Geometrical Aspects of Interpreting Images as a Three-Dimensional Scene

TAKEO KANADE

Invited Paper

Abstract – One of the fundamental aspects of vision is threedimensional geometry referring to the relationship between the scene which is depicted and the images which are projections of that scene. This paper presents computational theories which deal with geometrical aspects of interpreting images as a three-dimensional (3D) scene. The presentation covers three levels of descriptions: micro-surface level, volumetric-object level, and scene level. The micro-surface level concerns the relationship between surface orientations of small surface patches and image properties. Theories are presented which provide computational constraints for recovering surface orientations from geometrical image properties, such as symmetry, texture, shadows, and perspective distortions. The volumetric-object level handles shape representations of primitive objects: the relationship between 3D shape of an object and its projections onto images must be understood. The generalized cylinder is a popular volumetric shape representation in vision. We will discuss its formal properties. Finally, the scene level aims at constructing and maintaining the 3D description of the whole scene. We will present the Incremental 3D Mosaic system under development at CMU, which incrementally constructs the total 3D scene description of the task area from aerial photographs.

I. Introduction

VISION refers to sensing and interpreting of environments by means of images. It is widely recognized that vision is an important capability in enabling robots to behave intelligently and flexibly. Vision involves many aspects: appropriate sensors or cameras must be chosen to pick up the relevant physical phenomena; photometric and radioinetric'aspects must be considered in order to interpret the image intensity values in a sensible manner in accordance with the material reflection, illumination. and sensor characteristics; iniage-processing techniques must be developed to analyze images and to detect interesting events; algorithm efficiency is important for implementation. However, one of the fundamental aspects is three-dimensional (3D) geometry referring to the relationship between the scene and the images. In fact. if we consider that images are projections of a scene, vision. after all, amounts to backprojecting the images to the scene or to recovering the scene from the images.

It seems that successful applications of vision in robotics so far are limited to such tasks and techniques as optical detection inspecting two-dimensional (2D) patterns (e.g., PC board inspection), locating objects, and binary vision. Also we can note that the methods employed are mostly *ad hoc* and tuned to the particular tasks by careful engineering. For example, the task of locating and recognizing objects on a table is solved as region analysis of binary images (topological and geometrical properties of a region. such as connectivity. hole. and corners). This task is made possible by setting up a standard viewing angle (most typically overhead view) to convert the task into a 2D problem. and by using careful lighting to ensure high-contrast shadow-free images which provide a clean silhouette of objects after binarization. This type of technique used in today's robotics vision can be said to be oriented toward detection of events by 2D image analysis. While this technique will remain useful for many applications, another orientation we must seek as a goal of vision research is the derivation of 3D scene descriptions. A crucial difference between 2D image analysis and 3D scene description can be understood by considering the task of robot navigation using vision: the 3D description of the environment. rather than segmentation of its image, is necessary to enable the robot to move through the environment.

In order to achieve such a capability, which allows derivation of 3D descriptions of the environment, new directions in computer vision research are currently emerging, most notably in the Image Understanding community [1]. This research pursues more systematic and computational approaches than those of conventional research. In this paper, we will limit our discussion to computational theories which deal with geometrical aspects of interpreting images as a 3D scene. Also, this presentation is mostly based on the work done at Carnegie-Mellon University, and is not intended to be a survey.

The geometrical aspects of vision includes three levels of descriptions: micro-surface level, volumetric-object level, and scene level. The micro-surface level concerns the relationship between surface orientations of small surface patches and image properties. For example, a typical problem is how to estimate the slant and tilt of a surface from texture patterns. Theories are presented which provide constraints for recovering surface orientations from geometrical image properties, such as symmetry, texture, shadows, and perspective distortions, together with the gradient space which is a convenient tool to represent surface orientations. These theories are local in that they provide surface orientations based on the properties of a small portion in the image.

The volumetric-object level handles shape representations of primitive objects. The surface orientations obtained from the local analysis of an image must be integrated over a region to produce a shape description of an object, for which more global constraints must be imposed, such as smoothness, type of surfaces (planer, quadratic, etc), or type of volumes. To obtain such global constraints computationally, we must first establish the relationship between 3D shape representations and their projections onto the image. The generalized cylinder has been a popular volumetric shape representation in vision. We

Manuscript received December 15, 1982; revised February 17, 1983.
The author is with the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213.

0018-9219/83/0700-0789 $01.00

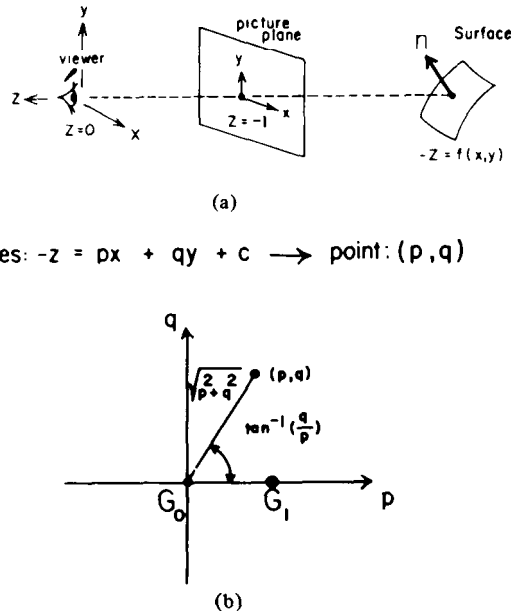


Fig. 1. (a) Imaging geometry including the surface, the picture, and the viewer. (b) Mapping of planes to a gradient.

will discuss its formal properties: specifically about the silhouette of a solid of revolution, which is an important class of the generalized cylinder.

Finally, the scene level aims at constructing and maintaining the 3D description of the whole scene. At this level, the main issues are how to combine various inputs of 3D information into a consistent scene description and how to revise it when new information is obtained. As an example of the systems that deal with this level of descriptions, we will present the Incremental 3D Mosaic system under development, which incrementally construct the total **3D** scene description of the task area from aerial photographs.

II. Surface Orientations from Image Properties

Historically, there are several annoying image phenomena which often cause vision programs to fail in image analyses. Some of these phenomena are of a geometrical nature: among them are deformed shape due to slanted views, perspective distortions, textural patterns, and shadow. That they are annoying may still be true in most applications, but it has begun to be understood that these are actually rich sources of information about the shape. An interesting class of theories has been developed to handle these sources. Their common nature is that they provide constraints on local surface orientations. This section will introduce such theories and tools which allow computational use of constraints obtained from geometrical properties observable in the image.

A. Coordinate System and the Gradient Space

Coordinate System: Let us first define the coordinate system which we will use in this paper. As shown in Fig. 1, the x and y axes in the scene are aligned in the image (x horizontal, y vertical), and the z axis points ***towards*** the viewer (i.e., a right-handed coordinate system) (Fig. 1). The ***eye*** (center of lens) is at the origin (0, 0, 0), and the ***image plane*** is $z = -1$: the focal plane is $z = 1$, but it is rotated around the origin to this image plane to preserve the sense of "up," "down," "left," and "right" from the scene.

In ***orthographic projection,*** the scene point (x, y, z) is mapped onto the image point (x, y). Conversely, the image point (x, y) can correspond to the set of scene points (x, y, a) for all values of a.

In ***perspective projection,*** the scene point (x, y, z) is mapped onto the image point $(-x/z, -y/z)$: the image point is the point at which a line through the origin (the eye) and the scene point (x, y, z) intersects the image plane. The unit of measure in the coordinate system is the focal length of the camera lens. An image point (x, y) corresponds to the set of scene points $(ax, ay, -a)$ for all values of a.

Gradient Space: Many image properties are related to the differential properties of 3D shape. For example, light reflection is mainly governed by the surface orientation relative to the orientation of illumination and the line of sight. The ***gradient space,*** introduced and popularized by Huffman [2] and Mackworth [3], provides a convenient way to represent surface orientations in relation to the imaging geometry. Let us first define it. In our coordinate system defined in Fig. 1, consider a surface

$$-z = f(x, y). \tag{1}$$

The gradient space is defined by (p, q) where

$$p = \frac{\partial f}{\partial x} \qquad q = \frac{\partial f}{\partial y}. \tag{2}$$

That is, p and q are the rate of change in depth on the surface along the x and y direction. We can easily see that $(p, q, 1)$ has the direction of the surface normal pointing toward the viewer; or $(p/\sqrt{p^2+q^2+1}, q/\sqrt{p^2+q^2+1}, 1/\sqrt{p^2+q^2+1})$ is the unit surface normal. Gradients are constant over a planar surface, and the gradient (p, q) corresponds to a set of parallel planes

$$-z = px + qy + c \tag{3}$$

where c is arbitrary.

The drawback of the gradient space is that the value of (p, q) approaches infinity as the surface orientation becomes tangential such as along occluding contours. There are other spaces that we can use for representing surface orientations: such as the stereographic space [4] or the Gaussian sphere [5]. Because of its simplicity we will use the gradient space throughout this paper.

Connect-Edge Relation Under Orthography: Under orthographic projection, the gradients of planes and the image line on the image have an interesting relationship. Referring to Fig. 2, let two planes P_1 and P_2 intersect in the space and let the intersection edge be depicted as a line l in the image. Then the line in the gradient space connecting the corresponding gradients $G_i = (p_i, q_i)$ is perpendicular to the line l. This can be seen in the following way. The normals of the two surfaces have directions $n_1 = (p_1, q_1, 1)$ and $n = (p_2, q_2, 1)$. Their cross product $n_1 \times n_2$ represents the direction of the intersection edge in the space. Since we assume orthographic projection, the direction of the line l in the image is given by the x and y components of this cross product; that is, $(q_1 - q_2, p_2 - p_1)$. This vector is obviously perpendicular to the vector $G_1 - G_2 = (p_1 - p_2, q_1 - q_2)$ which connects G_1 and G_2. Moreover, if the intersection edge is convex viewed from the viewer, then the positional order of G_1 and G_2 is the same as the order of the regions in the picutre corresponding to P_1 and P_2; if the edge is concave, the order is reversed.

This property of the gradient space has been extensively used

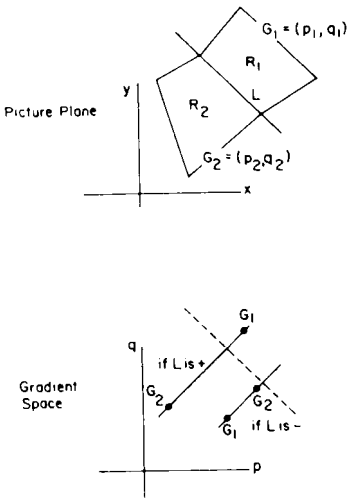


Fig. 2. Properties of dual lines. If two planes meet and the intersection edge is projected as a picture line L, then the gradients of the two planes are on a gradient-space line which is perpendicular to L.

