Order Number 9009744

Shape description and boundary restoration

Jankowski, Mariusz, Ph.D.

City University of New York, 1989

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

17

SHAPE DESCRIPTION AND BOUNDARY RESTORATION

by

MARIUSZ JANKOWSKI

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1989

This manuscript has been read and accepted by the Graduate Faculty in Engineering in satisfaction of the dissertation requirements for the degree of Doctor of Philosophy.

8/2/89
Date

Chair of Examining Committee

8/2/89
Date

Executive Officer

Professor M. Anshel

Professor J. Barba

Professor S. Basu

Professor L. Roytman

Dr. J. Keybl (UNISYS Corp.)

Examining Committee

The City University of New York

Abstract

SHAPE DESCRIPTION AND BOUNDARY RESTORATION

by

Mariusz Jankowski

Adviser: Professor George Eichmann

There is a growing interest in robotics, machine vision and computer graphics fields in developing efficient representations of three-dimensional (3D) objects. A choice of a representation scheme is critical to the efficient manipulation and storage of image data. Of special interest are geometrical representations. Since surfaces are what is seen, surface representations are important to computer vision. The enclosing surface, or boundary, of a well behaved 3D rigid solid unambiguously specifies the object. Most importantly, the boundary captures the notion of shape, which is an intrinsic property of solid objects. For purposes of pattern recognition, it is of great interest therefore, to represent shape numerically. The resulting so-called shape descriptors provide a canonical, viewer independent image representation that greatly simplifies the recognition task. The notion of similarity of objects, based on their shape characteristics, is simplified to a measurement of distance in an N-dimensional feature space. A new three-dimensional (3D) shape descriptor is calculated from the differential characteristics of a solid's bounding surface. Classification results for polyhedral and cylindrical solids are presented.

A common and mainly unsolved problem in image processing is occlusion. Error-correction that restores the boundary to its original shape makes shape description possible under conditions of partial boundary visibility. An error-correcting associative memory is used to solve the restoration problem. The associative memory is a special case of a neural network (NN). The generalizing and associative

properties of NNs are especially well-suited to handle problems which involve distorted, "blurry", and unpredictable data like the boundary reconstruction problem. Examples are given of boundary restoration for three-dimensional (3D) range data images of solids and two-dimensional (2D) boundary curves. Both, a Kohonen associative memory and a back-propagation NN are used.

Acknowledgment

I would like to take this opportunity to thank Prof. Goerge Eichmann for his guidance and support. His comments and suggestions were always greatly appreciated.

A special thanks to those closest to me for their encouragement and patience.

Table of Contents

Table of Figures

Table of Tables

1 Introduction.

There is a growing interest in robotics, machine vision and computer graphics fields in developing efficient representations of three-dimensional (3D) objects. A choice of a representation scheme is critical to the efficient manipulation and storage of object data. The different requirements and goals of the varied applications which manipulate 3D image data, result in a wide spectrum of available representation schemes [7,17,48,54,64]. Of special interest are geometrical representations, since they lend themselves to analysis with techniques based firmly on the mathematics of curves, surfaces and volumes. A further motivation for the use of geometric representations are the recent advances in data acquisition by ranging [12,37]. Ranging results in an image of the real world as a matrix of distance values, known as the range image or depth map. Due to the surface explicitness of this type of information, depth maps are finding increasing use in machine vision.

Since surfaces are what is seen, surface representations are important to computer vision. The enclosing surface, or boundary, of a well behaved 3D rigid solid unambiguously specifies the object. It frequently is the only information available. Significant local and global information can however be extracted from the raw data to facilitate the automatic analysis of multidimensional images, for purposes of detection, classification, restoration and coding. Most importantly, the boundary captures the notion of shape, which is an intrinsic property of solid objects. For purposes of pattern recognition, it is of great interest therefore, to represent shape numerically. Effective and efficient shape description capabilities are invaluable in computer vision, pattern recognition and image processing. From past two-dimensional (2D) image processing experience it is clear that raw image data is frequently unsuitable for analysis purposes. The amount of data manipulation, together with some of the real time requirements, taxes even the present day computing capabilities. It is important therefore, that higher-level shape representation schemes significantly decrease memory requirements. Additionally, as an important aid in the recognition

task, some or all of the following properties must characterize a representation scheme: invariance to scaling, translation and rotation. It has been demonstrated that the differential characteristics of a surface poses such invariance properties [19,50,69]. Differential invariants such as surface critical points (maxima, minima, and saddle points), zero-crossings of the first (Gaussian) curvature, principal curvatures, and geodesics have all been used with varying degrees of success. The coefficients of the first and second differential forms of a smooth surface uniquely characterize the shape of that surface. The local surface properties can be augmented by global functions of the surface in order to provide an even richer image description. These can in turn be transformed, using well known signal processing techniques. Transforms of global functions can yield significant reductions in data storage requirements, since significant shape information tends to be localized in small areas of the transform domain. Such canonical, viewer independent image descriptions greatly simplify the recognition task. Shape descriptors have been shown to be very effective in 2D imaging [9,52,54]. In Sect. 3 a new 3D shape descriptor is described. Examples of recognition of polyhedral and cylindrical solids are given.

A common and mainly unsolved problem in image processing is occlusion. There are a number of situations under which occlusion occurs. The most obvious, when one or more objects obstruct the view of the one being sensed. In 3D sensing objects invariably occlude themselves resulting in partial surface visibility. Finally, a part of the object might be out of range of the sensing equipment, again resulting in a partial view. Global boundary based representations are severely affected by occlusion [23,47,53,67]. The inability of global representation schemes to gracefully handle incomplete information is considered a major disadvantage, since occlusion of solids is an ever present phenomenon. The occluded boundary may be considered as an incomplete version of the original boundary. Error-correction that restores the boundary to its original shape makes shape description possible under conditions of partial boundary visibility. This motivated the research into the problem of boundary reconstruction under partial visibility and other forms of distortion, which is addressed

in Sect. 4. An error-correcting associative memory is used to solve the restoration problem. The associative memory is a special case of a neural network (NN). It is a NN with no hidden units and linear activation functions. The generalizing and associative properties of NNs are especially well-suited to handle problems which involve distorted, "blurry", and unpredictable data like the boundary reconstruction problem [1,30,41,46,48,58]. Results of computer simulations for 3D solids, planar images of handwritten Roman letters and numerals and airplanes are given.

2 Background.

The purpose of this section is to present some of the issues and survey some of the results pertaining to the representation, recognition, and reconstruction of 3D solid objects based on data derived from various sensor measurements. Current data collection techniques that allow the computation of 3D scene structure are laser and ultrasonic ranging, computed tomography, digital stereoscopy and scanning electron microscopy. This is the domain of 3D computer vision. Data acquisition techniques which provide explicit geometrical information about the sensed image will be discussed (Sect. 2.1). The availability of such data is fundamental to the 3D vision problem. Geometrical data directly describes surfaces which are natural for object identification. It contains shape information which is an intrinsic property of objects. Representational schemes used in image synthesis will also be reviewed (Sect 2.2). The fields of computer graphics and solid modeling are important because they share with machine vision the problem of representing real world objects by computer. In Sect 2.3, representational schemes currently in use in 3D imaging are presented. These are methods in which a object or scene description is derived directly from the raw data of an intensity or range sensor. Finally, transform domain shape description schemes are reviewed.

2.1 Data Acquisition.

Recently great deal of interest has been shown in the use of range data for the analysis of scenes and remote object recognition. The usefulness of range imagery in such applications is based on the explicitness of shape information. The range image is a matrix of distance values, which may be considered as noisy, discrete samples of some surface described by the function $z = f(x,y)$. Such a surface description is known as a depth map, and is also referred to as a Monge patch surface. A range image makes depth perception explicit, since the z values indicate directly how far or close a given surface point is to the viewer. It is

because of this explicit nature of the information about the 3D localization of surface points, that object recognition should be much less difficult using depth maps than intensity images [12]. The following is a listing of different types of range finding techniques [38]. Direct and active ranging includes ultrasonic and time-of-flight estimation and triangulation systems. All involve controlled energy beam and reflected energy detection. Passive range finding from single viewpoint includes texture gradient analysis, photometric methods (surface normals from reflectance), occlusion effects, size constancy and focusing methods. This leaves methods based either on motion or multiple relative positions of camera or scene: these include reconstruction from multiple views, stereo disparity, retinal flow and other motion related techniques. In addition a number of contrived lighting approaches are used, which include striped (a rotating mirror, vertical slit projector, and TV camera) and grid lighting, patterned lighting and Moire fringe analysis. A more detailed discussion of some of these techniques follows.

Texture gradient refers to the increasing fineness of visual texture with depth observed when viewing 2D image of a 3D scene containing approximately uniformly textured planes or objects. This is clearly a phenomenon useful in range analysis of natural outdoor scenes with large areas of uniform texture. It has been indicated that texture is very important to human depth perception [31]. With texture embedded on a planar surface depth cues can be obtained from the analysis of the changing size of the surface primitives (size becomes smaller as a surface recedes into background). The depth derivable from texture gradient is only relative unless the actual size of the texture element is known as basis of calibration. In one of the methods the 2D discrete Fourier transform was used for this purpose [6]. It is claimed that the Fourier descriptors obtained thereby vary in a manner consistent with surface geometries in 3D.

One obvious way to obtain a depth map is to use a simple one spot at time triangulation scheme [37]. Given the distance between light source and detector

and their orientation at the instant the detector "sees" the light, the range can be easily recovered. However, sometimes because of occlusion or surface absorbance no return light is encountered. This "missing parts" problem is common to all triangulation schemes.

Another obvious idea in 3D sensing is stereo vision which has clear biological justification [42,70]. Two cameras, or one camera from two positions, can give relative depth or absolute 3D location depending on the level of complexity of the processing and measurement. For two cameras simple geometry indicates that the image displacement (disparity) of the two views is inversely proportional to depth as measured from the camera. It is however necessary to establish correspondence or matching of points between the two images. Finding this correspondence is the hardest part of any stereo disparity scheme. If it is to be determined, there must be sufficient visual information at the matching points. Frequently this is not possible because of the presence of large areas of uniform intensity or some part of the scene appears in only one view of stereo pair because of occlusion effects.

An alternative to triangulation based range finding methods with their inherent "missing parts" problems and diminishing accuracy with range are ultrasonic and "time-of-flight" ranging techniques. No image analysis is involved, nor are assumptions about scene or object relevant. Furthermore, absolute range is directly available. There is however a significant problem with ultrasound ranging techniques, which does not allow measurements of range for large angles of incidence [18,38]. This occurs for a large number of commonly encountered surface materials and is a consequence of the intrinsic properties of acoustic waves and reflecting surfaces. For acoustic waves typical of ultrasonic ranging the Lambertian scattering phenomenon by which reflected energy scatters in all directions with equal probability, does not occur. The acoustic wave bounces off, mostly concentrated, in a direction where the angle of incidence and the angle of reflection are equal. Consequently little or no energy is reflected directly back to

the detector. Furthermore because of the wavelength of the beam (several millimeters) most small surface details are undetectable, i.e. resolution is poor. For these reasons, ultrasonic ranging is better suited for navigation purposes than highly accurate surface depth maps.

2.2 Solid Modeling Schemes.

The sensor data must first be organized into a representation form suitable for further processing and is strongly application dependent. Some representations are useful in describing biological organisms and are used in computer animation (i.e. image of a running animal or human). Other contain a wealth of geometric information and are therefore inherently useful in CAD/CAM applications. Additionally, the display of 3D images, necessary for man-machine interaction, presents its own set of distinct problems (hidden surface removal, shading, etc.) requiring specific representational forms. These however, need not be as sophisticated as the spatial-knowledge representation schemes needed for computer recognition and description. To accommodate the wide range of applications a large variety of representation schemes have been developed which is a consequence of the fact that no single representation is general and sophisticated enough to satisfy all the requirements.

Three-dimensional image representation techniques can be broadly categorized as belonging to one of the following three schemes: volumetric, generalized sweeps and surface based. The techniques belonging to the first category are very useful in 3D image synthesis and for that reason are popular in computer modeling and graphics, as well as in many planning (i.e. collision checking) and design problems. In its simplest formulation the spatial occupancy representation depicts an object as composition of many identical primitive elements, predominantly cubic elements called voxels (short for volume element) which are the 3D equivalent of a pixel. The cubes are of fixed size and lie in a fixed spatial grid. Each cell may

be represented in a purely geometrical fashion by the coordinates of a single point, such as the cell's centroid. The data structure is therefore a 3D binary array, marked as filled with matter or not. If necessary, additional information can be associated with each cell. This information could be non-positional in nature such as density, weight, temperature, stress, etc. In high resolution applications such arrays require much storage but it is also generally true that algorithms using this type of representation tend to be very simple. Another well known scheme based on a single primitive is the oct-tree [36]. This is a hierarchical representation. Volumes are decomposed into cubes of different sizes where the cube size depends on the distance from the root node. Each branching node of the tree structure represents a cube and points to eight other nodes that describe object volume occupancy in the corresponding octant sub-cubes of the branching node cube. This representation is clearly much more compact than the voxel representation. This compactness requires however more complicated algorithms. More sophisticated decompositions use cells that are more complex in shape [20,68]. However, as they still do not share volumes the only combining operation is "glue". At the highest level of generalization of the shape of the primitive element we have the hyper-patch: a generalization of bicubic surface patches [49]. This is a patch-bounded collection of points whose coordinates are given by continuous, three-parameter, single valued mathematical functions of the form,

$$x = x(u,v,w), \quad y = y(u,v,w), \quad z = z(u,v,w) \tag{2.1}$$

where the u,v,w parameters are constrained to some normalized interval, usually $u,v,w \in [0,1]$. This technique defines volume, surface area, and internal density variation of a solid. This is more general then other methods since they only allow uniform density distribution within a volume primitive. A price is paid for this generality in the complexity of the representation since 192 scalars are required for each volume element. A prominent technique in which this restriction of not sharing volumes has been removed is the Constructive Solid Geometry (CSG) scheme

[56,68]. Here solids are represented as Boolean constructions or combinations, via regularized set operations, of other solids which may have undergone rigid motions. At the lowest level are primitives such as arbitrarily scaled rectangular blocks, arbitrarily scaled cylinders and cones, and spheres of arbitrary radius. The data structure of a CSG representation is an ordered binary tree. The non-terminal nodes represent operators, which may be either rigid motions or regularized union, intersection, or difference. Terminal nodes are either primitive leaves, or transformation leaves which contain the defining arguments of rigid motion. CSG schemes are unambiguous but not unique, since an object will submit to a number of different decompositions. The domain of a CSG scheme depends on the set of primitive solids and on the available motional and combinational operators. If the primitive solids are well matched to the domain of objects to be depicted, the representation is very concise and easy to create by humans. The CSG representation is not an efficient source of detailed geometric data and because of its nonuniqness and type of data structure, is not very useful in automatic recognition. Currently, this representation is one of the two best understood and most important representation schemes for solid modeling and computer aided design.

Generalized sweeps, also known as Generalized Cylinders (GC) are popular category of representations in computer graphics [45,63]. They are intuitive since the volume of many biological and man-made objects can be naturally described as a "swept" volume of a planar, bounded area element moved along some curve in 3D. They appear to be quite useful in describing objects such as airplanes, animals, etc. which may be decomposed naturally into a few tubelike components. A common, simplifying restriction is to make the axis normal to the cross-section. Generalized Cylinders are appealing since relatively simple mathematics is adequate for representing them. A solid is defined by two mathematical functions, one defining the axis the other the cross sections. In a fixed Cartesian coordinate system x,y,z the axis may be represented parametrically as a function of arc length s:

$$r(s) = (x(s), y(s), z(s)) \tag{2.2}$$

If adequate in a given situation, this general sweep trajectory can be simplified to a simple translation or rotation of the cross-section element. This is known as translational and rotational sweeping, respectively, and the scheme reduces to a mathematical representation of the cross-sections. These are usually defined by the bounding curve in the local coordinate system with its origin on the curve *r(s)*. The cross-sectional element is usually defined to be in the plane perpendicular to the unit tangent of the sweep trajectory. The bounding curve may be expressed as a two parameter family:

$$\text{cross section boundary} = (x(r,s), y(r,s)) \tag{2.3}$$

where the dependence on s reflects the fact that the cross-section shape may vary along a GC curve. Frequently, if strict surface accuracy is not necessary, the cross-sections are constrained to circular shapes. Such a simplification is also used when fitting a GC to real data. While the GC representation gives simple descriptions for volumes of many biological and manufactured objects, it is not well suited to descriptions of non-elongated objects which have no preferred axis, or objects of arbitrarily deformed surfaces.

A third and final category, also intuitively very convincing consists of boundary surface representations. It is clear that the enclosing surface of a well-behaved 3D object unambiguously specifies the object. Since surfaces are what we see, these representations are important to computer vision, image analysis and recognition. Some of the more sophisticated surface representations are also fundamental to geometric modeling and computer graphics, as in the design of ship, airplane and automobile bodies, where computer graphics displays and software, are used to help visualize prototype shape and changes to existing designs. Similarly as in the volumetric representations, the enclosing surface of a solid is represented by primitive entities. These could be unbounded mathematical surfaces, curves

and points, which together could be used to define "faces" also known as "patches". In general then, a boundary representation will consist of a list of faces each of which is a mathematical surface together with information about its own connection to other faces (continuity, relative position, etc.). The data structure for this representation is a directed graph containing object, face, edge, and vertex nodes plus adjacency relations between them. It is assumed that each boundary representation scheme has a finite number of generic primitive surfaces. Similarly to the diversity of 2D curve approximation methods, where straight line segments (polygonal approximation), finite degree polynomials and splines are used, the individual patches can be either planar, straight edged (bilinear) or arbitrarily curved, but usually four sided. The computational simplicity of planar patches makes them, on occasion, useful as an approximate representation of non-polyhedral objects. However, when accuracy and boundary smoothness is desired, as in ship and aircraft building, curved patches must be used.

