

Research

Reverse engineering of geometric models—an introduction

Tamás Várady, Ralph R Martin* and Jordan Cox†

In many areas of industry, it is desirable to create geometric models of existing objects for which no such model is available. This paper reviews the process of reverse engineering of shapes. After identifying the purpose of reverse engineering and the main application areas, the most important algorithmic steps are outlined and various reconstruction strategies are presented. Pros and cons of various data acquisition techniques are described with related problems of boundary representation model construction. Specific issues addressed include characterization of geometric models and related surface representations, segmentation and surface fitting for simple and free-form shapes, multiple view combination and creating consistent and accurate B-rep models. The limitations of currently known solutions are also described, and we point out areas in which further work is required before reverse engineering of shape becomes a practical, widely-available engineering tool. © 1997 Elsevier Science Ltd. All rights reserved.

Keywords: CAD, geometric modelling, reverse engineering, scanning, segmentation, surface fitting, boundary models

INTRODUCTION

Reverse engineering is a rapidly evolving discipline, which covers a multitude of activities. In this paper we will only be concerned with reverse engineering of shape, but a broader interpretation of the term to involve understanding of design intents and mechanisms is also possible. While conventional engineering transforms engineering concepts and models into real parts, in reverse engineering real parts are transformed into engineering models and concepts. The advantages of the extensive use of CAD/CAM systems need not be

reiterated here. The existence of a computer model provides enormous gains in improving the quality and efficiency of design, manufacture and analysis. Reverse engineering typically starts with measuring an existing object so that a surface or solid model can be deduced in order to exploit the advantages of CAD/CAM technologies.

There are several application areas of reverse engineering. It is often necessary to produce a copy of a part, when no original drawings or documentation are available. In other cases we may want to re-engineer an existing part, when analysis and modifications are required to construct a new improved product. In areas where aesthetic design is particularly important such as in the automobile industry, real-scale wood or clay models are needed because stylists often rely more on evaluating real 3D objects than on viewing projections of objects on high resolution 2D screens at reduced scale. Another important area of application is to generate custom fits to human surfaces, for mating parts such as helmets, space suits or prostheses.

It seems important to clearly distinguish between the concepts of a 3D copier and a 3D scanner. A photocopier takes a piece of paper and produces another piece of paper just like the original. A 3D copier is a device which takes a solid object and makes another one of just the same shape (let us ignore material). In fact, copy machining has been a well established technology for a long time. A scanner however, in 2D, not only inputs a page of text into the computer, but can also recognize the characters and figures, thus providing a text file and graphical structures. Similarly, a 3D scanner will not only capture raw data from the object, but the data will be interpreted and some computer model will be created. Now, not only may a single copy be generated, but knowledge of the shape is obtained, and thus we can derive new shapes, make variations, analyse properties and determine characteristic quantities such as volume or surface area.

The ultimate goal of reverse engineering systems is to realize an intelligent 3D scanner. However, there is a long way to go. Even capturing shape and translating it into a CAD model is a difficult and complex problem. In spite of several encouraging partial results in particular

Computer and Automation Research Institute, Hungarian Academy of Sciences, 1111 Budapest, Kende u. 13-17, Hungary

* University of Wales, Cardiff, PO Box 916, Cardiff CF2 3XF, UK

† Brigham Young University, 133 CB, Provo, UT 84602, USA

Paper received: 15 October 1995. Revised: 15 May 1996

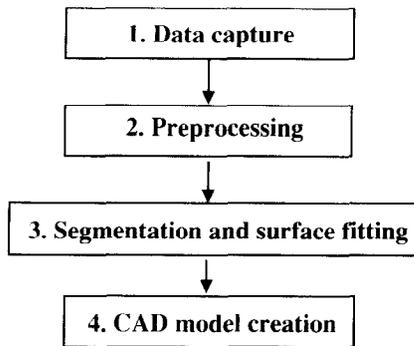


Figure 1 Basic phases of reverse engineering

areas, a fully automatic solution to build a complete and consistent CAD model is still a goal. The purpose of this paper is to describe the most important elements of a reverse engineering system and to identify problems, which still require further research. At the same time, we attempt to summarize the basic achievements of current reverse engineering research, as well. The reverse engineering procedure can be characterized by the flowchart in *Figure 1*.

Of course this sequence is fairly notional. In fact, these phases are often overlapping and instead of the sequential process shown, several iterations are required. Nevertheless, this outline may help the reader to understand the information flow and serves as a basis for organizing the content of our paper.

A crucial part of reverse engineering is *data acquisition*. After reviewing the most important measuring techniques, the relative merits and difficulties associated with these methods are discussed. Often, methods for reverse engineering are developed based on simulated data acquisition only. Our experience is that a certain amount of reservation is needed in such cases, as actual physical measurements may display many problems and undesirable side effects not present in artificial data.

As was indicated earlier, the main topic of this paper is the geometric part of reverse engineering. Data structures for representing shape can vary from point clouds to complete boundary representation models. We give later a *hierarchy of shape models*. This is particularly important since the representation chosen fundamentally determines the computational algorithms applied to the data sets.

The most critical parts of reverse engineering are *segmentation* and *surface fitting*. By means of these processes, data points are grouped into sets to which an appropriate single surface can be fitted. We believe that segmentation and surface fitting methods must be carefully matched to each other. A range of techniques and problems will be described in the following sections, including methods for various surface representations used in CAD ranging from planes and quadrics to composite free-form surfaces.

Data acquisition systems are constrained by physical considerations to acquire data from a limited region of an object's surface. Hence, multiple scans must be taken to completely measure a part. See the section on *combining multiple views*.

The problems of *creating geometric models* will be discussed in the last section. There are various representations providing approximate or incomplete models which may be sufficient for certain applications, such as computer vision, animation, collision checking, etc.

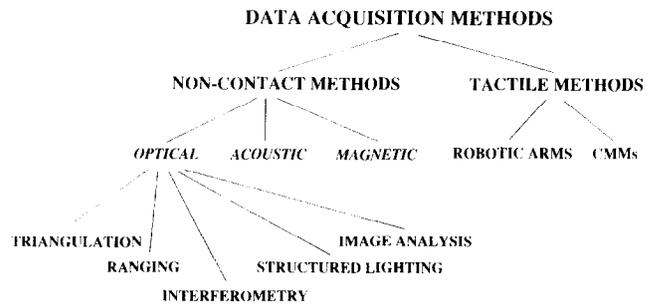


Figure 2 Classification of data acquisition methods

For CAD purposes these will not be adequate. Here we restrict our scope of interest and concentrate on accurate and consistent boundary representation models, using standard surfaces acceptable by commercial CAD/CAM systems. Identifying sharp edges, adding blends, providing proper continuity where smooth connections are needed, tidying up the model and enforcing constraints are all part of the problem.

Finally, in the conclusion we present our view of the current status of this technology and what are the most important research issues. This paper is an overview, so we do not attempt to describe individual topics in detail. We try to concentrate on important conceptual issues, while the reference list at the end will help readers to find the most relevant research contributions.

DATA ACQUISITION

There are many different methods for acquiring shape data, as shown in *Figure 2*. Essentially, each method uses some mechanism or phenomenon for interacting with the surface or volume of the object of interest. There are *non-contact methods*, where light, sound or magnetic fields are used, while in others the surface is touched by using mechanical probes at the end of an arm (*tactile methods*). In each case an appropriate analysis must be performed to determine positions of points on the object's surface from physical readings obtained. For example, in laser range finders, the time-of-flight is used to determine the distance travelled, and in image analysis the relative locations of landmarks in multiple images are related to position. Each method has strengths and weaknesses which require that the data acquisition system be carefully selected for the shape capture functionality desired. This section will discuss the principles of various methods and the next section will address the practical problems of acquiring data. Jarvis' paper²⁹ is a very good survey on the different methods of data acquisition.

Optical methods of shape capture are probably the broadest and most popular with relatively fast acquisition rates. There are five important categories of optical methods we discuss here: triangulation, ranging, interferometry, structured lighting and image analysis.

Triangulation is a method which uses location and angles between light sources and photo sensing devices to deduce position. A high energy light source is focused and projected at a prespecified angle at the surface of interest. A photosensitive device, usually a video camera, senses the reflection off the surface and then by using geometric triangulation from the known angle and distances, the position of a surface point relative to a

reference plane can be calculated. The light source and the camera can be mounted on a travelling platform which then produces multiple scans of the surface. These scans are therefore relative measurements of the surface of interest. Various different high energy light sources are used, but lasers are the most common. Triangulation can acquire data at very fast rates. The accuracy is determined by the resolution of the photosensitive device and the distance between the surface and the scanner. Motavalli and Bidanda⁴² present a reverse engineering strategy using laser triangulation. Moss *et al.*⁴¹ present a detailed discussion of a classic laser triangulation system used to capture shape data from facial surfaces. A discussion of accuracy and applications is also included. The use of laser triangulation on a coordinate measuring machine is presented by Modjarrad³⁸. These references give a broad survey of methods, approaches to and limitations of triangulation.