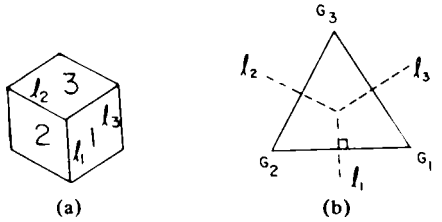


Fig. 3. (a) A drawing of a "cube." (b) The constraints on the gradients of the three surfaces due to their interconnection: the gradients form a triangle in the gradient space whose shape and orientation should be as shown, but the location and the size are arbitrary. This means that (a) need not he a cube, but can correspond to many shapes.

for understanding line drawings of planar surfaced objects in orthographic projection, such as polyhedra [3] and the Origami-world objects [6]. As the simplest example, if the drawing of Fig. 3(a) is interpreted as depicting a convex corner (i.e., the image lines l_1, l_2, and l_3 are interpreted as convex), the gradients G_1, G_2, G_3 of the three surfaces must form a triangle in the gradient space shown in Fig. 3(b). Notice that the direction and order from G_1 to G_2, for example, is constrained by that of l_1. Thus the shape and orientation of this triangle is constrained. but the location and the size are not. This constraint exactly represents all the possible shapes that Fig. 3(a) can depict.

B. Skew Symmetry

2D shapes of regions (i.e., projection of surfaces) in the image convey information about 3D shape of an object in the scene. This is illustrated by the simple line drawings in Fig. 4. Fig.4(a) and (b) are topologically tlie same and the slight difference in the 2D shape of the lower two regions results in the viewer seeing one as a cube and the other as a trapezoidal block. In fact, in terms of surface orientations of tlie three surfaces involved, Fig. 4(a) and (b) has exactly the same constraints as those shown in Fig. 3(b). Furthermore. it is interesting to observe that even such simple figures as Fig. 4(c) and (d) already invoke the perception of surface orientations. Certain geometrical properties should be the source of that perception. These phenomena may have been often studied in their psychological

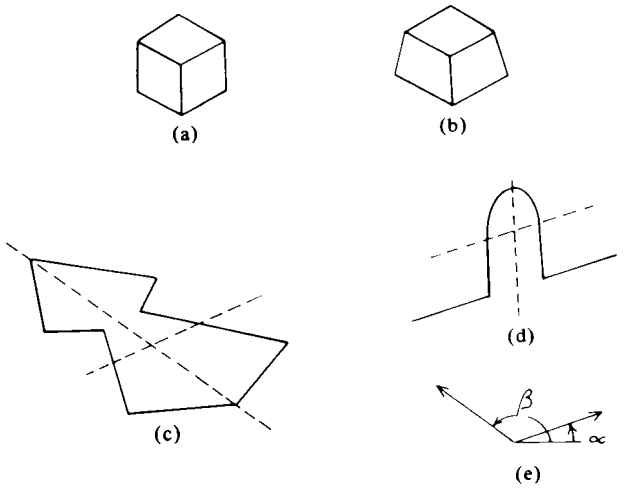


Fig. 4. (a) (b) Simple line drawings: (a) "cube"; (b) "trapezoidal block"; (c) (d) skewed symmetry; (e) axes of the skewed symmetry of (c).

aspects, but until recently very little has been formulated in a manner usable for machine vision.

In a previous paper [7], the author demonstrated a systematic method to recover 3D shapes from a single view by mapping image properties into shape constraints. The 2D shape property of Fig. 4(c) and (d) is one of the properties that were studied. It was named ***skewed symmetry,*** because such figures are interpreted as symmetric figures viewed obliquely. In other words, whereas in the usual symmetry reflective correspondences are found along the direction perpendicular to the symmetry axis, in the skewed symmetry they are found along the direction not necessarily perpendicular to the axis, but at a fixed angle to it. Formally, such shapes can be defined as 2D affine transforms of real symmetries.

There is a good body of psychological experiments which suggests that human observers can perceive surface orientations from figures with this property. This is probably because such qualitative symmetry in the image is often due to real symmetry in the scene.

Now let us associate the following assumption with this image property: ***"A skewed symmetry depicts a real symmetry viewed from some unknown view angle."*** Note that the converse of this assumption is always true in orthographic projection. A skewed symmetry defines two directions: let us call them the skewed-symmetry axis and the skewed-transverse axis. As shown in Fig. 4(e) that depicts the two axes for Fig. 4(c), let us denote their directional angles in the picture by α and β, respectively. Further let $\boldsymbol{C} = (p, q)$ be the gradient of the plane which includes tlie skewed symmetry. In general, the 2D unit vector in the direction γ is $\boldsymbol{e}_\gamma = (\cos\gamma, \sin\gamma)$. From (3), the 3D vector corresponding to $\mathbf{e}$ on the plane in the space is given as

$$\boldsymbol{u}_\gamma = (\cos\gamma, \sin\gamma, -(p\cos\gamma + q\sin\gamma)). \quad (4)$$

The assumption about the skewed symmetry demands that the two 3D vectors $\boldsymbol{u}_\alpha$ and $\boldsymbol{up}$ be perpendicular in the 3D space: that is, thier inner product vanishes, $\boldsymbol{u}_\alpha \cdot \boldsymbol{up} = 0$ or

$$\cos(\alpha - \beta) + (p\cos\alpha + q\sin\alpha)(p\cos\beta + q\sin\beta) = 0. \quad (5)$$

By rotating the p-q coordinates by the amount $\lambda = (\alpha + \beta)/2$ into the p'-q' coordinates so that the new p'-q' axes are the bisectors of the angle made by the skewed-symmetry and

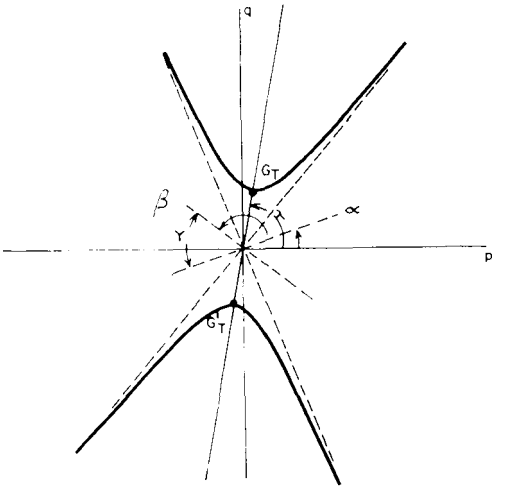


Fig. 5. The hyperbola of (5) corresponding to Fig. 4(e). The axis of the hyperbola is the bisector of the obtuse angle made by α and β. The asymptotes make the same angle as the acute angle made by α and β. The tips or vertices G_T and G_T' of the hyperbola represent special orientations with interesting properties. Especially, since they are closest to the origin of the gradient space, and since in general the distance $\sqrt{p^2+q^2}$ from the origin to a gradient (p. 9) represents the magnitude of the surface slant, G_T and G_T' correspond to the least slanted orientations that can produce the skewed symmetry in the picture from a real symmetry in the scene.

skewed-transverse axes, it is easy to show that (5) represents a hyperbola in the gradient space shown in Fig. 5. That is, the skewed symmetry defined by α and β in the picture can be a projection of real symmetry ***if and only if*** its surface gradient (p,q) is on this hyperbola. The skewed symmetry thus imposes a one-dimensional family of constraints on the underlying surface orientation. Fig. 6 illustrates how this skewed symmetry constraint can be used to recover the shape of "cube" from the image: the constraint shown in Fig. 3(b) was not strong enough to uniquely determine the shape, but the additional constraint from the skewed symmetry of the regions enables us to determine the shape of the cube.

The same approach was extended to other properties: parallel lines, affine-transformable patterns, and textures [8]. We can summarize the assumptions used in these cases: ***regular properties observable in the picture do not occur by accident, but are projections of some preferred corresponding 3D properties.*** Fig. 7 lists instances of this principle of ***nonaccidental image regularity*** and the corresponding theories. Since the mapping from the picture domain to the scene domain is one-to-many (ambiguous), we need to rely on these types of general assumptions which result in useful computational constraints in vision.

C. Texture

Perception of depth and surface from texture gradient has been studied by psychologists. Kender [9] developed a very powerful computational paradigm for deriving shape from texture . His central idea is to use a normalized texture property map (NTPM), which is again for mapping image features (texture in this case) into constraints on surface orientations.

Let us use again a simple example to show the NTPM for length. Suppose we have a texture pattern shown in Fig. 8, which is made up of line segments with two orientations: the horizontal ones with length L_2 and the diagonal ones (45°)

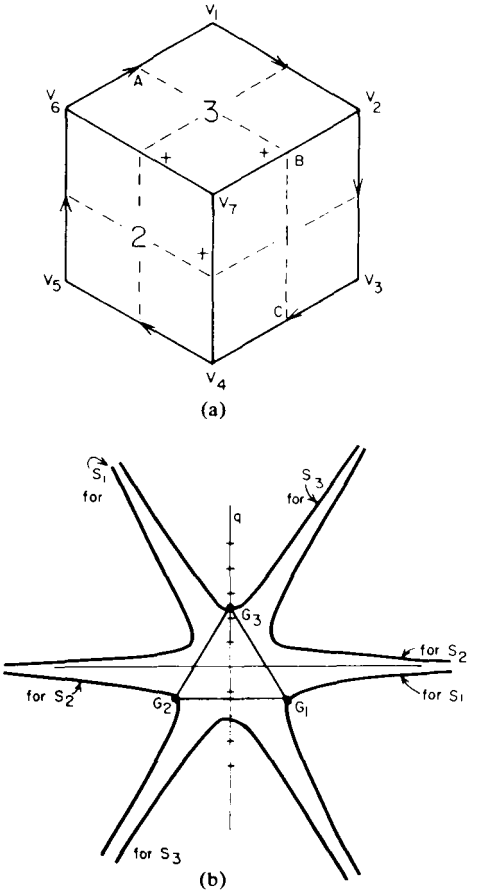


Fig. 6. (a) A simple labeled line drawing: the dotted linesshow the axes of skewed symmetry. (b) The hyperbolas shown correspond to the skewed symmetries of the three regions. The problem is thus how to place the triangle of Fig. 3(b), by translation and scale change, so that each vertex is on the corresponding hyperbola. The location shown is proven to be the only position, and the resultant shape is a cube.

