Representation and Control in Vision

Takeo Kanade

Department of Computer Science
Carncgic-Mcllon University
Pittsburgh, Pa. **15213**

1. Introduction

Onc of thc central issucs in vision is how to reprcscnt and usc knowlcdgc relevant to undcrstanding thc imagc. Partly bccausc vision is so difficult. and partly bccausc evcn thc chcapcst solutions can still bc **so** uscful. npproachcs to vision problcms havc had a tcndcncy to bc *ad hoc* and hcuristic. Rcccntly, howcvcr. ncw thrusts in computcr vision arc emerging. most notably in thc Image Understanding community, that try to pursue morc systcmatic and computational approaches. **[1]**

This article attcmpts to introduce such ncw Image Understanding approachcs to vision. It first prescnts thc author's view of structure of vision: what typcs of information must bc dealt with and what lcvcls of knowlcdgc are involved in transforming one type of information into another. Thcn, rcprcscntativc rcsearch progress in Imagc Undcrstanding is rcvicwed which addrcsscs usc and rcprcscntation of knowlcdgc in vision. Specifically. wc will discuss:

e Formalization of physical knowledge into computational forms

e Usc of **3D** models

e Construction of scciic descriptions from images

e Organization and control of vision systems

2. Structure of Vision

Lct us considcr what vision rcally is. Vision involves visual scnsing and intcrprctation. Visual scnsing is a *projcction* of a physical cnvironmcnt into a form of rcprcscntation callcd *images* Projcction can vary from thc most ordinary picturc takcn by cameras **to** activc scnsing such as by lascr rangc findcrs. Imagcs can also rangc from a singlc-point light **flux** mcasurcmcnt **to** 3-dirncnsional rangc data. Thcn vision is dcfincd as a process ***of*** undcrstanding thc cnvironmcnt througli thc projcctcd imagcs, or in short *inverse projection.*

l'hc vision proccss must involvc various typcs of information. Figurc 1 is a schcmatic diagram showing onc possihlc dichotomy of such typcs and thcir primary rclationships. [14] 'l'hc gcncral idca is as follows: givcn an imagc, cucs (*picturc domain cucs* or *scene domain cues*) arc cxtractcd: thcsc arc thcn uscd to acccss thc *model* of thc task world and to instantiatc it: thc *instantiated model* is verified by projecting it back to thc picturc lcvcl (*picturc interpretation*) and by matching it with thc input imagc. 'l'his positivc fccdback loop is thc basic drivc in thc analysis proccss.

Scvcral points should bc notcd about this mtdcl. First ***of*** all, thcrc is an important distinction hctwccn thc picturc domain and thc sccnc domain. In **short.** a *picturc* is a projcction of a *scene*. Thus thc picturc domain cucs arc thc fcaturcs obscrvcd in thc picturc. such **as** linc scgmcnts. homogcncous rcgions, intcnsity gradicnts, ctc. 'l'hc sccnc domain cucs arc thc fcaturcs which arc thc causc of thc picturc domain cucs, such as cdgc configurations, surfacc oricntations, rcflcctanccs, volumcs ctc. 'l'his distinction prcvciits oiic from confusing fcaturcs in thc picturc domain with thosc in thc sccnc domain. For cxample, thc "abovc" or "adjaccnt" rclationship hctwccn rcgions in thc picturc docs *not* necessarily corrcspond to the "on" or "ncxt-to" relation bctwccn objects. Though thc most basic and important Scene domain cues are spatial three-dimensional **(3-D)** configurations, thcrc arc other important ones, such as rcflcctanccs and lighting conditions. Thc modcl contains thc gcncric descriptions of thc task world. *It* can be thought of as containing information about thc gcncric shapcs of objccts and rclations among thcm. Notc that gcncnc

NATO ASI Series, Vol F4
Pictorial Data Analysis. Edited by K M. Haralick

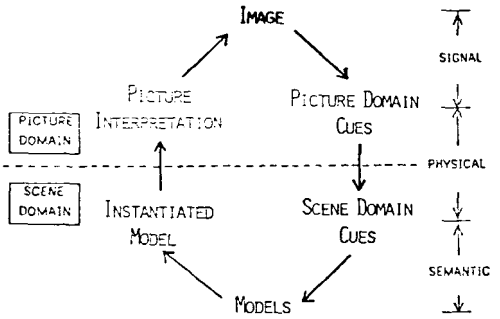


Figure I: A modcl of vision process. Roughly speaking, the right half of the cycle corrcsponds to the data-driven, bottom-up. hypothesis-generation phase, and the lcft **half** to the model-driven, top-down. verification phase **of** vision.