Ranging methods measure distances by sensing time-of-flight of light beams; practical methods are usually based on lasers and pulsed beams. *Interferometry* methods measure distances in terms of wavelengths using interference patterns. This can be a very accurate method of measurement since visible light has a wavelength of the order of hundreds of nanometres, while most reverse engineering applications distances are in the centimetre to metre range. In principle, other parts of the electromagnetic spectrum could also be used. In practice, a high energy light source is used to provide both a beam of monochromatic light to probe the object and a reference beam for comparison with the reflected light. Moring *et al.*⁴⁰ describe a range finder based on time-of-flight calculations. The article presents some information on accuracy and performance. Jarvis³⁰ presents an in-depth article on time-of-flight range finders giving detailed results and analysis.

Structured lighting involves projecting patterns of light upon a surface of interest and capturing an image of the resulting pattern as reflected by the surface. The image must then be analysed to determine coordinates of data points on the surface. A popular method of structured lighting is shadow Moiré, where an interference pattern is projected onto a surface producing lighted contour lines. These contour lines are captured in an image and are analysed to determine distances between the lines. This distance is proportional to the height of the surface at the point of interest and so the coordinates of surface points can be deduced. Structured lighting can acquire large amounts of data with a single image frame, but the analysis to determine positions of data can be rather complex. Will and Pennington⁶³ use grids projected onto the surface of objects to determine point locations. Wang and Aggarwal⁶¹ use a similar approach but use stripes of light and multiple images.

The final optical shape capture method of interest is *image analysis*. This is similar to structured lighting methods in that frames are analysed to determine coordinate data. However, the analysis does not rely on projected patterns. Instead, typically, stereo pairs are used to provide enough information to determine height and coordinate position. This method is often referred to as a *passive method* since no structured lighting is used. *Active methods* are distinguished from passive methods in that artificial light is used in the acquisition of data. Correlation of image pairs and landmarks within the images are big difficulties with this method and this is

why active methods are preferred. Another image analysis approach deals with lighting models, where an image is compared to a 3D model. The model is modified until the shaded images match the real images of the object of interest. Finally, intensity patterns within images can be used to determine coordinate information. There is a vast amount of literature on stereo imaging, and we just cite four papers that address this technique. Nishihara⁴³ uses a real-time binocular stereo matching algorithm for making rapid range measurements. Posdamer and Altschuler⁴⁵ describe a method for real-time measurement of surface data using stereo methods. Also, see Woodham's work⁶⁵ on shape from shading. Finally, a contribution by Rockwood and Winget⁴⁷ in this special issue describes an energy minimization approach of a mesh to match a collection of 2D images.

Tactile methods represent another popular approach to shape capture. Tactile methods touch a surface using mechanical arms. Sensing devices in the joints of the arm determine the relative coordinate locations. These methods are mostly limited by the measuring device limitations. For example, a 3-axis milling machine can be fitted with a touch probe and used as a tactile measuring system. However, it is not very effective for concave surfaces. There are many different robotic devices which are used for tactile measurement. These methods are among the most robust (i.e. less noise, more accurate, more repeatable, etc.), but they are also the slowest method for data acquisition.

Probably the most popular method is the use of coordinate measuring machines (CMM). These machines can be programmed to follow paths along a surface and collect very accurate, nearly noise-free data. Xiong⁶⁶ gives an in depth discussion of measurement and profile error in tactile measurement. Sahoo and Menq⁴⁹ use tactile systems for sensing complex sculptured surfaces. Butler⁶ provides a comparison of tactile methods and their performance.

The final type of data acquisition methods we will examine are *acoustic*, where sound is reflected from a surface, and *magnetic*, where a magnetic field touches the surface. Acoustic methods have been used for decades for distance measuring. Sonar is used extensively for this purpose. Automatic focus cameras often use acoustic methods to determine range. The method is essentially the same as time-of-flight, where a sound source is reflected off a surface and then distance between the source and surface is determined knowing the speed of sound. Acoustic interference or noise is often a problem as well as determining focused point locations. Dynamic imaging is used extensively in ultrasound devices where a transducer can sweep a cross-section through an object to capture material data internal to an object.

Magnetic field measurement involves sensing the strength of a magnetic field source. Magnetic touch probes are used which usually sense the location and orientation of a stylus within the field. A trigger allows the user to only record specific point data once the stylus is positioned at a point of interest. Magnetic resonance is used in similar applications to ultrasound when internal material properties are to be measured. MRI (magnetic resonance) activates atoms in the material to be measured and then measures the response. Watanabe⁶² uses an ultrasonic sensor for object recognition and Tsujimura *et al.*⁵⁷ place the ultrasonic device on a manipulator.

To sum up, all measuring methods must interact with

the surface or internal material using some phenomenon, either light, sound, magnetism or physical contact. The speed with which the phenomenon operates as well as the speed of the sensor device determines the speed of the data acquisition. The amount of analysis needed to compute the measured data and the accuracy are also basically determined by the sensor type selected. On the technical parameters of various commercial 3D digitizers see the table in Reference 64.

PRACTICAL PROBLEMS OF DATA ACQUISITION

There are many practical problems with acquiring usable data, the major ones being:

- calibration,
- accuracy,
- accessibility,
- occlusion,
- fixturing,
- multiple views,
- noise and incomplete data,
- statistical distributions of parts, and
- surface finish.

Calibration is an essential part of setting up and operating a position measuring device. Systematic sensing errors can occur through lens distortions, non-linear electronics in cameras, and similar sources. Any sensing must be calibrated so as (i) to accurately determine parameters such as camera points and orientations, and (ii) to model and allow for as accurately as possible systematic sources of error. Most of the papers cited present some discussion of accuracy ranges for the various types of scanners, but all methods of data acquisition require accurate calibration. Optical scanners' accuracies typically depend largely on the resolution of the video system used. Distance from the measured surface and accuracy of the moving parts of the scanning system all contribute to the overall measurement error.

Accessibility is the issue of scanning data that is not easily acquired due to the configuration or topology of the part. This usually requires multiple scans but can also make some data impossible to acquire with certain methods. Through holes are typical examples of inaccessible surfaces.

Occlusion is the blocking of the scanning medium due to shadowing or obstruction. This is primarily a problem with optical scanners. However, acoustic and magnetic scanners may also have this problem. Multiple scanning devices are one approach to obviate this problem. See Rioux⁴⁶ and Koivunen³² for methods of eliminating occlusion in optical systems. As well as self-occlusion, occlusion may also arise due to fixturing—typically parts must be clamped before scanning. The geometry of the fixtures becomes a part of the scan data. Elimination of fixture data is difficult and often requires multiple views. Multiple views introduce errors in acquired data because of registration problems (see more details later).

Noise elimination in data samples is a difficult issue. Noise can be introduced in a multitude of ways, from extraneous vibrations, specular reflections, etc. There are many different filtering approaches that can be used. An important question is whether to eliminate the noise

before, after, or during the model building stage. There are times when the noise should not be eliminated at all. Noise filtering, though, is often an unavoidable step in reverse engineering, but note, that this also destroys the 'sharpness' of the data, i.e. typically sharp edges disappear and are replaced by smooth blends, which in some cases may be desirable, but in other cases may lead to serious problems in identifying features. For an example of noise elimination see Koivunen³¹.

A similar problem is restoration of missing data. This is partly necessary due to the above mentioned inaccessibility and occlusion problems. Moreover, because of the nature of optical and even tactile scanning, the data close to sharp edges are also fairly unreliable. Finally there are situations where only parts of a certain surface can be measured, there are missing parts or parts obscured by other elements, but we need to reconstruct the whole surface from just the visible parts. Further ideas on surface extensions, intersections and patching holes are given in the last part of the paper.

Statistical distribution of parts deals with the fact that any given part which is scanned only represents one sample in a distributed population. When reverse engineering methods attempt to reproduce a given shape, the tolerance distribution of the scanned part must be considered. This gives rise to multiple part scans and the averaging of the resulting data. However, it may be somewhat impractical to attempt to sample many parts from a population, and indeed, often only one is available.

The final issue we bring up is surface finish of the part being measured. Smoothness and material coatings can dramatically affect the data acquisition process. Tactile or optical methods will produce more noise with a rough surface than a smooth one. Reflective coatings also can affect optical methods. When scanning human faces noise is often introduced when the light reflects off the eye or spectacles. Hair is an example of a rough surface which presents very difficult problems in scanning.

Imagine an ideal scanner: the object is 'floating' in 3D space, so it is accessible from all directions. The data are captured in one coordinate system with high accuracy, with no need for noise filtering and registration. Possibly, the measurement is adaptive, i.e. more points are collected at highly curved surface portions, etc. Unfortunately, such a device does not exist at present. But, despite the practical problems discussed, it is possible to obtain large amounts of surface data in reasonably short periods of time even today using the methods described. Once the measured data are acquired, the process of recognition and model building can begin. The imperfect nature of the data, particularly inaccuracy and incompleteness, however, makes these steps fairly difficult as will be seen in the following sections.

GLOBAL CHARACTERIZATION OF SHAPE

As indicated earlier the main purpose of reverse engineering is to convert a *discrete* data set into a *piecewise smooth, continuous* model. In this section various aspects of this conversion are described. The discrete data set typically consists of (x, y, z) coordinate values of measured data points. Concerning the *organization of data*, we may classify the point set as either scattered or regular. In the former case the points come from random or manual pointwise sampling. In the latter case the

measurements may have taken place along contours or slicing planes, which results in a sequence of scanlines; within each scanline there is a sequence of points. Alternatively the measurement may provide regularly organized points in two dimensions, as for example does the grid structure of range finders.