	Scene
Parallel in 2D	Parallel in 3D
Skewed symmetry	Symmetry
Matched T	Interruption by surface
Affine-transformable patterns	Similar or congruent patterns
Similar edge profile	Similar physical edge property
High/Low correlation of intensity across an edge	Shadow edge/Occluding edge
Gradient in	Uniformity in
Spacing	Spacing
Length	Length
etc.	

Fig. 7. Instances of the principle of nonaccidental image regularity. Notice that the rules are mostly true when going from the scene to the picture, but the other direction is heuristic.

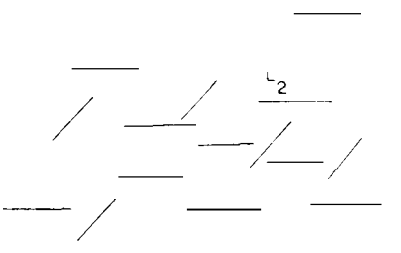


Fig. 8. A texture image of line segments. When we assume that all the line segments are on a plane surface and that their real lengths are the same, what orientations are possible?

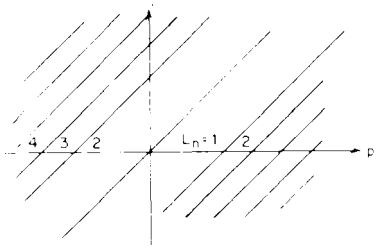


Fig. 9. A normalized texture property map (NTPM) for line length, $L_N = L(45^\circ, 1)$.

with length L_1. Consider, in general, a line segment in the image whose direction is of angle γ with the x axis and whose length is l. If that line segment is on a plane whose gradient is (p, q), what is the real length of the 3D line in the scene? From (4), $l\boldsymbol{u}_\gamma$ is the corresponding 3D vector, and its 3D length is

$$L(\gamma, l) = l|\boldsymbol{u}_\gamma| = l\sqrt{1 + (p\cos\gamma + q\sin\gamma)^2}. \quad (6)$$

$L(\gamma, l)$ is the NTPM for a line segment with slopey and length l. As with the reflectance map for shape-from-shading [10], we can represent (6) as a set of contours in the gradient space, $L(\gamma, l) = const$, each corresponding to such (p, q)'s that the 3D lines which are on the surface and which will be projected onto the image as a line of length l and angle γ are of length $const$. Fig. 9 shows the NTPM for $\gamma = 45^\circ$ and $l = 1$. In general. the NTPM is a function of surface orientations, and its value represents a scene property (e.g., 3D line length) that the scene constituent in that surface orientation should take if the observed image property (e.g., 2D line length) comes from it. In other words, it represents ***deprojected*** scene properties from image.

In Fig. 8, let us assume that all the line segments are on the same length in the space. The possible orientations of the surface can then be obtained by

$$L(0^\circ, L_2) = L(45^\circ, L_1) \quad (7)$$

or by graphically intersecting the two NTPM surfaces. Interestingly. this represents a hyperbola in the gradient space, again, and thus does not determine the surface orientation uniquely, but can be used in combination with other constraints in the same way as we did for skewed symmetry. Though this is a very simple example, the same technique can be used to estimate the surface orientations. for such cases as walls of buildings in outdoor scenes or of mechanical parts with textures.

D. Shadow Geometry

Shadow gives good clues to the spatial relationship between objects and surfaces. Aerial photo interpreters make much use of it in estimating. for example, the height of the objects. Lowe and Binford [11] demonstrated the reconstructing of the shape of an airplane by using shadow information. They first paired shadow-making edges and cast-shadow edges on the ground by assuming the ground plane and the sun angle. The height of the airplane along the contour made of shadow-making edges was then estimated. and this gave the shape of the airplane.

Shafer and Kanade [12] investigated a general and compact gradient-space representation of geometrical constraints given by shadow. Fig. 10 shows a basic shadow problem. It consists of the parallel-light illuminator I. the shadow-making (occluding) surface S_o, and the shadow-cast surface S_s. The problem includes six parameters to be computed: the gradient

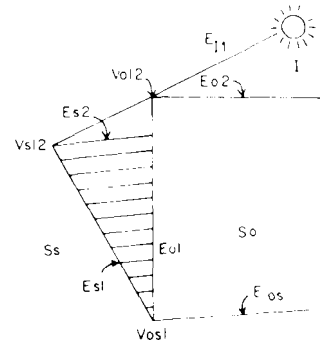


Fig. 10. Basic shadow problem: the suffixes are given to show the correspondences; for example, E_{o1} is a shadow making edge and E_{s1} is a corresponding shadow edge.

$\boldsymbol{C}$, $= (p_o, q_o)$ of S_o, the gradient of $\boldsymbol{G}_s = (p_s, q_s)$ of S_s, and the direction of illumination (p_I, q_I).

This problem can be studied by considering two other surfaces S_{I1} and S_{I2} (and their gradients $\boldsymbol{G}_{I1}$ and $\boldsymbol{G}_{I2}$), each of which includes a pair of shadow-making and cast-shadow edges: for example, E_{o1} and E_{s1} define S_{I1}. Note that S_{I1} and S_s make a concave edge along E_{s1}, and so do S_{I2} and S_s along E_{s2}. Now three constraints are provided from the basic shadow problem geometry: 1) the angle $\boldsymbol{C}$, - $\boldsymbol{G}_{I1}$ - $\boldsymbol{G}_s$ in the gradient space should be determined by the angle between E_{o1} and E_{s1} in the image; 2) the angle $\boldsymbol{G}_o$ - $\boldsymbol{G}_{I2}$ - $\boldsymbol{G}_s$ is determined by the angle between E_{o2} and E_{s2}; 3) the direction of the line L_{illum} (containing $\boldsymbol{G}_{I1}$ and $\boldsymbol{G}_{I2}$) is determined by the direction of E_{I1} (line containing the twovertices V_{o12} and V_{s12}). We would, therefore, expect that three parameters must be given in advance, and the other three can be computed from the shadow information. Fig. 11 shows a construction for the case in which the direction of illumination (actually the relative depth component of illumination vector—one parameter is given) and the orientation of shadow-cast surface $(\boldsymbol{G}_s)$ are known,

The basic shadow geometry provides three constraints, and thus three parameters have to be given by other means to solve the problem. It is interesting to compare thissituation with the situation without shadows an image which only depicts S_o and S_s intersecting along E_{os}. Here, there are four parameters ($\boldsymbol{G}_o$ and $\boldsymbol{G}_s$) to compute, and one constraint from the image (E_{os}), so three pieces of information are still needed in advance. With shadows, the same number of ***a priori*** parameters are needed, and in this sense, shadows do not make the problem easier. However, the geometrical significance of shadows is that they allow information about the light source to be used to solve the problem as a substitute for information about the surface orientations themselves. Practically, the information about the light source, such as the sun angle, is more available than the surface orientations in the scene in question.

E. Perspective Distortions and Gradient Space

The gradient space has proved to be a useful tool in analyzing image properties in orthographics. It also possesses useful and interesting properties in perspective projections, such as those developed and summarized by Mackworth [13] and Shafer, Kanade, and Kender [14]. The difference between the orthographic and perspective projections is that in perspective projection the backprojection from image to scene (i.e., projecting the image properties back to scene properties) is position-dependent: that is, the identical image feature found in an image corresponds to different scene properties depending on the

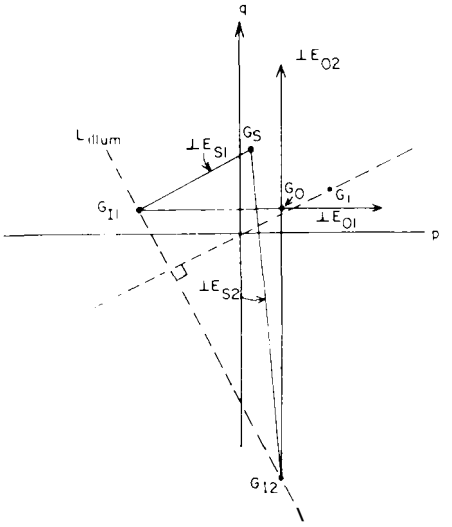


Fig. 11. The construction in the gradient space for the basic shadow problem. Given G_I and G_S, the computation of G_O proceeds as follows:

1) Draw the line parallel to E_{I1} through the origin. Plot the given G_I (it should be on this line). Let k be the distance from the origin to GI. Draw the line L_{illum} so that it is perpendicular to E_{I1}, opposite to G_I with respect to the origin, and at the distance of $1/k$ from the origin.

2) Plot G_S, which was given. Through this point, draw a line perpendicular to E_{S1}. Where it intersects L_{illum} must be G_{I1}. Through G_{I1}, draw a line perpendicular to E_{O1}. G_O must lie on this line.

3) From G_S, draw a line perpendicular to E_{S2}. Where it intersects L_{illum} will be G_{I2}. From there, draw a line perpendicular to E_{O2}. Since G_O must lie on this line, the intersection of this line with the final line from step 2) above must he G_O.

location of the feature in the image. The vanishing point of a line and the vanishing line of a plane are the fundamental notions in perspectives. Let us first see how they are related to the gradients.

In its original definition, the gradient (p, q) means a surface orientation whose surface normal vector is $(p, q, 1)$. In the following discussion, it is convenient to extend this notion and think of the gradient (p, q) as generally representing an orientation vector $(p, q, 1)$; that is, when we have a **3D** vector $\boldsymbol{A} = (Ax, Ay, Az)$, we can think of

$$\boldsymbol{G}_\Delta = \left(\frac{\Delta x}{\Delta z}, \frac{\Delta y}{\Delta z}\right)$$

as representing the direction of $\boldsymbol{A}$. (In [12], $\boldsymbol{G}_\Delta$ is called a vector gradient.)

