

Chapter 11

Object and Workspace Modeling Based on a 3-D Vision System

HELAI YAO, RON PODHORODESKI, and ZUOMIN DONG
Department of Mechanical Engineering, University of Victoria

11.1 Introduction



A description of the geometric characteristics of a working environment is a prerequisite for planning the motion of robots, CNC machine tools, coordinate measuring machines, and other manufacturing facilities. Intelligent robotic and manufacturing systems further require the geometric modeling of objects and workspace to be conducted in conjunction with the machine vision system and to be performed on-line to support on-line motion planning and decision making.

Geometric models for the working environment, including objects and workspace, may come from a CAD model built by a designer or from the geometric data acquired by a machine vision system. Traditionally, geometric modeling is mainly concerned with geometric description. Many geometric modeling techniques have been developed and used in computer-aided design and manufacturing (CAD/CAM), computer graphics, graphical animation, and other related applications over the last three decades. A geometric modeling system is supported by a geometric representation, algorithms for geometric manipulation, and a database for geometric entities. Such a system provides applications with object models that convey various geometric information.

11.1.1 Information Embedded in a Geometric Model

A geometric model describes objects by their shape, dimension, area, volume, position, orientation, and topology. In general, various geometric models for objects can be classified into three major categories: wireframe model, surface model, and solid model.

A wireframe model has the simplest model form, composed of object edges that are surface intersections and/or surface outlines. Wireframe may include straight lines, analytical curves,

and free parametric curves. Object descriptions provided by a wireframe model are primarily limited to edge-related features. To overcome the ambiguity introduced by such a model human intelligence is often needed.

In contrast, a surface model is able to intrinsically provide more detailed characteristic data, such as surface shape, curvature and surface points. A surface model can be built using either regular analytic surfaces of a plane, a cylinder or a sphere, or sculptured surfaces defined by spline functions or surface meshes. Both wireframe and surface models supply few of the topological relationships among their component elements: points; boundaries and surfaces.

A solid model, as a more sophisticated geometric model, provides a complete description of an object with sufficient geometric and topological information. The dimensions, areas, volumes, surface shape descriptions, surface neighborhood relationships, and relative object positions and orientations can be obtained from the database of a solid model. The inner and outer spaces of an object can also be detected.

Different types of geometric models have different capabilities of representing geometric information. A geometric model with a simple form is usually at a low level in terms of the geometric information it can supply, while a sophisticated geometric representation at high level can provide much more.

11.1.2 Traditional Approach to Geometric Model Generation

Construction of a geometric model requires initial data input. Sophisticated modeling approaches such as solid modeling require more detailed information about the object to be modeled. These initial data include the geometry and topology information of the object.

Traditionally, geometric models for various objects are created using a CAD/CAM system. The modeling approach has four primary components: (1) symbol structures which represent solid objects; (2) processes which use such representations for answering geometric questions about the objects; (3) input facilities for creating and editing object representations and evoking processes; and (4) output facilities and representations of results [19].

The creation of a geometric model is usually done interactively by users of the geometric modeling system through graphics input and output devices such as a mouse, a 2-D digitizer, a trackball, a joystick, a lightpen, a keyboard, a graphics terminal, or a plotter. Since the users of the graphics systems well understand the geometry and topology of objects that they want to model, they can effectively use the input devices to provide the object's geometry and topology to the geometric modeling system.

This modeling procedure can be illustrated using the example in Figure 11.1. When a solid model for the flange is built, the shape and location of various surfaces representing the part, surface adjacency relationships indicating object topology, and surface orientation identifying the interior and exterior of the object are specified by a designer. This traditional approach to geometric model generation is covered by many CAD and computer graphics textbooks (e.g., [8, 14, 25]).

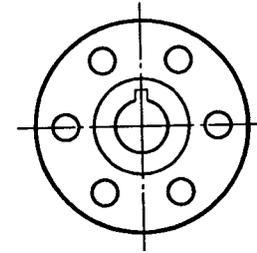


Figure 11.1: Modeling with Known Object Geometry and Topology

11.1.3 The Need for More Flexible Modeling Approaches

Although the traditional geometric modeling approaches have been used broadly in design and manufacturing, they cannot satisfy some key requirements of geometric model generation imposed by intelligent robotic and manufacturing systems. These requirements include: (1) modeling objects and workspace with known geometries, but in unstructured environments and (2) modeling existing objects whose geometries are not known.

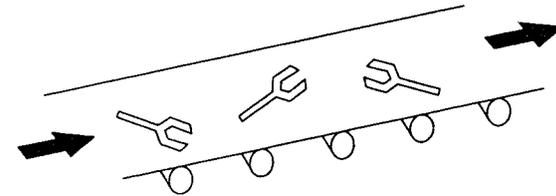


Figure 11.2: Objects with Known Geometry in an Unstructured Environment

Figure 11.2 shows an example for case one. A cast part has been designed and a geometric model for the ideal geometry of the part has been created. In manufacturing, the shape and dimensions of produced parts need to be measured automatically and be compared with design dimensions given in the ideal model. However, no direct match between the real cast part and its ideal model exists due to the different orientation and/or scale between these two. On-line creation of a geometric model based upon the real geometry of the produced part is needed for inspecting the dimensions of the manufactured part. Without human interaction the traditional modeling approaches are not able to construct this type of real geometric model. Other similar situations may include robots picking up parts from a bin and automatic vehicles moving in unstructured surroundings.

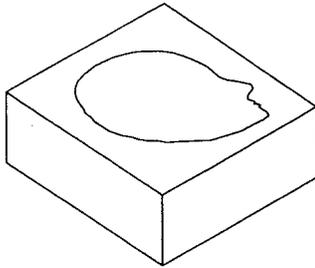


Figure 11.3: An Object with Unknown Geometry

An example of the second case is illustrated in Figure 11.3 where an injection mold of a face sculpture is desired. Since the shape is irregular and the exact geometry and topology of the object are unknown, the initial object description cannot be explicitly input into a CAD/CAM system with traditional approaches. Without a geometric representation, planning for machining and generation of CNC codes cannot be conducted. The solution to this problem is to model the object based on object surface measurements. Although coordinate measuring machines (CMM) are used for measuring object profiles in industry, applications of CMM are limited by its mechanical contact working principle and the degrees of freedom of the machine and its touch probe.

Apparently a noncontact surface measurement-based, on-line geometric modeling approach is needed. To measure object surfaces for such a geometric modeling system, vision sensors have been applied and relevant modeling algorithms have been developed. To understand how a vision sensor works with geometric modeling, a brief review of the basic functions of a machine vision system will be helpful. The remainder of this section serves this need. For further details on the subject a reader can refer to [2, 4, 6, 9, 11, 18, 22].