Another important issue is the *neighbourhood information*. For example, a regular grid implicitly provides neighbourhood information, which implicitly gives connectivity except at step discontinuities. In a scattered data set the neighbourhood information will be usually absent unless a specific triangulation is also associated with the data points.

Triangulated models are particularly important in computer graphics and animation, which explains why research work in this area has obtained a large momentum recently. Related work on mesh optimization and multi-resolution analysis of triangular meshes can be found among others in Schroeder⁵¹, Hoppe²⁷, Rossignac⁴⁸ and Eck¹⁰. Veron and Leon⁵⁹ describe an algorithm for polyhedron modification using error zones. Guo²¹ prepares approximating polyhedra by using 3D α -shapes¹¹ to capture the topological structure before surface fitting. (See References 21 and 59 in this special issue.)

The exact type of model created depends on the intended use envisaged for the model. One alternative is just to generate a collection of faces, i.e. planar facets or possibly higher order surface patches, without enforcing connectivity or continuity between neighbouring elements and without explicitly storing topological relationships. In this situation small gaps between the elements will not matter, assuming the bulk of the data points is compactly covered, and provided that the neighbourhood information is not needed for further computations. Typical examples include representations using superquadrics⁵⁵ or $z = f(x, y)$ surfaces^{33,50} which are adequate for many computer vision tasks or for surfaces to be used in medical applications (see, for example Reference 41). A variety of surface representations is considered in the review work by Bolle and Vemuri⁵.

In the majority of CAD/CAM applications, particularly in mechanical engineering, connectivities and continuity in the data structure are also very important. For example, many manufacturing methods depend strongly on surface type, which requires a perfect representation of the related data set. Thus it would not be sufficient to represent a plane or a cylindrical hole by approximating facets, since these may be functional faces presumably to be manufactured by grinding or drilling. Gaps larger than a given tolerance may also cause problems since small areas of superfluous material may remain on the surfaces of the machined object. Moreover, aesthetic parts such as car body panels require at least second order continuity between the constituent surface elements, which again must be assured by the constructed mathematical model.

Our interest in this paper is mainly directed towards these higher-level geometric models. We assume an abstract boundary representation (B-rep) model^{24,35}, where the skin of the object is represented as a set of faces, each bounded by one or more loops of edges. Each edge separates two faces and the edges meet in vertices. (We ignore non-manifold data-structures, since we want to deal with objects which have been and can be physically realized.) Within B-reps we may distinguish between partial models and complete models⁵⁶. In the

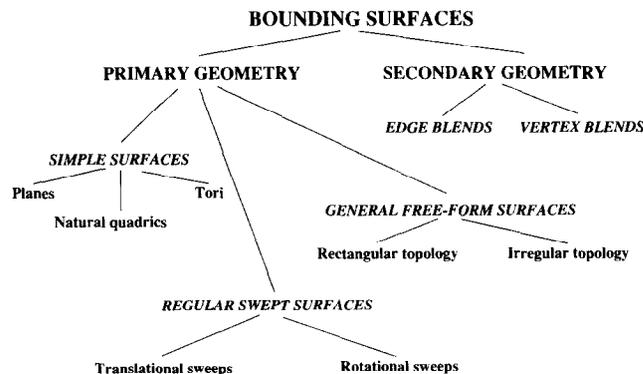


Figure 3 A hierarchy of surfaces

former case the result of the reverse engineering procedure will be a model of just some part of the object, for example, the visible portion of a panel or a well-defined functional detail. In the latter case, which will generally need multiple views to be merged (see later section), a complete solid model is created which contains all boundary elements of the object.

Although ideally we would like to have a fully automatic reverse engineering system which can make decisions and classifications without any user interaction, we believe that at the present state of the art it is very important to have as much *a priori* information about the object as possible. For a human carrying out reverse engineering, it is easy to recognize whether an object has planar faces or not, or whether the whole object is smooth everywhere. For the computer this sort of information is crucial, since it will determine the final model to be constructed and have a significant effect on the efficiency of the computations. The *a priori* information required here may specify what sort of surface elements occur in the object—global characterization—and may provide certain threshold values. For example, suppose we are given a set of measured points from a cylindrical surface (these data are noisy, and so inaccurate). If we are told that cylindrical surfaces may be present, and set our measurement tolerances correctly, we will successfully try to fit a cylinder to the data before considering more general surface types. However, if we do not look for cylinders, or if we set our tolerances incorrectly, this feature is unlikely to be recognized, and the area will be represented as some free-form surface element.

The above reasoning explains that it is important to specify a hierarchy of surface types in order of geometric complexity (see Figure 3). Note that we can extract simple types more reliably than more complex types, so if we think we have found a plane, it is quite likely we are right. This is less certain for more complex types, which also gives us a sensible order in which to look for them.

This hierarchy may be coupled with the following useful abstraction when describing complex objects. (The following concept reflects the authors' own view, which is similar to that held by many other researchers working in this area.) We may say that objects are bounded by relatively large *primary* or functional surfaces. The primary surfaces may meet each other along *sharp* edges or there may be *secondary* or *blending surfaces* which may provide smooth transitions between them⁶⁰. (See a simple example in Figure 4.) Of course, the notion of a sharp edge is also an abstraction: in physical terms a sharp edge does not exist, but one may say that for all edges

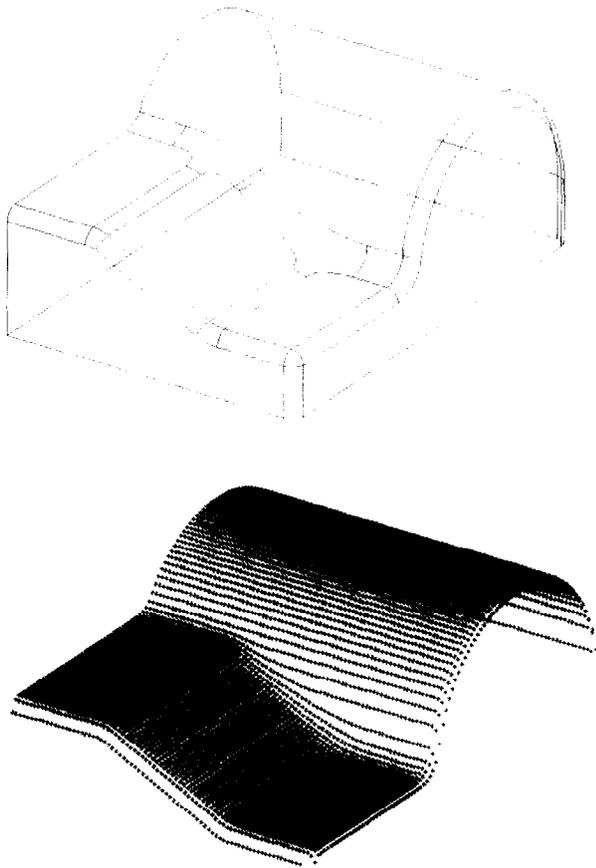


Figure 4 A simple solid object and measured data

where the radius of curvature is less than a very small value, our model should contain sharp edges. (This threshold value will be presumably smaller than the resolution of our measurements.) Alternatively, the user may require that we always fit small blends between neighbouring faces due to the fact that measured data are particularly unreliable along discontinuities.

Returning to the hierarchy of primary surface elements, we restrict our interest to surface types which are used as standard representations in the majority of CAD/CAM systems. We start with simple surfaces which have both simple implicit and parametric representations and conclude with more general piecewise parametric surface representations. The advantage of the simple surfaces, which are *planes*, *natural quadrics* and *tori*, is that they have a simple representation in terms of a few geometrically meaningful parameters. This makes segmentation and surface fitting relatively easy (see later). We prefer to consider the subclass of natural quadrics (spheres, cylinders and cones) since these are the most important in mechanical engineering. General quadrics other than the natural ones rarely occur. Furthermore they are specified by algebraic coefficients without direct geometric meaning, making them more difficult to segment.

Considering more general free-form surface elements, it is worth also specifically considering simple *translational* and *rotational sweeps*, characterized by some profile curve and a sweeping direction or an axis of rotation, respectively. This classification can be extended by recognizing more general sweeps. Finally our most general surface class is that of *composite surface elements* where no geometric or topological regularity can be

recognized. This composite surface element may be a collection of patches or possibly trimmed patches, across which internal smoothness is assured.

We mentioned earlier the surface class of blends, which play a special role in connecting primary geometry. As will be explained later, blends are important not only from a functional point of view, but they are also important in stitching or 'healing' gaps between neighbouring surface elements, and thus providing overall consistency of the geometric model. From a representational point of view we remark that it is worth distinguishing between *edge blends*, which run along edges between surfaces, and *vertex blends*, which serve to provide smooth transition surfaces at junctions where several edge blends meet.

The above classification is also somewhat related to conventional machining operations, which are such that natural motions generate planar cuts (face cutting on a mill) or rotational cuts (lathe) with high accuracy, and the remaining mostly free-form elements are typically machined on 3- and 5-axis milling machines with less accuracy.