Vanishing Point of a Line: Referring to Fig. 12, suppose a **3D** line in the scene is defined by $(x, y, z) + a(\Delta x, \Delta y, \Delta z)$ for all values of a, where (x, y, z) is a point on the line and $(Ax, \Delta y, Az)$ is a direction vector of the line. For any a, the corresponding point on the line is projected onto the image point $\boldsymbol{P}_a$

$$\boldsymbol{P}_a = \left(\frac{-x - a\Delta x}{z + a\Delta z}, \frac{-y - a\Delta y}{z + a\Delta z}\right).$$

As a grows larger, the image point $\boldsymbol{P}_a$ converges to some point $\boldsymbol{P}_\infty$ in the image if $Az \neq 0$ (that is, if the line is not parallel to the image plane).

$$\boldsymbol{P}_\infty = \lim_{a \to \infty} P_a = \left(\frac{-\Delta x}{\Delta z}, \frac{-\Delta y}{\Delta z}\right) = -\boldsymbol{G}_\Delta.$$

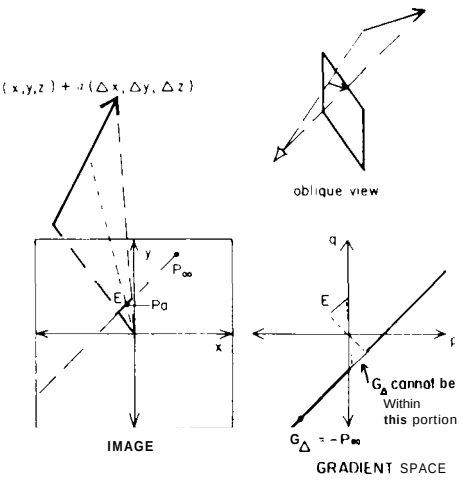


Fig. 12. Vanishing point P_∞ of a line and the vector gradient G_Δ.

The image point $\boldsymbol{P}_\infty = (x_\infty, y_\infty)$ is called the *vanishing point* of the line.

There are three interesting relationships between gradient and vanishing point:

1) $\boldsymbol{P}_\infty = -\boldsymbol{G}_\Delta$, that is, the location of vanishing point in the image $(x\text{-}y$ plane at $z = -1)$ for a **3D** line is the same as $-\boldsymbol{G}_\Delta$ (negative of the vector gradient) in the gradient space $(p\text{-}q$ plane).

2) Since $\boldsymbol{P}_\infty$ depends only on the direction vector $(Ax, \Delta y, Az)$, parallel lines have the same vanishing point. Conversely, each point in the image is the vanishing point for a family of parallel lines.

3) Suppose we have a finite-length **3D** line. It is depicted as a finite-length image line E. Obviously, the vanishing point $\boldsymbol{P}_\infty$ of the **3D** line must lie on the extension of image line E (Fig. 12). The vector gradient $\boldsymbol{G}_\Delta$ must, therefore, lie on the line $-E$ in gradient space, where $-E$ is the line which is 1) parallel to E, 2) at an equal distance from the origin as **E,** and **3)** on the opposite side of the origin from $\boldsymbol{E}$. However, as shown in Fig. 12, an important point here is that $\boldsymbol{G}_\Delta$ *cannot* lie within the portion directly corresponding to E but only within the portions corresponding to the extensions of E. This means that the longer the image line E, the narrower the range for $\boldsymbol{G}_\Delta$. In other words, in perspectives, by observing a longer part of the **3D** line, we do obtain more information about the orientation of the line, whereas in orthography the length of the line does not provide additional information about the orientation of the line.

Vanishing Line of a Surface: Suppose a surface S has a gradient $\boldsymbol{G}_S = (p_S, q_S)$. Since any vector $\boldsymbol{A} = (Ax, Ay, Az)$ on S must be perpendicular to its surface normal (p_S, q_S), we have a relationship $(p_S, q_S, 1) \cdot (\Delta x, Ay, Az) = 0$ or $\boldsymbol{G}_S \cdot \boldsymbol{G}_\Delta = -1$. Since the vanishing point of $\boldsymbol{A}$ is $\boldsymbol{P}_\infty = -\boldsymbol{G}_\Delta$

$$\boldsymbol{G}_S \cdot \boldsymbol{P}_\infty = 1$$

or

$$p_S x_\infty + q_S y_\infty = 1. \tag{8}$$

This equation means that given $\boldsymbol{G}_S$, then $\boldsymbol{P}_\infty = (x_\infty, y_\infty)$ moves on a line in the image. This line, the locus of vanishing points for all vectors on the surface, is called the *vanishing line* L_∞ of the surface (see Fig. **13).** The vanishing line L, has the following relationships with the surface gradient $\boldsymbol{G}_S$:

L_∞ is perpendicular to the line from $\boldsymbol{G}_S$ to the origin;

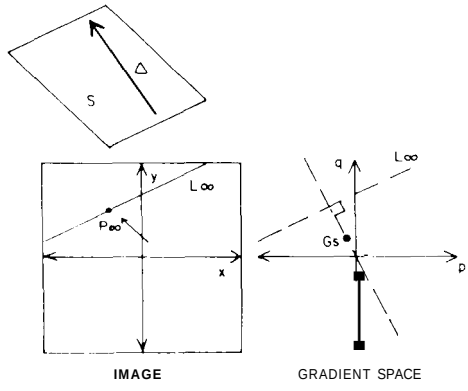


Fig. 13. Vanishing line L_∞ of a plane and the surface gradient G_S.

the distance from L_∞ to the origin, $1/\sqrt{p_S^2 + q_S^2}$, is the reciprocal of the distance from G_S to the origin;

L_∞ is on the same side of the origin as G_S.

Since L_∞ depends only on G_S, parallel surfaces have the same vanishing line. Thus each line in the image is the vanishing line for a family of parallel surfaces. Suppose E is a line in the image. There exists a family of parallel surfaces for which E is the vanishing line. These surfaces all have the same gradient (since they are all parallel), which might be called the ***vanishing gradient*** for E, denoted by G_E^∞. Let E be defined by the equation: $1 = ax + by$. Then by comparing this with (8), we see $G_E^\infty = (a, b)$. Thus for a line E in the image, we can determine the associated ***vanishing gradient*** G_E^∞ (the gradient of the surfaces for which E is the vanishing line). Mackworth [13] calls the surface through E and the origin the ***plane*** of ***interpretation*** of E; the "vanishing gradient" of a line is thus the gradient of its plane of interpretation.

This interrelationship among the surface orientation, the vanishing line. and the vanishing gradient is a useful concept for recovering surface orientations from perspective distortions in the image. Most of the image properties due to perspectives give constraints on the vanishing line. However, rather than trying to obtain the vanishing line directly, which tends to be less reliable and to be combinatorial, we can use the Hough transform technique in order to produce a histogram on the possible vanishing gradients. This type of method has been applied to images of convergent lines [15] and to the image of textures [16]. The Gaussian sphere representation, rather than the gradient space, can be also used for the same problem [17].

Connect-Edge Relation Under Perspective: We have seen in Section II-A that the gradient space has a nice property which relates the orientation of a connect edge in the image with the gradients of constituent surfaces which make the edge in the scene. We now consider what constraints we obtain from a connect edge in perspective projection.

Suppose a 3D edge $\mathbf{A} = (Ax, \Delta y, \Delta z)$ is the intersection of two surfaces with gradients G_1 and G_2. Then

$$G_1 \cdot G_\Delta = -1$$
$$G_2 \cdot G_\Delta = -1. \tag{9}$$

Now also suppose that $\mathbf{A}$ is visible in the image as a line E as shown in Fig. 14. By the definition of G_E^∞ and (8)

$$G_E^\infty \cdot G_\Delta = -1. \tag{10}$$

Combining (9) and (10), we obtain

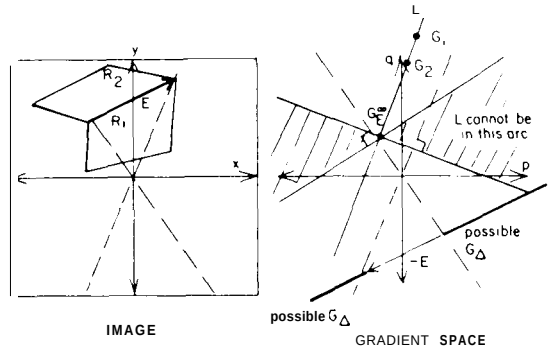


Fig. 14. The connect-edge rule under perspectives.

$$0 = (G_1 \cdot G_\Delta) - (G_E^\infty \cdot G_\Delta) = (G_1 - G_E^\infty) \cdot G_\Delta$$

so

$$(G_1 - G_E^\infty) \perp G_\Delta.$$

Similarly

$$(G_2 - G_E^\infty) \perp G_\Delta.$$

Thus the vectors $G_1 - G_E^\infty$ and $G_2 - G_E^\infty$ are parallel in the gradient space; that is, G_E^∞, G_1, and G_2 are collinear and on a single line L. Also, this line L, in the gradient space which contain G_1, G_2, and G_E^∞ must he perpendicular to the line through the origin and G_Δ. Fig. 14 illustrates these relationships.

There is further an interesting restriction on the position of the line L. It must pass through G_E^∞, which is completely determined by the location of the edge E in the image. Its slope depends on the gradient G_Δ, which, as described previously, must lie on the line $-E$, but not within the portion corresponding to the edge itself. This constrains the orientation of L, such that the line passing through the origin and perpendicular to L cannot pass through the inhibited portion of $-E$. Hence. the position and length of an edge in the image constrain the gradients of surfaces containing the corresponding vector in the scene.

This is the connect-edge relation under perspective; it is the perspective counterpart to the rule under orthography in Section II-A. As for the relative positions of G_1 and G_2 on the line L there is also a specific ordering relation depending on the convexity or concavity of the edge [9]; it is more complex than the case of orthography. since the line of sight (i.e., position in the image) must be taken into account under perspective projection.

III. Theory of Generalized Cylinders and Contour Analysis

A type of volume description often used in vision is the ***generalized cylinder.*** It was first introduced by Binford [18]. Informally, the generalized cylinders are the volume swept by a cross section moving along the axis (or the spine). An ordinary cylinder is defined with a circular disc and a straight line axis. In generalized cylinders, the cross section may be some planar figure and its size may vary as it moves along the axis by the rule called a sweep function. The spine may be a space curve. By a different combination of cross section, sweep rule, and spine curve, we can represent a fairly large class of shapes as generalized cylinders.

In spite of their popularity, little work has been done on the formal properties of generalized cylinders. Shafer and Kanade

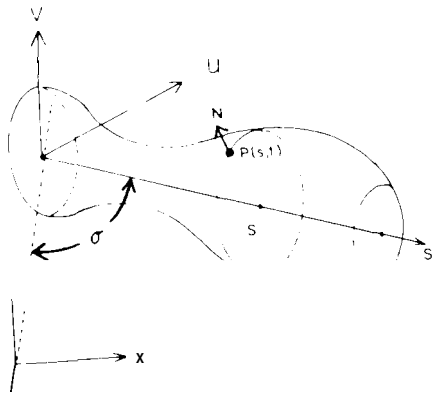


Fig. 15. The coordinate axes u-v-w of a Right Circular Straight Generalized Cylinder (RCSGC). the viewing angle σ, and the viewer's coordinates x-y-z.