modcls arc to bc represented in terms of sccnc domain cues. The instantiatcd modcl is the spccific situation being dcpictcd in the picturc: the specific objccts. their sizes, their placcmcnt, etc. The picturc intcrprctation is a projection (an anticipated picturc) of the instantiated modcl under the spccific conditions (view direction. lighting conditions, ctc.). It gives an explanation of the imagc by the descriptions such as what lincs and rcgions of what propcrtics should appear in the image. what set of lincs or regions constitute the objects. and so on. An example of this typc of information is an image segmented into rcgions with scmantic labels attached to cach. It is diffcrcnt from an *image* which is only a collection of image fcaturcs. such as color and intcnsity.

The model of Figure 1 involves thrcc levels of knoxlcdge -- signal. physical. and semantic. Roughly speaking, the signal level involves manipulating imagc fcaturcs as 2-D patterns, the physical level involves bridging the picturc and scene domains so that image features can be interpreted as scene features, and the semantic level involves working in the sccnc domain to exploit the task world constraints.

The diffcrcncc among these levels can be illustrated by the following cxamplc. Suppose that a histogram of the gray levels of an imagc shows a clear bimodality. The value at the valley is often sclcctcd as a good threshold. This is based on the *signal level* knowledge that in such a case often two sourccs of monomodal distribution arc prcscnt. Therefore, a threshold value at that valley will give a good splitting of the image. We are also assuming that a salt-and-pcppcr imagc (which can produce the same histogram) is rare. This signal-level histograming tcchniquc is actually based on *physical level* knowlcdge. We know that two surfaces, whose reflectivity characteristics arc diffcrcnt from each othcr but constant within cach. or which have different oricntations. can produce two regions in the image whose fray levels arc different but relatively constant within each. Why is it significant to extract those surfaccs for understanding the imagc? Because we know that surfaces with different rcflectancc or oricntations arc the *semantically* meaningful units which constitute the objects in our world.

As another example. the extraction of straight lincs by detecting suddcn changcs of intensity on the signal level is justified by the physical-level observation that such intensity changes arc caused by suddcn change of oricntations (convexity and concavity). distancc (occlusion). illumination (shadow). and rcflcctancc (material change), all of which arc usually importnnt clues in the scmantic analysis **of** the scene content.

It is *not* meant that the analysis should always follow the routes shown in Figurc 1. In fact. one can combine several steps into one, or prccompilc ncccssnry knowlcdgc ahout some entities into a processing procedure, so that those entities arc not explicitly trcatcd. However, the meaning and limitations of many methods can hc illustrated by considcring what types of knowledge are incorporated into them, according to this paradigm.

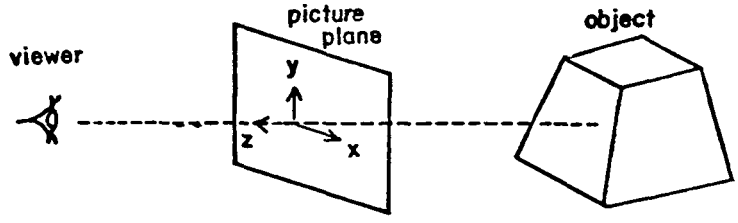
Notice that essential difficulties in vision iirc in crossing the houndary hctwccn the scene and picturc domains from picture to scciic: it is ambiguous, thus additional constraints arc to bc used. Generality of vision depends on how to derive and use necessary constraints.

3. Converting Physical Level of Knowledge into Computational Constraints

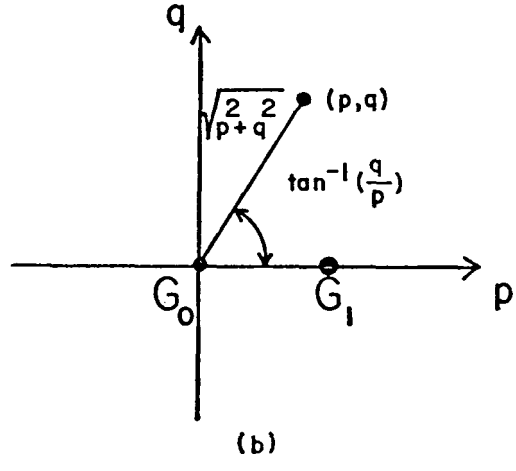
Historically, there are several annoying image phenomena which often cause vision programs to fail in image analysis: they include deformed shape due to slanted views, shadings due to curved surfaces and lighting, and textural patterns. and shadow. That may be still true in most applications, but it has begun to be understood that these are rich sources of information about object shape. An interesting class of theories have been developed to handle these sourccs. They arc all for relating picture domain cues with scene domain cues so that they can be used in recovering 3-dimensional scene features from images. This section will briefly introduce such theories which allow computational usage of constraints from image intensity and geometry in order to exploit 2D-shape, shading, texture. and shadow in the image.