To conclude this section, with the current state-of-the-art, it is important to have an *a priori* global characterization of the shape to be reverse engineered, and to have a prior understanding of the measurement process. These are to specify the class of possible surface elements and to set parameters for making decisions about whether a point belongs to a surface. It should be noted, however, that a sophisticated system might be able to adaptively adjust tolerance values for surface fitting.

SEGMENTATION AND SIMPLE SURFACE FITTING

In this section we consider the related problems of segmentation and surface fitting. A good general overview of segmentation is provided by Besl and Jain's survey paper²; another useful source is Shirai⁵². We assume here that the initial data are in the form of a dense set of points sampled from the surface of the object; rather different methods will be required if, for example, points have only been digitized along certain key curves on the surface of the object. (A fully automatic method is likely to produce dense data, as human interaction would be required to choose key curves.) The aim is to produce a higher level representation of the shape of the object, in the form of a set of surfaces. Possibly their boundary curves will also be produced at this stage, but it is also possible that further processing will need to be done at later stages to ensure each surface is properly joined to its neighbours by a boundary curve lying in each surface.

We also assume that the surface of the object can be 'naturally' broken down into various component surfaces, which meet along sharp or smooth edges. As was explained earlier, some of these will be simple surfaces, such as planar or cylindrical surfaces; others will need to be represented by more general free-form surfaces. The tasks to be solved at this stage of shape reconstruction are:

segmentation—to logically divide the original point set into subsets, one for each natural surface, so that each subset contains just those points sampled from a particular natural surface;

classification—to decide to what type of surface each subset of points belongs (e.g. planar, cylindrical); and **fitting**—to find that surface of the given type which is the best fit to those points in the given subset.

It should be clearly noted that these tasks cannot in practice be carried out in the sequential order given above, as, for example, deciding whether a point belongs to a given subset requires some measure of how well it matches the underlying surface the points in the subset represent. Thus, in practice, some approach is needed where each of these problems is solved simultaneously, and either backtracking, iterative, or probabilistic methods are used to finally converge on a consistent answer. A good discussion of these control problems is given by Leonardis *et al.*³³; their approach is to consider several possibilities in parallel, and to find the final answer by solving an optimization problem.

Two basically different approaches to segmentation may be considered, namely *edge-based* and *face-based* methods. The first works by trying to find boundaries in the point data representing edges between surfaces. If sharp edges are being sought, we must try to find places where surface normals estimated from the point data change direction suddenly, while if smooth (tangent-continuous) edges are also possible, we will need to look for places where surface curvatures or other higher derivatives have discontinuity. This technique thus basically attempts to find edge curves in the data, and infers the surfaces from the implicit segmentation provided by the edge curves. A representative example of this approach is described by Smith and Kanade⁵⁴. Another edge-based segmentation technique is presented in this special issue by Milroy *et al.*³⁷, where wrap-around objects are segmented. In this approach, several, user specified 'seed loops' are inflated to obtain edge-loops of faces.

The second technique goes in the opposite order, and tries to infer connected regions of points with similar properties as belonging to the same surface (e.g. groups of points all having the same normal belong to the same plane), with edges then being derived by intersection or other computations from the surfaces. Besl and Jain's³ work is a classical example of the latter approach.

Comparing these two approaches, we may make the following observations (see also Marshall³⁶ for further details of some of the points below). Edge-based techniques suffer from the following problems. Sensor data, particularly from laser-based scanners, are often unreliable near sharp edges, because of specular reflections there. The number of points used for segmenting the data is small, i.e. only points in the vicinity of the edges are used, which means that information from much of the data is not used to assist in reliable segmentation. In turn, this means a relatively high sensitivity to occasional spurious data points. Finding smooth edges, which are tangent continuous, or have even higher continuity, is very unreliable, as computation of derivatives from noisy point data is error prone. On the other hand, if smoothing is applied to the data first to reduce errors, this distorts the estimates of the required derivatives. Thus sharp edges are replaced by blends of small radius which may complicate the edge-finding process; also the positions of features may be moved by noise filtering.

On the other hand face-based techniques have the following advantages. They work on a larger number of points, in principle using all available data. Deciding

which points belong to which surface is a natural by-product of such methods, whereas with edge-based methods, it may not be entirely clear to which surface a given point belongs even after we have found a set of edges (3D edges do not surround a region of space). Typically, this type of approach also provides the best-fit surface to the points as a by-product. Overall, the authors believe that face-based rather than edge-based segmentation is preferable.

In fact, segmentation and surface fitting are like the 'chicken and egg' problem. If we knew the surface to be fitted, we could immediately determine those sample points which belonged to it, by just picking those points which were within a small distance of the surface. If we knew for certain the exact set of points which belonged to the surface, it would be easy to find the best surface class and the best fitting surface within this class. Unfortunately, neither of these holds. The difficulties of automatic segmenting can be seen even when processing the data points from the simple object shown in *Figure 4*. (Note that this is for illustration only, and the density of sample points shown may not be sufficient to actually determine the existence or radii of blends.) To overcome this problem we must either ask for interactive help or apply iterative methods. Within the latter we may distinguish between *bottom-up* and *top-down* methods.

Let us assume that we adopt a face-based approach to segmentation. The class of *bottom-up* methods initially starts from *seed points*. Small initial neighbourhoods of points around them, which are deemed to consistently belong to a single surface, are constructed. Local differential geometric or other techniques are then used to add further points which are classified as belonging to the same surface. Growing stops when there are no more 'consistent' points in the vicinity of the current region. Typically, several such regions may be grown in parallel, and a later step may then be required to merge regions which have grown until they touch, and are found to be compatible in that they represent the same surface.

In practice, during the growing phase, we may also have to be prepared to update our idea of what surface the region represents. For example, a set of points comprising a region may initially be well represented by a plane, but as more points are added, it may be necessary to assume instead that the points in the region belong to a cylinder of large radius. The region is kept, but our idea of to what underlying surface the region belongs can change in type as well as in parameters. Good examples of the state-of-the-art in region growing can be found in Leonardis *et al.*³³ and Sapidis and Besl³⁰. In Fitzgibbon *et al.*¹⁸ (see this special issue) after the segmentation of planar and quadric surfaces the adjacency information is also recovered and converted into a special B-rep data structure.

On the other hand, the class of *top-down* methods starts with the premise that *all* the points belong to a single surface, and then tests this hypothesis for validity. If the points are in agreement, the method is done, otherwise the points are subdivided into two (or more) new sets, and the single-surface hypothesis is applied recursively to these subsets, the process continuing until all generated subsets satisfy the hypothesis. Most approaches to surface segmentation seem to have taken the bottom-up approach, e.g. Besl and Jain³, and while the top-down approach has been used successfully for

image segmentation, e.g. Pietikäinen *et al.*⁴⁴, its use for surface segmentation is less common.

Various problems exist for both of these approaches. In the bottom up case, they include the following: choosing good seed points from which to start growing the surface can be difficult –obviously, a seed point lying on an edge will not be suitable. We need to decide whether to distribute the seed points uniformly, or in some more sophisticated way. If more than one type of surface is being considered, choosing which surface type to use for a region, and possibly changing this surface type as the region grows, requires careful thought. Also, updating the hypothesis based on the points which currently belong to the region must be done carefully –if bad points are wrongly added to the region, this will distort current estimates of the nature of that surface. We do not wish region growing to prematurely stop if an occasional bad point is encountered. Deciding whether to add points to the region can be difficult, as again, the decision is generally based only on local geometry, which is susceptible to noise.

A major problem associated with the top-down approach is choosing where and how to subdivide. Often, the subdivision will be done along a straight line rather than a ‘natural’ boundary, and so in practice, merging steps are also required to re-combine pieces. This leads to edges which are rather jagged in nature. When surfaces slowly and smoothly blend into one another, the subdivisions chosen may be quite some way from the real boundary, and so a lot of extra work may be done subdividing in the wrong place. Another major problem with the top-down approach is that after splitting, surface parameter estimates must be recomputed *ab initio*, whereas with the bottom-up approach, as new points are added to a region, the parameter estimates can be incrementally updated, resulting in a much lower computational effort. At best this problem will waste time, and at the worst, will result in poor segmentation results. Finally, both approaches, unless carefully controlled, are likely to end up representing a complex free-form surface of a type which is not included in the model as many small pieces of say planar or quadric surfaces, which is not the desired result.

Surface fitting for simple surface types generally is done during the segmentation process, as noted earlier, as we can only determine whether an entire subset of points belongs to the same surface if we have an underlying model for that surface which includes both its type, and some estimate of its parameters. Fitting planes to point data, using least-squares methods, is a stable and fairly reliable process. The vision community have next looked to quadrics as an obvious generalization to planes^{3,14}, and even superquadrics⁵⁵. However, a general quadric has many free parameters, and the fitting problem can often be numerically ill-conditioned. In practice, attempting to fit a general quadric to noisy, real world data can lead to poor results, especially in the context of a segmentation scheme. Better results can be obtained by restricting attention to only the natural quadrics, since—as mentioned earlier—each of these can be represented by a smaller number of parameters, which furthermore have some direct geometric (rather than algebraic) significance.

