data acquired using three-dimensional position digitization techniques. Part-to-CAD reverse engineering allows up to date NC fabrication plus easier modification of the design than would otherwise be possible. Successful instances include everything from sporting goods to aircraft parts.

Reverse engineering of solid objects traces its roots back to the pantograph, which uses a mechanical linkage to duplicate arbitrary geometric shapes at any predetermined scale. Copy lathes and mills are more contemporary and automated versions of the pantograph. In a copy lathe, a mechanical stylus is moved along a template specifying a 1–D profile. The position of the cutter is adjusted based on this template, producing a

This work was supported by the Advanced Research Projects agency under Army Research Office grant number DAAH04-93-G-0420 and by the Office of Naval Research under grant N00014-93-1-0950.

revolute object with the same profile. A copy mill typically moves a stylus over a surface, using the height of the surface to set the z-axis in a 3-axis mill, thus making a copy of the original object. Several vendors have produced copy mills which use non-contact sensors. These systems have the added advantage of storing the sensed profile, so that an object can be duplicated many times without repeated scanning. Copy lathes and mills duplicate a physical part without producing any intermediate model of the geometry of the part, other than stylus position or 3–D points acquired with a non-contact sensor. While some can produce NC code capable of driving other lathes and mills, none can produce a CAD model of an existing part. Such models are desirable for a number of reasons. Modifications to the part cannot easily be done at the level of NC code. Even if the part is to be duplicated as is, refixturing and hidden concavities often lead to situations in which multiple scans of an object's shape must be combined into a single, consistent representation. Some shape properties such as deep holes will not be accurately measured by either mechanical styli or non-contact sensors.

The most straightforward approach to generating a reverse engineered geometric model of a mechanical part involves a designer or engineer making measurements using traditional devices such as calipers and gauges and entering the results into a standard CAD system. When high precision is required, contact coordinate measuring machines (CMMs) are often used. Positional accuracy on the order of ± 3 microns locally and ± 14 microns corner to corner is possible, but sensing of a large number of points is extremely slow and expensive damage can be done if the probe is not maneuvered towards the object along an appropriate path. More recently, non-contact CMMs produced by companies such as Cyberware, Digibotics, and Laser Designs have significantly increased the speed with which data can be collected. These devices project a spot or line of light and use triangulation to determine range. While less accurate than contact CMMs, the best are capable of positional accuracy exceeding ± 50 microns. Non-optimal surface properties can degrade this, while deep concavities, discontinuous surface orientation, surface geometries forcing oblique viewing angles, or outright occlusion will cause data to be missing entirely. For comparison, commonly available NC milling machines can achieve precisions of $\pm$2-10 microns for hole and bore spacings and can produce cutting accuracies on the order of $\pm$50-250 microns depending on the feature being cut and the tool being used, though special measures can be used to obtain higher precision.

Many of the commercially available systems for the reverse engineering of mechanical parts using automatically acquired three-dimensional position data use rather unsophisticated geometric models. Often, a digitizer is moved along parallel scanning paths and NC code generated to move a cutter along the same 3–D path. In effect, no model other than the raw scan data is used, though preprocessing to remove noisy data points, align scan lines from multiple scans, etc., is usually necessary. More recently, techniques have been developed for fitting parametric surface patches to 3–D position data.[1] The geometric primitives that are used range from simple planes and cylinders[2] to piecewise smooth surface parametric surface patches.[3–5] Sometimes, triangulated meshes are used as an intermediate representation[6–11] Several software *surfacing* packages, including Imageware Surfacer™, Parametric Pro/SCAN-TOOLS™, and Cyberware Cyserf™, have recently become available. These packages fit spline patches to raw data points and format the result for importation of the surfaces into commercial CAD systems.

The current practice of creating models by fitting generic surface patches to scanned data is most appropriate for parts consisting largely of sculptured surfaces. Representing geometry in terms of surface points or collections of parametric surface patches is adequate to describe positional information, but cannot capture any of the higher level structure of the object. It is thus quite difficult to make modifications or to generate efficient and effective process plans automatically. For example, these representations might be able to cap-

ture the shape of a hole, but the fact that it is actually a true, cylindrical hole is not made explicit. As a result, it can be difficult for a designer to do something as simple as change the diameter of the hole. Modification of more complex manufacturing features is even more difficult.

In this paper, we describe an alternate approach for efficiently creating a CAD model of a part with a significant number of specialized manufacturing features. The system is interactive, since some aspects of the reverse engineering process cannot be done based on the part alone and other aspects of the process can benefit significantly from a small amount of human intervention. In a sense, we provide a set of *electronic calipers* to be used as a smart measuring tool, specialized to the job of creating CAD/CAM models. The system is effective because it analyses 3–D sensor data using knowledge of manufacturing processes and modeling techniques.