[19] recently presented an analysis of the generalized cylinders based on a formal definition and classification.

A. Right Circular Straight Generalized Cylinders

A generalized cylinder can he specified by a three tuple $(\boldsymbol{A}, \boldsymbol{E}, \alpha)$: $\boldsymbol{A} = (x_A, y_A, z_A)(s)$ is the axis which is a space curve defined with the parameter s; α is the angle of inclination which defines the u-v plane of the cross section relative to the tangent to the axis at $\boldsymbol{A}(s)$; $\boldsymbol{E} = (u_E, v_E)(t, s)$ is a planar curve which specifies the cross section with parameter t on the u-v plane at $\boldsymbol{A}(s)$. While the generalized cylinder, in this general definition. can include a large class of shapes, interesting properties are found for its subclasses. One such useful subclass is a *Straight Homogeneous Generalized Cylinder (SHGC).* In the SHGC, the axis $\boldsymbol{A}(s)$ is a straight line, and the shape of the cross section remains the same while its size may vary. Thus the cross section $\boldsymbol{E}$ can be decomposed into two functions by $\boldsymbol{E}(s, t) = r(s)\boldsymbol{C}(t)$, where the *contour function* $\boldsymbol{C}(t) = (u_C, v_C)(t)$ describes the *shape* of the cross section. and the radius function $r(s)$ describes its *size*. Furthermore. when the cross section is always perpendicular to the axis (i.e., the angle of inclination $\alpha = \pi/2$) and also the cross section is circular. it is called a *Right Circular Straight Generalized Cylinder (RCSCC).* The contour function $\boldsymbol{C}(t)$ is given by

$$\boldsymbol{C}(t) = (\cos 2\pi t, \sin 2\pi t). \tag{11}$$

The RCSGC is an important subclass of generalized cylinder, because it represents a shape of revolution which is common in industry, and because it is sufficiently constrained to allow interesting analysis from imagery without additional knowledge sources. In fact. most of the generalized cylinders used in vision are in this subclass [20]-[24]. Although the formulations in the paper by Shafer and Kanade [19] are given for more general cases, most of our discussion in this paper will be for the case of RCSGC's.

For an RCSGC, we can define a natural u-u-s object-centered coordinate system as shown in Fig. 15: the s-axis is along the axis of the generalized cylinder and the u-v plane defines the plane of cross sections; the u-axis is chosen to provide a right-handed u-v-s coordinate system. Let us first derive expressions of points and surface normals of an RCSGC.

For any values s and t, the point $\boldsymbol{P}(s, t) = r(s)\boldsymbol{C}(t)$ on the surface of the RCSGC is represented in u-u-s coordinates (Fig. 15) as

$$\begin{aligned}\boldsymbol{P}(s, t) &= (u_C(t)\, r(s),\ v_C(t)\, r(s),\ s)_{uvs} \\ &= (r(s)\cos 2\pi t,\ r(s)\sin 2\pi t,\ s)_{uvs}. \end{aligned} \tag{12}$$

Thus the outward-pointing surface-normal vector $\boldsymbol{N}(s, t)$ at $\boldsymbol{P}(s, t)$ is therefore given by

$$\boldsymbol{N}(s, t) = \frac{\partial \boldsymbol{P}}{\partial t} \times \frac{\partial \boldsymbol{P}}{\partial s} = r(s)\left(\alpha \frac{du_C}{dt}, \frac{dv_C}{dt}, 0\right) \times \left(u_C(t)\,\alpha \frac{dr}{ds}, v_C(t)\frac{dr}{ds}, 1\right)_{uvs}$$

$$(s)\{\cos 2\pi t, \sin 2\pi t,$$

B. Contour Generators of RCSGC's

Suppose we are looking at a RCSGC's from a certain angle. A question arises about what part of the generalized cylinder is observed as its contour or silhouette. In our standard viewing coordinate shown in Fig. 1, the s-axis and the y-axis are. respectively, aligned horizontally and vertically on the image plane, and the z-axis points towards the eye (or camera). We will limit our discussion here to orthographic projection. in which a world point $(x, y. z)_{xyz}$ is mapped onto the image point $(x, Y)_{xy}$.

We can always assume that the axis of the generalized cylinder is horizontal and aligned with the x-axis in the image. because otherwise we can rotate the camera. That is. we can assume the s-axis of the generalized cylinder lies on the x-z plane. The angle from the viewer's line of sight (z-axis) towards the cylinder's axis (s-axis) is named the viewing angle σ. The relationship between the s-y-z viewer's coordinate frame and the u-u-s cylinder's coordinate frame now becomes as shown in Fig. 15.

The u-u-s and the x-y-z coordinates are related as

$$(u, v, s)_{uvs} = (u\cos\sigma + s\sin\sigma,\ v,\ -u\sin\sigma + s\cos\sigma - d)_{xyz}.$$

A point $\boldsymbol{P}(s, t)$ on an RCSGC is. therefore.

$$\begin{aligned}\boldsymbol{P}(s, t) &= (u_C(t)\, r(s),\ v_C(t)\, r(s),\ s)_{uvs} \\ &= (u_C(t)\, r(s)\cos\sigma + s\sin\sigma,\ v_C(t)\, r(s), \\ &\quad -u_C(t)\, r(s)\sin\sigma + s\cos\sigma)_{xyz}\end{aligned}$$

and its image under orthography is

$$\boldsymbol{I}(s, t) = (x_P, y_P)_{xy} = (u_C(t)\, r(s)\cos\sigma + s\sin\sigma,\ v_C(t)\, r(s))_{xy}$$

In general, suppose we project an object along the direction of a vector $\boldsymbol{F}$. The contours of those points on the object where the surface is tangent to the line of projection (i.e., $\boldsymbol{N}\cdot\boldsymbol{F} = 0$) are called *contour generators* [25]. A contour of the object in the image is the projection of a contour generator. (Of course, if the object is opaque, some part of the contour generators may be hidden from view.) Also, if $\boldsymbol{F}$ is the illumination direction, the contour generator is the self-shadow boundary.

In our viewing geometry shown in Fig. 15, the viewing direction $\boldsymbol{F}$ is given as

$$\boldsymbol{F} = (0, 0, 1)_{xyz} = (-\sin\sigma, 0, \cos\sigma)_{uvs}. \tag{14}$$

From (13), (14), and $\boldsymbol{N}\cdot\boldsymbol{F} = 0$, we obtain the equation of the contour generators for an RCSGC in the u-u-s coordinates as

follows:

$$\sin\sigma\cos 27rt + \cos\sigma\frac{dr}{ds} = 0$$

that is,

$$t = \frac{1}{2\pi}\cos^{-1}\left(-\cot\sigma\frac{dr}{ds}\right). \tag{15}$$

This equation represents t (the parameter of the cross section) as a function of s (the parameter of the axis) along the contour generator. In other words, at a particular point (s) on the axis, we can tell which points (t) on the contour constitute the contour generator. The points $P(s, t)$ along the contour generator can now be specified as $P_{CG}(s)$ by substituting t in (12) by (15)

$$P_{CG}(s) = \left(-r(s)\cot\sigma\frac{dr}{ds},\ r(s)\sqrt{1-\cot^2\sigma(dr/ds)^2},\ s\right)_{uvs}. \tag{16}$$

Thus finally, we have obtained the image contour of an **RCSGC** (i.e., the image of the contour generator) which is

$$I_{CG}(s) = (x_{CG}, y_{CG})_{xy} = \left(-r(s)\frac{\cos^2\sigma}{\sin\sigma}\frac{dr}{ds} + s\sin\sigma, r(s)\sqrt{1-\cot^2\sigma(dr/ds)^2}\right)_{xy} \tag{17}$$

This is the equation of the image contour (or silhouette) of an RCSGC when viewed from an angle σ with respect to the cylinder axis. Note that in (16) the contour generator is defined for such points that satisfy

$$1-\cot^2\sigma\left(\frac{dr}{ds}\right)^2 \geqslant 0$$

$$\left|\frac{dr}{ds}\right| \leqslant |\tan\sigma| \tag{18}$$

This condition reveals very interesting properties of contours of the RCSGC that are somewhat different from our intuition. To see this, suppose we begin with a side view ($\sigma = \pi/2$) of the object, and study the behavior of the contour generator as we rotate the object towards an end view ($\sigma = 0$). Figs. 16–19 illustrate how the contour generators move with the viewing angle σ. At the start when $\sigma = \pi/2$, $\tan\sigma$ is infinite and $|dr/ds| \leqslant \tan\sigma$ for all s. As shown in Fig. 16, there is a single continuous contour generator on the object, which will in fact be planar (running along the top and bottom of the object).

As we rotate the object slightly, σ decreases and hence $\tan\sigma$, but as long as $|dr/ds| \leqslant \tan\sigma$ everywhere, the contour generator will still be continuous (Fig. 17). However, it will no longer be planar in general. As was defined in (11), the value of the parameter t specifics the position along the circular cross section: starting from the front $(t=0)$, the top $(t=\frac{1}{4})$, the rear $(t=\frac{1}{2})$, the bottom $(t=\frac{3}{4})$, back to the front $(t=1)$. From (15), we see that where dr/ds is 0, $t=\frac{1}{4}$. That is, where the radius of the cylinder is either minimum or maximum the contour generator is on "top" of the object: in other words, we see the real radius there. Where $dr/ds < 0$, $t > \frac{1}{4}$ and the contour generator is pushed away from us; where $dr/ds > 0$, the contour generator is pulled towards us.

Let us presume for the moment that the object is thinner at the near end, i.e., $dr/ds \leqslant 0$ toward the near end. Eventually, we rotate the object so much that $dr/ds = -\tan\sigma$ at some value of s, say s_m, where dr/ds is at a minimum (Fig. 18). At this point, the contour generator is tangent to the line of sight.

If we rotate the object yet farther, as shown in Fig. 19, there will be an interval (s_a, s_b) around s_m in which $dr/ds < -\tan\sigma$, that is, for which no contour generator points exist. What has happened is that the former, single-contour generator has been split into two separate contour generators, corresponding to values of s such that $s \geqslant s_b$ and $s \leqslant s_a$. Along the contour for $s \geqslant s_b$, all points will bevisible in the image. Meanwhile, along the contour for $s < s_a$, the object itself will occlude part of the contour generator, for values of s above some value s_c (where $s_c \leqslant s_a$) (segment X in Fig. **20).** What we have seen is a single image contour splitting into two parts connected by a sort of "T" junction; the split occurred at the point at which the contour generator was tangent to the line of sight.