11.1.4 A Brief Review of Machine Vision

Machine vision is an industrial practice of computer vision that is concerned with automating and integrating a wide range of processes and representations used for vision perception. Its goal is to allow a machine to understand its environment and perform required tasks using perceived information. The major tasks for machine vision are image processing and scene analysis.

Image processing takes an image as its input in order to generate another image with noise suppressed and useful image regions enhanced. In Figure 11.4, several application examples of image processing are shown. The original image (top left) looks very gloomy because of poor illumination and has noise. Applying histogram equalization to this image improves image visibility (top right). Reducing noise is one of major purposes for image processing. In this example, high frequency noise is suppressed by a low-frequency-pass filter (bottom left). There are many types of more complex image noises that affect information extraction

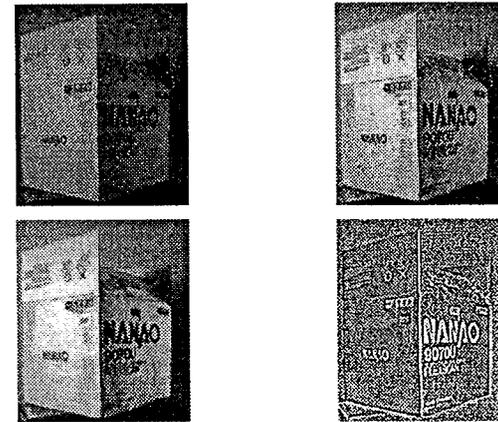


Figure 11.4: Top Left: Original Image; Top Right: Histogram Equalized Image; Bottom Left: Image after Using Median Filter; and Bottom Right: Image after Using Laplacian Filter

from images. Therefore, noise suppression is very important and is sometimes difficult. Another purpose of image processing is to enhance and extract useful information, such as edges and image pixels representing the same scene surface. Usually, a significant change in brightness in an image corresponds to an object's surface boundaries. This attribute is used for enhancing and extracting information. Using a Laplacian filter, which is a differential operator, enhances image region border (bottom right).

Scene analysis provides applications with structured description of a scene. This description is often concerned with a three-dimensional (3-D) scene. In industry applications, geometric descriptions in both low and high level are needed. Scene analysis approaches obtain information from a description in relatively low level, and generate a more sophisticated description at high level. For instance, the low-level wireframe description shown in Figure 11.5 can first be obtained from the images given in Figure 11.4. From this description, middle-level surface representation can be obtained by analyzing which edges enclose a surface. From the wireframe representation, a high-level solid model might also be built depending upon the completeness of the wireframe model.

Three-dimensional scene analysis relies on 3-D vision sensors which are able to generate initial descriptions for a scene. A 3-D vision sensor belongs to either the active or the passive type [18]. A passive sensor detects reflections of illumination provided by the environment lights, and an active sensor receives reflected light that is controlled by a special light source such as a laser. Passive sensors are usually appropriate for military use or applications where active light sources are not allowed for security or environmental constraints. However, active sensors can be used in many industrial applications. Since it is very difficult and not robust

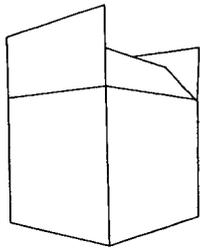


Figure 11.5: Wireframe-based Geometric Model is Built from the above Image

to extract 3-D information using a passive sensor, the research described in this chapter has considered the use of active sensors. More details on active sensors will be found in the next section.

11.2 Active 3-D Surface Data Acquisition

Since an active sensor usually measures the distance (range) between the sensor and object surface, it is referred to as a range sensor, a range finder, a range-imaging sensor, or a laser range scanner. In the following discussion, the term range sensor will be used. A range sensor is capable of generating a range image of a scene with dedicated hardware and software. A range sensor generates a range image that is a 2-D array. Each element of the image array records a number to represent a range (a coordinate value). This element is indicated by two subscripts to represent the range measurement direction. Commonly used range sensors are the triangulation-based and time-of-flight- (or phase-shift-) based sensors.

11.2.1 A Triangulation-based Range Sensor

The triangulation principle is straightforward, and is frequently used for measurement. The principle can be mathematically represented by the law of sines as

$$\frac{\sin(\theta_{laser})}{S} = \frac{\sin(\phi)}{T} = \frac{\sin(180^\circ - \theta_{laser} - \phi)}{B} \quad (11.1)$$

The triangulation-based range sensor shown in Figure 11.6 employs a HeNe laser source with an optical straightline generation lens, a charged coupled device (CCD) camera, and a nodding mirror. The laser source, object surface point, and the camera form a triangle. By applying the law of sines the distance between an object surface point and the camera is obtained.

The laser beam is extended into a line-structured light (not shown in the Figure 11.6) by going through the straightline generation lens. This line-structured laser light is projected by the nodding mirror which rotates to perform horizontal scanning of the laser light. Then

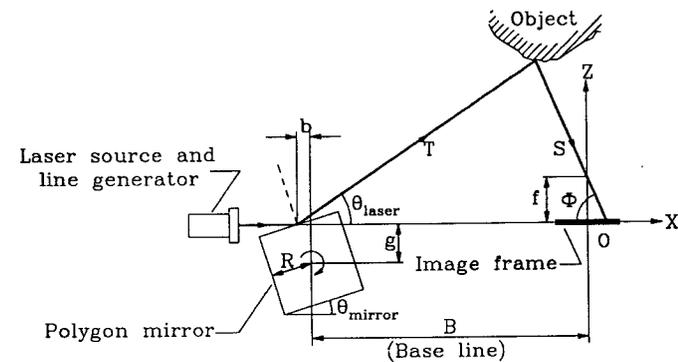


Figure 11.6: Geometry of a Triangulation-based Range Sensor

the reflected laser light, after hitting the object surface, is caught by the CCD camera. An image of an object with the laser light is shown in Figure 11.7.

The laser light is extracted from the image by subtraction from an image of the same scene without the laser light and by taking thresholds. The resulting laser light image is scanned row by row to find each pixel corresponding to the laser light. The image frame's row and column numbers of each found pixel are used for the range calculation.

The camera is considered as a pinhole camera whose geometry is simple. The geometric position, in terms of the image plane, of a pixel corresponding to laser light is obtained by the transform

$$x_{plane} = (x_{center} - x_{frame}) \cdot aspect_ratio_x \quad (11.2)$$

$$y_{plane} = (y_{center} - y_{frame}) \cdot aspect_ratio_y \quad (11.3)$$

where, x_{center} and y_{center} represent the center of the image frame, and $aspect_ratio_x$ and $aspect_ratio_y$ are the interval distances between two pixels in the horizontal and vertical directions, respectively.