Our main innovation is to use manufacturing features as the geometric primitives fit to scanned data, rather than using triangulated meshes or parametric surface patches. This leads to four important advantages:

- *Appropriateness for complex parts.*

  Many complex parts can be described naturally and compactly in terms of manufacturing features. A feature-based reverse engineering system can more easily generate models of such parts than can a system intended for more general free-form geometries.

- *Ease of importation into existing feature-based CAD systems.*

  Several commercially available CAD systems allow parametric modification of manufacturing features in their models. This functionality is lost, however, if imported models consist only of surface patches, without the additional semantics and topology inherent in feature-based representations.

- *Reduced need for complete, robust geometric computations.*

  Substantial effort is involved in converting a collection of surface patches obtained by fitting to scanned data into a form usable by a solid modeler. Topology and other aspects of patch adjacency must be determined, a process often involving substantial hand editing. By generating an object representation in terms of higher-level manufacturing features, correct lower-level B-rep solid models can be generated by existing CAD packages and their generation need not be the responsibility of the reverse engineering system.

- *Accuracy.*

  Non-contact position digitizers are subject to errors which can exceed the tolerances needed in recreating many parts. The local smoothing that is implicit in methods based on fitting surface patches to position data may not be optimal for reducing this sensing noise. The use of manufacturing features as primitives can substantially increase the accuracy of the generated models.

REPRESENTING PART GEOMETRY IN TERMS OF MANUFACTURING FEATURES

A number of modern CAD/CAM systems support some form of *feature-based design*, allowing designers to specify a shape in terms of complex primitives.[12] Design systems of this sort have two clear advantages over

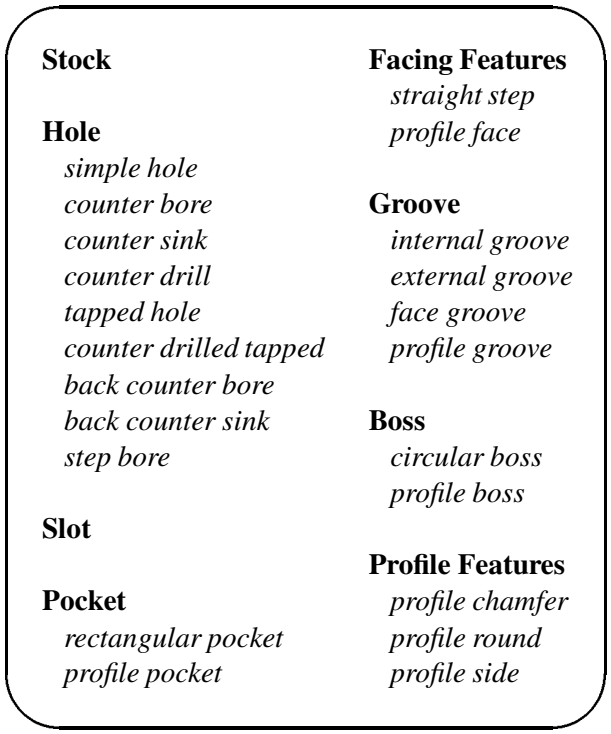

Figure 1: Manufacturing features in the Alpha_1 CAD system.

modeling solely at the level of detailed geometry. They provide a more natural interface for machinists and they allow much more sophisticated automated process planning, since the intent of the designer is clearer. There is as yet no concensus on what specific modeling primitives should consist of in such systems. In an ideal feature-based design environment, the primitives would specify nominal geometry, tolerances, materials and finishes, assembly properties, and other aspects of intent. In commercial CAD packages such as Parametric's Pro/ENGINEER™and Bridgeport's EZFeatureMILL™, and in full function research CAD/CAM systems such as the University of Utah's Alpha_1,[13] the emphasis is on form features that typically have a close association with machining operations.

Figure 1 shows the manufacturing features available in Alpha_1. Each of these feature types has associated with it the appropriate geometric information plus manufacturing specifications such as fillets and chamfers. Free-form surfaces can be freely mixed with these features. Alpha_1 automatically creates NURBS representations for all features and free-form surfaces, intersects surfaces appropriately to create a topologically valid B-rep, and is able to generate with a minimum of human intervention high-quality NC code from models specified using these primitives. Our current reverse engineering system uses a subset of the features in Figure 1. Extending the system to the full set of features listed there will require substantial engineering effort, but is in principle straightforward.

Several methods have been proposed for automatically extracting a high-level, feature-based description from lower-level models of part geometry.[14–16] The goal is usually to start with a conventional volumetric representation of part geometry, derive an alternate representation in terms of features, and then use this

information as an aid in process planning. All of these systems start with an exact representation of surface shape. While they provide useful ideas applicable to creating high-level models from sensed data, none begin to deal with the error and variability present in such data. As described below, we use an interactive approach in our modeling system which avoids the need for automated recognition of manufacturing features.