A 3D surface can be written as $F(x,y,z) = 0$, where x,y,z represent the Cartesian projections. This is known as the implied representation of a surface [19,25,49]. If the gradient of F exists, and is continuous and non-zero then it is a smooth surface. For a low-order surface, such as a plane or a quadratic surface, it can be characterized and classified by a few classical parameters. For example, quadratic (second order) surfaces include such familiar objects as spheres, cylinders, cones, ellipsoids, paraboloids and hyperboloids. They can be represented by the general quadratic equation:

$$F(x,y,z) = ax^2 + by^2 + cz^2 + 2hxy + 2gzx + 2fyz + 2ux + 2vy + 2wz + d \qquad (2.4)$$

Only three coefficients are needed to describe the shape of a quadratic surface of a given type and six parameters locate and orient the surface in space. If a quadratic surface is properly translated and rotated at least six of the ten coefficients will be zero. All quadratic surfaces can be classified as one of the following

six types using the three or four non-zero coefficients in that particular coordinate system: ellipsoid (a>0, b>0, c>0, d=-1), elliptic paraboloid (a>0, b>0, w=-1), hyperbolic paraboloid (a>0, b<0 w=-1), hyperboloid of one sheet (a>0, b>0, c<0, d=-1), hyperboloid of two sheets (a>0, b<0, c<0, d=-1), quadratic cone (a>0, b>0, c<0). Unfortunately, the implicit representation is not generally useful for an arbitrary surface because as the order of the polynomial increases, the polynomial surface function is difficult to deal with. As a further difficulty the implicit surface equation provides no simple, direct means of determining whether a geometric entity is bounded and if so, what are its extreme coordinates. A standard alternative approach is to use an explicit parametric surface representation of the form: $x=h(u,v)$, $y=g(u,v)$, $z=f(u,v)$ where h,g,f are smooth, bivariate functions of the parameters u,v, defined over some closed interval, usually $u,v \in [0,1]$, making the parameter domain rectangular. The form of the functional relationship between coordinates x,y,z and the parameters u,v can be quite general. Mostly it is a tensor product (cross or Cartesian) of finite degree polynomials, rational polynomials, polynomial splines or rational polynomial splines [25,57]. Such polynomial representations are comparatively simple and are easily stored as ordered sequences of coefficients in a suitable polynomial basis. From the computational viewpoint, finite-degree polynomials are in many respects, ideal forms for representing and approximating functions. They are easily and rapidly evaluated, can be differentiated and integrated, arithmetic operations of addition, subtraction, multiplication and division are all well defined. Parametric representations are extremely versatile in defining complex geometries and unambiguous in their interpretation. They provide a natural means of digitizing and computing differential properties of geometric items. The differences in the representations are the various choices for the functions f,g, and h or more specifically the type of basis used in the polynomial formulation. The well known and frequently referenced Coons'

representation is expressed in the Hermite basis [25,28]. The parametrized surface patch, $S(u,v) = (x(u,v), y(u,v), z(u,v))$ is a tensor product of the following univariate, cubic Hermite functions:

$$
\begin{aligned}
b_0(u) &= 2u^3 - 3u^2 + 1 \\
b_1(u) &= -2u^3 + 3u^2 \\
b_2(u) &= u^3 - 2u^2 + u \\
b_3(u) &= u^3 - u^2
\end{aligned}
\tag{2.5}
$$

These functions are also referred to as the blending functions, since they "blend" the effect of the various boundary conditions that uniquely define the surface patch. The boundary conditions, known as control points are specified in a square matrix which multiplies the tensor product of the basis functions. For a given surface patch all elements of the matrix are constants and can be classified in the following way:

$$
[Q] = \begin{bmatrix} \text{corner} & \text{v-tangent} \\ \text{coordinates} & \text{vectors} \\ \text{u-tangent} & \text{twist} \\ \text{vectors} & \text{vectors} \end{bmatrix}
\tag{2.6}
$$

The shape of the Coons' patch is controlled by the contents of this 4x4 matrix. The form of the bicubic Coons' surface patch can now be written as follows:

$$
S(u,v) = B(u)[Q]B(v)^t
\tag{2.7}
$$

where $B(u) = (b_0, b_1, b_2, b_3)$ and t stands for transpose. Although such a bicubic surface patch has been used to design surfaces this formulation exhibits several disadvantages. There are three different quantities, generally of different orders of magnitude, which must be manipulated to obtain the desired surface shape. The process of changing the elements of the control matrix is very confusing since typically there is a lack of intuitive feel for how these elements affect the shape

of the patch. In order to eliminate these problems to some degree, a simplified version of the bicubic surface patch with zero twist vectors has been formulated [27,57]. On occasion this leads to surfaces which are not sufficiently smooth for some applications. Other types of surfaces currently in use are: Bezier [13], B-Spline [16,21], Rational B-Spline [65], and Beta-Spline [8]. Such surface representations, with stitched together patches have proven extremely useful in computer graphics and modeling, they are appropriate for surface synthesis, they however, are not easy to use for analysis. This is due to the difficulty of registering the patch edges with image data. A given surface will admit to many patch decompositions, as is the case with volume decomposition schemes.

2.3 3D Data Representation Schemes in Image Analysis.

In analysis applications it is of interest to convert image data into a mathematical, or other form of representation suitable for subsequent recognition processes. Here, the most typical methods of obtaining a mathematical description of a 3D object are discussed. The review reflects the fundamental dichotomy in image analysis between local and global descriptions. In the former objects are represented as compounds of primitive surface elements, in the latter the solid's whole surface is a single entity.

2.3.1 Segmented representations.

A frequently used recognition process consists of the following steps: first the 3D coordinates of the surface points are calculated by triangulation or obtained directly from the appropriate ranging technique [38], they are then grouped in small surface elements (usually planar), which are later merged into homogeneous regions. These regions are classified as belonging to some elementary surface type (planar, curved, etc.). Alternatively they are fitted with second order (quadratic) surfaces, which in the case of large, smooth areas leads

to a compact surface description in terms of the coefficients of equation (2.4). Finally, recognition is achieved by matching the generated description with the descriptions available in a library of prototype object models [14,15,26,51]. Clearly some method for matching the model data to the sensor data must be available. The higher the degree of invariance of the representation to perturbations and the more compact its form, the easier is the matching task.

Another possible scheme represents the visible surface section by topologically invariant features. There are significant advantages of using such features. Any recognition process that employs a surface model that is not invariant to basic geometric transformations such as rotation, translation, and scaling must treat each individual view of an object as distinct. This requires a potentially large number of stored descriptions in exchange for a reduction in the magnitude and complexity of the computations that would otherwise be necessary to compensate for the effects of the various transformations. The alternative to relying on an exhaustive enumeration of all possible appearances is to use a topologically invariant set of surface features. These can be calculated directly from the depth map [10,11]. The technique is both motivated and justified by well known facts from mathematics of differential geometry. Two useful local surface characterizations are the Gaussian (second) curvature and mean (first) curvature. Both of these surface characteristics are invariant to arbitrary changes in surface parametrization and to arbitrary translations and rotations of object surfaces. They are both local surface properties. This allows surface curvature to be used in situations where occlusion is a problem. Other possibilities include surface critical points (local maxima, minima, and saddle points) and large metric determinant points (depth map discontinuities), to name a few. They can all be combined to yield a rich and redundant surface description for purposes of view independent object recognition from noisy range data. The recognition is achieved by comparing depth map surface region characteristics with pre-computed model object region characteristics.

An interesting variant is the Extended Gaussian Image (EGI), which has been used for object attitude determination and recognition from depth maps [33,34]. Objects are represented by their surface normal orientation histograms, parametrized according to the desired tessellation of the Gaussian sphere. Given a pre-computed library of prototype histograms, classification is achieved by comparing this to the unknown objects orientation histogram of its visible side. The best match determines which object is represented by the input data and how it is oriented in space.

Real data has also been fitted with GCs in order to produce descriptions of biological forms. The process involves a step in which the GC axes are inferred from the original 3D data. Subsequently circular cross sections were iteratively fitted to the data to produce a GC description of the viewed object. In this scheme a major difficulty arises when an important image axis is obscured, either because it is foreshortened or is hidden behind another part of the shape [44,45]. One possible solution would require multiple descriptions of a shape, including likely degenerate views. A similar scheme used spline based axis and cross sections in a GC representation of the organs of the human abdomen [62].

2.3.2 Global representations.

In an alternative approach, an attempt is made to describe objects analytically by various functions of the boundary, defined over the entire object, as opposed to an individual surface patch or structural element. In earlier 2D image analysis, analytic shape descriptors have received a significant amount of attention, with descriptors in terms of moments and coefficients of a Fourier expansion of a function of the boundary being the most popular. Various FDs can be generated from a number of different analytic representations of the boundary curve. Zahn and Roskies [71] assume that the boundary forms a

clockwise oriented simple closed curve with a parametric representation $(x(s), y(s)) = z(s)$, where s is an arc length parameter, $0 \le s \le L$ and L is the total length of the closed boundary. Let $\phi(s)$ denote the angular direction and $\theta(s)$ the cumulative (net) angular bend of the boundary as measured from the starting point on the curve to the running arc length s. This is recognized as the trace of the tangent indicatrix (see Sect. 3.2.2). It is desirable to normalize the arc length to a value such that the total normalized arc length is 2π . Using this normalized arc length parameter t, an intrinsic shape function $f(t)$ is defined as

$$f(t) = \theta(t) + t, \qquad 0 < t < 2\pi \tag{2.8}$$

This function is invariant to both geometrical translation and scaling. This pattern vector is an intrinsic shape function since it does not depend on how the underlying curve is embedded in the 2D space of the image plane. Rotation causes a linear shift in the shape function. The linear term added to the normalized net angular bend function insures that for a circle the intrinsic shape function is identically zero. This function is therefore a measure of the deviation of a curve from the circle. Since $f(t)$ is periodic, it may be expanded in a Fourier series, in polar form, as

$$f(t) = A_0 + \sum_{k=1}^{+\infty} A_k \cos(kt - \alpha_k) \tag{2.9}$$

The Fourier coefficients (A_k, α_k) are the FDs of the curve. The zero term represents the starting position on the curve and is therefore suppressed. Since the amplitude coefficients A_k do not depend on the linear translation of $f(t)$ and are therefore independent of translation, rotation, scaling and starting position on the curve, thus the A_k's, for k not equal to zero, are true shape descriptors. For most types of curves the high harmonics rapidly vanish leading

to efficient data compression of the boundary representation. While knowledge of the magnitude is sufficient for classification, boundary reconstruction requires information about both amplitude and phase.

In yet another representation, only the magnitude of the complex $z(s)$, the so called radius vector $r(s)$ is used [40]. The parameter need not be arc length. The radius vector is measured from the centroid of the planar figure to a point on the contour. The required discrete function values are the equi-spaced samples of the radius vector $\{r(1), r(2),....r(N)\}$ of points located on the curve. The accuracy of the representation depends on the sampling parameter N. The centroid calculation and the subsequent translation provide for the translation invariant characteristic of the representation. Other possible arc length parametrizations of the boundary curve such as the local tangent angle, the relative angular bend [43], the curvature or the radius of curvature [7], and the polar angle (see sec. 3.1) are also possible.

In 3D image analysis a number of methods are used, which are extensions of earlier 2D techniques. The radial representation is a simple and easily visualized example. The distance (radius) from a viewing point to a boundary point is a function of the direction from the point to the surface. It is natural to describe the direction in space in terms of position on a unit sphere centered about the viewing point, with the radius to the surface given by a continuous function on that sphere. By choosing a polar coordinate system on the sphere, the tessellation yields two parameters frequently referred to as the elevation and azimuth. In Ballard and Schudy [60], to represent an arbitrary closed surface, this spherical radius was subsequently described as a linear sum of spherical harmonics,

$$r(\theta,\phi)=\sum_{m=0}^{M}\sum_{n=0}^{m} A_{m,n}u_{m,n}(\theta,\phi)+B_{m,n}v_{m,n}(\theta,\phi) \qquad (2.10)$$

where the spherical harmonic functions $u_{m,n}(\theta,\phi)$ and $v_{m,n}(\theta,\phi)$ are defined in polar coordinates by:

$$u_{m,n}(\theta,\phi) = \cos(n\theta)\sin^n(\phi)P(m,n,\cos(\phi)) \tag{2.11}$$

$$v_{m,n}(\theta,\phi) = \sin(n\theta)\sin^n(\phi)P(m,n,\cos(\phi)) \tag{2.12}$$

with $m = 0,1,\ldots,M$ and $n = 0,1,\ldots,m$. Also $P(m,n,\cos(\phi))$ is the nth derivative of the mth Legendre polynomial as a function of ϕ. The real constants ($A_{m,n}$, $B_{m,n}$) represent a quality of the closed surface. Similarly to the 2D FDs, the low order zonal harmonic coefficients represent gross shape characteristics. This method has been used to approximate heart volumes from ultrasound data. Using only five zonal harmonic coefficients good results have been obtained.

3D moments are yet another analytic description of an object [59]. The second central moments measure moments of inertia of a closed surface. The eigenvectors of the matrix of second central moments give the directions about which a surface has maximum and minimum moments of inertia. Ratios of the eigenvalues describe fatness and thinness in different directions. Given this kind of information it is possible to discriminate between a parallelepiped, pyramid and cylinder in continuous space.

2.4 Summary.

A multitude of representational schemes found in computer vision and related fields of computer graphics and geometric modeling have been presented. The distinction between synthesis (computer graphics, solid modeling) and analysis (3D image processing, machine vision) methods has been emphasized, stressing their relative advantages, disadvantages and incompatibilities. It is apparent that there are no general-purpose representation schemes. For this reason characteristics of any individual representation must be carefully investigated as to storage requirements, ease of manipulation, and inherent restrictions. The applicability of a representation scheme to the problem of pattern analysis and recognition must be carefully considered in view of these characteristics since it is critically important to correctly match a representation scheme with its destined application area.

3 Shape Description.

In this section, global analytic boundary representations with desirable invariance properties will be discussed. Methods for obtaining functions of the bounding curves and surfaces of three-dimensional (3D) solids will be described. Shape descriptors will be calculated and classification results presented.

Topologically invariant representations are extremely useful in pattern matching algorithms since they significantly decrease the number of comparisons, calculations and model prototypes. Scaled, rotated or translated versions of an image result in identical representations. There is no need in such cases to enumerate in memory topological variants of the same object shape. This results in significant savings in storage requirements and computational complexity. A method of tackling the invariance problem is to control the coordinate system in which the object is embedded. Fundamentally there are two types of coordinate systems of interest to us. If locations are specified relative to the viewer, we talk of a viewer-centered representation. If however, locations are specified in a coordinate system defined by the viewed object, the representation uses an object-centered coordinate system. For recognition tasks, viewer-centered descriptions are easier to produce but harder to use than object-centered ones, because they depend on the vantage point from which they are built. Object-centered descriptions are more difficult to derive since the objects coordinate system must be derived from the image before further processing can take place. If this is done however, canonical descriptions are available by normalization of features. In the absence of an object-centered coordinate system the image must be interrogated for intrinsic features, such that do not depend on how the object is embedded in the coordinate system.

It is intuitively clear that surface geometry uniquely determines a homogeneous solid. Surface characteristics should therefore be useful in shape dominated pattern recognition. Both local and global analytic properties of curves and surfaces will be discussed. Local properties describe the local behaviour of curves and surfaces only.

They vary from point to point and are only computed at specific points. For curves they include curvature, torsion, the principal vectors (tangent, normal, and binormal), and the principal planes (normal, osculating, and rectifying). The local properties of a surface include the normal vector, tangent plane, Gaussian, principle, and mean curvatures, to name a few. Global properties are those that depend on the overall characteristics of a geometric element. The smallest local changes of curves or surfaces affect the global characteristics. Universally familiar global properties are arc length, surface area and volume. A well known global function of a planar boundary curve is the cumulative slope function, which measures the total rotation of the tangent vector as one moves along the boundary. This is the trace of the so-called tangent indicatrix [19]. The value of the function at any one point depends on the values at all the preceding points of the curve. This particular function has the additional property of being an intrinsic function of the boundary, since it does not depend on how the curve is embedded in the two dimensional space of a plane. There are a number of surface properties which are invariant under many of the following conditions: changes in the viewing position, object translation, rotation, scaling, and surface parametrization. A 3D representation model with all the above characteristics would be an invaluable tool in image description and recognition. It is the purpose of this section to describe new boundary functions for closed surfaces, which exhibit the properties listed above. New analytic surface representations based on the solid angle and Gaussian curvature will be discussed. A new 3D shape descriptor will be presented and results of coding and classification of will be shown. First, however, a new intrinsic curve function, the polar angle, which is the 2D equivalent of the solid angle, is presented.

3.1 Planar Shape Functions.

The new intrinsic boundary function, the polar angle, has a simple geometrical interpretation, as a projection of units of arc length of the boundary onto a unit

circle. The appropriate ratio of length defines the polar angle and hence its name. Alternatively, the polar coordinate angle is $p = \arctan\left(\frac{y}{x}\right)$, so the differential change dp is

$$dp = d\left(\arctan\left(\frac{y}{x}\right)\right)$$
$$= \frac{-ydx + xdy}{x^2 + y^2} \tag{3.1}$$

The net change in polar angle as a function of arc length, suitably normalized, is an intrinsic measure of the shape of the 2D boundary. Using the normalized arc length t, as defined in Eq. (2.8), we define the net increase in polar angle, measured from some arbitrary starting position, as

$$p(t) = \int_0^t \frac{-ydx + xdy}{x^2 + y^2} \tag{3.2}$$

where $p(t)$ varies continuously on a path C, with the curve centered on the origin. For any closed path not through the origin $p(t) = 2\pi$, therefore it is independent of scale changes in the image. The intrinsic function is defined as in Eq. (2.8) with $p(t)$ replacing $\theta(t)$.

$$f(t) = p(t) - t. \qquad 0 < t < 2\pi \tag{3.3}$$

Again, for a circle, the intrinsic function is identically equal to zero, since for a circle $p(t) = t$. The coefficients of the Fourier series coefficients of the expansion of $f(t)$ in Eq. (3.3), are the new FDs.

$$f(t) = A_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k) \tag{3.4}$$

All coefficients, except for A_0, are invariant to rotation, scaling, translation, and choice of starting point. In Fig. 3.1 the new intrinsic function for closed, planar curves depicting handwritten Roman letters is plotted. Its smoothness contrasts

with the sawtooth nature of the cumulative angular function. The absence of discontinuities in the definition of the polar angle results in a reduced spectral bandwidth. In Fig. 3.2, the amplitude Fourier spectrum of one of the functions in Fig. 3.1 is shown. Only very low spectral components have significant amplitudes. The new FDs are very effective data compressors. Gross shape characteristics are captured by a few low order FDs. These capture the "lobedness" of the boundary. The higher order FDs representing high spatial frequency variation of the boundary capture the detail in the boundary. For most classification purposes, only the gross features are important and therefore only a few FDs are needed. For polygonal approximations to the letters of the Roman alphabet, 90% of the total signal energy was confined to the first two or three spectral components. This compares very favorably with the cumulative slope FDs, where the number of components ranged from a low of 4 to a high of 27. A total of 27 handwritten Roman letters and numerals were used to collect this data. A comparison of normalized amplitude spectra, of the new FDs and those defined by Eq. (2.9), for a single curve, indicates a significant data compression improvement for the new descriptors (see Fig. 3.3).

3.2 Surface Shape Functions.

The explicit parametric form of a surface S is expressed by a triplet of functional relationships between the Cartesian coordinates x,y,z and the parameters u,v:

$$X = \{(x,y,z) \ni x = x(u,v), y = y(u,v), z = z(u,v)\} \tag{3.5}$$

As previously described this relationship can be quite general, and is mostly unobtainable for an arbitrary surface, unless it is made up of well behaved patches. Certain forms of representing image data result in a natural parametrization, i.e. depth maps. In situations when the complete surface geometry is known a global parametrization can be attempted. There is a class of surfaces for which there is a natural choice of parameters. For closed, convex surfaces the pair of direction

coordinates on the unit sphere, the azimuth u and the elevation v, are a natural choice. The entire surface can now be represented by the three individual functions from Eq. 3.5, defined over: $0 \leq u \leq 2\pi, 0 \leq v \leq \pi$. These functions are numerically obtained by the following process. For a given point u_0, v_0 on the unit sphere, a ray emanating from the origin and passing through this point is assumed to intersect the enclosing surface at only one point. The Cartesian coordinates of this intersection are the values of the respective functions: $x(u_0, v_0)$, $y(u_0, v_0)$, $z(u_0, v_0)$. If this process is repeated for all the discreet values of the parameters u, v we obtain an approximate, sampled version of the functional relationships defined by Eq. 3.5 [23]. As an example Fig. 3.4 depicts the $x(u,v)$ function sampled at 64 x 64 points. Given spherical parameters, the three coordinate functions are uniquely defined by the single bivariate radius function $r(u,v)$. In Fig. 3.5, the radial representation of a polyhedral, convex solid is shown, both as a surface plot and an intensity image.

A global parametrization allows the calculation of bivariate surface functions, which in some sense characterize it, over the entire object. They can therefore be used for purposes of effective shape discrimination. The solid angle, area density (inverse solid angle), cumulative curvature can all be sources of useful shape descriptors. These global characterizations can be enhanced for partial view recognition by quantities with only local influence, such as curvature and differential area. Both convey a significant amount of information about an object. For polyhedral solids peaks indicate vertices, ridges signify straight edges, where two faces meet.