The angle ϕ is found as

$$\phi = \tan^{-1}\left(\frac{f}{x_{plane}}\right) \quad (11.4)$$

The coordinates of a surface point with respect to the frame $X - Y - Z$ are obtained from

$$X = \left(1 - \frac{S}{\sqrt{x_{plane}^2 + f^2}}\right) \cdot x_{plane} \quad (11.5)$$

$$Y = \left(1 - \frac{S}{\sqrt{x_{plane}^2 + f^2}}\right) \cdot y_{plane} \quad (11.6)$$

$$Z = \frac{S}{\sqrt{x_{plane}^2 + f^2}} f \quad (11.7)$$

where, according to the law of sines, S is found as

$$S = \frac{(B + b + x_{plane}) \cdot \sin(\theta_{laser})}{\sin(\theta_{laser} + \phi)} \quad (11.8)$$

To a triangulation-based range sensor, “missing data” are a significant drawback which is caused by separation of the laser source and the camera. Part of the laser light may be occluded from the viewing direction of the camera by other objects in the scene. In Figure 11.7 discontinuous laser light can be seen, since the scene hides part of the light. This disadvantage exists in almost all sensors which are based on the structured light and/or triangulation measurement.



Figure 11.7: Image of a Scene with the Laser Light

11.2.2 A Time-of-Flight-based Range Sensor

The structure of a typical time-of-flight range sensor, the ERIM (Environmental Research Institute of Michigan) range finder [21], is illustrated in Figure 11.8. A laser beam is generated by a laser diode and is projected onto a polygon mirror through a collimator and an expansion telescope. The polygon mirror performs horizontal scanning by reflecting

the beam onto the nodding mirror. The nodding mirror controls the vertical scanning of a scene. The reflected beam from the object surfaces is received by a receiver through the two mirrors and a similar optical path.

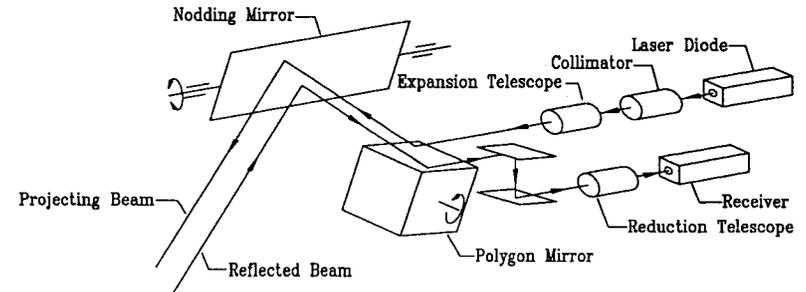


Figure 11.8: Structure of a Time-of-flight-based Range Sensor

The ERIM range sensor uses active image acquisition with the laser scanning. The range data are acquired at equal angular intervals of the nodding and polygon mirrors. The laser beam is sent out and reflected back to the range sensor, and the time of flight is measured to determine the range of the projected point.

The optical paths of the range sensor can be divided into exterior and interior portions, defined by the extrinsic and intrinsic parameters of the time-of-flight sensor. The exterior working space of the sensor is formed by rotations of the polygon and nodding mirrors. The interior optical paths can be ignored to consider the sensor as a point, if the sensor is not located very close to the object surface [23].

This introduces a spherical coordinate system, as illustrated in Figure 11.9, where, α , β , and r are extrinsic parameters of the sensor. The parameter r is the distance between the projected point on the scene and the sensor. The coordinates of the projected point P on the object, specified in this spherical coordinate system, can be transferred back into a Cartesian coordinate system aligned with the range sensor by

$$x(\alpha, \beta, r) = r \sin(\beta) \cos(\alpha) \quad (11.9)$$

$$y(\alpha, \beta, r) = r \sin(\alpha) \quad (11.10)$$

$$z(\alpha, \beta, r) = r \cos(\beta) \cos(\alpha) \quad (11.11)$$

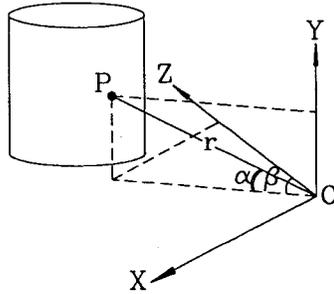


Figure 11.9: Cartesian and Spherical Coordinate Systems of the Range Sensor

11.2.3 Pseudo Range-image Generation

In research on geometric modeling with range data, a tool for generating pseudo range data is very helpful. The generated pseudo range data allow: (1) algorithm testing with images of objects of various shapes in a variety of desired positions and orientations, and (2) the control of the affect of image noise for the purpose of research.

The pseudo range-data generation system includes a solid modeling subsystem, a range-data generation subsystem, and a display subsystem as shown in Figure 11.10. The solid modeling subsystem, which has the basic functions of geometric modeling and interfaces to a CAD system, provides geometric models of desired objects and workspace [24]. Primitive solids of various types are represented by B-rep using the winged-edge data structure [8]. More complex objects can be created using a CAD system in polyhedra form and incorporated into the system through metafiles. The logical operations of OR on the primitives are conducted by a modified z-buffer algorithm. This approach greatly simplifies the generation and manipulation of a solid model.

The modified z-buffer algorithm, called *r*-buffer, is applied to generate the depth map of surface points, that is, range data. It simulates a scanner composed of two scanning mirrors that perform horizontal and vertical scanning. Therefore the *r*-buffer works in the spherical coordinate system. The *r*-buffer approach consists of a matrix, the elements of which are indexed by the polygon mirror angle (column) and the nodding mirror angle (row). Each element of the matrix corresponds to a "light beam" with fixed orientation. In range-data generation, the beam scans through the surfaces of all pertinent objects. When the beam strikes a surface, the range of the struck point is calculated. If this range is less than the previously calculated range (of another surface) stored in this matrix element slot, the newly calculated range replaces the old one in the slot. Eventually, each element of the matrix contains the minimum range, *r*, along the light beam between surfaces of the pertinent objects and the range sensor. The range data of an example scene generated by the simulation system are illustrated in Figure 11.11. The scene is about a composite object composed of a polyhedron and an ellipse-shaped cone. The display subsystem uses a floating horizons algorithm or intensive image display to illustrate the range-data values.