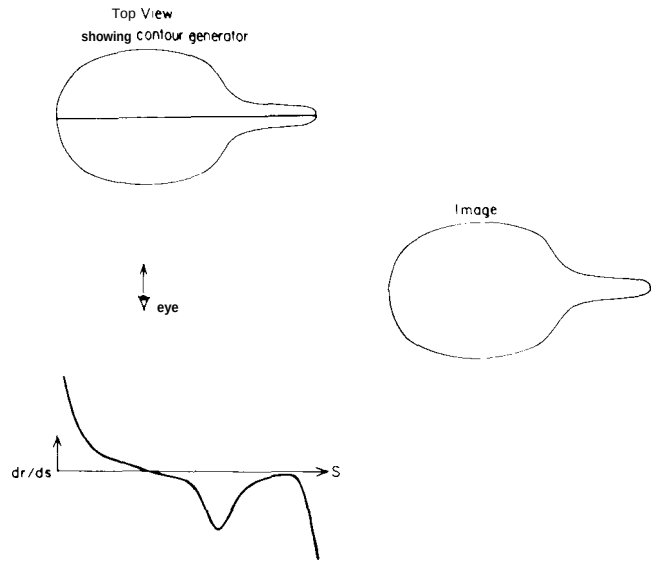


Fig. 16. The contour generator of an RCSGC in its side view.

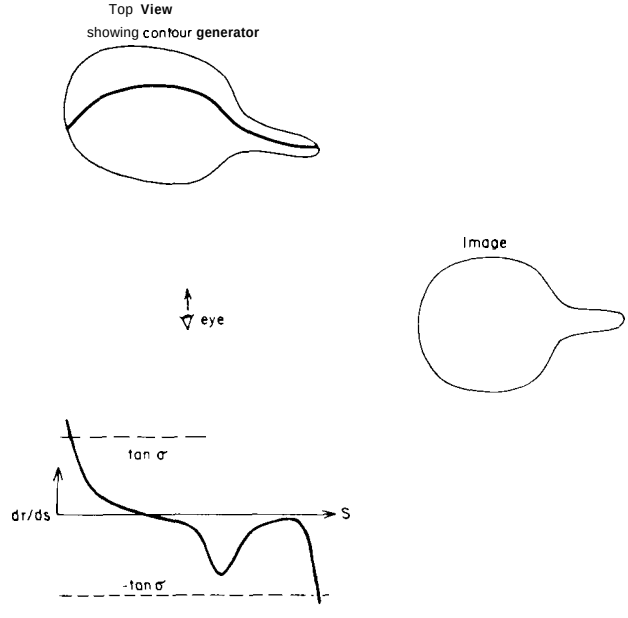


Fig. 17. The contour generator in a near-side view.

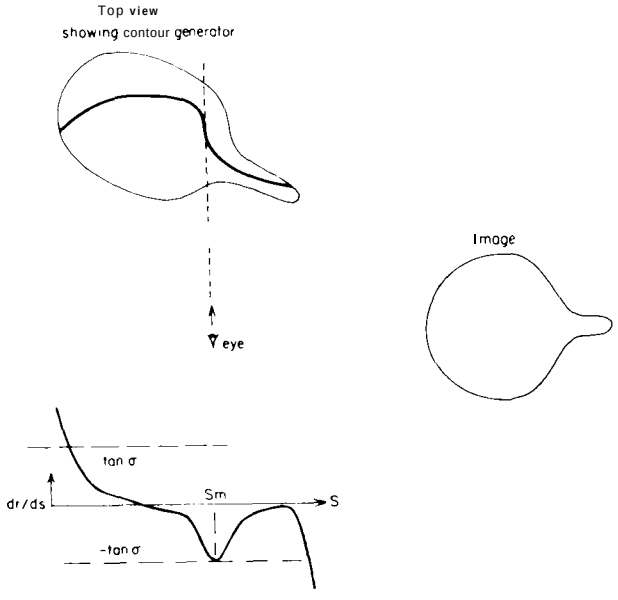


Fig. 18. The contour generator when tangent to the line of sight.

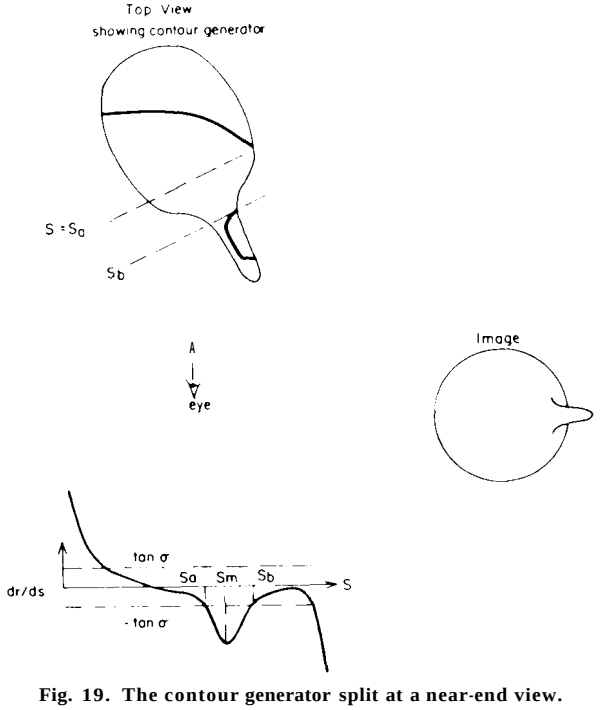


Fig. 19. The contour generator split at a near-end view.

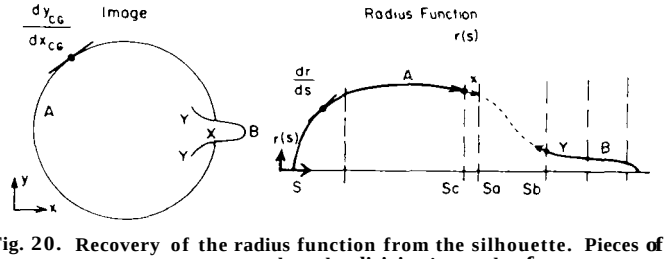


Fig. 20. Recovery of the radius function from the silhouette. Pieces of contour correspond to the disjoint intervals of s.

When we look at a vase (which is an RCSGC) we tend to think that the silhouette portrays the farmost points on the diameter at each cross section, and thus it represents the radius function. Contrary to this intuitive perception, the phenomena illustrated in Figs, 16–20 show that the situation is much more complicated.

C. Contour and Silhouette Analysis for RCSGC's

Now let us consider the reverse problem of projecting an **RCSGC:** that is, the problem of reconstructing a description of the solid shape portrayed from a given visible (i.e., nonoccluded) contours of an **RCSGC** such **as** Fig. 20.

Recovering the Radius Function; For the time being, we assume that the viewing angle σ (the angle between the line of sight and the axis of the generalized cylinder) is known. For an **RCSGC,** we need only determine $r(s)$, the radius function, to have a complete description of the shape. We may be tempted to simply use the width of a contour **as** representing the radius function $r(s)$, but the discussion above on the contour generator has shown that it is wrong to do so.

Our task here is to determine s and $r(s)$ for any given contour point $(x_{CG}, y_{CG})_{xy}$ in the image. By doing this for all contour points, we can determine as much as possible about $r(s)$. We have shown that the contour point $(x_{CG}, y_{CG})_{xy}$ is given by (16).

First, we can determine dr/ds as a function of dy_{CG}/dx_{CG} by the following derivation:

$$\frac{dy_{CG}}{dx_{CG}} = \frac{dy_{CG}}{ds}\,\frac{ds}{dx_{CG}} = (1/\sqrt{\sin^2\sigma - \cos^2\sigma(dr/ds)^2})\,\frac{dr}{ds}.$$

Therefore

$$\frac{dr}{ds} = (\sin\sigma/\sqrt{1+\cos^2\sigma(dy_{CG}/dx_{CG})^2})\,\frac{dy_{CG}}{dx_{CG}}.$$

This is the relationship between the slope of the contour in the image and the slope of the radius function along the axis. Now substituting for dr/ds in the definition of $y_{CG}(s)$ and $x_{CG}(s)$ and solving them for $r(s)$ and s, we obtain

$$r(s) = y_{CG}(s)\sqrt{1+\cos^2\sigma(dy_{CG}/dx_{CG})^2}$$

$$s = \frac{x_{CG}(s) - y_{CG}(s)\cos^2\sigma(dy_{CG}/dx_{CG})}{\sin\sigma}$$

Thus we have expressions for s, $r(s)$, and dr/ds, in terms of $x_{CG}(s)$, $y_{CG}(s)$, and dy_{CG}/dx_{CG}. The point $(x_{CG}(s), y_{CG}(s))$ is the image contour itself, **and** dy_{CG}/dx_{CG} is the slope of the contour in image, and thus both can be measured in the given image. Note, however, that the expression for s has a singularity when $\sin\sigma$ is near 0: that is, in an end view.

Of course, as shown in (18), there is a contour generator only where $|dr/ds| \leq |\tan\sigma|$; values of s for which $|dr/ds| > |\tan\sigma|$, therefore, do not correspond to any points on a contour generator. **Also** part of the contour generator may be occluded. The radius function $r(s)$ cannot be determined for those values by examination of image contours. Interestingly, however, for an interval of s over which $r(s)$ cannot be computed, say (s_i, s_j), we can stillcompute $r(s_i)$, $r(s_j)$, $dr/ds|_{s_i}$, and $dr/ds|_{s_j}$ (the values and slopes of $r(s)$). Therefore. in practical image analysis, it is possible to interpolate $r(s)$ over (s_i, s_j) by fitting a function which conforms to these boundary constraints.

Viewing Angle: The preceding analysis has presumed that

we know the viewing angle a and have aligned the image of the axis with the image x-axis. We will now address the problems of aligning the image and determining σ. This is in fact a very difficult problem. If the viewing angle cannot be determined, the reconstruction of $r(s)$ may contain substantial errors.

Suppose we are given an image of the contours of an **RCSGC,** arbitrarily scaled, rotated, and translated, and viewed from an unknown angle. We can immediately determine the image of the axis, since this will be an axis of symmetry in the image. By rotating and translating the image, this axis can be nade to line up with the x-axis. The use of the axis of the projected image contour is actually done in a few systems. ACRONYM [24] uses the axis of an image contour, which is called a ribbon, to align the image with the object model. Marr and Nishihara [23] also use the axes for object matching. Extremely foreshortened images seem to be difficult for perception even by humans because they do not provide this cue of axis oriertation.