SEGMENTATION AND SURFACE FITTING FOR FREE-FORM GEOMETRY

While it is not easy to do segmentation even for simple surfaces, such as planes and cylinders, further difficulties arise when trying to segment free-form shapes. For the following discussion let us assume that we have a *composite free-form* face, which is smooth internally and bounded by edge loops. These edges may be partly or entirely the boundaries of some underlying patches or they may also be trimming curves cutting across the patch structure. Such trimming curves may have been determined by higher level operations such as intersections, Boolean operations, or blending. For example we may want to reconstruct just a large, single surface with high geometric complexity or we may wish to deal with a composite free-form surface, which has already been separated from the rest of a solid object.

The most widely used parametric surfaces such as Bézier patches and NURBS surfaces map a rectangular parametric domain into 3D, resulting in surface patches with four boundary curves, but complex free-form shapes cannot be represented by a single surface. A composition of several surface pieces is required while maintaining appropriate continuity between the constituent elements. The key issue here is how to do this additional free-form segmentation, i.e. how to find appropriate internal boundaries, which delineate such local regions, and which are representable by single surface patches.

There are four important approaches for free-form segmentation which we will refer to as *global approximating surfaces*, *curve network based surfaces*, *arbitrary topology surfaces* and *functionally decomposed surfaces* as described in the following paragraphs.

In the first approach a *large* roughly approximating four-sided surface is chosen. Its movable boundaries and corner points are chosen in such a way that all the data points of interest lie within the boundary of the surface. (We assume here that the point data represent just some part of the boundary of the complete object, and not an entire closed surface.) After choosing corresponding points on the approximating surface, the distances between the surface points and the measured data points can be minimized by well-known least-squares methods²⁸. By means of these procedures the free parameters of the surface are computed. Having found a best fit surface in some sense, the actual distances of the measured points from the surface can be evaluated. If these are too large, we try to iteratively improve the parametrization and refit. If the result is still too bad, a new approximating surface needs to be created, which has more free parameters to provide a better fit. This surface can be constructed by subdividing the previously obtained surface or by making local refinements, which may lead to some quadtree-like surface representation. The outside parts of the approximating surface will behave in an uncontrolled way, since outside there are no positional constraints. However, this does not matter, because internally a good fit is reached and the trimming curves constructed around the measured data points will help to separate the useful surface area from the rest. This sort of surface representation is illustrated in *Figure 5*.

The basic advantage of this technique is its relative simplicity, but it is not necessarily straightforward to construct a sufficiently good initial surface which can then be refined as described. By repeated least-squares

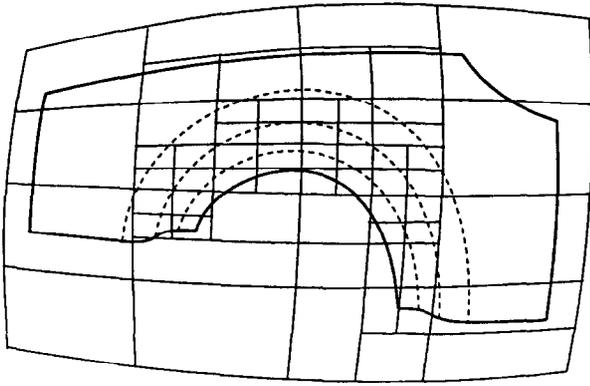


Figure 5 Global approximating surface

fitting and subdivision a surface model will be obtained at the end and practically no user interaction will be required. The basic disadvantage from an engineering point of view, however, is that no attempt is made to interpret the underlying structure of the shape. For example ribs or free-form pockets, which could be represented otherwise in a *natural manner* as a single surface, now may belong to several, separate small surface elements, depending on how the initial surface was oriented and the subdivision was performed.

The second approach is based on a curve network, which divides up the surface by means of a series of 'characteristic curves', which might be sharp edges of the boundary, lines of extremal curvature, lines of symmetry, etc. Constructing the curve network is tricky, and also potentially unreliable. Firstly, only local parts of the data are used to determine the curve network and secondly, estimating differential properties is highly susceptible to noise. *Automatic patching* of such networks is thus not usually done, but instead interactive solutions are used where the user helps to specify the curve network. Furthermore, choosing which 'characteristic' curves are of engineering significance and hence are useful, is not obvious. The user has to pick characteristic vertices and define connecting edge curves, and in this way the internal face structure is explicitly defined (see *Figure 6*). Typically further patching is applied and finally the whole surface is covered by exclusively four-sided patches. This representation does not require internal trimming since the boundaries of the composite free-form surface are explicitly incorporated into the curve network.

Fitting faces across the boundaries of the regions obtained is relatively easy. There are several publications on how to subdivide n -sided regions into four-sided ones, how to restructure the curve net if T-nodes occur, and how to smoothly join adjacent parametric patches along

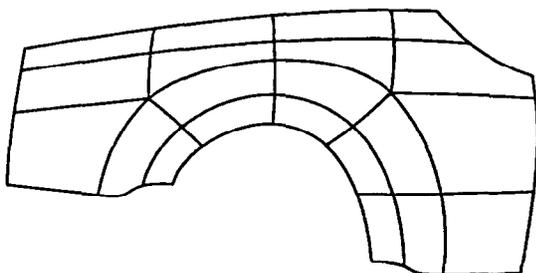


Figure 6 Curve network based surface

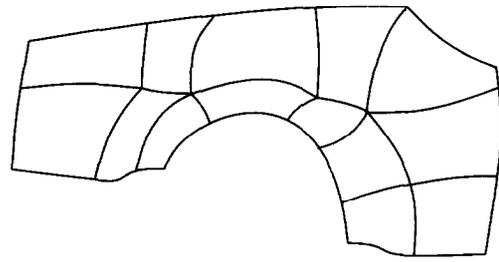


Figure 7 Arbitrary topology surface

their boundaries: see e.g. References 12 and 28. Typically compatible cross-derivative functions are assigned to each curve segment and in the final phase patches are generated locally which satisfy the previously determined positional and tangential constraints along the boundaries. The advantage of this approach is obviously that the patch structure reflects the user's concept of the structure of the surface, and this technique will guarantee overall G^1 or G^2 continuity for the whole surface. At the same time the enforcement of a curve network with four-sided patches creates artificial boundaries as well, i.e. parts which naturally would belong together may get separated. This is a particularly crucial issue, for example, when a global surface fairing is required, but the artificial internal segmentation separates functionally connected pieces. Note that a small change to the underlying geometry may lead to relatively large changes in the structure of the curve network. Furthermore, at times it may be difficult to incorporate the trimming curves into the network in a natural way.

The third approach—arbitrary topology surfaces—represents a combination of the previous two approaches (*Figure 7*). Here a global approximating surface with arbitrary topology is created in an automatic manner. The patch structure is generated by simplifying the underlying triangulation, and this provides a particular parametrization for the final surface. Based on this, first a general topology free-form curve network is created, which will be interpolated by piecewise polynomial surfaces. Special constructions are applied to satisfy overall cross-boundary continuity. Typical representatives for this approach are the publications by Eck and Hoppe⁹ and Guo²¹. The advantages and disadvantages of arbitrary topology surfaces also combine those of the previous two approaches. In this case we get a collection of smoothly connected, untrimmed patches automatically, which globally reflects the topological structure of the shape. At the same time, this collection cannot reflect the functional structure of the shape, since the basis of these methods is the triangulated data structure, which does not hold information on how one should subdivide the shape in a sensible manner.

Of course, the fourth method of functional decomposition also holds its difficulties. The main idea here is to try to somehow discover the design intent behind a given shape. If there is information on how the shape was possibly created, i.e. what the original functional surface elements were and what modelling operations were performed to get the final shape, it may be possible to find the original surfaces and redo the operations using the individually re-engineered surfaces. An example is shown in *Figure 8*, where the overall shape is the result of

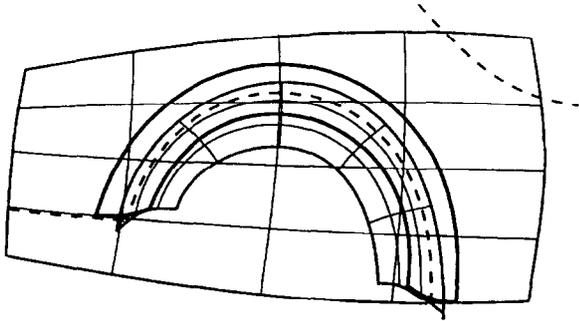


Figure 8 Functionally decomposed surface

an intersection operation between a translational-like surface and another surface with rotational symmetry, which have been smoothed by a third blending surface. This sort of representation seems ideal from an engineering point of view, but several problems may arise. First of all, with our present knowledge it is very difficult to discover general functional elements without user interaction, unless there is an underlying information base which somehow a priori describes the CAD history of the given model. Moreover the surface fitting problem is also difficult, since we have to fit a complete surface, but only a well-defined part of it may be represented by measured data. In our example, there are missing portions which disappeared during the intersection and blending operations. The basic advantage of this method is that it provides an appropriate engineering decomposition of the shape. The basic disadvantage is that it requires a potentially large amount of interaction or a fairly detailed *a priori* knowledge about the part. In comparison with the previous methods, here the surface elements join each other along internal trimming curves, so it is difficult to assure exact mathematical continuity. The surface elements will join with numerical continuity, and special care is required to guarantee that the connections are within prescribed system tolerances relating to matching positions and tangent planes.