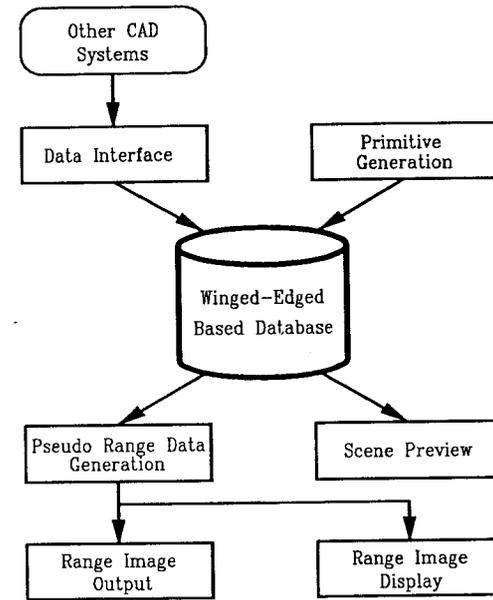


Figure 11.10: Structure of the Pseudo Range-data Generation System

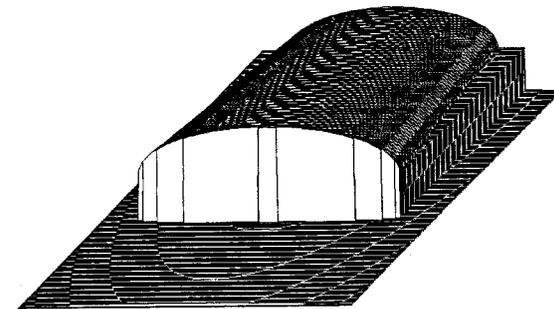


Figure 11.11: Pseudo Range-data Values Illustrated in a Floating Horizons Representation

11.3 Object and Workspace Modeling from Range Data

To obtain 3-D geometric models for existing objects and workspace, the initial description of a scene is required. In many situations, on-line creation of geometric models may be required. Active sensors are able to directly provide the required initial description by providing discrete range data representing object surface points. A key point is how to convert these discrete surface points into geometric models. This requires a transformation from a 3-D image representation of objects to a relatively concise geometric representation of the objects.

Due to the limited viewing field of a sensor and due to the possibility of object occlusion, acquiring range images for objects requires viewing from multiple range sensor locations to allow a complete geometry description to be obtained. Therefore, the modeling system needs to perform multiple-view fusion in its modeling processing. Flexible selection of viewing locations is quite important in order to handle objects and workspace with complex surface shapes. Guaranteeing this flexibility increases the difficulty of modeling from range images.

11.3.1 Review of Previous Methods

Previous work on geometric model generation from active range sensor image data utilizes geometric representations that require no sophisticated topologic relationships for modeling and facilitating the conversion from discrete 3-D range data to geometric models. Previous research has used mesh surface based representation, cross-section-based representation, and segmentation associated (surface or feature) representation. Most of these approaches manipulate range data in a Cartesian frame.

A mesh surface-based representation can have either regular mesh node distribution or irregular distribution. Faugeras and colleagues used a triangular facet-oriented polyhedral approximation to model sculptured surfaces [7]. The range data were represented and processed in a Cartesian coordinate system. Henderson used a graph—a spatial mesh to organize discrete surface points in 3-D space and obtain a surface representation through range data segmentation based on this graph [10]. To generate a workspace model for mobile robots, Asada developed a height map that was a mesh representation with regularly distributed nodes [1]. Despite the differences of their methods, these researchers all used multiple views to cover the entire object. Each of these views was usually taken using an auxiliary view selection device on a fixed trace. Examples of auxiliary devices include turntables, linear moving platforms, or high marks.

A cross-section-based representation is frequently used to generate a 3-D description in the volumetric rendering of computer tomography (CT) and/or magnetic resonance (MR). Since the CT/MR scanners conduct well-structured scanning, the acquired data are in ideal slice form. The 3-D volume can be formed by stacking all slices and interpolating between two neighboring slices [16]. A similar work on approximating 3-D surfaces using serial microscopic sections was reported in the research of biomedical engineering [13]. Theodoracatos used a

laser plane-of-light projector for obtaining range surface contours and for building the model of 3-D objects [20]. In these applications the form of the acquired image data is consistent with the cross-section representation. Different from the geometric representation using polyhedrons, the cross-section-based representation simplifies a complex 3-D description problem by handling the task in 2-D space.

Geometric modeling with image data is also found in the area of close range photogrammetry [5]. In the work of [5] one cross section or profile was captured by an angular scanning of the laser beam. With the step motion of the measuring system a tunnel surface was described by a series of profiles.

To process the raw range images for obtaining surface models, various segmentation algorithms are used. These algorithms divide an image into several local regions. All surface points in each of these local regions belong to a certain surface function. A surface-based representation such as a bicubic spline surface can thus be used to fit the regional image. Processed region by region, the object surface, initially recorded by the range images, is transformed into the surface-based representation. In-depth discussion of this type of technique can be found in [15] and [17].

11.3.2 Device Frame of Active Range Sensors

Both time-of-flight and triangulation-based range sensors work primarily by a central projection mechanism. In effect, this generates a device frame. The coordinates of the device frame can be two scanning angles and a range. Range data for each acquired surface point is a distance value between the sensor and the surface point and is associated with the direction pointing from the sensor to the projected surface point in the device frame. The direction can be defined by two angular scanning components, horizontal and vertical. When scanning, the angle changes equally in each scanning direction, and range data are stored into a rectangular array, called a depth map or a range image. Each element of this 2-D range image array is addressed by the horizontal and vertical scanning angles. In general, the device frame can be considered to be basically a spherical frame due to the central projection mechanism [12, 21, 23]. Hence, each surface point is expressed by its range value and two scanning angles in the object range image array. In the device frame array the data are dense and regularly distributed. The global coordinates of surface points can be calculated by transforming the corresponding range data from the device coordinate frame to the global coordinate frame.

In present practice, range image processing and/or range-data-based modeling are usually conducted in a Cartesian coordinate frame. Surface points are transformed from the device frame to the Cartesian frame, and are indicated by their x , y and z coordinates. Because such a transformation is nonlinear, the transformed surface points with x , y and z coordinates can no longer be uniformly represented in a 2-D array. Although some alternative data structures (such as a sparse 2-D array) can be used to represent these discrete points, computation costs will increase due to the complexity of handling a sophisticated data structure. A sparse 2-D array or a specially designed data structure also makes many present image processing methods either inapplicable or very costly to use. The coordinate transformation of the

massive amount of range data also contributes to high computation costs.

The remainder of this section discusses a representation allowing the characteristics of the data in the device frame to be exploited. Further details on basic representations can be obtained from [3, 10].