3.2.1 Solid Angle.

The solid angle is the 3D analog of the polar angle. In its simplest form it is a projection of unit surface areas on a unit sphere. For a formal definition consider a plane surface patch S with an outward unit surface normal n. The

solid angle G of S with respect to a point O not on S is defined as (see Fig. 3.6)

$$G(O,S) = \pm \text{projection area of } S \text{ on } S_1 \tag{3.6}$$

where S_1 is a unit radius sphere centered on the point O. The sign of the solid angle is determined by the sign of the outward normal of the surface patch S. In terms of an analytic formula, the solid angle is defined by the following surface integral:

$$G(O,S) = \int\int_S \frac{rn}{|r|^3} ds \tag{3.7}$$

where r is the vector from the origin O to the incremental surface element ds and where $G(O,S)$ varies continuously on S. The vector n is the unit normal. The argument under the double integral sign is the incremental solid angle. It can be shown that when the origin is inside the volume and if S forms the boundary of a bound, closed, simply connected region R, then the total solid angle is $+4\pi$ [39]. Clearly the sum of all nonoverlapping projections onto the unit sphere, of a closed surface will add up to the total area of the sphere, which is 4π.

While there is a unique, ordered traversal of a planar curve, there are infinitely many traversals of a general closed surface. To simplify the traversal convention, the spherical coordinates of the surface S, denoted as (u, v), where $0 \le u \le 2\pi$ and $0 \le v \le \pi$ will be used. The net reciprocal solid angle $S(u_0, v_0)$ is now defined as equal to the area of the surface that projects to an elemental solid angle on the sphere, such that $0 \le u \le u_0$ and $0 \le v \le v_0$. It is necessary that the solid angles be equal, or equivalently, the sphere must be tessellated into equal elementary areas, $ds = d\xi \cdot d\eta$. However, using spherical coordinates u and v we get

$$ds = \sqrt{EG - F^2}\,du\,dv$$
$$= \sin(v)\,du\,dv \tag{3.8}$$

where E,G and F are coefficients of the first fundamental form. The required reparametrization is

$$d\xi = du, \qquad d\eta = \sin(v)\,dv$$
$$\text{so,} \quad \xi = u, \qquad \eta = -\cos(v) \tag{3.9}$$

The new 3D intrinsic function is now a function on the sphere of the form:

$$f(\xi,\eta) = \frac{4\pi}{S^T}\ S(\xi,\eta) - \xi\eta \tag{3.10}$$

where $0 \le \xi \le 2\pi$ and $-1 \le \eta \le 1$ and S^T is the total area of the surface S. The normalization conditions, in analogy to the 2D case, insure that f is size invariant and identically vanishes for a sphere. This intrinsic surface function can now be represented by a 2D Fourier series expansion

$$f(u,v) = \sum_m \sum_n A_{m,n}\cos(m\xi - \alpha_m)\cos(n\eta - \beta_n) \tag{3.11}$$

where the coefficients $A_{m,n}$ are the new 3D FDs. These however do not exhibit a rotational invariance to a general 3D rotation, since such a rotation does not naturally decompose into two linear shifts in the u and v directions. In order to achieve rotational invariance they must be restricted to rotations around the North-South pole axis. This can be achieved by fixing the major axis of an object along the Cartesian z-direction. In Fig. 3.7 (top), the intrinsic surface function for a square based parallelepiped perpendicular to the x-y plane is shown. The result for an ellipsoid with parametric equation,

$$r(u,v) = \sin(v)\cos(u)\vec{i} + \sin(v)\cos(v)\vec{j} + 3\cos(v)\vec{k} \tag{3.12}$$

is shown in Fig. 3.7 (bottom). The amplitude spectrum $A_{m,n}$ of the parallelepiped solid body is shown in Fig. 3.8, where the low pass characteristic of the spectrum is very apparent.

3.2.2 Cumulative curvature.

It is clear from the theory of differential geometry of curves [16,50,69], an investigation of Frenets' equations, that the cumulative slope function is a normalized integral of the curvature function parametrized with respect to arc length. Let $\theta(s)$, be the angle the tangent $t(s)$ makes with the x-axis, that is, $x'(s) = \cos\theta$, $y'(s) = \sin\theta$, and s is arc length. Since $t(s) = (x'(s), y'(s))$,

$$\begin{aligned} \frac{dt}{ds} &= \frac{d}{ds}(\cos\theta, \sin\theta) \\ &= \theta'(-\sin\theta, \cos\theta) \\ &= \theta' n \end{aligned} \tag{3.13}$$

where n is the normal. Also, from Frenet's equations: $\frac{dt}{ds} = kn$. This means that $\theta'(s) = k(s)$ and suggests a global function of a planar curve by

$$\theta(s) = \int_0^s k(s)ds \tag{3.14}$$

This global function agrees with the cumulative slope function reviewed in Sect. 2.3.2.

In a similar fashion, the bivariate (two-parameter) curvature function of a surface will be integrated and normalized to yield a global function of the boundary. Gaussian curvature is defined as:

$$k(u,v) = \frac{eg - f^2}{EG - F^2} \tag{3.15}$$

where E, F, G and e, f, g are scalar functions of u, v and are the coefficients of the first and second fundamental forms, respectively. In Fig. 3.9, the parametrized surface curvature for a polyhedral solid is shown. Gaussian curvature is a local surface property that is intrinsic to the boundary. It indicates the surface shape at individual points. When $k(u,v) > 0$ at a point on the surface S, then the surface is shaped like an ellipsoid in the neighborhood of that point. When $k(u,v) < 0$, the surface is saddle shaped. When $k(u,v) = 0$, the surface is flat, cylindrical or conical in shape. The cumulative curvature global surface function is defined by the following double integral:

$$\theta(u,v) = \int_0^u \int_0^v k(\xi,\eta)\,d\xi\,d\eta \tag{3.16}$$

where $0 \le u \le 2\pi$, $0 \le v \le \pi$. The total curvature C is, $C = \theta(2\pi,\pi)$. This double integral can be numerically evaluated yielding a global surface function of a 3D solid object (Fig. 3.10). As a normalization step, this function is scaled with respect to total curvature C. The desired intrinsic surface function $f(u,v)$ is,

$$f(u,v) = \frac{2\pi^2}{C}\theta(u,v) - uv \tag{3.17}$$

It is identically reduced to zero for spherical objects and can be interpreted as a normalized curvature density function. Fig. 3.11 depicts $f(u,v)$ for a polyhedral solid. Discontinuities in the function f indicate abrupt changes in the orientation of the faces of a solid. The intrinsic function $f(u,v)$ can now be represented, in analogy to FD's of 2D boundary curves, by a Fourier series expansion

$$f(u,v) = \sum_m \sum_n A_{m,n}\cos(mu - \alpha_m)\cos(nv - \beta_n) \tag{3.18}$$

The 16x16 low-order spectral components A_{ij}, are shown in Fig. 3.12.

3.3 Rotational Invariance.

Topologically invariant surface representations are invariant to translation, rotation, scaling of the solid object and the choice of the surface starting point. Such shape invariant descriptions lead to efficient matching algorithms and savings in storage requirements. None of the 3D representations developed so far have all of these properties. They lack rotational invariance with respect to arbitrary rotations in space around the origin (three degrees of freedom). This severely complicates the recognition task. This problem was solved by a normalization of the radial representation of a solid. A primary means of achieving such a normalization is the extraction of the object's "significant axis," followed by a rotation step which aligns it with the fixed z-axis. This results in radial representations of solid objects that differ solely in their azimuthal position. It has already been shown earlier that variations due to rotations around the z-axis are eliminated by the subsequent Fourier transformation. A set of descriptors that is invariant to scaling, translation and rotation, all in a 3D coordinate system, has been obtained. An unanswered question concerns the choice of a "significant" axis. There is clearly a number of possibilities such as, maximum or minimum momentum, a least squares, straight line fit to the surface points, finally axes of maximum or minimum length. For reasons of computational simplicity the longest axis of the object is chosen for normalization purposes. The longest axis is found by calculating the maximum radial distance between surface points for all the spherical directions. This is followed by two rotations which align the axis with the z-axis of the Cartesian coordinate system.

3.4 Classification.

In this section, classification results for the cumulative curvature based 3D shape descriptors are presented. The data set for these experiments consisted of

polyhedral and cylindrical solids. It will be shown that the shape descriptors introduced in Sect. 3.3.2 are very effective in discriminating between objects that differ in their gross shape characteristics.

The solid objects were modeled using the IDEAS solid modeling software package available on City College VM computing environment. Finite element analysis of the computer synthesized solids provided the surface mesh data which was subsequently used as inputs to the algorithm. Altogether, seventeen objects were modeled according to the following groupings: rectangularly faced polyhedrons (5), pyramidal polyhedrons (7), and cylindrical/conical solids (5). The numbers in brackets denote the number of objects in each category. The three categories of objects differ with respect to major surface shape features such as "roundness", number and relative location of edges and vertices, and the shape of individual faces. In Fig. 3.13, examples of some of the modeled polyhedral and cylindrical objects are shown. As can be seen, to obtain the various objects, relatively simple solids were modified by using Boolean geometric cuts to obtain the variety necessary for meaningful classification simulations. As described earlier, the surface data was radialized, normalized and Fourier transformed to yield the shape descriptors, as in Eq. 3.3. The shape descriptors were based on the low-order coefficients of the amplitude spectrum. In the classification experiments the smallest possible number of descriptors that gave correct results was used. Out of a total of 32x32 spectral components results presented below are based on the 3x3 lowest order coefficients: $\{A_{ij},\ i=0,1,2,\ j=0,1,2\}$. Classification was based on a hierarchical cluster analysis of the distribution of the seventeen observations, with each observation an 8 element vector (A_{00} not used) representing one solid [4]. In the first step the distances (dissimilarities) between all pairs of observations are calculated. Two different methods were used to calculate the dissimilarities:

Euclidean distance (l_2 norm), $\sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}$

angle in radians ϕ, where $\cos(\phi) = \frac{x \cdot y}{|x||y|}$

and x,y denote any pair of observations. This results in an upper diagonal distance matrix. Each observation is initially considered a single point cluster. The two clusters which are closest to one another in distance are merged, and the distance of the new cluster to all the remaining clusters is computed. This continues until only one cluster remains. Ward's method is used to update the distance matrix. Clusters are formed so as to minimize the increase in the within-cluster sums of squares. The distance between two clusters is the increase in these sums of squares if the two clusters were merged. This method tends to yield spherical clusters.

In Table 3.1 the distance matrix for the l_2 norm is shown. It can be seen that one of the pyramidal solids is very distant from all the other solids and will probably not be classified as belonging to any one of the three anticipated clusters. In Table 3.2 the angles between the observation vectors are shown. To aid in the visualization of the clustering results Figs. 3.14, 3.15 show the cluster trees that resulted from the analysis. From Fig. 3.14 it is easy to see that object no. 7 is indeed misclassified, with all the other objects forming expected groupings. In all probability this wedge shaped solid was sufficiently different from the other objects to cause a fourth grouping (class) to appear. In clustering based on the angle measure, again all but one object are correctly grouped. This time object no. 17, which is of conical shape was misclassified as a pyramid (see Fig. 3.15).

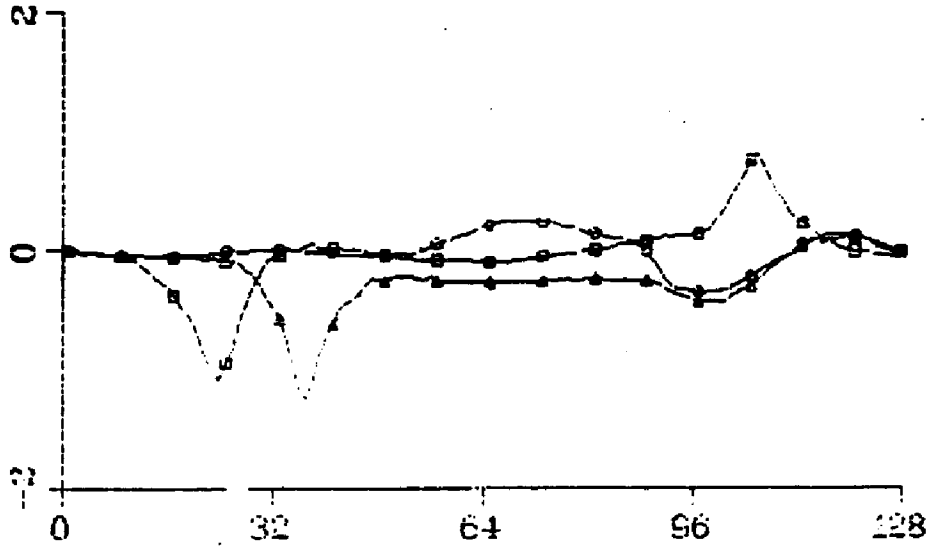


Figure 3.1 Polar angle boundary functions of three handwritten characters.

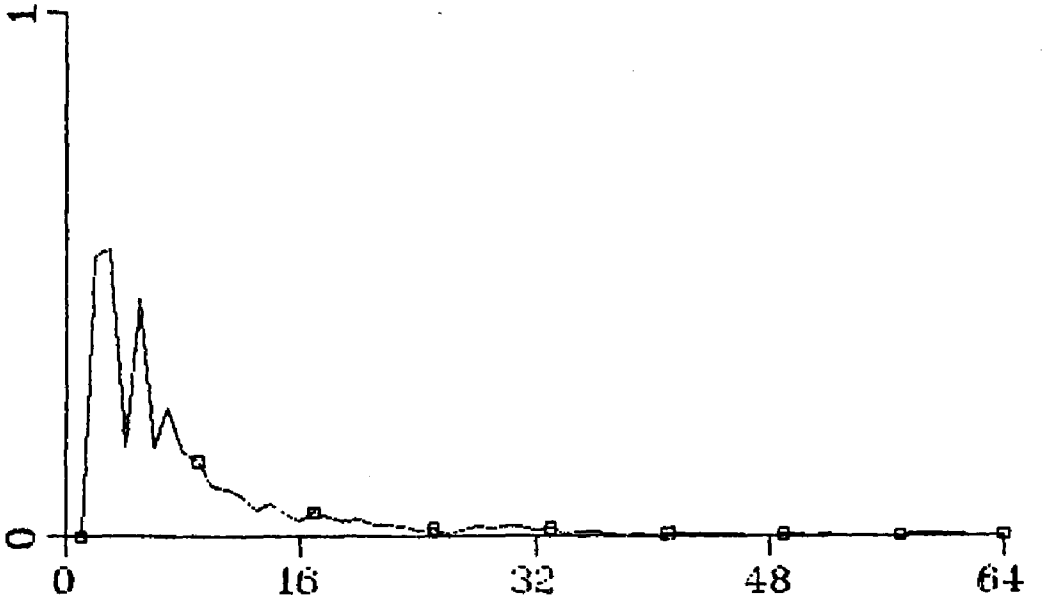


Figure 3.2 Fourier coefficients of a boundary function in Fig. 3.1.

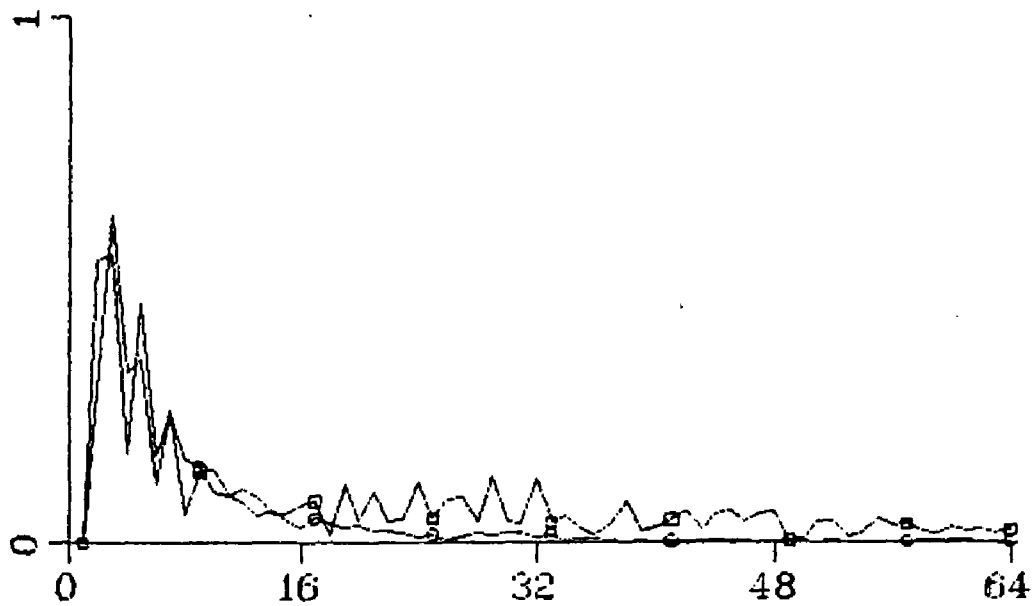


Figure 3.3 Comparison of Fourier spectra: cumulative slope (squares), polar angle (circles).

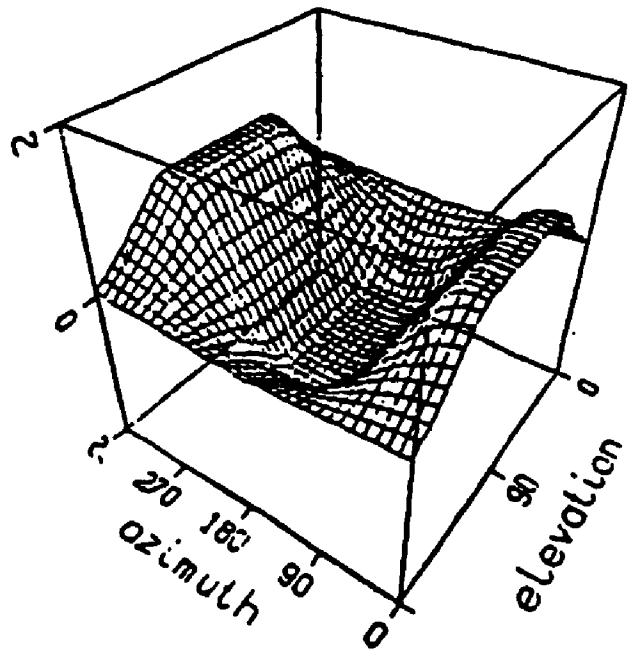
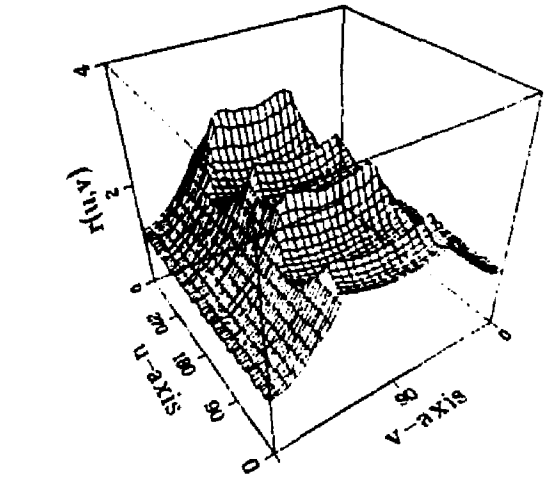


Figure 3.4 Spherical parametrization of the x coordinate of a polyhedral solid.



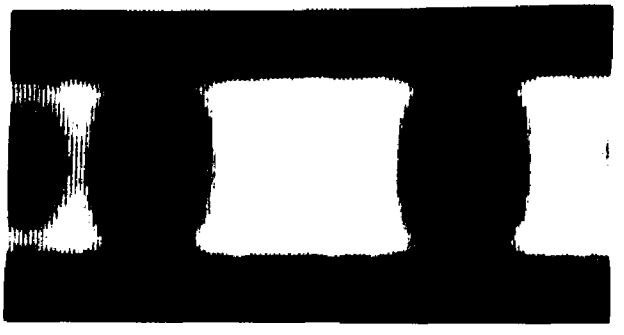


Figure 3.5 Radial representation of a polyhedral solid, surface plot (top), intensity image (bottom).