We note that in principle a really good system might combine the above four approaches in different regions of the surface.

The majority of related reverse engineering publications deal with fitting four-sided surfaces to measured data. Further practical problems include the following. Given a surface, how do we find good parameter values for the measured data points to put them into correspondence with points on the surface: see related works by Hoschek and Lasser²⁸ and Ma and Kruth³⁴. Another issue is whether it is sufficient to minimize only a function of distances, or whether more advanced surface fairing techniques are needed to minimize some sort of *surface energy*, taking into consideration higher order derivatives as well. Such methods generally lead to a non-linear system of equations which is computationally very expensive. The literature on surface fairing is very extensive, and methods differ in whether an ideal shape with certain constraints needs to be created or an existing or approximating surface representation needs to be improved (see some of the related work in References 7, 13, 19, 28 and 39). A third issue is what to do when there are missing regions in the measured data set or the data are very unevenly distributed (see, for example, Dietz⁸ and Hermann *et al.*²²).

The techniques described above can help in creating local topology for a composite free-form surface, but further steps are required to construct a complete and consistent B-rep model. Before we look at that, the problem of merging multiple views is discussed in the next section.

MULTIPLE VIEW COMBINATION

While some applications may only be interested in reverse engineering part of an object, many will require a geometric model of the whole surface of the object. This leads immediately to the following problem: ideally we would have the object 'floating' in 3D space (in a fixed position with a fixed orientation), so that the scanner could move around the object from all sides to capture data in a single coordinate system. In practice, the object will have to rest on some surface, so part of it will be inaccessible to the scanner. Furthermore, if the scanner is fixed in position, at any one time, it will be able to capture data from an even more limited region of the object's surface. Thus, generally, it will be necessary to combine multiple views taken with the object placed in different orientations in front of the scanner. We outline some of the important issues involved in view combination in this section; many are discussed in further detail in Marshall and Martin³⁶.

There may also be other good reasons for combining views—for example, some parts of the object may have more details than others, and so a low-resolution scan may be taken of the whole object, while higher-resolution scans may also be taken of the detailed areas. Other considerations may also be important in deciding how many scans are taken. For example, the more scans that are taken, the longer the whole scanning process will take, and the scanner may be an expensive resource. On the other hand, the more nearly face-on the scanner views a given surface, the more accurate the data collected from that surface are likely to be. Clearly, some compromise must be made between these conflicting requirements.

Naturally, sufficient views must be taken so that every part of the object's surface is visible in at least one view. This may not be trivial to arrange if the object has many concavities. Indeed, it may be necessary for some sort of geometric model to be made from the initial scan of the object to determine those places where further scanning is required. This again demonstrates the interplay between the different phases of an overall reverse engineering system.

The major problem in combining multiple views is *accurate registration*, i.e. finding the transformation (rotation and translation) which relates information in one view with information in another view, so all sets of information can be combined. Various approaches exist for capturing multiple views in such a way that registration is possible.

The simplest approach is to place the object on a turntable; this must have a high accuracy to infer registration reliably. Even so, the bottom of the object will not be visible, and if a model of the complete object is required, the object will have to be turned over. It may be possible to use a robot to do this, but compliance in the robot gripper is likely to lead to positioning errors which are too large for the registration to be computed directly from the robot's commanded movements. Instead,

it will be necessary in such a case to determine the registration directly from the multiple views themselves, by correlating the information in the views. Once this is deemed necessary, the object may as well be turned over and placed in roughly the right position for the next view by hand.

When determining the registration directly from multiple views, it is necessary to ensure that there is sufficient overlap of information between successive views. Various sophisticated approaches have been suggested for comparing object models, most of them for use in the field of object recognition where accuracy is not an issue, although they have also been used for inspection³⁶. Such methods can match either higher level information²⁶, (partial geometric models constructed for each view) provided enough 'simple' surfaces can be seen in each view to uniquely determine the registration, or the point data can be matched directly^{4,15,17}.

A rather more pragmatic approach to provide overlapping data for registration of multiple views, which has been adopted in commercial environments¹, is to clamp some reference objects (e.g. three spheres) to the object being scanned. This ensures that enough simple surfaces *are* present in each view, and a direct algorithm can then be applied to compute the registration once these surfaces have been found in each view. It is much easier and more reliable to use specific surfaces known to be present than to use a general algorithm. Unfortunately, clamping the reference objects to the object to be scanned will obscure part of the latter, so this trick does not directly solve the problem of producing a complete object model.

A relatively untried technique is to use multiple sensors to view the object from several sides at once⁵². This has the obvious disadvantage of the requirement for extra equipment. Also, calibration of several sensors at the same time will lead to a large set of simultaneous (probably non-linear) equations, whose solution is likely to be time consuming, and numerically unstable. However once the system is calibrated, many objects can then be scanned—in effect, it is the sensors which are registered rather than parts of the objects being scanned. Another disadvantage of this approach, however, is that it is never possible to guarantee that there are sufficient sensors to see the whole of the object at once—consider a cog wheel with many deep teeth.

Having captured several views of the object, with the appropriate registration data, another choice still remains to be made—whether to merge the data at the point set level, before segmenting and further processing it, or whether to process each view to produce a corresponding partial geometric model, and then to merge the latter. There are two relevant issues here. Merging extracted features rather than the point sets may seem computationally attractive, as smaller data sets need to be processed at any one stage. However, features extracted from different views may be inconsistent. At best, estimates of parameters for the same feature will differ and must be combined. Worse still, features may be unexpectedly absent from a view, or a single feature in one view may be segmented as multiple features in another view. A single point set leads to a large, but consistent data structure, while the merging of partial models is likely to need complex methods to resolve these inconsistencies. Furthermore, the approach of processing a single point set has the advantage that any face which appears in multiple views can be estimated from the

complete subset of original points belonging to it. If higher level, partial face descriptions are constructed before merging, and the lower level data are discarded, this is likely to lead to less reliable parameter estimation for the whole face, as demonstrated by Fisher's recent work¹⁶.

B-REP MODEL CREATION

In previous sections we described various alternatives for extracting geometric information from a dense set of measured data points. Due to the diversity of surface representations and applicable algorithms, the input for the model creation can also be in various formats at different levels of geometrical and topological completeness. For this reason to present a detailed uniform approach for the final model creation would be very difficult, but various relevant ideas will be discussed here.

The purpose of this final phase is to create a consistent and contiguous model of vertices, edges and faces, where both the adjacency relationships between the constituent elements and the mathematical equations of underlying edge curves and surfaces are explicitly stored^{24,35}. In certain approaches some boundary curve structure is created simultaneously with the segmentation (particularly for edge-based methods and free-form curve network driven algorithms), but generally we have to start model building just from a set of *untidy* elements. These are created by local region growing or fitting algorithms and usually no effort has been made to provide consistency between them. For example, region growing algorithms provide subsets of points to which well-defined surfaces are fitted, but there may be gaps (i.e. unassigned data points) between them or they may partly overlap each other. The surfaces may not be consistent, e.g. two almost adjacent parallel planes may be segmented with a step between them; or we may not have the desired/expected degree of continuity. Similarly edge detection algorithms may produce a sequence of curve segments which may cross each other and which do not necessarily form a network of smooth connected edges. On boundary tracking see the paper of Fitzgibbon *et al.*¹⁸ (this issue); on the problems of building polyhedral boundary models, see also the work of Hoover *et al.*²⁶

If there are no explicit edges defined in the previous steps we have to compute these by means of extending the regions. Assuming the underlying surfaces can be extended beyond the boundaries of their segmented regions, surface–surface intersections will provide proper edge curves, which need to be limited by two end vertices. Intersection is not always possible or not even the right thing to do. In these cases we may need to insert blends or adjust the parameters of the surfaces to make them meet smoothly: for example, consider a plane meeting a semicylinder along a smooth edge.

To locate a vertex where three surfaces meet seems relatively simple if they meet in a sharp intersection, since then a unique intersection point can be determined. The situation is more difficult when we have to compute vertices where smooth edges meet or more than three edges meet, due to numerical instability problems (see related work by Higashi *et al.*²³). An example of the latter case is shown in *Figure 9*. *Figure 9a* shows a pyramid object to be reverse engineered, while *Figure 9b* depicts possible regions identified looking at it from a top view. It can be seen that data along the edges will be fairly

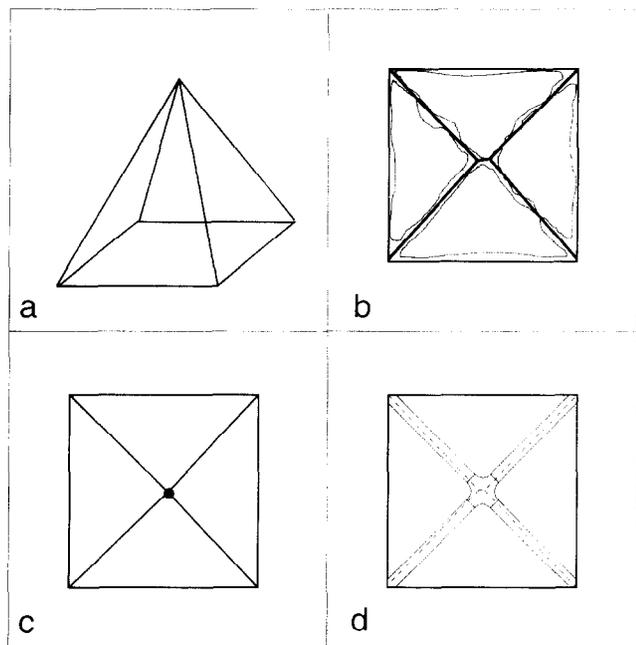


Figure 9 Fixing a solid pyramid model

disordered and unreliable. If we extend the four planes of the side faces, due to inaccuracies the edges will not meet in a single vertex, but a little artificial edge segment will be created. In other cases not only small edge artifacts but small facets will be generated. Obviously these are undesirable and when possible should be removed. One may choose a single vertex as in our example and adjust the fitted planes to a small extent to go through it (see *Figure 9c*). This can be done in many cases, but in other situations other constraints will prevent this.