11.3.3 Selection of Representation

Different representations handle geometric data on different levels. To describe objects and workspace based on discrete range data of surface points and to provide applications with relatively concise geometric models of objects and workspace, an appropriate geometric representation must be considered and applied in the modeling system. The criteria for selecting such a representation primarily include (1) compatibility with discrete range data; (2) suitability for modeling in the device frame; (3) capacity for facilitating multiple-view fusion and model updating; and (4) efficiency of converting range data into the representation.

With the above requirements various representations, classified into several types with respect to their levels to represent objects, can be evaluated.

Point based— such as 3-D voxel array representation (a digitized space). Objects are depicted by a set of discrete surface pixels in low level. One voxel is adjacent to its neighbors in the digitized space; otherwise the description is not complete. It is difficult to adjust the resolution of the digitized space by changing the distance between the range sensor and surface to be scanned. This kind of adjustment is important to cope with surface details carried in the range images. Point-based representations also require a large amount of memory.

Line based— such as wireframe representation. Wireframes are intersections of adjacent surfaces. Surface properties are normally not described in this representation. The major disadvantage of a line-based representation is that range data do not explicitly provide intersection lines of surfaces and end points of each line.

Surface based— such as mesh, sweep, generalized cylinder (GC), and cross-section contour representations. Surface-based representation is harmonious with surface range data.

Mesh represents a surface by surface points and spatial grids. With the other representations, object shapes are primarily represented by stacking a series of planar contours at intervals. These representations are a combination of discrete and continuous modeling. Mesh can be used by linking the surface points with a graph to form a polyhedron of triangular- or rectangular-shaped boundary elements. Object modeling based on this representation requires relatively extensive computations. Cross-section contours can be formed from object surfaces described by surface points with moderate computations. The representation is compatible with range data. Difficulties of handling shape details, data condensation, and fusion are reduced due to the use of the 2-D space of cross-section planes as the working space. Sweep and

GC are similar to the cross-section form except GC contains fewer geometric details and sweep is inadequate for representing irregular shapes.

Volume based— such as octree representation. Not only the surface but also the volume of an object is modeled by this representation in a discrete form. The major problem with using this type of representation with range data is that the involved recursive spatial subdivision requires complete data acquisition before modeling. However, this requirement cannot in general be satisfied.

Boundary based— such as B-rep representation. A boundary-based representation of an object includes surface facets and surface borders. Linkages between borders and facets are required. Models in this representation are concise and in high level. However, this representation is not compatible with range data due to the required linkages.

Object based— such as CSG representation. This type of representation describes an object using geometric primitives. The geometric and topologic properties of the object are incorporated into the representation. The topologic relations at all levels of the model are explicitly required. These requirements make an object-based representation unsuited for direct use with range data.

Based on this comparison, the surface-based representations are most appropriate for representing range data. Of surface-based representations the cross section is superior to others in terms of the computational efficiency in multiple-view fusion and the ability to describe shape details and complex surfaces. Therefore, a cross-section-based approach is used in this work to serve as the representation for building geometric models from range data.

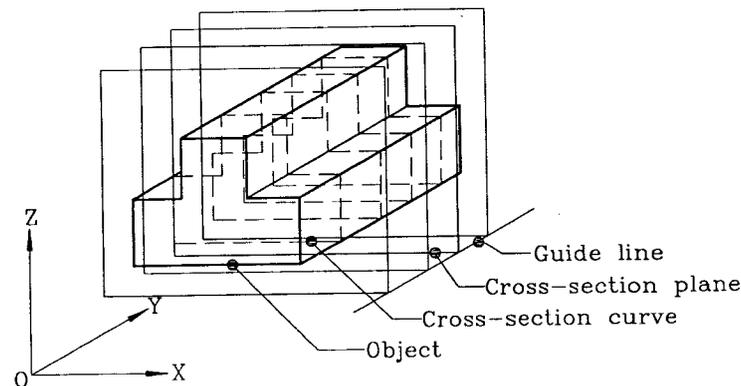


Figure 11.12: Cross-section-based Representation

Cross-section representation uses 2-D surface contours lying on a series of planes to represent object shapes. These contours are called cross-section curves, and the planes are called cross-section planes. Cross-section planes are arranged along a guide curve. The planes are arranged to be always perpendicular to a guide curve. If the guide curve is a straight line, the planes form a group of parallel planes.

A modeling frame, which is a global 3-D Cartesian coordinate system, is introduced for using the representation. The guide curve and the cross-section planes are specified in this Cartesian modeling frame.

A 2-D cross-section curve on a cross-section plane is the intersection curve of the plane and the surfaces of the 3-D objects to be modeled. If the modeled object is bounded, the corresponding cross-section curves will be closed when complete. The cross-section curves for an unbounded surface, for example, the workspace surroundings, will be open. The surfaces of modeled objects can be approximated by interpolating the cross-section curves. Since the outer and inner sides of object boundary are identified by the curves, the volume of the modeled objects can be computed. When the object surface changes greatly, surface details can be preserved by decreasing the interval of the cross-section planes.

A simple object represented by cross sections is illustrated in Figure 11.12 where the cross-section curves, the cross-section planes, and the guide curve are shown.

11.3.4 Modeling in Device Frame with Cross-section-based Representation

The generation of cross-section curves can be conducted in the Cartesian modeling frame. However, the range data are nonuniformly distributed in a Cartesian frame, and the processing will be complex. Therefore, in this work cross-section curve generation in the device frame is introduced to preserve the uniform distribution of the surface points (raw range data) in a 2-D array. This approach requires the cross-section planes to be transformed into the device frame in order to find the intersections (cross-section curves) of these planes and the object surfaces represented by the range data.

Three coordinate systems are involved in this modeling approach: the global Cartesian modeling frame; a device Cartesian frame attached to the range sensor; and the device frame of the range sensor, which is in spherical form. Mapping between the global Cartesian modeling frame and the device Cartesian frame can be conducted by a homogeneous transformation, that is,

$$\{ X, Y, Z, 1 \} = \{ x, y, z, 1 \} \begin{bmatrix} [R]_{3 \times 3}^T & 0 \\ \{t\}_{1 \times 3}^T & 1 \end{bmatrix} \quad (11.12)$$

where $[R]$ is a 3×3 rotation matrix describing the orientation of the range sensor with respect to the global modeling frame, $\{t\}$ is a vector describing the location of the range sensor in

the global modeling frame, and $\{X, Y, Z\}$ and $\{x, y, z\}$ represent the coordinates of a point in the global Cartesian modeling frame and the device Cartesian frame, respectively.

The coordinates of the device frame are the two scanning angles, θ and ϕ , and the range, r . Transformation from the device Cartesian frame to the device frame has the analytical form given by Eqs. (11.9), (11.10), and (11.11).