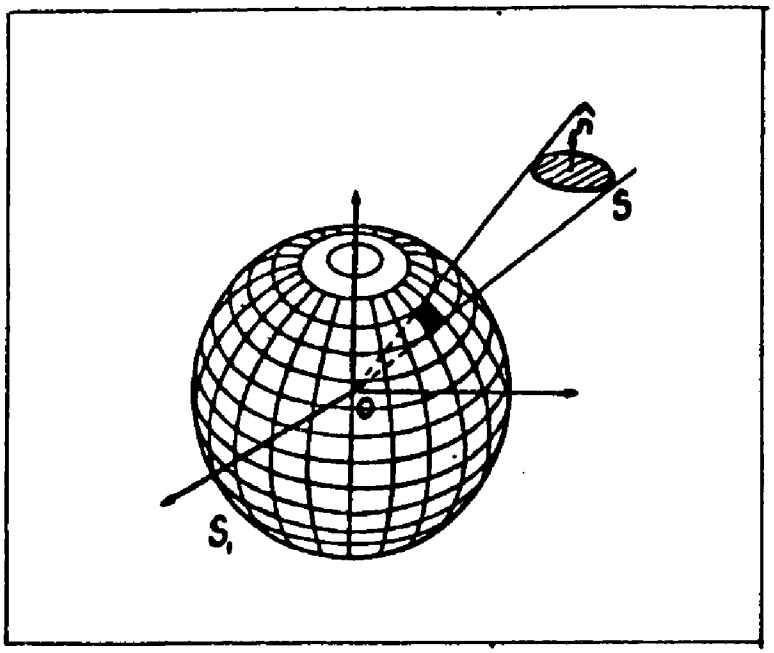


Figure 3.6 Solid angle.

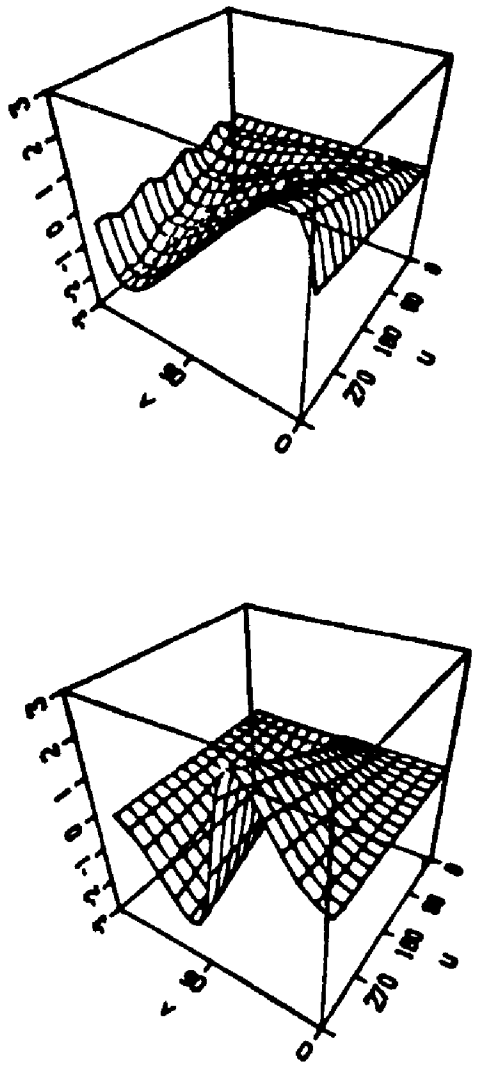


Figure 3.7 Inverse solid angle, square based polyhedral solid (top), ellipsoid (bottom).

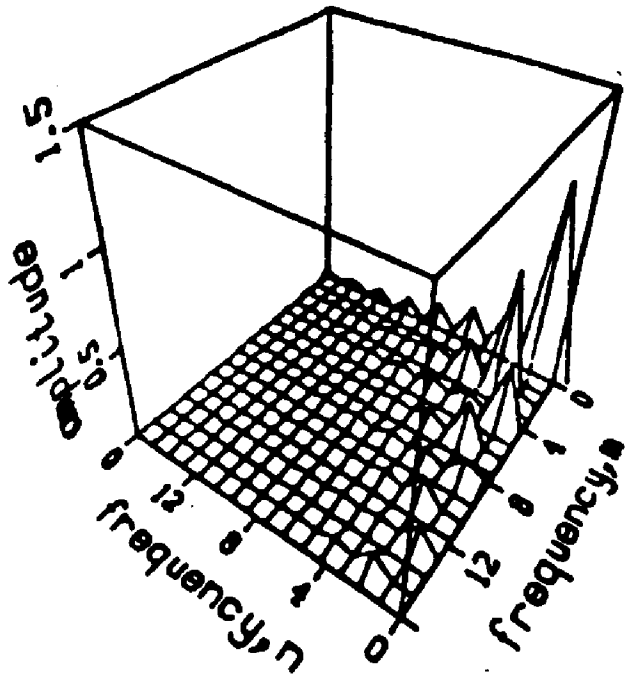


Figure 3.8 Fourier coefficients of top boundary function in Fig. 3.7.

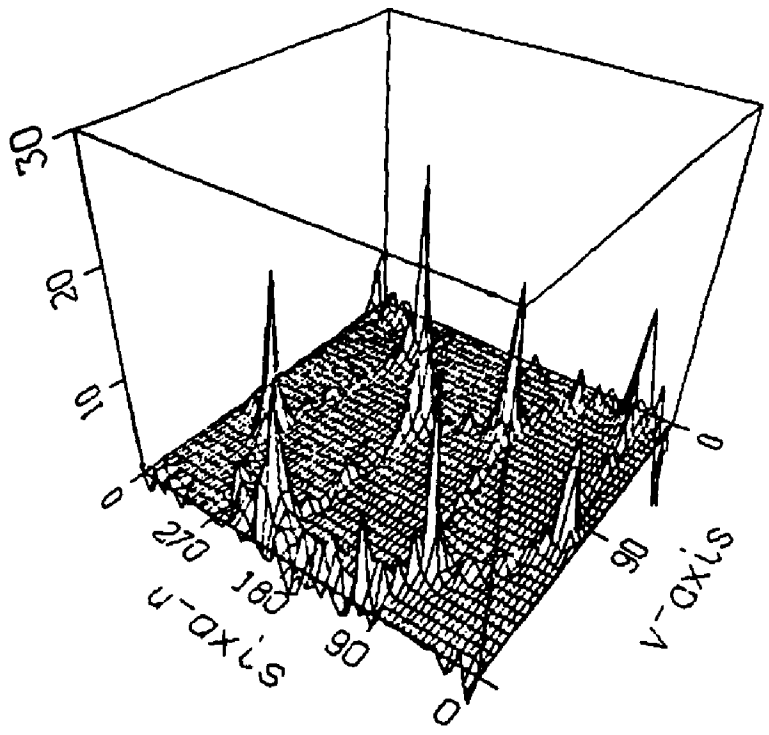


Figure 3.9 Surface curvature of a polyhedral solid.

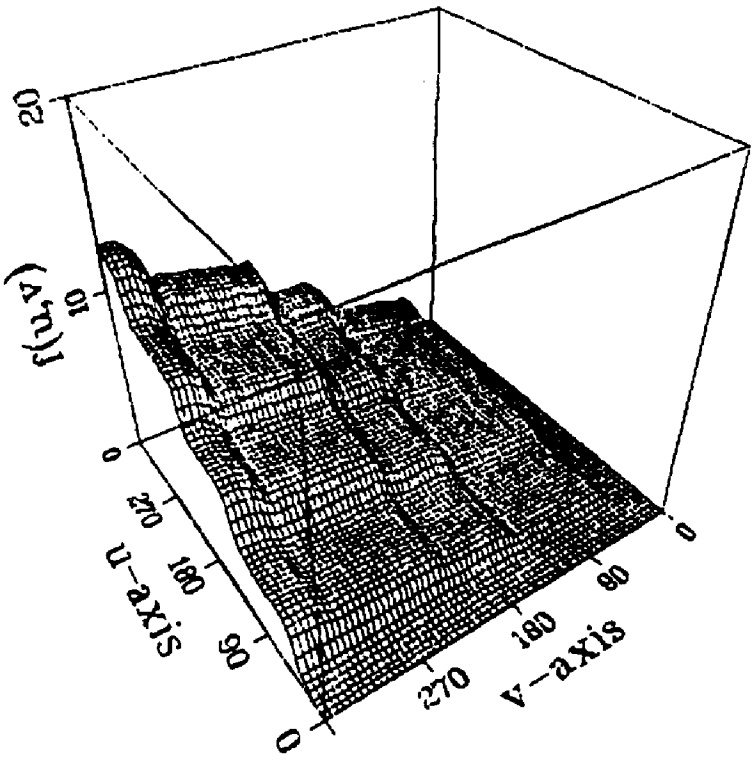


Figure 3.10 Cumulative curvature function.

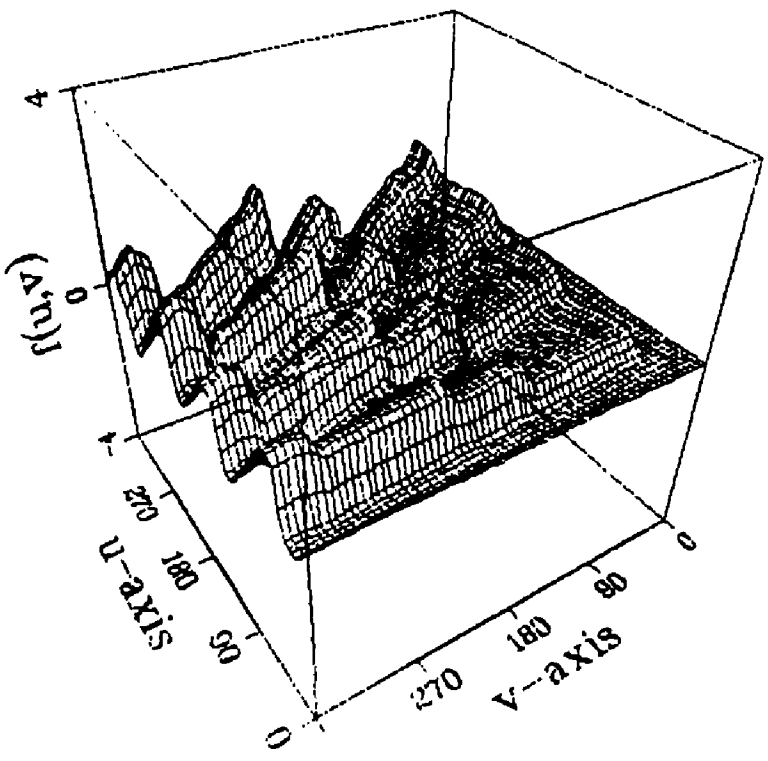


Figure 3.11 Intrinsic surface function.

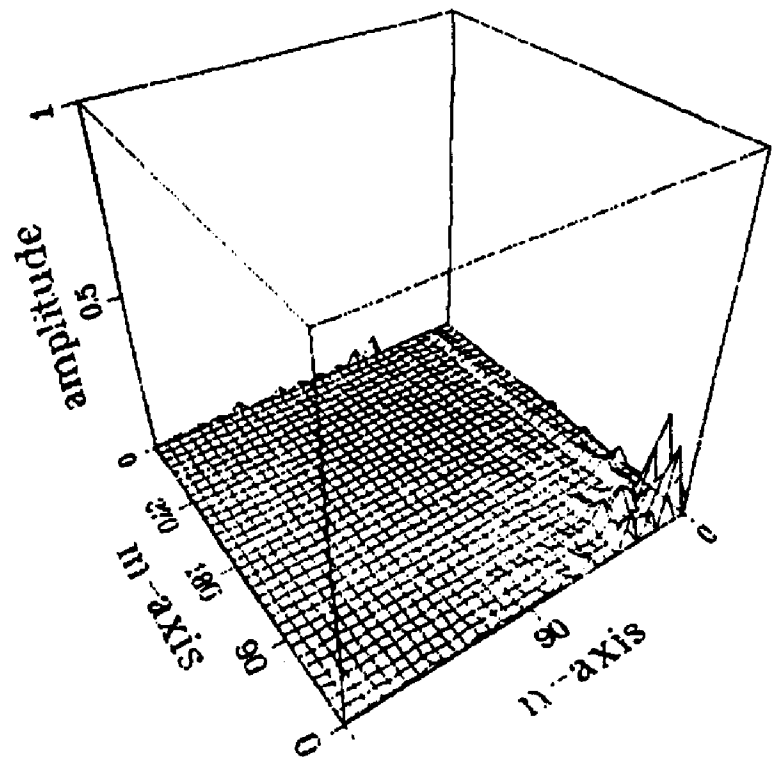


Figure 3.12 Fourier coefficients.

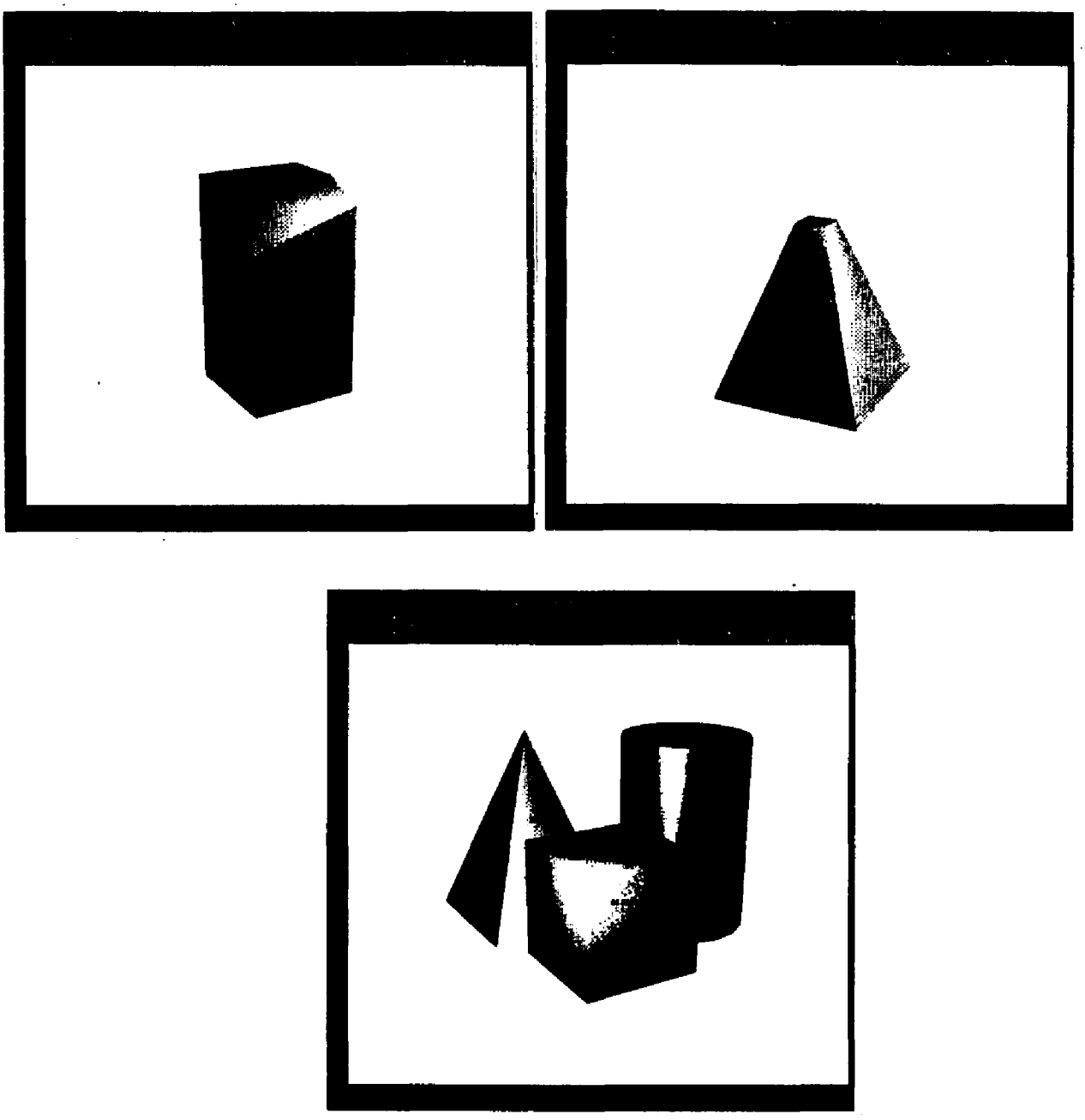


Figure 3 13 Examples of computer modeled objects.

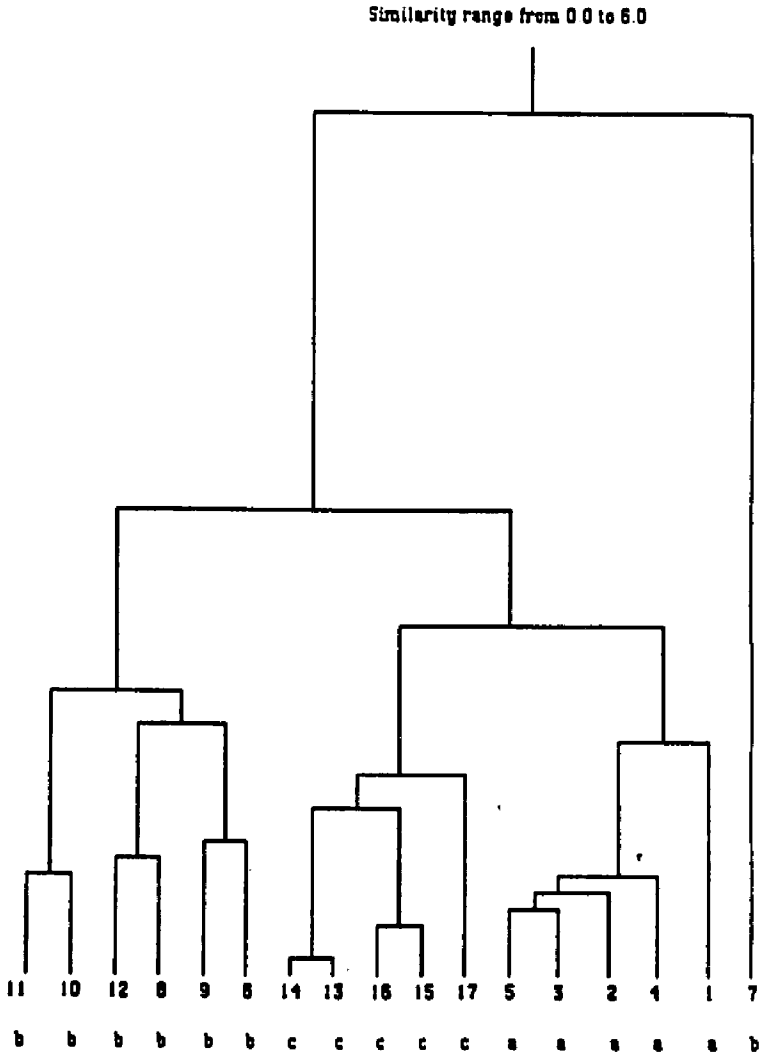


Figure 3.14 Cluster tree, l_2 norm measure of dissimilarity.