The generation of blending surfaces should also be based on primary surfaces. The blends may be explicitly present on the surface of the measured object, or we may create these artificially to bridge the gaps due to the imperfection of data close to edges. Either the intersection curve of two surfaces will provide a spine curve for some range-based blend, or the intersection of the related offset surfaces will provide a spine curve for the centre point of a rolling ball which sweeps out a blending surface. More general blends with variable parameters can also be derived based on the primary surfaces together with the data points in the *blending region* which can give us some clue as to the extent of a varying blend radius, etc. Details of blending methods are given in References 60 and 28. Narrow blending strips can be used to solve certain inaccuracy problems, such as those shown in the previous pyramid model (see *Figure 9d*), whereupon there is no longer a need to adjust the side faces.

Beside edge blends, smooth vertex blends need to be created at the junctions of edge blends. One technique is to create n -sided patches when n blend surfaces run together. The corners are computed as the intersection points of the neighbouring trimlines lying on the same face. Another possibility is to apply *setback type* vertex blends (see for example, Reference 58). These are $2n$ -sided patches which are bounded by small curve pieces running across the primary faces to be blended; these pieces smoothly connect the adjacent trimlines. Both vertex blend types can be seen in *Figure 4*.

We have mentioned earlier the problem of generating smooth segmenting curves within large free-form

regions. This may take place with some interactive help or automatically, but to generate projected curves or geodesic curves or curvature lines, etc., is not trivial without having an explicit representation rather than just the underlying point clouds. In order to create *fair* curves the known curve fairing techniques^{12,28} need to be extended.

Assuming we have managed to create a consistent B-rep model, there are further tasks to make the representation acceptable from an engineering point of view as well. Particularly for man-made objects, there are many important geometric properties, such as symmetry, parallelism, orthogonality, concentricity, etc., which represent essential information^{20,25}. We may want to enforce such constraints on the model, when it is likely that they hold, but this should not be done without careful consideration, as the following example shows. If due to the inaccuracies of measurement and surface fitting the angle between two planar faces is almost 90° , we might decide to set the faces to be orthogonal. But in mould making, small draft angles are a necessary feature for the *quasi-vertical* faces in order to be able to easily remove the part from the mould. Thus, the issue of where and how to 'improve' the model is very difficult, and requires higher level information to determine. User interaction or artificial intelligence techniques may help.

CONCLUSION

Reverse engineering of geometric models for CAD/CAM is a rapidly evolving discipline in which interest is currently high. This is due in no small part to the relatively recent commercial availability of active stereo based laser scanners which are of sufficient accuracy for many applications. Nevertheless, we are still at a similar stage to the early days of computer graphics displays—although various people sell the raw hardware, the range of software to make full use of the hardware is still lacking in many cases, and underdeveloped. Current commercial software systems often only allow simple point cloud processing and single surface fitting with interactive help; the production of complete B-rep models is only possible for very simple objects or polyhedral approximations (see also the paper by Skifstad⁵³). On the other hand, users wish to automatically process a wide range of objects, possibly from a variety of data capture devices with differing characteristics, to produce models in a variety of representations and accuracies.

In summary, while systems exist which can perform the simple operation of 3D copying, the goal of extracting higher level information which can be edited and analysed is still some way off. Key research areas which still need further work before general-purpose reverse engineering becomes widely available include: improving data capture and calibration, coping with noise, merging views, coping with gaps in the data, reliable segmentation, fair surface fitting, recognizing natural or human-intended structure of the geometry of the object, and finally ensuring that consistent models are built.

ACKNOWLEDGEMENTS

This overview is based on one originally presented at a minisymposium on Reverse Engineering at the Fourth

SIAM Conference on Geometric Design, 6–9 November 1995, Nashville, USA. Special thanks are due to the conference organizers, who invited the authors to arrange this minisymposium and present their views.

Ralph Martin and Tamás Várady would like to acknowledge the support of the European Union for their work into reverse engineering under Copernicus grant RECCAD 94-1068. The authors would like to thank to the members of the Copernicus team for many helpful discussions. The National Science Foundation of the Hungarian Academy of Sciences also supported this work under grant 16420.