The analytical expression of a cross-section plane in the global modeling frame can be represented by

$$Ax + By + Cz + D = 0 \quad (11.13)$$

Transformation to the device Cartesian frame yields

$$ax + by + cz + d = 0 \quad (11.14)$$

where $\{a, b, c\} = \{A, B, C\} [R]$ and $d = D - \{t\}^T \{a, b, c\}^T$.

Substituting for x, y and z from Eqs. (11.9) to (11.11) yields

$$ar \cos \theta \cos \phi + br \cos \theta \sin \phi + cr \sin \theta + d = 0 \quad (11.15)$$

Due to the nonlinear form of Eq (11.15), it is difficult to use it directly for finding the plane-to-object intersections and for conducting range image interpolation. In particular, searching the solution all over the image would be a very time consuming task. In this work, a quick estimation method for finding plane-to-object intersections using discrete values is introduced. This method is explained below.

A cross-section plane transformed to the device frame forms a range image in a 2-D array. This image will be named the plane image. The plane image can be generated with identical size and the same subscripts of the object range image using the two scanning angles as indices. As illustrated in Figure 11.13a, the plane and object can be assumed to simultaneously exist in 3-D space. Since the existence of a cross-section plane is hypothetical, the range images of the object and the cross-section plane can be generated separately as illustrated in Figure 11.13b and c. It should be noticed that the plane range image is not produced by the sensor with physical perceiving, but by the modeling system.

With these two identical indexed range images, it is easy to estimate the object surface to cross section plane intersection, that is, the cross-section curve. A simple solution would be scanning the two range images pixel by pixel and connecting these pixels where the discrete object image and plane image share a same value, as shown in Figure 11.13c.

In practice, the object-to-plane intersection may occur at a subpixel location. When the range of the plane changes from larger (or smaller) than the range of the surface to smaller (or larger) at two neighboring pixels, an intersection exists somewhere between these two pixels. The intersection can be generated by interpolating the two continuous surface patches formed by the range values of the two images at the four neighboring pixels, as shown in Figure 11.14d.

The generated curve section is then transformed to the global Cartesian modeling frame to update the present model. Through successive viewing and updating multiple-view fusion

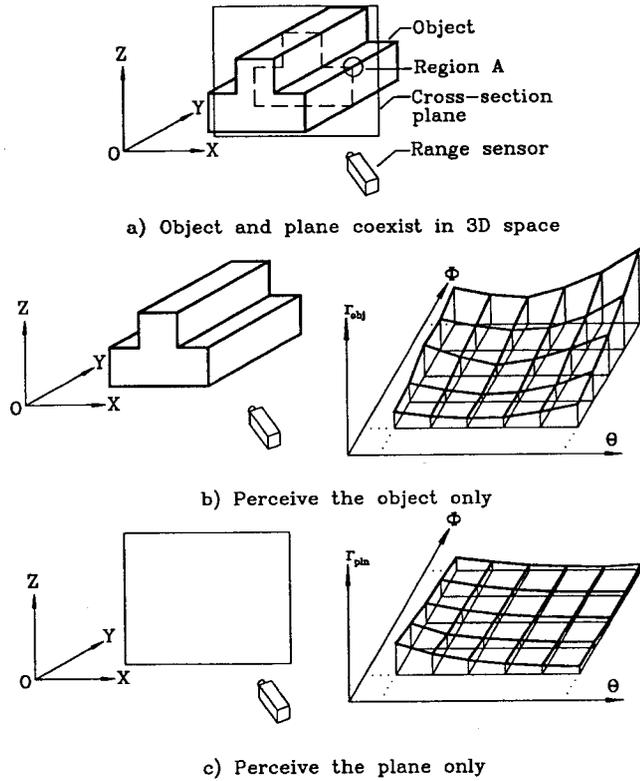


Figure 11.13: Generation of the Object and Cross-section Plane Range Images

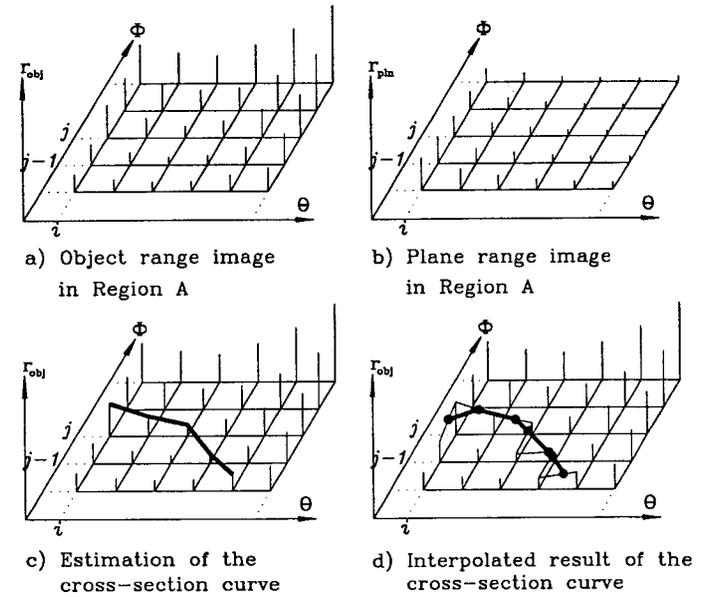


Figure 11.14: Estimation and Interpolated Solution of the Cross-section Curve

and a complete geometric description can be achieved. Successive viewing and updating also allow basing the next view to be selected on interpretation of the updated model (intelligent multiple-view selection).

11.3.5 Advantage of Range-data Processing in Device Frame

In previous research, range-data processing and object modeling using a cross-section representation are conducted in a Cartesian coordinate frame. The computational cost of these approaches is quite high due to the need for searching data with indirect addressing and for massive data transformation, as discussed previously. The device-frame-based range-data processing method introduced in this work directly uses the device frame for range-data processing. Significant savings in computational effort can be found after comparing the data processing procedures of these two approaches. This saving is primarily due to (a) only the solved intersection curves (several control points), rather than all of the raw range data, need to be transformed between different coordinate frames; (b) the “transform” of the involved cross-section planes from the global Cartesian modeling frame to the device frame can be conducted using a simple analytical formula; and (c) the range data are preserved in their well-distributed 2-D array form. This well-distributed form can allow efficient existing image processing techniques to be applied.

11.3.6 Complete Description for Objects and Workspace

A complete geometric model for 3-D objects and workspace using the proposed cross-section representation can be built by generating all cross-section curves from multiple-view range data.