FEATURE-BASED REVERSE ENGINEERING

Sensor-based reverse engineering of mechanical parts must yield complete and accurate object models appropriate for computer-aided manufacturing. Current commercial practice, which represents geometry in terms of scan lines or meshes of scan points, is inflexible and requires careful coordination between scanning patterns, tool selection, and tool paths. Parametric model fitting techniques proposed to date do not use geometric primitives that are natural to most manufacturing operations. Methods for extracting manufacturing features from lower-level geometric representations are intended to work with existing CAD models, not imperfect sensed data.

Improvements can be made by specializing the recovery of object models to the manufacturing environment. Most machined parts are made using a relatively small number of manufacturing operations, each of a constrained form (Figure 1). Reverse engineering can be done using a form of parametric model fitting, where the primitives correspond to these features. This avoids inconsistencies between actual object shape and what the models are capable of representing, while leading in a natural and obvious way to representations usable in feature-based CAD/CAM systems. The approach we describe here is interactive, which improves performance and allows for human entry of information that cannot be acquired from sensed data alone.

To demonstrate the effectiveness of feature-based reverse engineering, we have created a prototype system called REFAB (**R**everse **E**ngineering – **F**e**A**ture-**B**ased). REFAB allows a user to interactively define a model composed of mechanical features from a set of 3–D surface points. The user specifies the types of manufacturing features present and the approximate location of each feature in the object. REFAB deals with the determination of precise, quantitative parameterization of each feature. The final output is a fully specified model usable by the Alpha_1 CAD/CAM system. Though we have not yet done so, it would be relatively easy to produce models suitable for other CAD packages supporting manufacturing features, such as Pro/ENGINEER™. The ability to create feature-based models in a more generic form awaits futher progress on standardization efforts such as PDES/STEP.

Figure 2 shows the user interface for the REFAB system. Though all modeling computations are done on 3–D point cloud data obtained from position digitizers, user interaction is facilitated by generating a triangulated mesh from the points,[6] and then using standard rendering techniques to create synthetic views from a variety of vantage points. While the triangulated mesh lacks the accuracy and structure of a high-quality CAD model, it generates rendered views with sufficient realism to allow users to easily indicate features of interest.

The series of small images along the top corresponds to alternate views of the same object and allows the user to specify a current working view. REFAB maintains a single, internal coordinate system and views can be switched at any time to provide a better perspective on whatever feature the user is currently interested in. The set of buttons at the lower left corresponds to the set of features the system is able to model. To model a

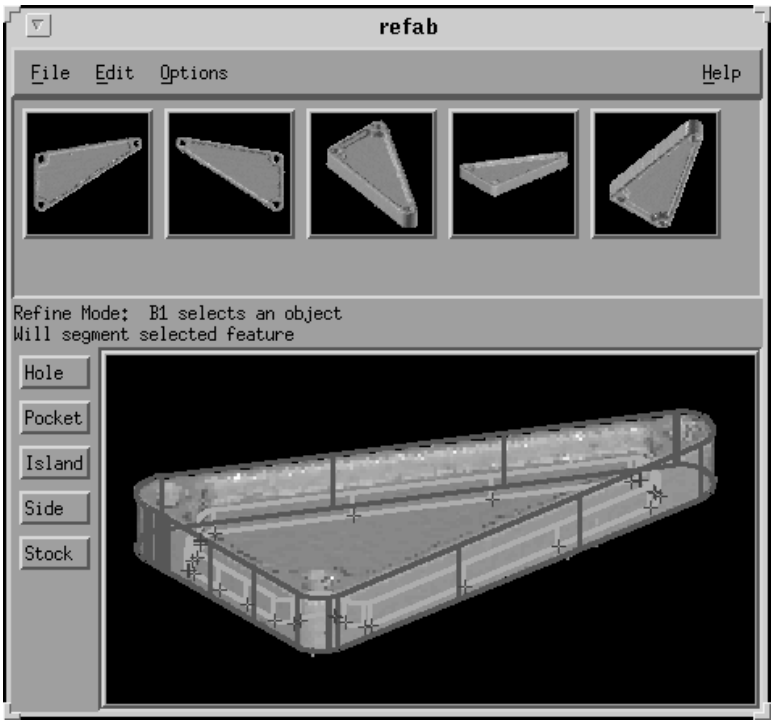

Figure 2: REFAB user interface.

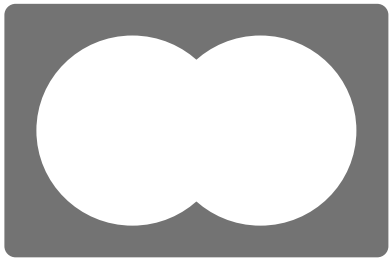

Figure 3: Two interacting holes or one pocket?

feature, the user selects a feature type and a view in which the feature can be seen on the object. The panel on the lower right will show the selected view and all previously modeled features. The mouse is used to specify enough points on the displayed image to indicate the approximate location and shape of the feature. REFAB will then analyze the 3–D position data to provide an optimal parameterization of the feature, render the feature on the display, and then prompt for the next feature to be modeled.