For determining the viewing angle σ together with knowing which end is closer, the image of the cross section at either end, if available, provides much information. If the closer end of the object **is** flat (i.e., $r(1) > 0$), then the edge of the cross section at that end will produce a contour in the image, which will be an elongated ellipse. We then know that the center of the ellipse must be the image of the axis endpoint $A(1)$. Further, we can compute the viewing angle a from the eccentricity of the ellipse, using $\cos\sigma = b/a$, where a and b are the semimajor and semiminor axis lengths, respectively. If the farther end of the object is flat and not occluded (i.e., $r(0) > 0$ and $dr/ds|_0 <$ tan a), then we will see half of an ellipse, which can be analyzed as above to determine the image of $A(0)$ and σ.

If neither end can be analyzed in this manner, outside information must be used. In practice, there are several potential sources for sufficient information to determine the viewing angle, or at least its range, such as knowledge of the permissible object's length or width. This **is** in fact what ACRONYM [24] does in estimating the object parameters.

IV. Construction of 3D Descriptions

The theories presented in the previous section extract natural geometrical constraints under reasonable, mostly task-independent assumptions, However, the information they provide is fragmental in the sense that it **is** about the orientations of local surfaces or about the shape of individual objects.

The goal of the visual process, however, is to construct a complete description of the scene which is sensed by visual methods. It must be recognized that this goal is different from classifying parts of an image, detecting the object existence, or segmenting the images. The difference is most easily understood in the scenario of robot navigation where classification or detection **is** not enough to plan actions of the robot: it must have a **3D** description of its environment.

A. Incremental Acquisition of 3D Descriptions

At Carnegie-Mellon University, Herman and Kanade [26] are developing a system called Incremental **3D** Mosaic, which builds a scene description from a sequence of (stereo) images. It is currently applied to an urban scene for building a **3D** model of the task area from low-altitude aerial photos. The reason for building the description incrementally is that a single pair of stereo photos can give only partial information about the scene: only limited portions are visible in both images and the stereo

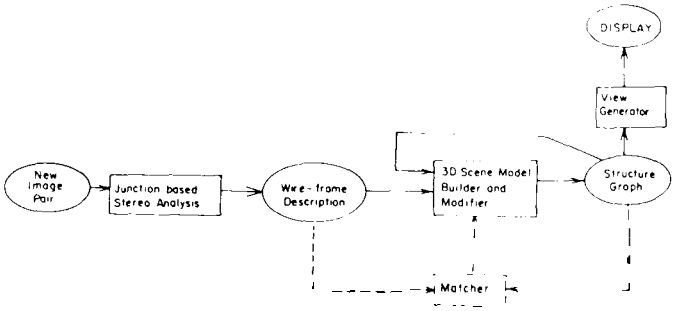


Fig. 21. The flowchart of the Incremental 3D Mosaic system, showing major modules (boxes) and data structures (ellipses).

system cannot be perfect anyway. As new images become available, new information must be incrementally added to a current description, while maintaining consistency: the information on the portions of the scene which become visible should be added, and the information on the previously known portion should be used to improve the description, either by correcting errors or by increasing the precision of the description. Also important **is** that the partial description still provides knowledge which is usable for interpreting the new images or planning actions.

Fig. 21 shows the general flow of the system. The system has the following components: a junction-based stereo analyzer that generates a **3D** wireframe representation of the scene; a **3D** model builder and modifier that takes the wireframe representation and constructs a surface-based description of the scene; and a view generator that synthesizes natural-looking images of the scene from the model, given an arbitrary viewing angle.

B. Edge and Corner Findings from Images

Given a stereo pair of images, the first thing to do is to extract 3D information from them. Fig. 22(a) shows examples of input stereo images. First, lines and junctions are extracted from them. The results are shown in **Fig.** 22(b). Then, junctions and lines are matched. They are mostly corners and edges of buildings. Here we use the fact that buildings are mostly block-shaped, and that they have vertical (gravitational) edges. This allows one to hypothesize the changes of junction appearances along the epipolar line of stereo, and to cope with the difficulty inmatching wide-angle stereo images that include large disparity jumps, such as urban scenes. We can compute the **3D** locations of the matched junctions and lines. **As** a result, we obtain **3D** wireframes. Fig. 22(c) shows their perspective view.

C. Representation of Model: Structure Graph

The wireframe representation does not yet identify surfaces. The next step is to perform reasoning on the surfaces. This is done by assigning planar surfaces so that an enclosed object is obtained with the wireframes as edges. The process is similar to obtaining solid objects from wireframes [27], but since we assume that our wireframes are not always complete, we must combine edges if they are close, generate web faces from a set of edges which partially enclose, merge partial faces, hypothesize vertical faces if necessary, etc. There are mutual relationships and constraints among the edges and faces concerning which evidence supports which hypothesis. In order to maintain such relationships, constraints on **3D** structure are represented in the form of a graph, called the ***structure graph.*** This graph **is**

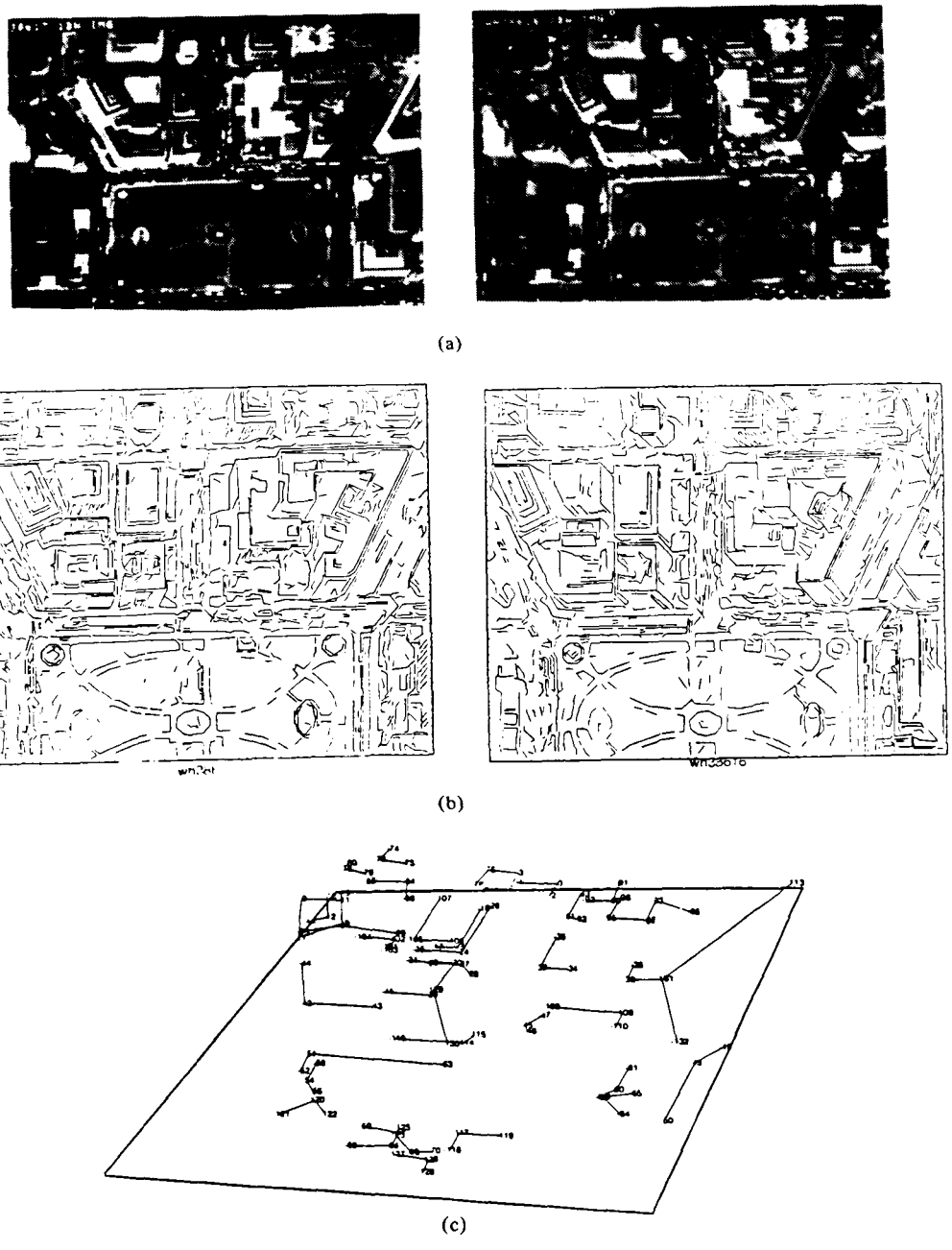


Fig. 22. Image analyses in Incremental 3D Mosaic. (a) Images of right and left views. (b) Line segments extracted. (c) Perspective view of 3D wireframes which are made by edges obtained by stereo matching.

incrementally constructed through the addition of topological and geometrical constraints which are represented by the nodes and links. As constraints are accumulated in the graph, their effects are propagated to other parts of the graph so as to obtain globally consistent interpretations.

The current structure-graph representation models surfaces in the scene as polyhedra whose components are the faces, edges, and vertices. We distinguish the topology of the polyhedral components from their geometry [28], [29]. The geometry involves the physical dimensions and location in 3D space of each component, while the topology involves connections between the components.

In the structure graph, nodes represent either primitive topological elements (faces, edges, vertices, objects, and edge groups which are rings of edges on faces), or primitive geometric elements (planes, lines, and points). Vertex, face, and edge nodes are tagged as either ***confirmed*** or ***unconfirmed.*** Confirmed means that the element represented by the node has been derived directly from the images. Unconfirmed means that the element has only been hypothesized.

The primitive geometric elements serve to constrain the 3D locations of faces, edges, and vertices. Plane and line nodes contain plane and line equations, respectively. Point nodes contain coordinate values. The graph contains two types of

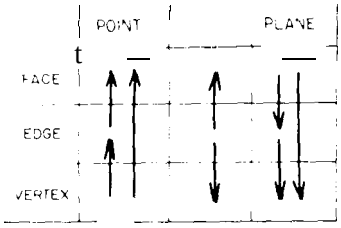


Fig. 23. Propagation of constraints in modifying the structure graph. Rectangular boxes indicate geometric constraints on topological nodes. Arrows indicate direction of propagation of constraints.

links: the *part-of* link, representing the part/whole relation between two topological nodes, and the *geometric constraint* link, representing the constraint relation between *a* geometric and topological node.

D. Modifications to Model

Modifications to the model will occur *as* part of the process of incremental construction. Deletions and changes are made when new information is found to conflict with information currently contained in the model. This happens most often with portions of the model that have been hypothesized. Additions *to* the model are made to incorporate the new information.