After a range image is acquired, a partial surface model that consists of a series of incomplete cross-section curves is constructed. This partial model is added to and integrated with the present object model, which was generated previously by the earlier processed partial models. The multiple-view fusion is implicitly accomplished. The partial model merging activity can eliminate any redundant surface data caused by overlaps of different range images and identify surface discontinuities caused by gaps between range images. The model updating procedure can be expressed by the *OR* Boolean operation

$$S_{obj_n} = S_{obj_{n-1}} \cup s_n \quad n = 1, 2, \dots$$

where S_{obj_n} is the updated object surface model, $S_{obj_{n-1}}$ is the previous object surface model, and s_n is a partial surface model. Initially, when $n = 1$, $S_{obj_{n-1}} = \emptyset$.

Overlaps between partial models (partial redundancy) exist when

$$S_{obj_{n-1}} \cap s_n \neq \emptyset$$

Gaps exist between partial models or s_n is perfectly adjacent with $S_{obj_{n-1}}$ when

$$S_{obj_{n-1}} \cap s_n = \emptyset$$

When the current surface measure is totally redundant to the previous ones, we have

$$S_{obj_{n-1}} \cap s_n = s_n$$

Specifically, updating the object model is performed on the involved cross-section planes in the global Cartesian modeling frame. On each cross-section plane any overlaps between the current curve section and the previously generated curve section(s) are tested. If an overlap of curve section is found, the redundant part is eliminated, and the newly generated curve section is connected to the curve section(s) generated previously. If the newly generated curve section is separated from the previously generated curve sections in the current object model, the new curve should be stored separately. For a complete object model, gaps between separated curve sections should all have been filled. Once the partial modeling for an acquired range image and the model updating using the newly created model are completed, the next range image can be taken and processed. By this updating approach, as long as the observation for object or workspace is complete, the geometric descriptions will be complete too.

11.4 An Example

Illustrated in Figure 11.15 is an object that will be modeled using the cross-section representation. The object consists of a truncated cone merged with a rectangular parallelepiped. Range data from three sensor locations will be used to construct the cross-section model.

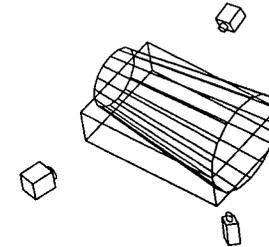


Figure 11.15: Three Different Sensor Locations

Figure 11.16 shows an intensity visualization of the range image for the first sensor location. Based on this range data the partial cross-section curves illustrated in Figure 11.17 were found. Based on data from a second viewing location (range intensity visualization in Figure 11.18) the model was updated yielding the curves illustrated in Figure 11.19. A final sensor location yielded sufficient range data (intensity visualization in Figure 11.20) to complete the cross-section model. The completed cross-section model based on all three views is shown in Figure 11.21.

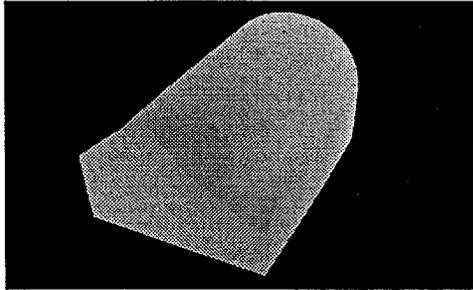


Figure 11.16: Range Image (Intensity Visualization) from a First Sensor Location

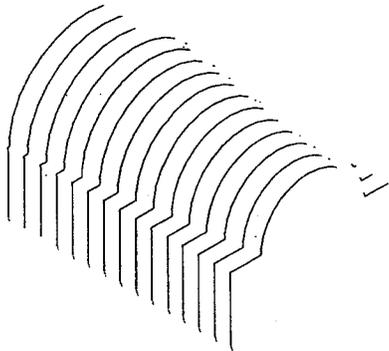


Figure 11.17: Partial Model Based on the Range Image from a First Sensor Location

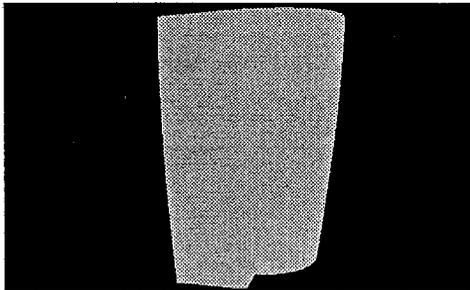


Figure 11.18: Range Image (Intensity Visualization) from a Second Sensor Location

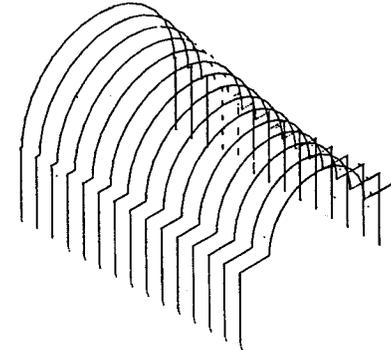


Figure 11.19: Updated Model Based on a Second Sensor Location

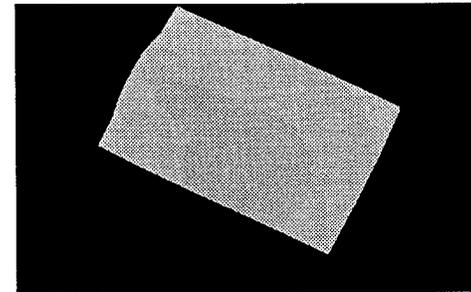


Figure 11.20: Range Image (Intensity Visualization) from a Third Sensor Location

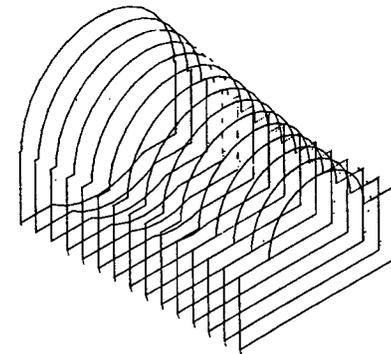


Figure 11.21: Complete Model Based on the Three Sensor Locations

11.5 Summary

Geometric data input is important for applications of intelligent robotic and manufacturing systems. In many cases, these applications depend on the understanding of the work environment. This dependence has strongly activated research and development of 3-D machine vision and geometric modeling based on vision data. This chapter has reviewed the basic principles of active range sensors and previous approaches of forming geometric representations based on range information. Furthermore, recent research on a new approach for generating a representation based on 3-D range information has been discussed.

The discussed research focused on a geometric modeling approach for representing multiple-view range data. Acquiring object range images requires viewing from various sensor locations (multiple views) to allow a complete geometric description. A flexible multiple-viewing approach requires a modeling technique that is able to integrate partial models together. Effective on-line application of a modeling approach for such range data requires the capability of simultaneous performance of acquisition and modeling.