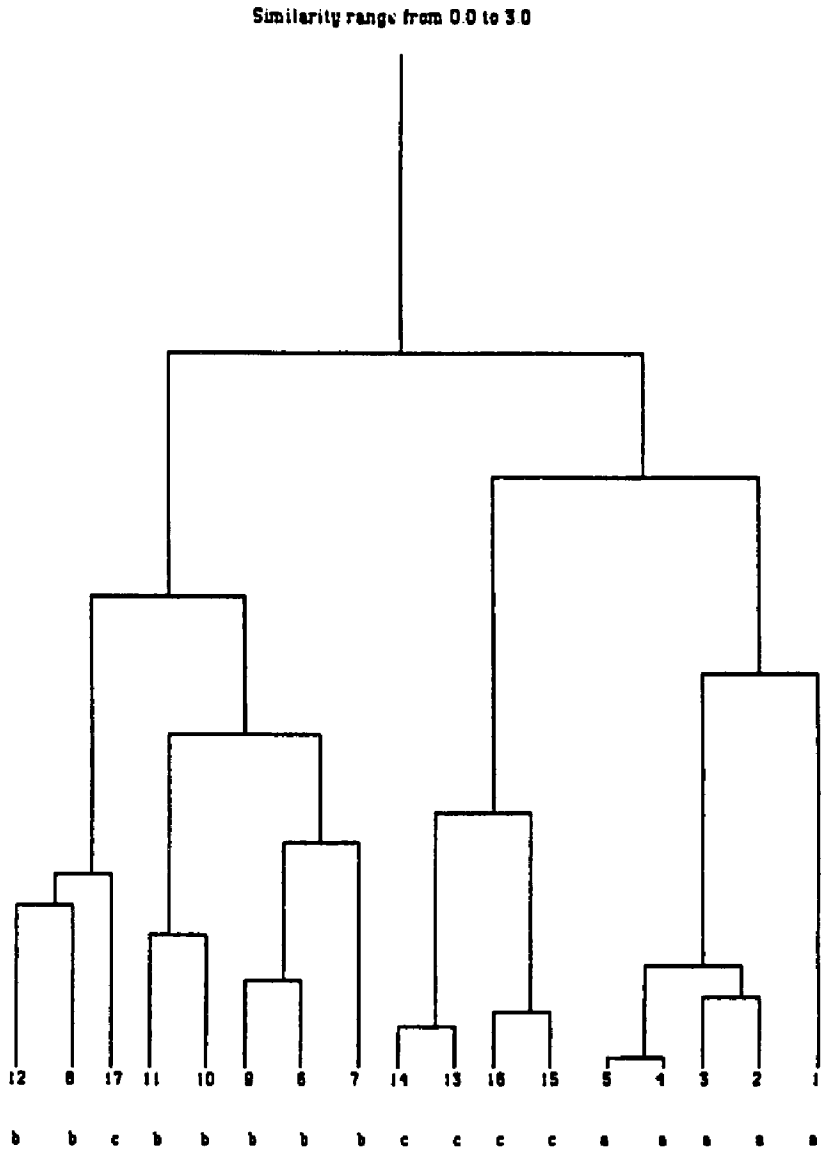


Figure 3.15 Cluster tree, angle measure of similarity.

id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1		0.65	0.62	0.72	0.59	1.55	5.87	1.42	1.33	1.08	1.40	1.25	0.94	0.93	0.91	0.84	0.96
2			0.24	0.28	0.26	1.31	5.59	1.04	1.17	0.68	1.04	0.87	0.61	0.63	0.43	0.40	0.85
3				0.38	0.22	1.18	5.54	0.93	0.99	0.67	0.97	0.77	0.60	0.61	0.44	0.35	0.67
4					0.24	1.28	5.43	1.15	1.22	0.60	0.97	0.96	0.85	0.86	0.65	0.63	1.02
5						1.19	5.49	1.07	1.05	0.58	0.94	0.85	0.72	0.72	0.59	0.52	0.80
6							4.64	0.84	0.57	0.82	0.47	0.70	1.41	1.39	1.26	1.19	1.15
7								0.14	5.09	5.06	4.73	5.13	5.82	5.82	5.61	5.60	5.63
8									0.80	0.96	0.89	0.45	0.97	1.00	0.76	0.73	0.89
9										0.91	0.76	0.59	1.12	1.08	1.09	0.98	0.69
10											0.44	0.66	0.95	0.94	0.79	0.76	1.00
11												0.64	1.21	1.20	1.06	1.01	1.12
12													0.77	0.77	0.67	0.62	0.68
13														0.09	0.39	0.39	0.60
14															0.45	0.43	0.56
15																0.13	0.73
16																	0.62
17																	

Table 3.1. Distance values (l_2 norm) for all pairs of objects.

id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1		0.63	0.64	0.60	0.57	1.19	1.27	1.25	1.33	0.90	1.08	1.22	1.10	1.10	0.93	0.88	1.19
2			0.20	0.17	0.24	0.94	1.01	0.85	1.12	0.52	0.73	0.80	0.54	0.54	0.39	0.37	0.87
3				0.19	0.19	0.82	0.90	0.76	0.97	0.48	0.64	0.73	0.66	0.66	0.44	0.37	0.72
4					0.10	0.89	0.93	0.90	1.06	0.45	0.67	0.80	0.66	0.65	0.52	0.48	0.84
5						0.83	0.89	0.89	0.99	0.42	0.63	0.78	0.73	0.71	0.56	0.51	0.80
6							0.37	0.54	0.22	0.53	0.29	0.38	1.05	1.02	0.89	0.81	0.42
7								0.70	0.43	0.62	0.43	0.51	1.11	1.08	1.00	0.94	0.44
8									0.63	0.72	0.61	0.32	0.76	0.78	0.59	0.53	0.40
9										0.72	0.48	0.52	1.23	1.20	1.06	0.97	0.44
10											0.27	0.50	0.72	0.66	0.62	0.56	0.67
11												0.37	0.86	0.81	0.75	0.68	0.52
12													0.73	0.71	0.62	0.57	0.39
13														0.10	0.29	0.39	0.96
14															0.34	0.42	0.97
15																0.11	0.76
16																	0.69
17																	

Table 3.2. Distance values (angle) for all pairs of objects.

4 Shape Restoration.

In this section a novel shape recognition and reconstruction scheme is proposed which uses the formalism of associative memory mappings [3,5,41]. The scheme will be shown to be highly successful in the restoration of distorted images. As shown earlier topological distortions such as scale change, translation and rotation can be easily managed by developing appropriately invariant representations. The problem of image distortions due to noise, and occlusion is non-trivial. Partial views, distortions due to noise and occlusion severely affect global boundary based representations. There are a number of situations under which occlusion occurs. The most obvious when one or more objects obstruct the view of the one being sensed. In 3D sensing objects invariably self-occlude themselves resulting in partial surface visibility. Finally, a part of the object might be out of range of the sensing equipment, again resulting in a partial view. Existing boundary analysis techniques are generally ill equipped to handle the problem of boundary occlusion. Scalar transform techniques are global and therefore do not give correct results when a segment of the boundary is missing or is incorrect (i.e. belongs to an occluding object). A possible solution is to resort to space domain techniques with localized representations. This, however, results in a loss of compactness of the representation and complex time consuming piece-by-piece matching schemes or complex data structures. Classification algorithms that are based on template matching techniques, require excessive amount of computation and storage [47,53,66]. It is therefore of interest to combine the compactness of scalar transform descriptions of images with space domain corrective preprocessing for purposes of efficient and successful pattern representation and recognition.

For corrective preprocessing, the distorted boundary function may be considered as an erroneous version of the original. In such a case, it is reasonable to consider error-correcting signal/image processing techniques. Subsequently the restored boundary is used as the actual representation of the viewed object. The use of numerical error correction techniques is possible because of the data structures involved. These

are vectors or matrices, where the elements can be considered as discrete, noisy samples of univariate or bivariate continuous functions of the boundary. The mathematical formulation of restoration results in an ill-posed problem, since there is no a priori guarantee that there is a solution or that a unique solution exists or that the solution is stable under small data perturbations. As an example consider the problem of constructing a description of the hidden surface of a non-transparent solid. Clearly, reconstruction of what is not seen is impossible. A solution must therefore be based on an "educated guess", which can yield correct results only if the hidden parts can be deduced through association with the visible part. It is this associative property, which mimics the capabilities of the human brain, that makes neural networks (NNs) attractive in applications were missing, non-deterministic and unpredictable data is a problem, such as image processing, computer vision, microscopy, seismic exploration, radioastronomy and so on [1,5,30,32,48,58]. The Kohonen Associative memory is a special case of a NN. It is a NN with no hidden units and a linear dependence of the outputs (activations) on the inputs (excitations). This restricts the use of the associative memory to problems were the set of inputs is linearly separable. This is not a severe restriction since the capacity of the associative memory is equal to the dimension of the image vector N, which can be thousands of elements. In addition, it is not always necessary to keep all the items distinct. Frequently, a many to one mapping is desired which implies a categorization of the inputs. The number of stored entities is more a function of the number of categories, then individual inputs. For random state vectors, however, the memory capacity has been shown to be less, about 20% of the dimensionality [32]. In what follows, an approach to shape restoration by association and generalization using an associative memory is shown.

4.1 Associative Memory Mapping.

Associative recall may in general be defined as a mapping in which a finite number of input (pattern) vectors is transformed into a given set of output vectors. In the case of incomplete or erroneous input vectors, it has been shown [40] that

this mapping is least square sense optimal. It is this error tolerance that suggests its applicability to pattern restoration and classification. There are two basic kinds of recall: the autoassociative (association) and the heteroassociative (generalization). In the former, an incomplete pattern is restored into a complete version of itself, while in the latter, an output pattern is produced in response to an input pattern. The mapping between the input-output pair of vectors is arbitrary and depends on the application requirements. Associative recall suggests the working mechanism of an error-correcting content-addressable memory. This means, that for all similar vectors, in the sense of some appropriate measure, to the set of recognizable input patterns, the recall will be similar to the corresponding output vector.

The mapping process is described by the following transfer relation:

$$y_k = M \cdot x_k \tag{4.1}$$

where the x_k's, ($x_k \in R^n$) are the input column vectors, the y_k's, ($y_k \in R^m$) are the output column vectors, and M is the unknown $m \cdot n$ associative memory matrix. The creation of a linear associative memory mapping is a process of learning a set of responses to a chosen set of input signals. It can be formulated as a problem of finding the optimal solution, in the sense of least squares, for the matrix M. Given M, recognition is achieved by linearly transforming an unknown input, according to Eq. (4.1), and therefore belongs to the scalar transform category.

An iterative technique for the determination of the matrix M entries will now be presented. Let $\{x_k\}$, $\{y_k\}$, $k=1,...,p$, be two sets of vectors. Forming the two rectangular matrices, Y and X, with the vectors y_k and x_k as their columns, as

$$X = [x_1, x_2, \ldots, x_p], \qquad Y = [y_1, x_2, \ldots, x_p] \tag{4.2}$$

every pair of vectors from sets $\{y_k\}$, $\{x_k\}$, can be related by the matrix equation

$$Y = M \cdot X \tag{4.3}$$

Using the so-called "Moore-Penrose" method [2], the least squares optimal solution of Eq. (4.3), is found as

$$M = Y \cdot X^{+} \tag{4.4}$$

where X^{+} is the pseudo-inverse of matrix X. The iterative solution of Eq. (4.4) will now be presented. The iterative solution [41] expresses the new optimal matrix, M_k, as a function of the previous optimal matrix M_{k-1} and the new observation vectors x_k and y_k. This takes the form of a difference equation for M_k:

$$M_k = M_{k-1} + (y_k - M_{k-1}x_k)c_k^t \tag{4.5}$$

where c_k^t is a gain vector that defines the correction and the index t denotes the transpose. The iterations may be initiated with $M_0 = 0$ or $M_0 = I$, where I is the identity matrix. In the simulations that follow however, the elements of matrix were randomly selected. As a consequence, slightly better results were observed, then with the other starting values. The expression for the gain factor is:

$$c_k^t = \frac{h_k^t}{|h_k|^2} \qquad h_k \neq 0 \tag{4.6}$$

where h_k is the orthogonal projection of x_k on the subspace of R^n spanned by the vectors $h_1,\ldots,h_{k-1}$. The orthogonal projection of the vector x_k to the set of previous columns of the matrix X can be obtained by using Gram-Schmidt orthogonalization [1]:

$$h_k = x_k - \sum_{j=1}^{k-1} \frac{(x_k, h_j) \cdot h_j}{|h_j|^2}, \qquad |h_j| \neq 0 \tag{4.7}$$

where (x_k, h_j) is the vector inner product, and $|h_j|$ is the Euclidean norm of the vector h_j.

The classification or reconstruction task with an associative memory is a two stage process. In the first, a learning phase, patterns representative of given classes

are used to induce the proper output responses. According to the presented algorithm, first the matrices X and Y need to be generated. This is done by concatenating the pattern vectors in a column-wise fashion, with the columns of X representing the set of training patterns and columns of Y, the desired response. The mapping that relates the input-output pairs is obtained by recursively applying Eq. (4.5) to the columns of X and Y. Depending on the specific application and requirements, a variety of input-output pairs can be used during the learning stage. In the second, or recognition stage, an unknown or degraded pattern vector is applied to the input of the associative memory filter. This filter then produces a response similar, in the least square sense, to one of the trained responses.

4.2 Associative Memory Restoration of Planar Curves.

It is assumed that the image domain is that of planar images. The silhouettes of planar images contain information about the shape of the projected real objects. For purposes of shape-based pattern recognition it is useful to only consider the border of the silhouette. These borders as planar curves can be efficiently coded and used as an intermediate, substitute representation of the image. The extraction of a representation from the raw image data is known as the feature extraction step of a pattern recognition process. The image processing with NNs described in this section is performed at the feature level, with images represented by the cumulative shape function.

The extracted boundary forms an oriented, simple, and closed curve. For purposes of computer processing, the continuous, smooth curve is approximated by a finite number of straight, not necessarily of fixed length, line segments. A possible boundary representation is in the form of a pair of column vectors $((x_1, x_2)$, $x_1 \in R^n$, $x_2 \in R^n)$, with projections $(x_{1,1},\ldots,x_{1,n})^t$ and $(x_{2,1},\ldots,x_{2,n})^t$ that are the Cartesian coordinates of the n line segment endpoints and t stands for transpose. From this two-variable data set, a number of efficient one-dimensional data sets

can be formed. In all of the examples the cumulative slope, described in Sect. 2.3.2, has been used to represent shape. Therefore, all the boundaries are 1-D sequences, $x(n)$, of uniformly spaced samples of the trace of the tangent indicatrix. In the examples, the contours have been hand digitized using a 64x64 square grid (Fig. 4.1), the boundary interpolated and then uniformly resampled to yield a desired number of equidistant points. This process introduces a substantial amount of boundary noise. However, this did not impair the performance of the pattern recognition scheme. The length of the sequence, 128 samples, is a compromise between the accuracy of the representation and storage and computational requirements. Finally, two different data sets have been chosen for the computer experiments. The data set of handwritten characters (data set 1, see Fig. 4.2) is characterized by large interclass variations and relatively uncomplicated boundary curves, while the airplane silhouettes (data set 2, see Fig. 4.2), are characterized by small intraclass variations and complex boundary curves. These data sets were chosen to test the scope of the shape recognition scheme. There are three self-evident pattern classes in data set 1 and two pattern classes in data set 2. They represent two basic types of planes: large passenger planes and small, light cargo planes. However, because of invariance of the shape representation, the size of a plane is not used as a discriminating feature.

4.2.1 Definition of distortions:

For purposes of testing the association and generalization capabilities of the mapping from sec. 3.2, a variety of pattern vector distortions were introduced. They are:

-linear, spatial truncation,

$$\hat{x}(i) = \begin{cases} x(i), & i \le N_t, \quad where \quad N_t \le N \\ 0, & otherwise \end{cases} \tag{4.8}$$

-uniform, additive noise,

$$\hat{x}(i) = x(i) + n(i), \quad for \quad 0 \le i \le N-1,$$
$$where \quad -\alpha \le n(i) \le \alpha \quad and \quad \alpha \quad is \quad real \tag{4.9}$$

-spectral bandlimiting,

$$\hat{x}(i) = \frac{1}{N} \sum_{j=0}^{N-1} X(j) H(j) W_N^{-ij},$$
$$where \quad H(j) = \begin{cases} 1, & for \quad j \le N_c, \quad where \quad N_c < N, \\ 0, & otherwise \end{cases}$$
$$and \quad X(j) = \sum_{i=0}^{N-1} x(i) W_N^{ij} \tag{4.10}$$

Fig. 4.3, shows the effect of the distortions on a representative shape function.

4.2.2 Computer experiments.

In the experiments that follow, the descriptive and reconstructive capabilities, for different shapes, of the associative memory filter are demonstrated.

To illustrate the versatility of an associative memory as a shape recognition and reconstruction tool, a number of useful associated pairs have been used. First, for efficient coding and classification, the boundary representation vectors (i.e., pattern vectors) were mapped into seven element feature vectors. These binary feature vectors correspond to the ASCII codes for alpha-numeric characters, where 'B' is {1,0,0,0,0,1,0), 'D' is {1,0,0,0,1,0,0}, and '8' is represented by {0,1,1,1,0,0,0). In the training stage 24 associated pairs were used, grouped in sets of 8 corresponding to the 3 pattern classes. It is expected that 16 such pattern classes could be accommodated since this would continue to satisfy the requirement that the number of patterns not exceed the dimensionality of the pattern vector. However, if more input training vectors must be used, their dimensionality can be increased by either sampling the boundary at a greater

number of boundary points or by non-linear preprocessing. A discussion of the issues of pattern dimensionality, linear invariance and associative memory matrix inversion is in [41]. In the two-class airplane identification problem a more compact two-element feature vector was used. Here, the code (0,1) represents passenger while the code (1,0) represents cargo planes.

The scatter diagrams of Fig. 4.4 demonstrate the high degree of tolerance of the heteroassociative recall to intraclass variability and interclass separability, as well as its tolerance to shape function degradation. The tight clustering of the shapes within each class and good separation between the clusters are noteworthy. For display purposes only the fifth and sixth digit of the seven digit ASCII output code have been used. They are the abscissa and ordinate of the diagram, respectively. Fig. 4.4(a) depicts the result of a recognition experiment in which 15 handwritten characters, not part of the training set, were heteroassociatively mapped into the ASCII feature space. The geometric separation in the 2-D space of Fig. 4.4(a) is an indication of the interclass separability of the clusters in the 7-D feature space. It is clear, that any one of a number of similarity measures [4,29] could be used with adequate classification results. Figs. 4.4(b), 4.4(c) illustrate the clustering in the feature space which results from a mapping of degraded boundary representations. To verify the effect of low-pass filtering in the frequency domain and a 32-point window bandpass filtering in the space domain, Figs. 4.4(b) and 4.4(c), respectively, should be compared with Fig. 4.4(a). There is no significant change in the separability between classes and there is continued tight clustering within classes indicating that the classification process has not been impaired.

Boundary reconstruction with an associative memory is illustrated in the next few figures. A variety of bandpass space domain filters have been used to distort the elements of the shape vector. These distorted patterns were subsequently applied as an input to an associative memory filter. The associative

memory matrix used in present experiments has been trained on undistorted representations, i.e. no a priori knowledge of the type and the effect of the degradation is assumed. The training pairs are shown in Fig. 4.5. In Fig 4.6, restoration results for truncated pattern representation vectors are shown. For all the input patterns, 25% (32 of 128 samples, N_t=96) of the shape vector was truncated to obtain the results in Fig. 4.6(a), and 50% (N_t=64) truncation was used in Fig. 4.6(b). Although slight boundary distortions are noticeable, gross shape features are clearly preserved. In Fig. 4.7, the results of associative recall with noisy data are shown. Noise power was in the order of the signal power. The signal-to-noise ratio (SNR) was calculated using:

$$SNR = 20\log_{10}\frac{\sqrt{x_1^2+x_2^2+\ldots+x_n^2}}{\sqrt{n_1^2+n_2^2+\ldots+n_n^2}} \qquad (4.11)$$

The SNRs for the results of Fig. 4.7 are given in Table 4.1. Finally, results of reconstruction from partial views will be shown. These are especially significant since partial views frequently occur in situations where the objects are forced to move with respect to a fixed viewpoint (i.e., in target tracking, in recognition of parts on a conveyor belt, etc.). In Fig. 4.8, the shape function of one of the Roman letters is shown together with its version resulting from occlusion. As this figure indicates, these occluded curves are actually new and different shapes. Occlusion causes lengthening (shortening) of the boundary. This effect combined with uniform sampling at a fixed number of points, destroys the match between corresponding vector elements of the original, full-view pattern, and its occluded version. The mismatch will occur even at the unaffected parts of the boundary. Fig. 4.9 illustrates the reconstruction of occluded boundaries. The occluded images were obtained by covering part of the image and recoding the visible boundary together with the resulting edge. These new shape vectors were used as inputs to an associative memory filter, a filter that had been trained on full-view boundaries. Despite the occlusion, the reconstructed boundary curves are clearly recognizable.