While a fully automated system might seem desirable, there are two aspects of modeling for manufacturing that are infeasible based on automatic processing of sensed data alone. Figure 3 shows a downward-looking view of a plate with an opening in the middle. The opening can be represented exactly using either two holes or a single profile pocket. To choose the preferable representation requires a rather complex understanding of dimensions, tolerances, and manufacturing costs. Next, consider a part which contains seven through holes of identical diameter, four of which mate with locating pins on another part, two of which

stack with holes on parts to either side to form a conduit for oil, with the remaining hole providing access for a flexible cable that runs from one side of the part to the other. The tolerances and finishes required vary enormously. Cost effective fabrication requires that this information be understood and accounted for in the manufacturing process plan. The REFAB system acknowledges the need for human intervention, but frees the user from most of the tedious, quantitative analysis that can be done faster, easier, and more accurately by automated tools.

The current version of REFAB is limited to five common types of $2\frac{1}{2}$-D features: *stocks*, *simple holes*, *profile pockets*, *profile islands*, and *profile sides*. Profile features are extrusions of arbitrary planar curves. A profile island is a special kind of boss. It is defined only within the context of a pocket and specifies a volume to be "skipped" when the pocket is milled. A profile side represents a simple side cut (no plunging), and is typically used to trim stock down to the outside shape of a part. The features are typical of those in parts machined using 3-axis mills for simple drilling and parallel sided cutting. Features can have different orientations, as would occur with refixturing with a 3-axis milling.

SEGMENTATION, FITTING, AND REFINEMENT

In REFAB, the first step in reverse engineering a machined part is to define the stock from which the part is to be cut. Currently, we support only block stock and determine the dimensions using a straightforward bounding box computation on the position data. Extensions to standard stock sizes and other stock shapes would be straightforward. The remaining features require a more careful fit to the position data. Three interrelated problems must be solved in order to accurately model a particular manufacturing feature: determination of feature type, segmentation of relevant 3–D points, and model fitting. The user specifies the feature type and approximate location using REFAB's control panel. Thus, no automatic feature recognition is required. The segmentation and fitting operations proceed automatically, using an iterative refinement process.

The accuracy of model fitting depends on the constraints that are used as part of the approximation process. When fitting surface patches to 3–D points, these constraints are expressed in terms of the functional form of the patches (i.e., planar, cylindrical, conic, etc.) or in terms of some measure of smoothness as for splines. An inappropriate choice of constraints or an incorrect partitioning of the original points prior to fitting individual surface patches can lead to models that are inaccurate representations of the underlying shape. REFAB avoids most difficulties associated with data partitioning by using a robust, top-down segmentation technique. Two classes of constraints are used in the model fitting process. The first exploits the $2\frac{1}{2}$-D nature of the part geometry. Once the orientation of each $2\frac{1}{2}$-D feature has been determined, the remainder of the analysis for that feature can be done in a two-dimensional space in which the 3–D geometry has been projected along the axis of feature orientation. The second class of constraints is based on the use of manufacturing features as primitives and knowledge about how designers typically express the geometry in such features when parts are initially created.

Each $2\frac{1}{2}$-D feature has an orientation. Currently, we allow for this orientation to be specified with respect to some flat portion of the part or with respect to the part's fixturing while being scanned. The user is able to specify the orientation of a flat surface by making mouse clicks on the surface in any of the synthetic views generated by rendering the triangulated mesh representation of the original 3–D data points. We save the 3–D location of every pixel in the rendered views and thus know the 3–D surface location corresponding to each mouse click. A plane is fit to these points, using the least median squares (LMedS) method, which

largely eliminates the effect of outliers in the selected points.[17] The same approach is used to allow the user to specify planar aspects of features such as pocket bottoms.

Once the user has specified the orientation of a $2\frac{1}{2}$-D feature, he or she then indicates a rough outline of the feature by clicking on a sampling of points along the contour of the feature, as indicated in any of the views available in the user interface. Each click is projected back into a line in the 3–D coordinate system in which the sampled data is represented, using the viewing model which generated the view on which the click was made. Each of these lines in three-space is intersected with the plane indicating the orientation of the feature, yielding a set of 2–D points corresponding to the contour.