A cross-section-based representation has been used for supporting implementation of the introduced modeling approach. This representation uses a series of object contours to describe geometric shapes. Each contour curve is built from partial model curve sections by model updating in a global Cartesian modeling frame. The partial model curve sections can be formed efficiently in the device frame. Low computational cost of range-data processing in the device frame has been found possible due to fast searching for adjacent surface points and transforming generated curves rather than raw data to the Cartesian modeling frame.

The research introduces a new geometric modeling approach for representing range data. The approach allows partial modeling and flexible multiple-view selection. The concept of modeling after the acquisition of each view allows decreased memory requirements. The approach should provide subsequent researchers with a platform for the consideration of intelligent view acquisition schemes. In addition, the new device-frame-based range-data processing method increases data processing speed by avoiding computationally expensive range-data processing in a Cartesian frame.

References

- [1] Asada, M. (1988). "Building a 3D World Model for a Mobile Robot from Sensory Data," *Proceedings of the IEEE International Conf. on Robotics and Automations*, Vol. 2, pp. 918-923.
- [2] Ballard, D.H., and Brown, C.M. (1982). *Computer Vision*, Prentice-Hall, Inc.
- [3] Besl, P., and Jain, R. C. (1985). "Three-Dimensional Object Recognition," *Computing Surveys*, Vol. 17, No. 1, pp. 73-145.
- [4] Besl, P. (1989). "Active Optical Range Imaging Sensors," *Advances in Machine Vision*, J. Sanz, ed., Springer-Verlag, pp. 1-63.

- [5] Clarke, T. (1990). "The Use of Optical Triangulation for High Speed Acquisition of Cross Sections or Profiles of Structures," *Photogrammetric Record*, Vol. 13, No. 76, pp. 523-532.
- [6] Fan, T.J. (1989). *Describing and Recognizing 3D Objects Using Surface Properties*, Springer-Verlag, London.
- [7] Faugeras, D., Hebert, M., Mussi, P. and Boissonnat, J. (1984). "Polyhedral Approximation of 3D Objects without Holes," *Comput. Vision, Graphics, and Image Process.*, Vol. 25, pp. 169-183.
- [8] Foley, J. D., Dam, A. V., Feiner, S.K. and Hughes, J.F. (1990). *Computer Graphics Principles and Practice*, Addison-Wesley, Reading, MA.
- [9] Gonzalez, R.C., and Wintz, P. (1987). *Digital Image Processing*, Addison-Wesley.
- [10] Henderson, H. (1983). "Efficient 3-D Object Representations for Industrial Vision Systems," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-5, No. 6, pp. 609-618.
- [11] Horn, B. K. P. (1986). *Robot Vision*, MIT Press and McGraw-Hill, New York.
- [12] Kaisto, I., Kostamovaara, J., Moring, I., and Myllyla, R. (1990). "Laser Rangefinding Techniques in the Sensing of 3D Objects," *SPIE Sensing and Reconstruction of Three-Dimensional Objects and Scenes*, Vol. 1260, pp. 122-133.
- [13] Mercer, R., McCauley, G., and Anjilvel, S. (1990). "Approximation of Surfaces in Quantitative 3D Reconstructions," *IEEE Trans. Biomed. Eng.*, Vol. 37, No. 12, pp. 1136-1146.
- [14] Mortenson, Michael E. (1985). *Geometric Modeling*, John Wiley.
- [15] Neik, S.M., and Jain, R.C. (1988). "Spline-Based Surface Fitting on Range Images for CAD Applications," *Proceedings CVPR '88: The Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 249-256.
- [16] Ney, D.R., Fishman, E.K., Magid, D., and Drebin, R.A. (1990). "Volumetric Rendering of Computed Tomography Data: Principles And Techniques," *IEEE Computer Graphics & Applications* (March), pp. 24-32.
- [17] Orr, M.J.L. (1991). "World Modeling from Range Images," *International Conf. on Control '91*, pp. 807-812.
- [18] Poussart, D., and Laurendeau, D. (1989). "3-D Sensing for Industrial Machine Vision," *Advances in Machine Vision*, J. Sanz, ed., Springer-Verlag, pp. 122-159.
- [19] Requicha, A.A.G. (1980). "Representations for Rigid Solids: Theory, Methods, and Systems," *Computing Surveys*, Vol. 12, No. 4, pp. 437-464.

- [20] Theodoracatos, V., and Katti, R. (1991). "An Automated and Interactive Approach for Fitting B-Spline Surfaces through 3D Planer Visual Data," *Proceedings of the 1991 ASME Advances in Design Automation*, Vol. 2, ASME, New York, pp. 23-31.
- [21] Veatch, P. A., and Davis, L. S. (1990). "Efficient Algorithms for Obstacle Detection Using Range Data," *Computer Vision, Graphics, and Image Processing*, Vol. 50, pp. 50-74.
- [22] Wang, Y.F., and Aggarwal, J.K. (1988). "An Overview of Geometric Modeling Using Active Sensing," *IEEE Control Systems Magazine*, Vol. 8, No. 3, pp. 5-13.
- [23] Yao, H., Podhorodeski, R., and Dong, Z. (1991). "Modeling and Simulation of a Range-Finding Device," *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, IEEE, New York, pp. 166-169.
- [24] Yao, H., Dong, Z., and Podhorodeski, R. (1991). "Simulation of Range Finding Devices by Geometric Modeling," *Proceedings of the 1991 ASME Computers in Engineering Conference and Exposition*, pp. 327-332.
- [25] Zeid, I. (1991). *CAD/CAM Theory and Practice*, McGraw-Hill, Hightstown NJ.

Library of Congress Cataloging-in-Publication Data (

Artificial intelligence in optimal design and manufacturing/Zuomin Dong (Ed.)

p. m.

Includes index.

ISBN 0-13-037540-3

1. CAD/CAM systems. 2. Production planning. I. Dong, Zuomin.

TS155.6.A77 1994

670'.285'63--dc20

93-552

CIP

ARTIFICIAL INTELLIGENCE IN OPTIMAL DESIGN AND MANUFACTURING

Zuomin Dong, Editor



©1994 by P T R Prentice-Hall, Inc.

A Simon & Schuster Company

Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-037540-3



P T R PRENTICE HALL

Englewood Cliffs, New Jersey 07632

Prentice-Hall International (UK) Limited, London

Prentice-Hall Australia Pty. Limited, Sydney

Prentice-Hall Canada Inc., Toronto

Prentice-Hall Hispanoamericana, S.A., Mexico

Prentice-Hall of India Private Limited, New Delhi

Prentice-Hall of Japan, Inc., Tokyo

Simon & Schuster Asia Pte. Ltd., Singapore

Editora Prentice-Hall do Brasil, Ltda., Rio de Janeiro