4.3 Neural Network Shape Restoration.

In this section, an approach to the shape distortion problem that is based on the application of a neural network (NN) is described. Here, the feed-forward, back-propagation learning algorithm for multilayered NNs is used. The associative character of NNs is especially well-suited to handling problems which involve distorted, "blurry", and unpredictable data. The attractive properties of NNs include a large degree of fault tolerance, an invariance to distortion, graceful degradation, and an adaptive behavior through learning. Simple associative networks, in which a set of input patterns is mapped directly into a set of output patterns, have proven useful in a wide variety of applications [3,24,41,64]. Due to their mapping of similar input to similar output patterns, they allow for reasonable generalizations on unknown patterns. These networks do not, however, allow for augmenting or recoding of the pattern representation codes leading to significant mapping problems with inputs that exhibit excessive and unwanted similarities. This, unfortunately, is a common problem when working with large data sets [32,48,58]. With the addition of hidden layers, the so-called internal representation, there is always a way of recoding the inputs, which supports any required mapping.

The classification or reconstruction task with a feed-forward back-propagation NN is a two stage process. In the first, a learning phase, patterns representative of given classes are used to induce the proper output responses. The back-propagation algorithm is a gradient descent scheme in the space of the connection weights (see Appendix). It minimizes the sum of squares of output errors. The connection weights of the network are modified until a desired minimum error criterion is reached, a mapping is obtained that relates the input-output pairs. In a second, or recognition stage, an unknown or degraded pattern vector is applied as an input to the NN. This filter then produces a response similar, in the least-square error sense, to one of the trained responses.

4.3.1 Computer simulations.

The neural network chosen for the computer simulations in this section was a three-layer network with 64 input and 64 output nodes and a 96 node internal layer. This resulted in a network with 12288 connections. The inputs and outputs were 64-sample, cumulative slope shape functions as defined in Sect. 2.3.2. Iteration time was approximately 23 seconds on an AT compatible computer with a math co-processor unit. The associated learning patterns are shown in Fig. 4.5. The restoration results from 32 and 48 samples of a truncated shape function are shown in Fig. 4.10. Results for partial views of a Roman letter 'B' are in Fig. 4.11. These results are as expected, based on the earlier solutions using a NN with no hidden unit. The association and generalization performance is comparable for this linearly separable set of input patterns and problem dimensionality. Interesting results were however obtained from an examination of the learning dynamics of the NN. It was noticed that the total mean squared output error and the maximum unit output error decreased very fast, that is to within acceptable levels in 75-125 iterations. In our case, this required only 8-14 presentations of the full set of training patterns. In Fig. 4.12, outputs after 25, 50, and 150 iterations are shown. A comparison with Fig. 4.5, reveals how small this error is. The total root mean squared (RMS) output error and maximum unit output error for different numbers of iterations are in Table 4.2. The total RMS output error is computed by summing the square of the difference of the training output (target) and actual output in every output unit for every pattern, averaging this, and then taking the square root. Such fast learning is a consequence of the linear independence of the training patterns. From observations of the network dynamics it is hypothesized that the space of the connection weights has the shape of a paraboloid with very steep sides far away from the global minimum and very flat close to it. The initial very steep, monotonic decline in the average total RMS output error was followed by an ever more gradual change. These dynamics suggest the following

comments. For the domain of linearly independent patterns there is no advantage in using a multilevel NN in place of an associative memory network. However, the penalty in longer learning and recall phases when using a multilevel NN is very small indeed. With decreasing memory costs and optical disk implementations, the more significant disadvantage of increased storage requirements may become a minor factor as well.

4.4 Associative Pre-Processing of Range Data.

A major disadvantage of global representation schemes is the inability to gracefully handle incomplete information. In 3D image analysis self occlusion of solids is an ever present phenomenon. For single view imaging and triangulation methods, only a part of the bounding surface of an object is sensed. A part remains invisible and therefore unavailable for further processing. Methods that require knowledge of the complete boundary cannot be directly applied. One such adversely affected image processing scheme is the shape description method discussed earlier. To overcome this problem, a pre-processing step based on the image processing formalism of associative memory mapping is used. It is assumed that a range data image of the sensed 3D object is available. In addition, a restricted world model is assumed with only a finite variety of shapes expected to occur. In this context, it is shown that complete boundary data in the form of a radial representation (see Sect. 3.2) can be computed from a single depth map. Subsequently, as in Sect. 3.3.2, shape descriptors of the unknown object are calculated. This is now possible in spite of the fact that only partial surface information is available. This approach, as in the earlier scheme for restoring planar curves, makes use of the generalizing capability of the associative memory. Finally, classification of objects viewed from a single position is possible.

The implementation details are as follows. For each solid model, range images from a number of different viewpoints were obtained. Degenerate viewpoints were

carefully avoided. These sets of depth maps for all solids were then mapped into the corresponding radial representations of the model objects. This was done following the algorithm in Sect. 3.2, with the depth maps as the x_k column vectors and radial representations as the y_k column vectors. The dimensionality of x_k is 1024 elements corresponding to column-wise concatenation of the depth map values, and y_k is 512 elements obtained similarly, from the matrix of radial values. The resulting associative memory matrix is then 512x1024 in size. In Fig. 4.13, an intensity image and a surface plot of a single depth map are shown. Bright points indicate surface points closer to the viewer. The range data set was synthesized from solids modeled with the IDEAS computer graphics software package. The radial data was obtained similarly as in Sect. 3.2. Two associated input-output pairs are shown in Fig. 4.14.

4.4.1 Computer simulations.

In this section classification results of unknown objects are presented. The unknown objects are represented by their depth maps. The classification assigns the 3D solid to one of three classes according to the clustering results from Sect. 3.5. An associative memory mapping was obtained as described above. 30 additional depth maps, not used during training, were computed, ten for each category of model shapes. Shape descriptors obtained from the associatively recalled radial representations were compared with the cluster centers. The root mean squared distance measure (l_2 norm) was the classification criterion. The unknown solid was assigned to the class for which the dissimilarity measure was minimum. As in Sect. 3.5, only eight shape descriptors were used. The classification results are tabulated in Table 4.3. It was observed from the simulation results that a high probability of correct classification occurred when the unknown input differed from a training input by not more than 5-10° (degrees) of rotation. Otherwise the results were unpredictable. This would suggest the need for a very large training set. Luckily, man-made objects posses

symmetries that would significantly reduce the number of needed viewpoints. Additionally, it is safe to predict that only a few of the remaining theoretically possible viewpoint would, in reality, occur. It is unlikely, that in a normal setting a pyramid would be viewed as standing on its tip. It is possible that such, and other restrictions would result in lowering the number of necessary viewpoints to a manageable quantity.

As an additional test of the associative mapping between depth maps and radial representations, incomplete versions of the former were used as inputs. A result of the auto-associative recall is shown in Fig. 4.15. The error-correcting character of the associative mapping is again in evidence.

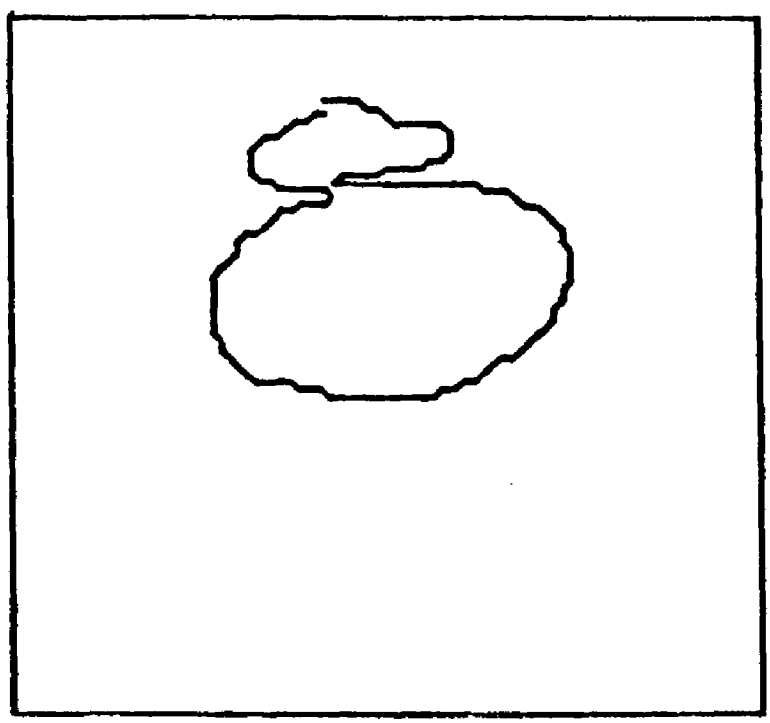


Figure 4.1. Silhouette of a handwritten numeral '8', 64x64 grid.

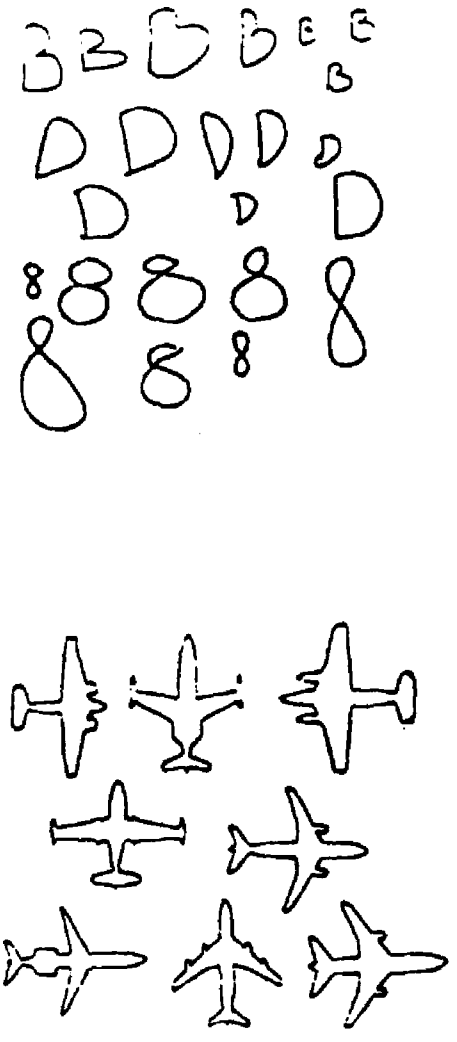
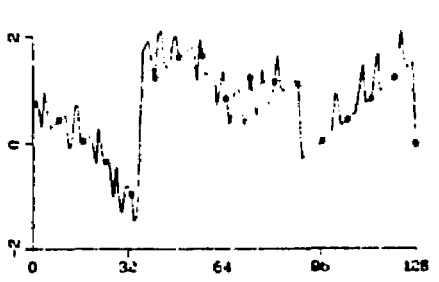
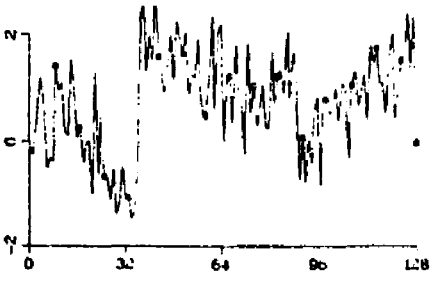


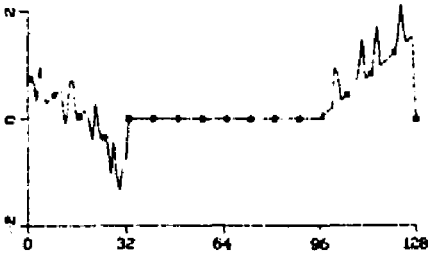
Figure 4.2 Data sets 1 (top) and 2 (bottom).



(a)



(c)



(b)

(d)

Figure 4.3 Shape function distortions (a) undistorted, (b) linear, spatial truncation, (c) uniform, additive noise, (d) spectral bandlimiting.

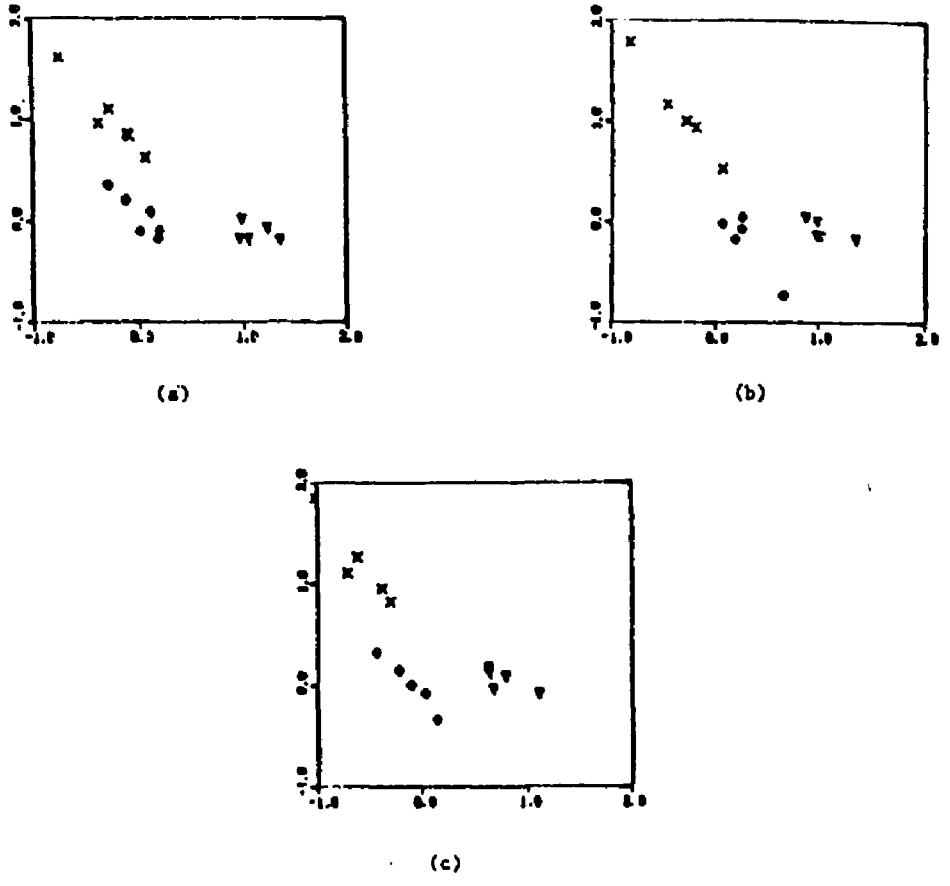


Figure 4.4 Scatter diagrams (a) unknown inputs, (b) low-pass filtered inputs, (c) truncated inputs.

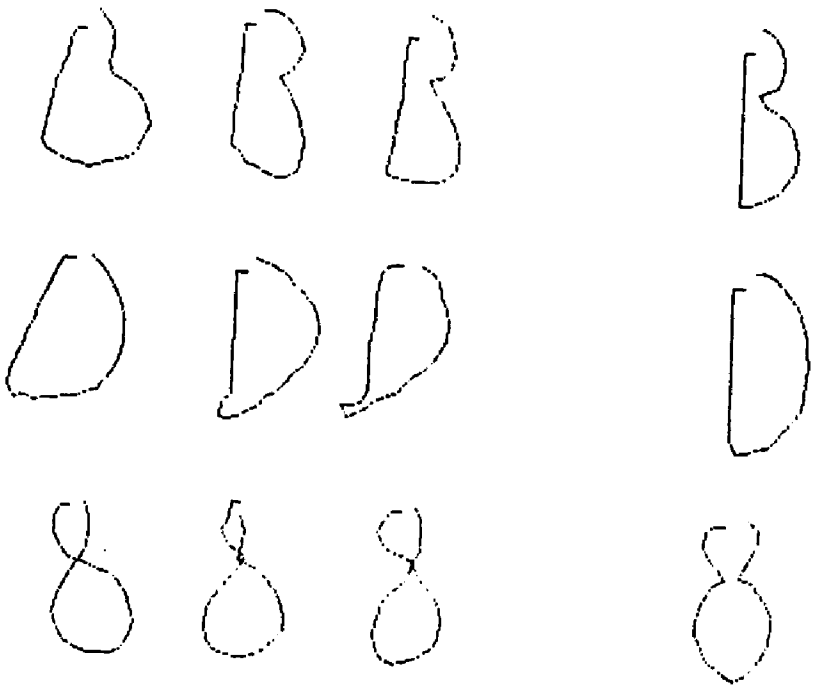


Figure 4.5 Associative memory training pairs.

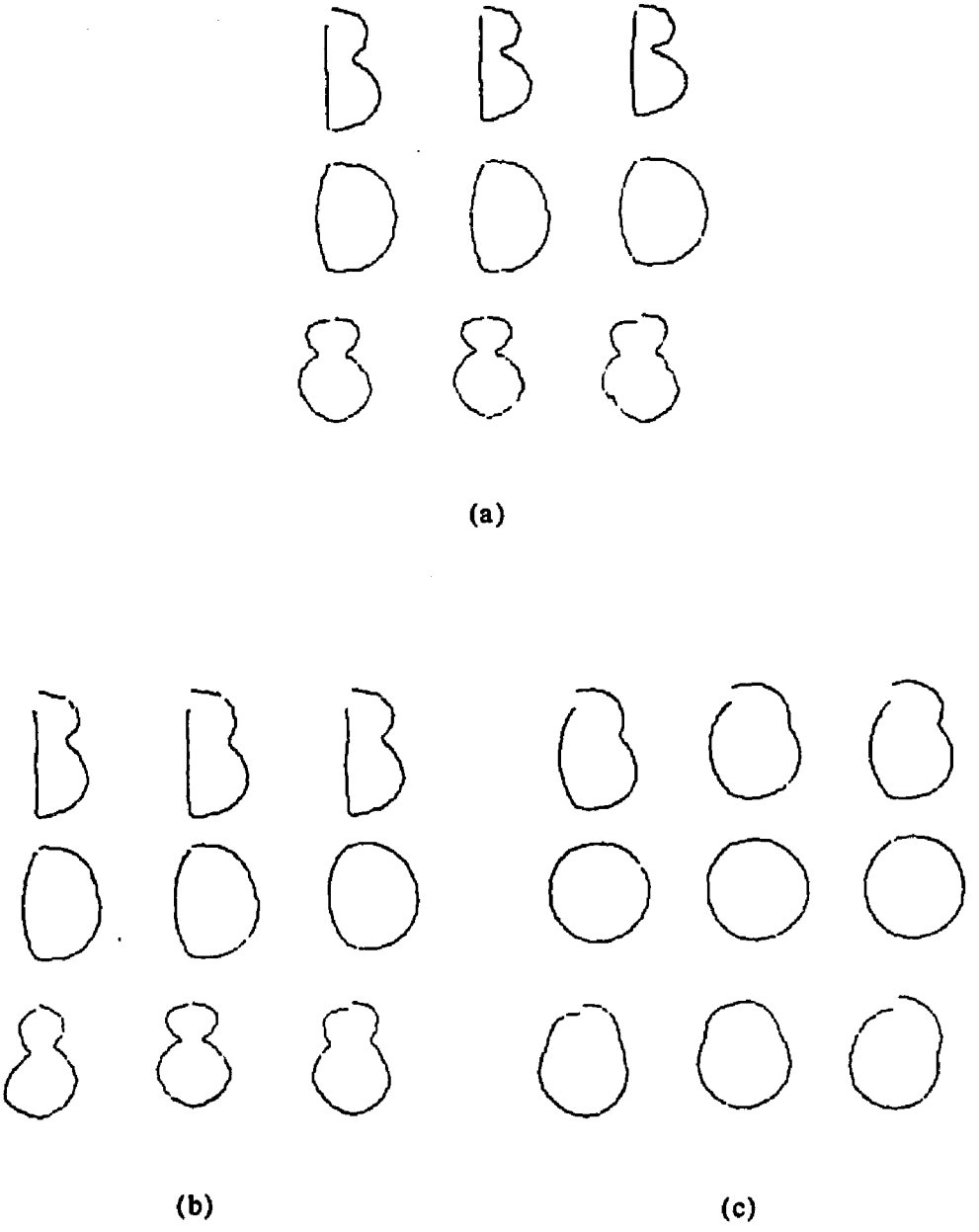


Figure 4.6 Associative restoration, linear spatial truncation (a) N_t=96, (b) N_t=64, (c) N_t=32.