Hole features are fit to these points by producing an initial estimate of the hole center based on the center of mass and an initial estimate of the hole radius based on the average distance to the selected points from the center of mass. These estimates are used as the starting point for a three degree-of-freedom (two for the hole center, one for the radius), non-linear optimization algorithm based on the generalized simplex method.[18] The criteria function that is minimized is the sum of the squared distances from each selected point to the circle. We have found no need to go to more sophisticated, maximum-likelihood data fitting.[19] Profile pockets, profile islands, and profile sides require a 2–D closed profile curve as part of their specification. This is found from the user indicated points by fitting a Bezier curve.[20]

Fitting a parameterized feature model to sensed data requires a decision as to what data points should be considered to lie on the feature and which values are parts of other features. Most other approaches to dealing with position data use some form of bottom up segmentation procedure.[21,22] Faces on polyhedral objects are found with plane fitting techniques. Curved faces are found using grouping operations which combine collections of points into surfaces, followed by detection of lines of orientation discontinuities. However, few mechanical parts are polyhedra. For curved surfaces, segmentation based on orientational discontinuities is problematic due to noise effects in most range sensors, which produce substantial local variations in surface normals. This problem is particularly acute at surface boundaries, where reliable information is essential for bottom-up processing.

Since in our case the user has specified an approximate feature type and location, we can use a much more reliable top-down segmentation approach. Given an approximate feature parameterization, we select those position points that are close to the surface of the estimated feature in *both* distance and orientation. The combination gives a much better indication of points that are really part of the feature than would either property alone. For example, consider the problem of finding those sensed points on the wall of a drilled hole. Clearly, we want to consider only those points near the expected location of the hole. Using only a distance check, however, will inevitably include some points on the surface through which the hole was drilled, near the rim of the hole. An orientation check quickly discards these points. Additional improvements are obtained by further restricting the distance check, based on per-feature information about where position data is most likely to be accurate. In the case of the hole, data near the rim and deep within the hole is most suspect. An initial segmentation is done using a large tolerance for distance and orientation, but only using those parts of the user-specified model which are expected to yield the best sensed data. As the estimate of feature parameters is refined, the position data can be re-segmented using tighter tolerances on distance and orientation, while reducing or eliminating the restrictions on which parts of the feature surface to consider.

Figure 4 illustrates the interaction between the top-down segmentation procedure and model fitting. A preliminary segmentation of the position data is done based on the user-provided initial feature description, using the distance and surface orientation criteria as described above. Individual surface points are associated

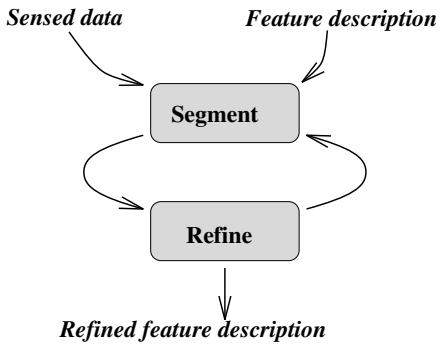

Figure 4: Iterating the segment/refine cycle.

with particular surfaces of the user-selected features. In our current implementation, three distinct surface types are possible. Separate methods exist for fitting each surface type to the segmented data:

- *Planar surfaces.* To avoid the computational complexity associated with LMedS robust plane fitting, we use a simpler trimmed distribution least-squares approximation. A least-squares fit to the data points is done using the familiar eigenvector method. We then compute the residuals associated with each data point and remove a percentage of the points that are furthest away from the fit plane. A second least-squares approximation is done to this reduced set of points, yielding the final plane fit. Combined with the initial data segmentation, this two-step process minimizes the effect of outliers almost as well as a full least-median-squares optimization.

- *Holes.* Holes are fit to data points in the same manner that they are fit to user indicated hole contours. First, the hole orientation relative to some planar surface on the part or relative to the part fixturing is determined. The data points are then projected along this direction. Finally, the center and radius of the circle best fitting the projected points are found using standard non-linear optimization techniques. Though we do not currently do so, the optimization can be made more robust to outliers by using a non-convex optimization function instead of the sum-of-squared distances currently employed.

  Extruded Profiles. As with simple holes, profile features are defined in terms of an orientation and a 2–D contour. The initial, user-specified contour is represented in terms of a Bezier curve. Segmented points likely to correspond to a particular profile side are projected into 2–D along the sweep direction of the profile feature. The data points are sorted based on the parameter value of the nearest point on the Bezier curve.[23] Sequences of points which can accurately be approximated by line segments are identified.[23] The remaining points corresponded to curved portions of the profile. An attempt is made to fit each of these segments using one, two, or three constant radius arcs of alternating curvature. If this fails, the segment is fit with a general Bezier curve .

Next, the position data is re-segmented by scanning the entire set of 3–D points for those that are consistent with the revised model, given a tighter set of tolerances than used in the previous iteration. Segmentation and refinement alternate until preset tolerance bounds are met for the segmentation process.