Modifications to the structure graph are made by adding or deleting nodes and links, or changing the equations of line and plane nodes, or the coordinates of point nodes. **All** effects of modifications are propagated to other parts of the graph.

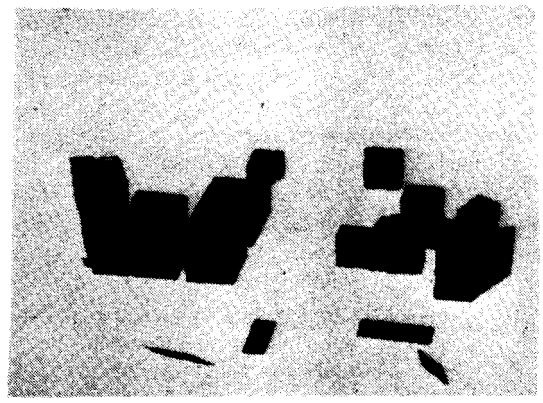
As an example, consider adding or deleting a geometric constraint link between *a* geometric and topological node. Any of the three geometric nodes—points, lines, and planes—may constrain any of the three topological nodes—vertices, edges, and faces. Fig. 23 shows how *a* constraint on one node may propagate to others. The arrows in the figure indicate the direction of propagation. For example, if *a* point constrains *a* vertex, it must also constrain all edges and faces containing that vertex. Similarly, *a* point that constrains an edge also constrains all faces containing that edge.

When *a* geometric constraint link **is** deleted, the rest of the structure graph must be made consistent with this change. Our approach to this problem is based on the TMS system [30], using the notion that when an assertion is deleted, all assertions implying it and all assertions implied by it should also be deleted, unless they have other support. Assertions that imply *a* given assertion are obtained by following backwards along the arrows in Fig. 23. Assertions implied by a given assertion involve following forward along the arrows.

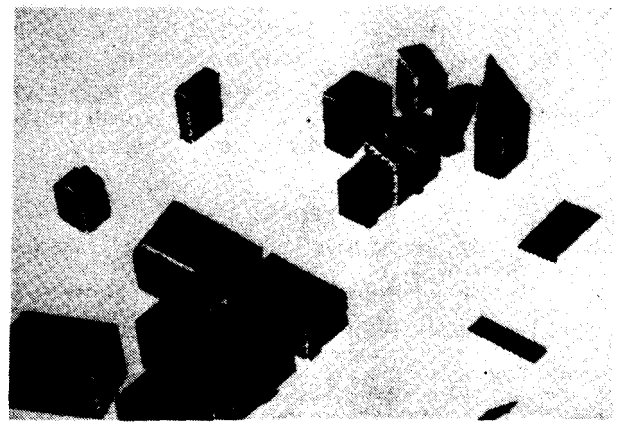
E. Display

In our examples, the 3D description of the buildings in the area is constructed **as** a set of polyhedra. **A** perspective view of the model is displayed as shown in Fig. 24(a). Once we have such a surface description, we can crop image patches from original images to know the normalized appearances of surfaces (e.g., window patterns). A natural looking display can be generated for the scene viewed from any angle by appropriately transforming such appearances according to surface orientations. Fig. 24(b) is an example of such synthesized images. In such a synthesized image, parts of surfaces which were not visible in the original images are displayed distinctively as such.

This kind of description is useful, for example, for planning



(a)



(b)

Fig. 24. The scene model from Incremental 3D Mosaic. (a) Plane-surfaced models of buildings. (b) Synthesized image of the scene from the angle different from the original angle.

the angles that the next images should take: it is generally better to cover as much of the nonvisible portions as possible to increase the knowledge of the task area. This scenario of Incremental 3D Mosaic is applicable to robot navigation and to change detection in *a* scene.

V. Conclusion

We have identified 3D geometrical aspects *as a* crucial issue in vision. The key feature to be achieved is the capability to construct scene descriptions from images. Theories and system, which deal with 3D shape information in three different levels—surface patch, volumetric-object shape, and scene description—were presented. Throughout the discussion, emphasis has been put on:

computational aspects of vision to handle the reaitionship between 3D shape and images;

converting the knowledge of projective geometry into constraint expressions in the appropriate representation space;

obtaining scene descriptions **as** the goal of an image-understanding program

Noteworthy is that the higher the level of description, the more it tends to use the task-specific knowledge: the theories

in the surface-patch level mostly depend on the task-independent physical-level knowledge such as projective geometry, whereas the Incremental 3D Mosaic system also relies on the domain knowledge such as polyhedral buildings in the urban scenes.

The techniques and approaches in obtaining and representing geometrical constraints from the physical level of knowledge are very important. The key idea is to model the projection process and represent it in a form that can be used for inverse projection in conjunction with other constraints involved in interpretation. It is noteworthy that the forward projection rules are simple and fairly well understood. But they are often so local or microscopic that their direct application may result in huge equations that are difficult to manipulate. Appropriate representational spaces, such as gradient space, enable them to be applied in a macroscopic manner.

ACKNOWLEDGMENT

Credits of the work at CMU described in this paper should go to M. Herman, J. Kender (currently at University of Columbia), S. Kuroe (Oki Electric, Inc), S. Shafer, and D. Williams. The author also thanks D. Ballard and A. Mackworth for their constructive criticism on the earlier version of this paper.

REFERENCES

[1] *Artificial Intell.* (Special Issue on Vision), vol. 17. New York: North Holland, 1981.

[2] D. A. Huffman, "Impossible objects as nonsense sentences," in *Machine Intelligence* 6, B Meltzer, and D. Michie, Eds. New York: American Elsevier, 1971 ch. 19. pp. 295-323.

[3] A. K. Mackworth, "Interpreting pictures of polyhedral scenes," *Artificial Intell.*, vol. 4, pp. 121-137, 1973.

[4] K. Ikeuchi, "Shape from regular patterns (an example of constraint propagation in vision)," Mass. Inst. Technol., Tech. Rep. A.I. Memo 567. Mar. 1980 (Artificial Intelligence Laboratory).

[5] J. K. Kender, "The Gaussian sphere: A unifying representation of surface orientation," in *Proc. ARPA Image Understanding Workshop,* pp. 157-160, Apr. 1980.

[6] T. Kanade, "A theory of origami world," *Artificial Intell.*, vol. 13. pp. 279-311, 1980.

[7] —, "Recovery of the three-dimensional shape of an object from a single view," *Artificial Intell.*, vol. 17, pp. 409-460, 1981.

[8] T. Kanade and J. R. Kender, "Mapping image properties into shape constraints: Skewed symmetry, affine transformable patterns and the shape-from-texture paradigm," Carnegie-Mellon Univ., Computer Sci. Dep., Tech. Rep. CMU-CS-80-133, July 1980.

[9] J. R. Kender, "Shape from texture," Ph.D. dissertation, Carnegie-Mellon Univ., Computer Sci. Dep., 1980.

[10] B.K.P. Horn, "Understanding image intensities," *Artificial* Intell., vol. 8, pp. 201-231, 1977.

[11] D. G. Lowe and T. O. Binford, "The interpretation of three-dimensional structure from image curves," in *Proc. 7th Int. Joint Conf. on Artificial Intelligence*, pp. 613-624, 1981.

[12] S. Shafer and T. Kanade, "Using shadows in finding surface orientations," Carnegie-Mellon Univ., Tech. Rep. CMU-CS-82-100, Jan. 1982.

[13] A. K. Mackworth, "On the interpretation of drawings as three-dimensional scenes," Ph.D. dissertation, University of Sussex, 1974.

[14] S. A. Shafer, T. Kanade, and J. K. Kender, "Gradient space under orthography and perspective," in *IEEE Workshop on Computer Vision: Representation and Control,* pp. 26-33, Aug. 1982.

[15] J. R. Kender, "Shape from texture: A brief overview and a new aggregation transform," in *Proc. ARPA Image Understanding Workshop,* pp. 79-84, 1978.

[16] Y. Ohta, K. Maenobu, and T. Sakai, "Obtaining surface orientation from texels under perspective projection," in *Proc. 7th Int. Joint Conf. on Artificial Intelligence,* pp. 746-751, Aug. 1981.

[17] S. T. Barnard, "Methods for interpreting perspective images," in *Proc. ARPA Image Understanding Workshop,* pp. 193-203, Sept. 1982.

[18] T. O. Binford, "Visual perception by computer," in *Proc. IEEE Conf. on Systems and Control,* 1971.

[19] S. Shafer and T. Kanade, "The theory of straight generalized cylinders," Carnegie-Mellon Univ., Computer Sci. Dep., Tech. Rep. CMU-CS-83-105, 1983.

[20] G. J. Agin, and T. O. Binford, "Computer description of curved objects," in *Proc. 3rd Int. Joint Conf. on Artificial Intelligence,* pp. 629-640, Aug. 1973.

[21] R. Nevatia and T. O. Binford, "Structured descriptions of complex objects," in *Proc. 3rd* Int. *Joint Conf. on Artificial Intelligence,* pp. 641-647, 1973.

[22] J. M. Hollerbach, "Hierarchical shape description of objects by selection and modification of prototypes," Mass. Inst. Technol. Tech. Rep. AI-TR 346, Nov. 1975.

[23] D. Marr and H. K. Nishihara, "Representation and recognition of spatial organization of three-dimensional shapes," *Proc. Roy. Soc. of London,* vol. B-200, pp. 269-294, 1978.

[24] R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *Artificial* Intell., vol. 17, pp. 285-349, 1981.

[25] D. Marr, "Analysis of occluding contour," *Proc. Roy. Soc. of London,* vol. B-197. pp. 441-475, 1977.

[26] M. Herman, T. Kanade, and S. Kuroe. "Incremental acquisition of a three-dimensional scene model from images," Carnegie-Mellon Univ., Computer Sci. Dep., Tech. Rep. CMU-CS-82-139, 1982.

[27] G. Markowsky and M. A. Wesley, "Fleshing out wire frames." IBM Res. Div., IBM Thomas J. Watson Res. Center, Yorktown Heights, NY 10598, Tech Rep. RC 8124, Feb. 1980.

[28] A. Baer. C. Eastman, and M. Henrion, "Geometric modelling: A survey," *Computer-Aided Des.,* vol. 11, Sept. 1979.

[29] C. M. Eastman and K. Preiss, "A unified view of solid shape modeling based on consistency verification." Carnegie-Mellon Univ., Tech. Rep., Sept. 1981.

[30] J. Dolye, "A truth maintenance system," *Artificial* Intell., vol. 12, pp. 231-272, 1979.