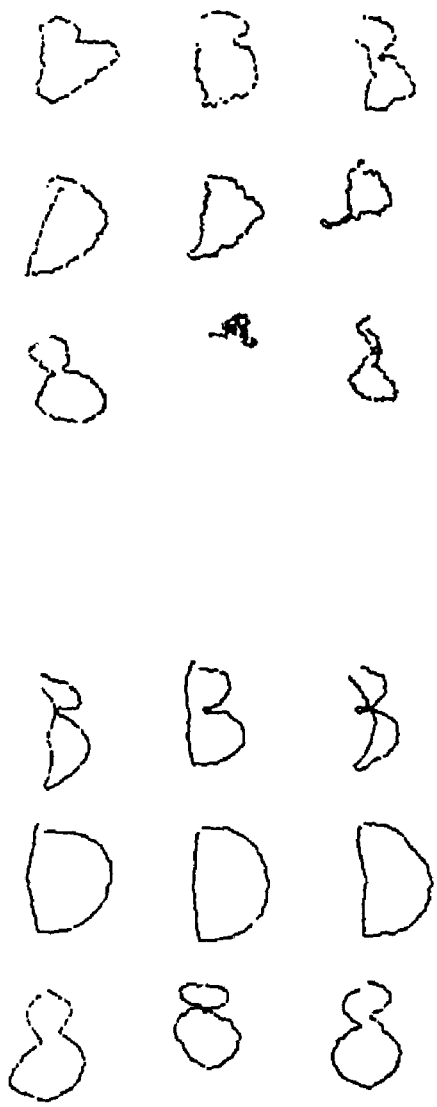


Figure 4.7 Associative restoration, noise, input (top), output (bottom).

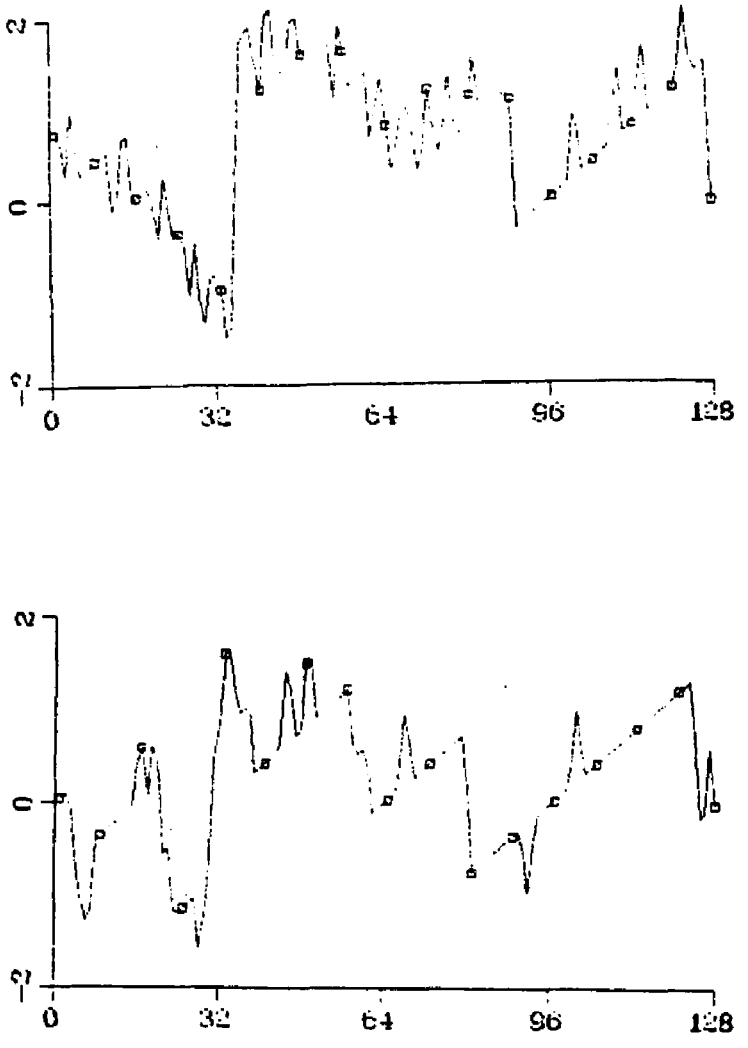


Figure 4.8 Shape function distortion due to occlusion, original (top), occluded (bottom).

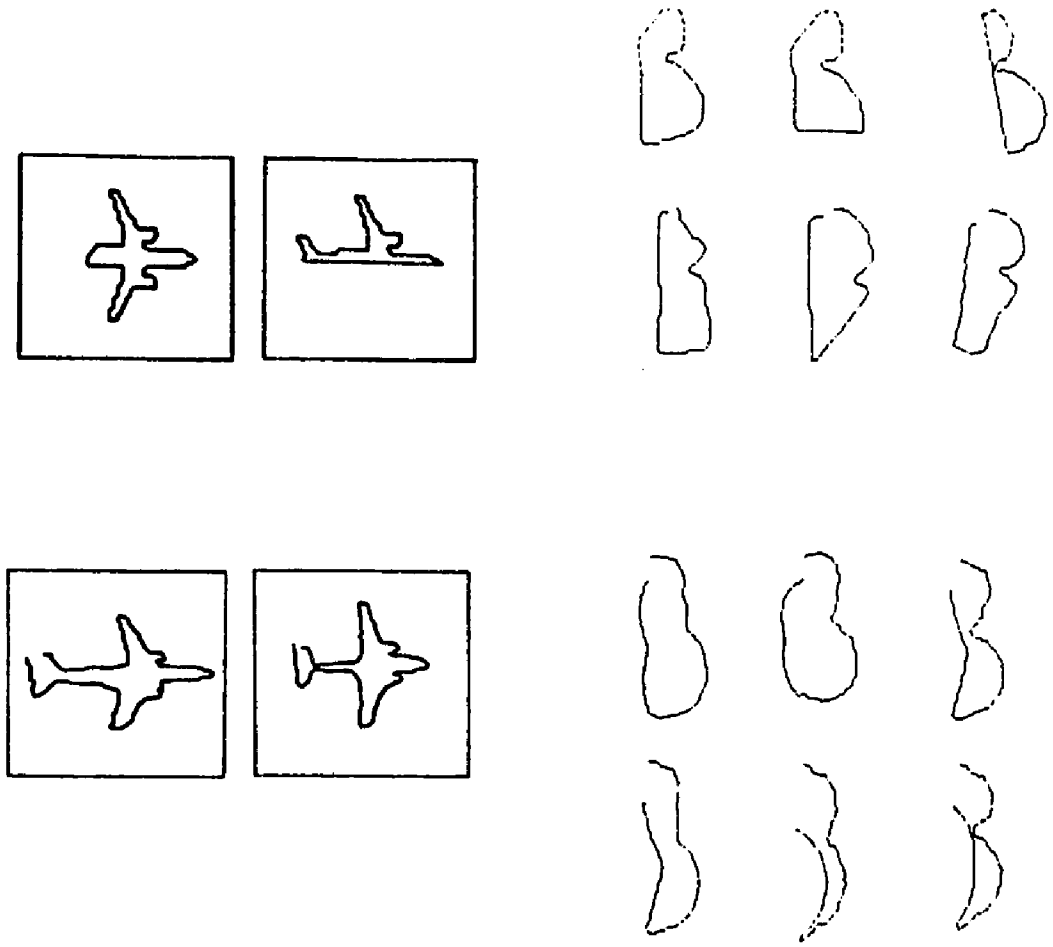


Figure 4.9 Associative restoration, occlusion, input (top), output (bottom).

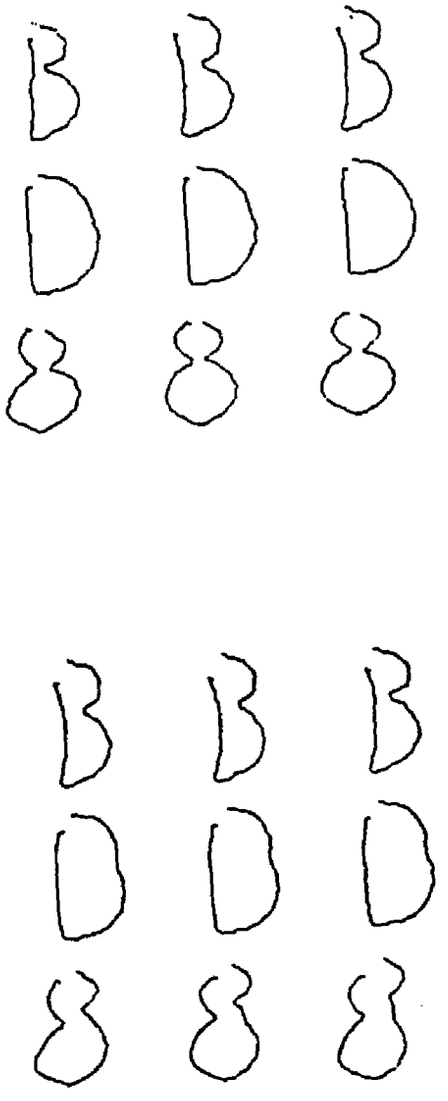


Figure 4.10 Restoration results, truncated sequences, 25% (top), 50% (bottom).

Figure 4.11 Restoration results, occlusion, input (top), output (bottom).

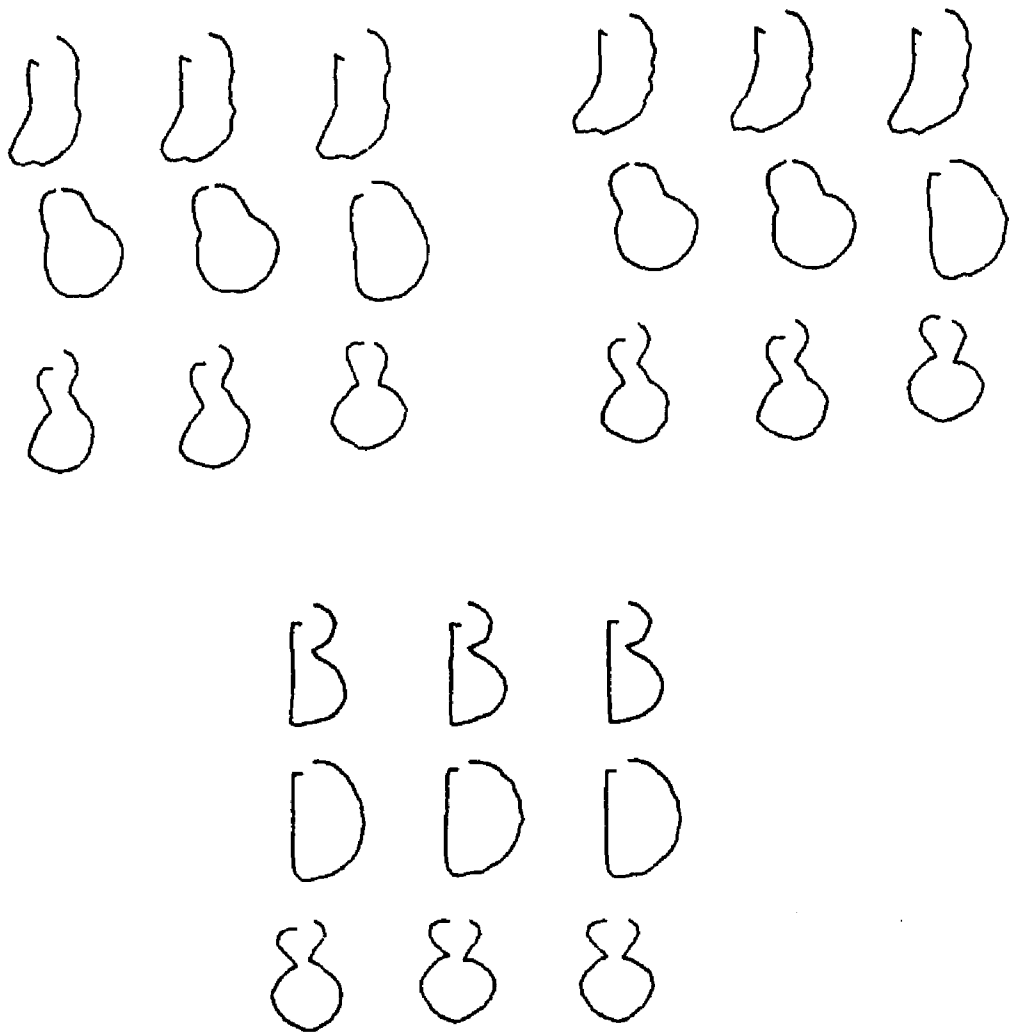


Figure 4.12 Learning dynamics, network output after 25 (top left), 50 (top right) and 150 (bottom) iterations.

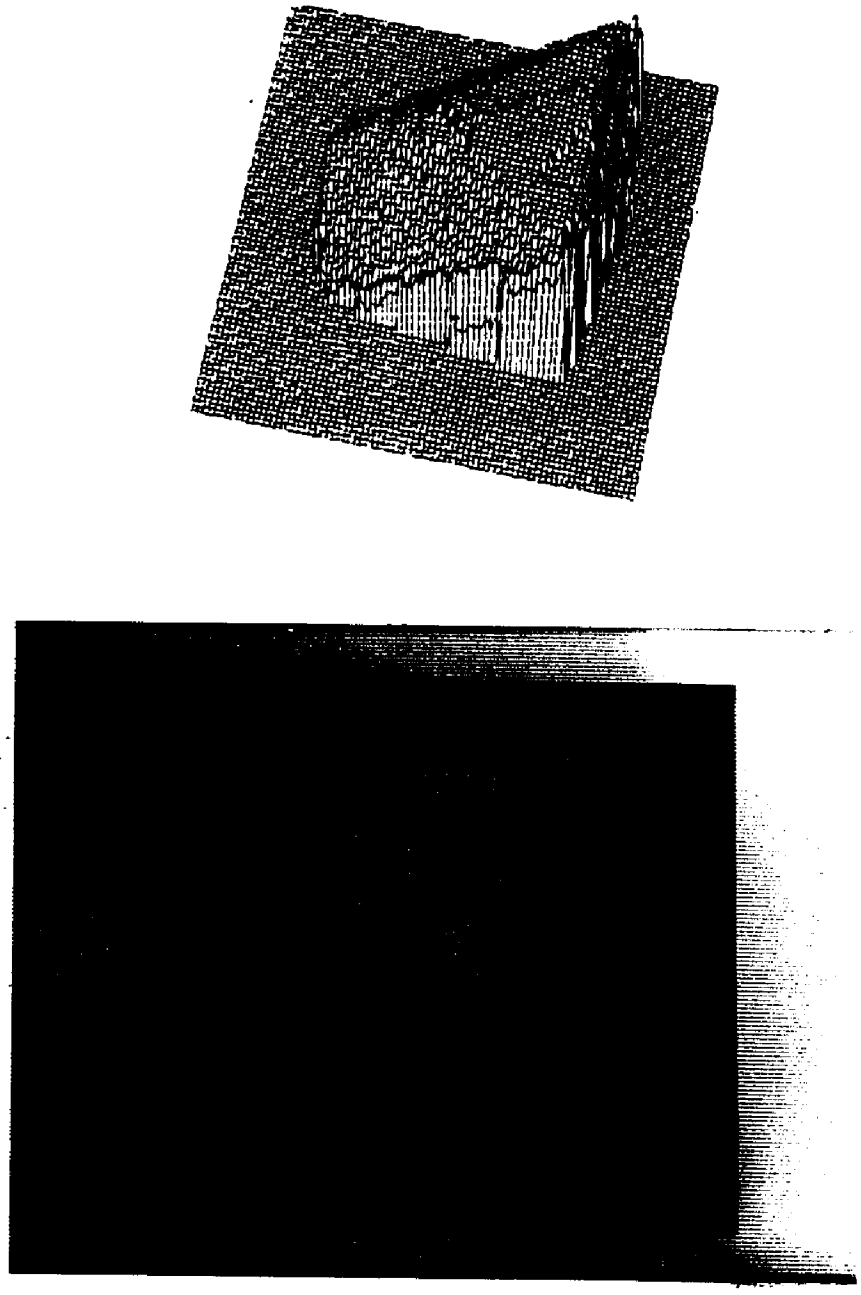


Figure 4.13 Depth map of a polyhedral solid as a surface plot (top) and intensity image (bottom).

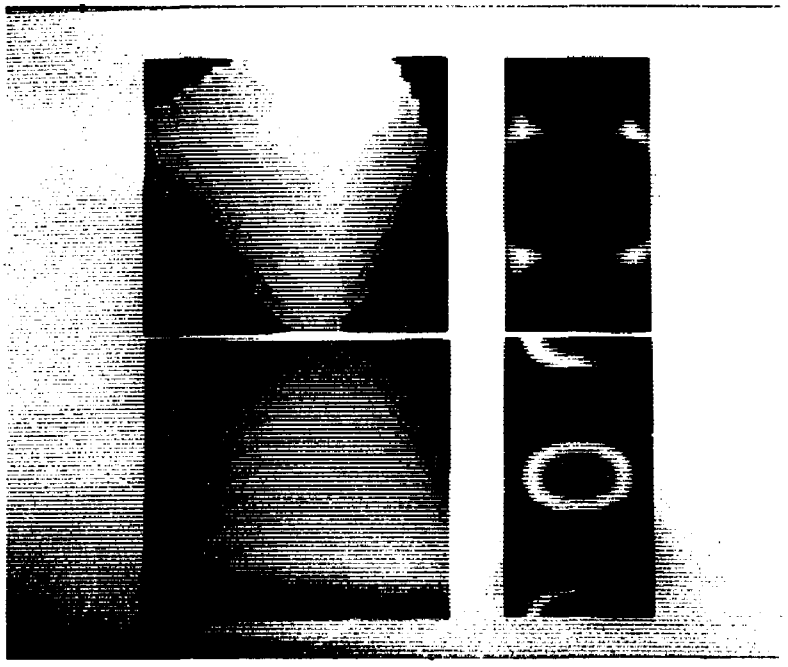


Figure 4.14 Examples of associated training pairs.