EXPERIMENTAL RESULTS

Few if any of the publications describing part-to-CAD reverse engineering address the issue of modeling accuracy, despite the critical role of design verification in the overall reverse engineering process.[24] In order to quantitatively evaluate the accuracy of the models obtainable using the feature-based modeling approach, we started with parts for which we had access to the original CAD models.[25] Instances of these parts were carefully machined out of aluminum using a 3-axis NC mill. Surface points on the parts were measured using a non-contact laser digitizer. New CAD models for each part were generated using the REFAB system. Finally, the geometric differences between the original and recovered models were computed. This was done by registering the two models based on lining up planar faces in each.[26] We then generated a dense, uniformly sampled set of points on the reverse engineered model. Standard CAGD techniques were used to find the distance to the closest surface point on the original model. RMS and worst-case distances were reported for each surface making up the reverse engineered model and for the model as a whole.

We have tested the REFAB system on several machined parts originally designed for the Utah mini-Baja and formula SAE racing vehicles. Results from two of these parts are presented here. While the parts are relatively simple, they provide an adequate test of the accuracy and usability of our system. Figure 5 shows the shock mounting plate that forms a linkage in one version of the rear suspension of the vehicle. To better fit our workspace requirements, a special plate was made that was three quarters the size of the one used on the vehicle itself, yielding a part that was approximately 17.75 cm x 7.5 cm x 2 cm. The second object is part of the vehicle's steering arm assembly and is approximately 10 cm x 5 cm x 2 cm (Figure 9).

Position data was acquired with a DIGIBOT II laser position digitizer. The DIGIBOT II has a nominal measurement accuracy of $\pm$50 microns (1 σ) under optimal conditions. In practice we have observed accuracies on the order of $\pm$50-300 microns, depending on the nature and shape of the surface at that point. For evaluation purposes, we produced special versions of both parts without chamfers and threads, which were too small to be accurately measured with the DIGIBOT system. To remove specularities that cause problems for most current range finding systems, parts were sprayed with a penetrant process developer (Sherwin DUBL-CHEK D-100), which leaves a thin, talcum-like coating. Multiple views were taken of each part and transformed into common point-cloud data sets, using a registration procedure similar to that in reference 26. The reverse engineering of the shock plate involved the use of 143,140 3–D points. 44,578 points were used for the steering arm. Figures 6 and 10 show samplings of both point sets, rendered so that nearer points are brighter.

Figures 8 and 12 are wire frame drawings generated from the reverse engineered CAD models produced by REFAB for the two parts. To emphasize that the recovered CAD models are feature-based and not just arbitrary surface representations, Figures 13 and 14 show exploded views of the two models indicating the separate features making up each object. The shock plate is a fairly simple object with an outer contour defined by a profile side, two symmetric profile pockets that serve to lighten the part, and three mounting holes. The steering arm has an outer profile side with both smooth contours and sharp corners, one large hole and one smaller hole drilled normal to the stock, and two small holes drilled in a perpendicular orientation. Tables 1 and 2 show the quantitative deviation between the reconstructions and the original CAD model.

Figure 5: Shock plate: original part.

Figure 9: Steering arm: original part.

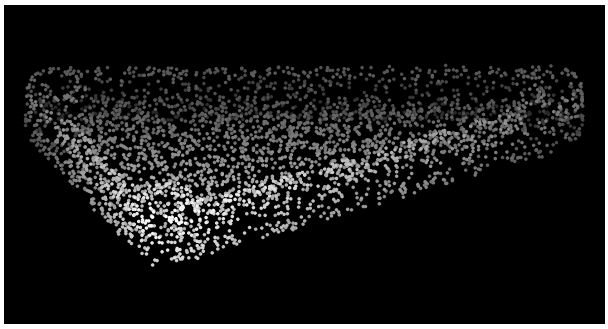

Figure 6: Shock plate: sensed 3–D position points.

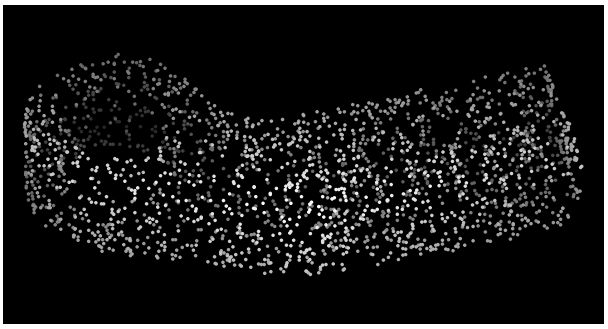

Figure 10: Steering arm: sensed 3–D position points.

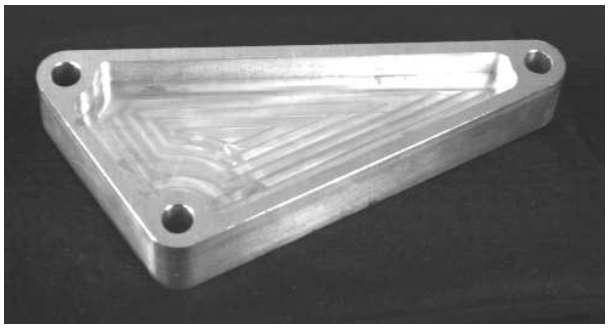

Figure 7: Shock plate: reverse engineered part.