REFERENCES

- 3D Scanners, London, UK, Private communication, 1995.
- Besl, P J and Jain, R C 'Three dimensional object recognition' *ACM Computing Surveys* **17**(1), 1985, 75–145.
- Besl, P J and Jain, R C 'Segmentation through variable-order surface fitting' *IEEE PAMI* **10**(2), 1988, 167–192.
- Besl, P J and McKay, N D 'A method for registration of 3D shapes' *IEEE PAMI* **14**(2), 1992, 239–256.
- Bolle, R M and Vemuri, B C 'On three-dimensional surface reconstruction methods, *IEEE PAMI*, **13**(1), 1991, 1–13.
- Butler, C 'Investigation into the performance of probes on coordinate measuring machines' *Industrial Metrology* **2**(1), 1991, 59–70.
- Celniker, G and Gossard, D 'Deformable curve and surface finite elements for free-form shape design' *Computer Graphics (Proc of SIGGRAPH'91)* **25**(4), 1991, 257–266.
- Dietz, U 'Erzeugung glatter Flächen aus Meßpunkten' Technical Report 1717, Dept of Mathematics, Technische Hochschule Darmstadt, 1995.
- Eck, M and Hoppe, H 'Automatic reconstruction of B-spline surfaces of arbitrary topological type' *Computer Graphics (SIGGRAPH '96 Proceedings)* **30**, 1996, 325–334.
- Eck, M, DeRose, T, Duchamp, T, Hoppe, H, Lounsbery, M and Stuetzle, W 'Multiresolution analysis of arbitrary meshes' *Computer Graphics (SIGGRAPH'95 Proceedings)* **29**, 1995, 173–182.
- Edelsbrunner, H and Mücke, E 'Three-dimensional alpha shapes' *ACM TOG*, **13**(1), 1994, 43–72.
- Farin, G *Curves and Surfaces for Computer Aided Geometric Design*, 2nd edn. Academic Press, London, 1990.
- Farin, G and Sapidis, N 'Curvature and fairness of curves and surfaces' *IEEE CGA* **9**, 1989, 52–57.
- Faugeras, O D, Hebert, M and Pauchon, E 'Segmentation of range data into planar and quadric patches' *Proc. 3rd Computing Vision and Pattern Recognition Conference* Arlington, VA, 1983, pp 8–13.
- Fisher, R B 'Best-first and ten other variations of the interpretation-tree model matching algorithm, Dept of AI, University of Edinburgh, Research Paper 717, 1994.
- Fisher, R B 'Personal communication' 1996.
- Fitzgibbon, A W, Bispo, E, Fisher, R B and Trucco, E 'Automatic acquisition of CAD models from multiple range views' *10th National Conf. on Manufacturing Research* Loughborough, 1994.
- Fitzgibbon, A W, Eggert D and Fisher, R B 'High-level CAD model acquisition from range images' *Computer-Aided Design* **29**, 1997, 321–330.
- Greiner, G 'Variational design and fairing of spline surfaces' *Computer Graphics Forum* **13**(4), 1994, 143–154.
- Grimstead, I J and Martin, R R 'Creating solid models from single 2D sketches' *Proc. 3rd Symposium on Solid Modeling and Applications* ed. C Hoffmann and J Rossignac, ACM Press, 1995, pp 323–327.
- Guo, B 'Surface reconstruction: from points to splines' *Computer-Aided Design* **29**, 1997, 269–277.
- Hermann, T, Kovács Z and Várady, T 'Special applications in surface fitting' *Geometric Modelling Studies*, GMS 1996/3, Computer and Automation Research Institute, Budapest, 1996.
- Higashi, M, Torihara, F, Takeuchi, N, Sata, T, Saitoh, T and Hosaka, M 'Face-based data structure and its application to robust geometric modeling' *Proc. 3rd Symposium on Solid Modeling and Applications* ed. C. Hoffmann and J. Rossignac, ACM Press, 1995, pp 235–246.
- Hoffmann, C M *Geometric and Solid Modelling, An Introduction*, Morgan Kaufman, San Mateo, CA, 1989.
- Hoffmann, C M and Juan, R 'Erep, an editable, high-level representation for geometric design and analysis' *Geometric and Product Modelling* ed. P Wilson, M Wozny and M J Pratt, North Holland, 1993, pp 129–164.
- Hoover, A, Goldgof, D and Bowyer, K W 'Extracting a valid boundary representation from a segmented range image' *IEEE PAMI*, **17**(9), 1995, 920–924.
- Hoppe, H, DeRose, T, Duchamp, T, McDonald, J and Stuetzle, W 'Mesh optimization' *Computer Graphics (SIGGRAPH'93 Proceedings)* **27**, 1993, 19–26.
- Hoschek, J and Lasser, D *Fundamentals of Computer Aided Geometric Design* A.K. Peters, 1993.
- Jarvis, R A 'A perspective on range finding techniques for computer vision' *IEEE PAMI*, **5**(2), 1983, 122–139.
- Jarvis, R A 'A laser time-of-flight range scanner for robotic vision' *IEEE PAMI*, **5**(5), 1983, 505–512.
- Koivunen, V 'Robust signal restoration and local estimation of image structure' Technical Report, MS-CIS-92-92, University of Pennsylvania, 1992.
- Koivunen, V and Bajcsy, R 'Rapid prototyping using three-dimensional computer vision' Technical Report, MS-CIS-92-70, University of Pennsylvania, 1992.
- Leonardis, A, Gupta, A and Bajcsy, R 'Segmentation of range images as the search for geometric parametric models' *International Journal of Computer Vision* **14**, 1995, 253–277.
- Ma, W and Kruth, J P 'Parametrization of randomly measured points for least squares fitting of B-spline curves and surfaces' *Computer-Aided Design* **27**(9), 1995, 663–675.
- Mäntylä, M *An Introduction to Solid Modelling*. Computer Science Press, Rockville, Maryland, 1988.
- Marshall, A D and Martin, R R *Computer Vision, Models, and Inspection*, World Scientific Publishing, Singapore, 1992.
- Milroy, M J, Bradley C and Vickers, G W 'Segmentation of a wrap-around model using an active contour' *Computer-Aided Design* **29**, 1997, 299–320.
- Modjarrad, A 'Non-contact measurement using a laser scanning probe' *SPIE, In-Process Optical Measurements*, **1012**, 1988, 229–239.
- Moreton, H P and Sequin, C 'Functional optimization for fair surface design' *ACM Computer Graphics (Proc of SIGGRAPH'92)* **26**, 1992, 167–176.
- Moring, I, Heikkinen, T and Myllylä, R 'Acquisition of three-dimensional image data by a scanning laser range finder' *Journal of Optical Engineering*, **28**(8), 1989, 897–902.
- Moss, J P, Linney, A D, Grindrod, S R and Mosse, C A 'A laser scanning system for the measurement of facial surface morphology' *Journal of Optics and Lasers in Engineering*, **10**, 1989, 179–190.
- Motavalli, S and Bidanda, B 'A part image reconstruction system for reverse engineering of design modifications' *Journal of Manufacturing Systems*, **10**(5), 1991, 383–395.
- Nishihara, H K 'Practical real-time imaging stereo matcher'. *Journal of Optical Engineering*, **23**(5), 1984, 536–545.
- Pietikäinen, M, Rosenfeld, A and Walter, I 'Split-and-link algorithms for image segmentation' *Pattern Recognition*, **15**, 1982, 287–298.
- Posdamer, J L and Altschuler, M D 'Surface measurement by space-encoded projected beam systems' *Journal of Computer Graphics and Image Processing*, **18**, 1982, 1–17.
- Rioux, M 'Laser range finder based on synchronized scanners' *Journal of Applied Optics* **23**(21), 1984, 3837–3844.
- Rockwood, A and Winget, J 'Three-dimensional object reconstruction from two-dimensional images' *Computer-Aided Design* **29**, 1997, 279–285.
- Rossignac, J and Borrel, P 'Multi-resolution 3D approximations for rendering complex scenes' *Modeling in Computer Graphics* Springer-Verlag, 1993, pp 455–466.
- Sahoo, K C and Menq, C H 'Localization of 3-D objects having complex sculptured surfaces using tactile sensing and surface description' *Journal of Engineering for Industry* **113**, 1991, 85–92.
- Sapidis, N S and Besl, P J 'Direct construction of polynomial surfaces from dense range images through region growing' *ACM Trans. on Graphics* **14**(2), 1995, 171–200.
- Schroeder, W J, Zarge, J and Lorensen, W E 'Decimation of triangle meshes' *Computer Graphics (SIGGRAPH'92 Proceedings)*, **26**, 1992, 65–70.
- Shirai, Y *Three-Dimensional Computer Vision*. Springer Verlag, Berlin, 1987.
- Skifstad, K 'Rising above the data clouds' *Computer Graphics World* **18**(2), 1995, S4–S12.

- 54 Smith, D R and Kanade, T 'Autonomous scene description with range imagery' *CVGIP* **31**(3), 1985, 322-334.
- 55 Solina, F and Bajesy, R 'Recovery of parametric models from range images: The case for superquadrics with global deformations' *IEEE PAMI* **12**(2), 1990, 131-147.
- 56 Stroud, I A 'Boundary modelling with special representations' *Computer-Aided Design* **26**(7), 1994, 543-550.
- 57 Tsujimura, T, Yabuta, T and Morimitsu, T 'Shape-reconstruction system for three-dimensional objects using an ultrasonic distance sensor mounted on a manipulator' *Transactions of ASME* **111**, 1989, 180-186.
- 58 Várady, T and Rockwood, A P 'Vertex blending based on the setback split' *Mathematical Methods for Curves and Surfaces* ed. M Daehlen, T Lyche and L L Schumaker. Vanderbilt University Press, 1995, pp 527-542.
- 59 Veron, P and Leon, J C 'Static polyhedron simplification using error measurements' *Computer-Aided Design* **29**, 1997, 287-298.
- 60 Vida, J, Martin, R R and Várady, T 'A survey of blending methods that use parametric surfaces' *Computer-Aided Design* **26**(5), 1994, 341-365.
- 61 Wang, Y F and Aggarwal, J K '3D object description from stripe coding and multiple views' *Proceedings of the 5th Scandinavian Conference on Image Analysis*, 1987, **60**(6), 1987, 669-680.
- 62 Watanabe, S and Yoneyama, M 'An ultrasonic visual sensor for three-dimensional object recognition using neural networks' *IEEE Transactions on Robotics and Automation* **8**(2), 1992, 240-249.
- 63 Will, P W and Pennington, K S 'Grid coding: a novel technique for image processing' *Proceedings of the IEEE*, **60**(6) 1972, 669-680.
- 64 Wohler, T '3D digitizing for engineering' *Computer Graphics World* **18**(2), 1995, S1-S3.
- 65 Woodham, R J 'Photometric stereo: a reflectance map technique for determining surface orientation from a single view' *Image understanding systems and industrial applications, Proc. SPIE 22nd Annual Technical Symposium*, San Diego, California, Vol. **155**, 1978, pp 136-143.
- 66 Xiong, Y L 'Computer aided measurement of profile error of complex surfaces and curves: theory and algorithm' *International Journal of Machine Tools and Manufacturing* **30**(3), 1990, 339-357.



Dr Ralph Martin has been working in the field of CAD/CAM since 1979. He obtained his PhD in 1983 from Cambridge University for a dissertation on 'Principal patches', and since then has been a Lecturer and Senior Lecturer at the University of Wales College of Cardiff. He has coauthored more than 65 papers and coauthored or edited 4 books covering such topics as solid modelling, surface modelling, intelligent sketch input, vision based geometric inspection and geometric reasoning. He is a Fellow of the Institute of Mathematics and its Applications, and a Member of the British Computer Society. He is currently working on an EU Copernicus grant on reverse engineering.



Dr Jordan J Cox received his Bachelor and Master's degrees from Brigham Young University in 1983 and 1984, respectively. After working in industry he returned to Brigham Young University to teach in the Design Engineering Technology Department until September of 1988 when he entered Purdue University to pursue his doctoral work. He received his PhD from Purdue University in December of 1991. He joined the faculty of the Mechanical Engineering Department of Brigham Young University in January of 1992 and has taught in that department since then. Dr Cox has worked in the aerospace industry at Garrett Turbine Engine Co and the Navy Labs. He has consulted with various companies ranging from the orthotics industry, software companies and is currently a special consultant to the Vice President of Aeromechanical Systems for the Aerospace Sector of AlliedSignal developing and directing the implementation of 3rd generation engineering graphics, product data management and knowledge based engineering. His academic emphasis is in engineering analysis and geometric modelling.



Dr Tamás Várady received a Diploma in electrical engineering from the Technical University of Budapest, in 1976. He obtained a doctorate degree in 1977 and an advanced degree of technical sciences (PhD) from the Hungarian Academy of Sciences in 1986 with a dissertation titled the 'Integration of free-form surfaces into a volumetric modeller'. He led the CAD Department at the Computer and Automation Research Institute between 1984 and 1990, where he directed several research projects on geometric modelling and software developments for various industrial applications. In 1990, he was the cofounder of CADMUS Consulting and Development, of which he is president. Since January 1991, he has been heading the Geometric Modelling Laboratory at the Institute. His recent research interests include general-topology surface modelling, vertex blending in B-rep solid modellers and reverse engineering.