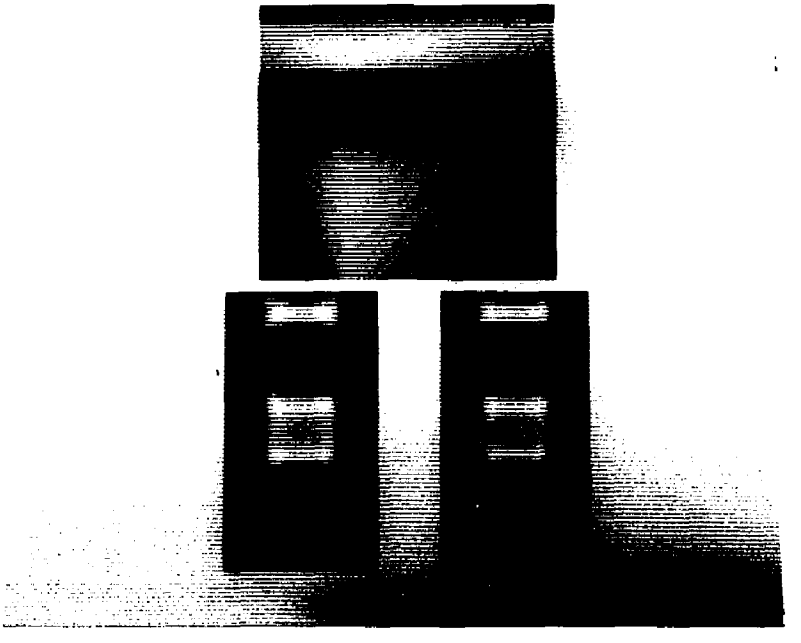


Figure 4.15 Auto-associative recall of a range data image, input (top), output (bottom left) and desired output (bottom right).

row number	column number 1	2	3
1	1.468	1.107	1.845
2	1.849	-1.164	0.869
3	4.592	-4.927	2.779

Table 4.1 SNRs for curves illustrated in Fig 4.7.

number of iterations	total RMS error	maximum unit error
25	0.089372	0.268489
50	0.060973	0.191378
75	0.047541	0.152073
100	0.020109	0.090823
200	0.005729	0.040298

Table 4.2 Total RMS output error and maximum unit output error during training cycle.

	assigned to:		
	c_1	c_2	c_3
c_1	7	3	0
c_2	3	6	1
c_3	0	2	8

Table 4.3 Classification results, c_i is class i.

5 Summary and Discussion.

In this thesis three new boundary-based representations were introduced. The boundaries were assumed to be closed curves (for 2D images) or closed, convex surfaces (3D). In a unified approach, global, analytic functions of such boundaries were obtained from differential, and therefore local, characteristics. As a result, canonical representations of curves and surfaces were defined. They are independent of how the geometric entity is embedded in the Cartesian coordinate system. Canonical representations are extremely useful in pattern recognition since they lead to relatively simple classification schemes. There is a unique representation for every geometric entity, and therefore no need to store its frequently numerous variants. The intrinsic curve and surface functions were subsequently mapped into the Fourier domain. It is interesting that similar objects, based on their shape characteristics, posses similar Fourier coefficients. These are the so-called shape descriptors. Usually only a few low-order descriptors are needed to encode prominent object features. In the classification example given in Sect. 3, only 8 descriptors were used. The resulting savings in storage and matching operations must be balanced against the increase in processing time involved in computing such a representation.

The computation of a shape invariant, canonical representation is, as shown, a non-trivial task. The problem of finding a mapping between a viewer-dependent image and a viewer-independent representation is ill-posed and does not, in general, have a solution (see also [43], pp. 296-300). This is a consequence of not having complete information about an object, mostly because of occlusion. In Sect. 4, a possible solution to the problem of occlusion was proposed. The image processing formalism of NNs was used to compute mappings with occlusion correcting capabilities. The associative character of NNs is especially well-suited to handling problems which involve distorted, "blurry", and unpredictable data. The attractive properties of NNs include a large degree of fault tolerance, an invariance to distortion, graceful degradation, and an adaptive behavior through learning. Examples were given of the restoration of

shape functions from a number of possible distortions, including additive noise, space domain truncation and partial views. For 3D imaging purposes, a linear associative memory was used to compute canonical surface representations from single range data images. Here, because of self-occlusion only a part of the bounding surface of an object was available. For the restoration of planar boundary curves, both a Kohonen associative memory and a multilayered NNs with non-linear activations, were used. The multilayered NN was trained using the back-propagation algorithm. Fast convergence rate of the least squares output error was observed during training with the back-propagation algorithm.

In the discussion that follows some of the problems left unsolved will be discussed and possible solutions proposed. This will serve as a basis for future research in shape description and image restoration.

The information preserving and non-preserving character of a shape description scheme is frequently an important issue. Information preserving descriptions allow for the reconstruction of the image data (silhouette, bounding surface) from the descriptors, with possible errors but without significant shape distortion. It would be of interest to investigate ways of reconstructing an object's boundary from the differential characteristics of its bounding curve (2D) or surface (3D). It has been shown that curvature is an intrinsic property of a curve or surface. It is frequently used as a discriminating feature between objects differing in shape. Earlier surface curvature was used to obtain a new set of shape descriptors for 3D objects. It would be desirable therefore, to be able to reconstruct the surface geometry from curvature.

As extensions to the shape description method described in this thesis the following would be interesting alternatives. Most importantly, the restriction of convex surfaces should be removed. This will require a modification of the parametrization scheme with the parameters dependent on the differential characteristics of the surface. For recognition purposes all object surfaces must be mapped into a common parameter domain. In an alternative approach to representation and shape description,

the 2D autoregressive moving average (ARMA) model could be utilized. The parameters of this model, unique for a given surface will provide the desired set of descriptors. Such compact, shape invariant surface descriptors can be easily and efficiently cataloged. A world in the form of a library of object descriptors allows for effective pattern recognition. The unknown object's descriptors will be matched with the pre-stored descriptors. The best match determines the classification of the new object. The best match may be obtained through the evaluation of similarity measures. The Euclidean distance metric is one example. Various other matching metrics could be investigated. Clustering algorithms may be utilized for the purpose of subdividing the feature space (the descriptor space) and classification based on feature vector direction.

Recently, Monte Carlo optimization techniques have been proposed as an alternative method of learning neural interconnections. They have also been successfully applied to problems in early vision such as surface reconstruction and stereo computations. The problem of shape invariant object description can be similarly formulated. Specifically, such an approach to the problem of parametrizing a non-convex closed surface is promising.

6 Appendix.

Generalized Delta Rule.

The Generalized Delta Rule is used to compute the interconnection weights in a multilayer neural network. It minimizes the square of the total error over all output nodes and all patterns.

$$E = \frac{1}{2} \sum_{p} \sum_{j} (y_{j,p} - t_{j,p})^2$$

where y is actual output and t the desired output (target). The weight change formula is

$$\Delta w_{i,j} = -\epsilon \frac{\partial E}{\partial w_{ij}}$$

where ϵ is the so called learning rate and w_{ij} is connection weight between nodes i and j in a given layer. In effect, this is a gradient descent algorithm. The partial derivatives $\frac{\partial E}{\partial w_{ij}}$ are obtained in the following manner. Starting with the output layer, the change in the total error due to the change in a single output (for a single pattern) is

$$\frac{\partial E}{\partial y_j} = y_{jp} - t_{jp}$$

Each node (neuron) in the network is characterized by a nonlinear relation between the inputs and the single output,

$$y_j = \frac{1}{1 + \exp(-x_j)}$$

where x_j is the total input into node j, given by $x_j = \sum_i y_i w_{ij}$ with the y_i's denoting outputs of an earlier layer. Using the chain rule it is now possible to express the change in error due to the inputs and subsequently to the weights as a function of the outputs and observed output error.

$$\frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dx_j}$$

$$= \frac{\partial E}{\partial y_j} y_j(1 - y_j)$$

then $$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial x_j} \cdot \frac{\partial x_j}{\partial w_{ij}}$$

$$= \frac{\partial E}{\partial x_j} y_i$$

To calculate the necessary weight modifications in layers other then the output, the change in total error due to outputs of nodes in those layers must be obtained. This is simply,

$$\frac{\partial E}{\partial y_i} = \frac{\partial E}{\partial x_j} \cdot \frac{\partial x_j}{\partial y_i}$$

$$= \frac{\partial E}{\partial x_j} \cdot w_{ij}$$

This completes the description of the algorithm.

7 References.

1. Ackley, D.H., G.E. Hinton and T.J. Sejnowski, *A Learning Algorithm for Boltzmann Machines*, **Cognitive Science 9**, 1985, pp. 147-169.

2. Albert, A., **Regression and the Moore-Penrose Pseudoinverse**, Academic Press, New York, 1972.

3. Amari, S.I., *Neural Theory of Association and Concept-Formation*, **Biological Cybernetics 26**, 1977, pp. 175-185.

4. Andenberg, M.R., **Cluster Analysis for Applications**, Academic Press, New York, 1973.

5. Anderson, J.A., J.W. Silverstein, S.T. Ritz and R.S. Jones, *Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model*, **Psychological Review 84**, 1977, pp. 413-451.

6. Bajscy, R. and L. Lieberman, *Texture Gradient as a Depth Cue*, **Computer Graphics and Image Processing 5**, March 1976, pp. 52-67.

7. Ballard, D.H. and C.M. Brown, **Computer Vision**, Prentice Hall, New Jersey, 1982.

8. Barsky, B.A., *A Description and Evaluation of Various 3D Models*, **IEEE Computer Graphics and Applications 4**, Jan. 1984, pp. 38-52.

9. Beck, J.M., R.T. Farouki and J.K. Hinds, *Surface Analysis Methods*, **IEEE Computer Graphics and Analysis 6**, Dec. 1986, pp. 18-36.

10. Besl, J.P., *Geometric Modeling and Computer Vision*, **Proceedings, IEEE 76**, Aug. 1988, pp. 936-957.

11. Besl, P. and R. Jain, *Intrinsic and Extrinsic Surface Characteristics*, **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, San Francisco, June 1985, pp. 226-233.

12. Besl, P. and R. Jain, *Range Image Understanding*, **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, San Francisco, June 1985, pp. 226-233.

13. Bezier, P., *Mathematical and Practical Possibilities of UNISURF*, in **Computer-Aided Geometric Design**, R.E. Barnhil and R. Riesenfeld, eds., Academic Press, New York, 1974.

14. Bhanu, B., *Representation and Shape Matching of 3D Objects*, **IEEE Transactions on Pattern Analysis and Machine Intelligence 6**, May 1984, pp. 340-351.

15. Bolle, R.M. and D.B. Cooper, *Bayesian Recognition of Local 3-D Shape by Approximating Image Intesity Functions with Quadric Polynomials*, **IEEE Transactions on Pattern Analysis and Machine Intelligence 6**, July 1984, pp. 418-429.

16. Boor, C. de, **A Practical Guide to Splines**, Springer Verlag, New York, 1978.

17. Brown, C.M., *Some Mathematical and Representational Aspects of Solid Modeling*, **IEEE Transactions on Pattern Analysis and Machine Intelligence** 3, July 1981, pp. 444-453.

18. Brown, M.K., *The Extraction of Curved Surface Features with Generic Range Sensors*, **The International Journal of Robotics Research** 5, Spring 1986, pp. 3-18.

19. Carmo, M.P. do, **Differential Geometry of Curves and Surfaces**, Prentice Hall, New Jersey, 1976.

20. Casale, M.S. and E.L. Stanton, *An Overview of Analytic Solid Modeling*, **IEEE Computer Graphics and Applications** 5, Feb. 1985, pp. 45-56.

21. Coons, S.A., *Surface Patches and B-Spline Curves*, in **Computer-Aided Geometric Design**, R.E. Barnhill and R. Riesenfeld, eds., Academic Press, New York, 1974.

22. Eichmann, G. and M. Jankowski, *Fourier Shape Descriptors of Multi-Dimensional Closed Volumes*, **SPIE Proceedings 659**, Sept. 1985.

23. Eichmann, G., M. Stojancic and M. Jankowski, *Shape Description with an Associative Memory*, **SPIE Proceedings 638**, April 1986.

24. Eichmann G. and M. Jankowski, *Surface Representation and Shape Description of Solid Bodies*, **OSA, Second Topical Meeting on Machine Vision**, March 1987.

25. Farouki, R.T. and J.K. Hinds, *A Hierarchy of Geometric Forms*, **IEEE Computer Graphics and Applications** 5, May 1985, pp. 51-78.

26. Faugeras, O.D. and M. Hebert, *The Representation, Recognition, and Locating of 3D Objects*, **The International Journal of Robotics Research** 5, Fall 1986.

27. Ferguson, J., *Multivariable Curve Interpolation*, **J. Association of Computing Machinery** 11, April 1964, pp. 221-228.

28. Forrest, A., *On Coons' and Other Methods for the Representation of Curved Surfaces*, **Computer Graphics and Image Processing** 1, Dec 1972, pp. 341-359.

29. Fu, K.S., Editor, **Digital Pattern Recognition**, Springer Verlag, New York, 1980.

30. Geman, S. and D. Geman, *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*, **IEEE Transactions on Pattern Analysis and Machine Intelligence** 6, 1984, pp.721-741.

31. Gibson, J.J., **The Scenes Considered as Perceptual Systems**, Houghton Miflin, Boston, M. A.,1966.

32. Hopfield, J.J., *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, **NAS Proceedings** 79, 1982, pp. 2554-2558.

33. Horn, B.K.P., *Extended Gaussian Images*, **IEEE Proceedings** 72, Dec. 1984, pp. 1671-1686.

34. Horn, B.K.P., **Robot Vision**, MIT Press, Cambridge, Massachusets, 1986.

35. Ikeuchi, K., *Recognition of Objects Using the Extended Gaussian Image*, **Proc. IJCAI-81**, Vancouver, Canada, Aug. 1981, pp. 595-600.

36. Jackins, C.L. and S.L. Tanimoto, *Oct-Trees and Their Use in Representing Three-Dimensional Objects*, **Computer Graphics and Image Processing 14**, Nov. 1980, pp. 249-270.

37. Jarvis, R.A., *A Computer Vision and Robotics Laboratory*, **IEEE Computer**, June 1982, pp. 8-24.

38. Jarvis, R.A., *A Perspective on Range Finding Techniques for Computer Vision*, **IEEE Transactions on Pattern Analysis and Machine Intelligence 5**, March 1983, pp. 122-139.

39. Kaplan, W., **Advanced Calculus**, Addison Wesley, Reading, Mass., 1952.

40. Kashyap, R.L. and R. Chellapa, *Stochastic Models for Closed Boundary Analysis: Representation and Reconstruction*, **IEEE Transactions on Information Theory 27**, Sept. 1981, pp. 627-637.

41. Kohonen, T., **Self-Organization and Associative Memory**, Springer Verlag, New York, 1984.

42. Levine, M.D., D.A. O'Handley, and G.M. Yagi, *Computer Determination of Depth Maps*, **Computer Graphics and Image Processing 2**, Oct. 1973, pp. 131-150.

43. Levine, M.D., **Vision in Man and Machine**, McGraw Hill, New York, 1985.

44. Marr, D., **Vision**, W.H. Freeman and Company, New York, 1982.

45. Marr, D. and H.K. Nishihara, *Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes*, **Proceedings, Royal Society of London B200**, 1978, pp. 269-294.

46. Marr, D. and T. Poggio, *Cooperative Computation of Stereo Disparity*, **Science 194**, 1976, pp. 283-287.

47. McKee, J.W. and J.K. Aggarwal, *Computer Recognition of Partial Views of Curved Objects*, **IEEE Transactions on Computers 26**, Aug. 1977, pp. 790-800.

48. Minsky, M.L. and S.A. Papert, **Perceptrons**, MIT Press, Cambridge, Massachusets, 1988.

49. Mortenson, M.E., **Geometric Modeling**, John Wiley & Son, New York, 1985.

50. O'Neill, B., **Elementary Differential Geometry**, Academic Press, Florida, 1966.

51. Oshima, M. and Y. Shirai, *Object Recognition Using Three-Dimensional Information*, **IEEE Transactions on Pattern Analysis and Machine Intelligence 5**, July 1983, pp. 353-361.

52. Pavlidis, T., *Algorithms for Shape Analysis of Contours and Waveforms*, **IEEE Transactions on Pattern Analysis and Machine Intelligence 2**, July 1980, pp. 301-312.

53. Perkins, W.A., *A Model-Based Vision System for Industrial Parts*, **IEEE Transactions on Computers 27**, Feb. 1978, pp. 126-143.

54. Persoon, E. and K.S. Fu, *Shape Discrimination Using Fourier Descriptors*, **IEEE Transactions on Sytems, Man and Cybernetics 7**, March 1977, pp. 170-179.

55. Requicha, A.A.G., *Representations for Rigid Solids: Theory, Methods, and Systems*, **Computing Surveys 12**, Dec. 1980, pp. 437-464.

56. Requicha, A.A.G. and R.B. Voelcker, *Constructive Solid Geometry*, **Tech. Memo**, Production Automation Project, Univ. of Rochester, Rochester, New York, Nov. 1977.

57. Rogers, D.F. and J.A. Adams, **Mathematical Elements for Computer Graphics**, McGraw Hill, New York, 1976.

58. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning Internal Representations by Error Propagation*, **Parallel Distributed Processing 1**, D.E. Rumelhart and J.L. McClelland (Eds.), MIT Press, Cambridge, MA, 1986.

59. Sadjadi, F.A. and E.L. Hall, *Object Recognition by 3D Moment Invariants*, **Proceedings, IEEE Conference on Pattern Recognition and Image Processing**, 1979, pp. 327-336.

60. Schudy, R.B. and D.H. Ballard, *A Computer Model for Extracting Moving Heart Surfaces from Four-dimensional Cardiac Ultrasound Data*, **Proceedings 6th Conference on Computer Applications in Radiology and Computer Aided Analysis of Radiological Images**, 1979, pp. 366-376.

61. Sederberg, T.W., D.C. Anderson, and R.N. Goldman, *Implicit Representation of Parametric Curves and Surfaces*, **Computer Vision, Graphics, and Image Processing 28**, 1984, pp. 72-84.

62. Shani, U., *A 3D Model-Driven System for the Recognition of Abdominal Anatomy from CT Scans*, **Proceedings, 5th International Joint Conference on Pattern Recognition**, Miami Beach, Florida, Dec. 1980, pp. 585-591.

63. Soroka, B.I., *Generalized Cylinders from Parallel Slices*, **Proceedings, IEEE Computer Society Conference on Pattern Recognition and Image Processing**, Chicago, Aug. 1979, pp. 421-426.

64. *Special Issue on Neural Networks*, **Applied Optics**, Vol. 26, Dec. 1987.

65. Srihari, S.N., *Representation of Three-Dimensional Digital Images*, **Computing Surveys 13**, Dec. 1981, pp. 400-424.

66. Tiller, W., *Rational B-Splines for Curve and Surface Representation*, **IEEE Computer Graphics and Applications 3**, Sept. 1983, pp. 61-69.

67. Turney, J.L., T.N. Mudge, and R.A. Volz, *Recognizing Partialy Occluded Parts*, **IEEE Transactions on Pattern Analysis and Machine Intelligence 7**, July 1985, pp. 410-421.

68. Voelcker, R.B. and A.A.G. Requicha, *Geometric Modeling of Mechanical Parts and Processes*, **IEEE Transactions on Computers 10**, Dec. 1977, pp. 48-57.

69. Weatherburn, C.E., **Differential Geometry of Three Dimensions**, Cambridge University Press, 1927.

70. Yakimovsky, Y. and R. Cunningham, *A System for Extracting Three-Dimensional Measurements from a Stereo Pair of Cameras*, **Computer Graphics and Image Processing** 7, April 1978, pp. 195-210.

71. Zahn, C.T. and R.Z. Roskies, *Fourier Descriptors for Plane Closed Curves*, **IEEE Transactions on Computers** 21, March 1972, pp. 269-281.