Figure 11: Steering arm: reverse engineered part.

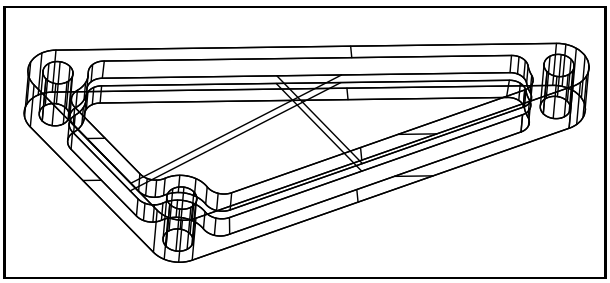

Figure 8: Wire frame rendering of reverse engineered shock plate.

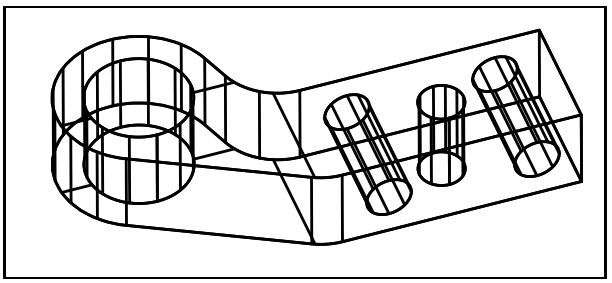

Figure 12: Wire frame rendering of reverse engineered steering arm.

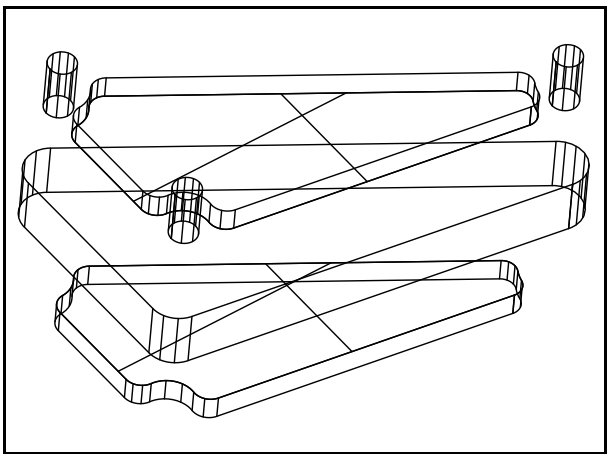

Figure 13: Exploded view of the features making up the reverse engineered shock plate.

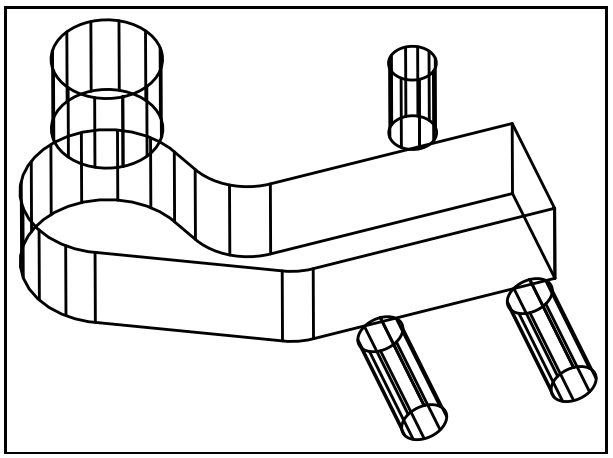

Figure 14: Exploded view of the features making up the reverse engineered steering arm.

Shock plate	RMS	worst-case
overall	61	350
outer profile	48	118
top cap	6	6
bottom cap	29	29
pocket 1 bottom	6	6
pocket 1 side	114	350
pocket 2 bottom	43	43
pocket 2 side	141	350
hole 1	86	119
hole 2	72	104
hole 3	103	151

Table 1: Modeling error – reverse engineered shock plate (microns).

Steering arm	RMS	worst-case
overall	59	127
top	66	66
bottom	66	66
outer profile	63	127
large hole	35	57
small top hole	22	57
side hole 1	48	88
side hole 2	35	80

Table 2: Modeling error – reverse engineered steering arm (microns).

DISCUSSION AND CONCLUSIONS

The use of manufacturing features as geometric primitives in part-to-CAD reverse engineering systems provides substantial advantages in accuracy and usability. In a prototype system, we were able to reverse engineer CAD models with an accuracy often exceeding that of the precision of the sensor used to acquire raw data about part shape. The models which are produced are feature-based, providing a higher level description of part geometry and allowing easy importation into feature-based CAD systems.

Our system does not yet deal with secondary feature properties such as taps, chamfers, fillets, and rounds. Each of these involves small scale geometry that requires specialized gauges for accurate measurement. Once measured, however, the feature based representation allows for easy addition of this information to the model without the tedious surface blending that would be necessary if large-scale geometry were represented only as an unorganized set of surface patches.

It is important to note that the technique we are proposing here deals with only one aspect of the part-to-CAD reverse engineering process. Better methods are needed for deciding what sensors to use (CMMs, laser scanners, x-ray tomography, etc.), squeezing accuracy out of sensors that are available, and registering multiple scans into a common coordinate system. Open problems remain in combining free-form surfaces with manufacturing features, particularly with regards to segmentation and surface blending. Finally, almost no attention has been paid to automating tools for the production of Technical Data Packages (TDP's) specifying ancillary information such as materials, finish, tolerances, etc.

ACKNOWLEDGMENT

Thomas C. Henderson suggested to us the idea of interactive feature specification. Peter-Pike J. Sloan created the user interface. Alyosha Efros and Brian Morris provided software support. Samuel Drake gave us important insights into the manufacturing process and assisted in the creation of test objects. The Alpha_1 group provided technical support on geometric modeling and manufacturing.

References

[1] R. Broacha and M. Young, "Getting from points to products," *Computer-Aided Engineering*, July 1995.

[2] P. N. Chivate and A. G. Jablokow, "Solid-model generation from measured point data," *Computer-Aided Design*, September 1993.

[3] B. Sarkar and C. H. Menq, "Smooth-surface approximation and reverse engineering," *Computer Aided Design*, vol. 23, pp. 623–628, November 1991.

[4] L. Piegl, "On NURBS: A survey," *IEEE Computer Graphics & Applications*, pp. 55–70, January 1991.

[5] M. Lounsbery, S. Mann, and T. DeRose, "Parametric surface interpolation," *IEEE Computer Graphics and Applications*, September 1992.

[6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Computer Graphics, SIGGRAPH '92*, vol. 26, July 1992.

[7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," in *Computer Graphics, SIGGRAPH '93*, vol. 27, August 1993.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Piecewise smooth surface reconstruction," in *Computer Graphics, SIGGRAPH '94*, vol. 28, August 1994.

[9] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *International Journal of Image and Vision Computing*, vol. 10, pp. 145–155, April 1992.

[10] Y. Chen and G. Medioni, "Integrating multiple range images using triangulation," in *Proc. ARPA Image Understanding Workshop*, pp. 951–958, April 1993.

[11] G. Turk and M. Levoy, "Zippered polygon meshes from range images," Tech. Rep. CSL-TR-94-609, Stanford University, February 1994.

[12] J. J. Shah, "Assessment of features technology," *Computer-Aided Design*, June 1991.

[13] S. Drake and S. Sela, "A foundation for features," *Mechanical Engineering*, vol. 111, January 1989.

[14] A. Fischer and M. Shpitalni, "Encoding and recognition of features by applying curvatures and torsion criteria to boundary representation," in *ASME 1992 Winter Annual Meeting – Symposium on Concurrent Engineering*, pp. 69–84, November 1992.

[15] J. Vandenbrande and A. Requicha, "Spatial reasoning for the automatic recognition of machinable features in solid models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1269–1285, December 1993.

[16] W. Regli, S. Gupta, and D. Nau, "Feature recognition for manufacturability analysis," Tech. Rep. ISR TR94-10, University of Maryland, February 1994.

[17] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York: John Wiley & Sons, 1987.

[18] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C*. Cambridge, MA: Cambridge, 1991.

[19] R. M. Bolle and B. C. Vemuri, "On three-dimensional surface reconstruction methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1–13, January 1991.

[20] P. J. Schneider, "An algorithm for automatically fitting digitized curves," in *Graphics Gems* (A. S. Glassner, ed.), Academic Press, 1990.

[21] P. Besl and R. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 167–192, March 1988.

[22] M. Suk and S. Bhandarkar, *Three Dimensional Object Recognition from Range Images*. New York: Springer-Verlag, 1992.

[23] P. J. Schneider, "Solving the nearest-point-on-curve problem," in *Graphics Gems* (A. S. Glassner, ed.), Academic Press, 1990.

[24] K. A. Ingle, *Reverse Engineering*. McGraw-Hill, 1994.

[25] W. B. Thompson and J. C. Owen, ""Hard-copy" benchmark suite for image understanding in manufacturing," in *Proc. ARPA Image Understanding Workshop*, November 1994.

[26] H. Y. Shum, K. Ikeuchi, and R. Reddy, "Virtual reality modeling from a sequence of range images," in *Proc. ARPA Image Understanding Workshop*, pp. 1189–1198, November 1994.