Sincc wc will use *gradient space* [32] [20] throughout the section as a convenient tool to represent surface oncntations. lct us first dcfinc it. Our coordinate system x-y-z is placed so that the optical axis is the z axis and the image plane is the x-y plane. (See Figure 2.) Consider a surface,

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



planes: $-z = px + qy + c \longrightarrow$ point: (p,q)



(b)

Figure 2: (a) Imaging geometry including the object, the picture, and the viewer. (b) mapping of planes to a gradient.

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

$$p = \frac{\partial f}{\partial x}, \quad q = \frac{\partial f}{\partial y} \tag{2}$$

That is, p and **q** arc the rate of changc in depth on the surfacc along the **x** and **y** dircction. Wc can easily see that (p,q,1) has the direction of the surfacc normal.

Gradients arc constant over a planar surfacc, and the gradicnt (p,q) corrcsponds to a set of parallcl plancs:

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

where c is arbitrary. Throughout this section we will assume orthographic projection for simplicity rather than pcrspcctive projcction.

Under orthographic projcction. the gradients of plancs and the image linc on the image have an intcrcsting rclationship. Referring *to* Figure 3, let two plancs P_1 and P_2 intersect in the spacc and let the intcrscction cdgc be dcpictcd as a linc *l* in the image. Then the linc in the gradient space connecting the corrcsponding gradicnts $G_1 = (p_1,q_1)$ *is* perpendicular to *the* linc *l*. This can bee seen in the following way. The normals of the two surfaccs havc dircctions $n_1 = (p_1,q_1,1)$ and $n = (p_2,q_2,1)$. Their cross product $n_1 \times n_2$ rcprcscnts the dircction of the intersection cdgc in the space. Since wc assume orthographic projcction the dircction of the linc *l* in the image is givcn by the x and y components of this cross product: i.e., $(q_1 - q_2,\ p_2 - p_1)$. This vcctor is obviously pcrpcndicular *to* the vcctor $G_1 - G_2 = (p_1 - p_2,\ q_1 - q_2)$ which connects G_1 and G_2. Morcovcr if the intcrscction cdgc is convex vicwcd from the viewer, then the positional ordcr of G_1 and G_2 is the samc as the rcgions in the picture corrcsponding to P_1 and P_2: if concavc. the ordcr is rcvcrscd.

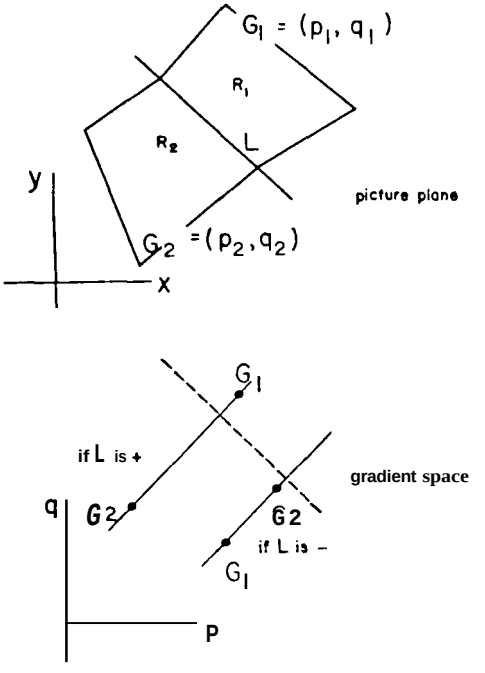


Figure 3: Properties of dual lines. If two plancs meet and the intcrscction linc is projcctcd as a picture line *L*, then the gradicnts of the two plancs are on a gradient-space line which is pcrpcndicular to *L*

3.1. Image Intensity

3.1 _ ISShading

Grey-level shading has been neglected or even thrown out in most robotics vision. though it is well known that it gives information on surface curvature. This is partly due to the dominance of binary imagc processing for fast proccssing and partly due to lack of adcquatc theory and techniques for exploiting shading information. Horn [9] did a pioneering work in shape-from-shading theory. Figurc **4** shows a simple modcl of imaging: it consists of a point light source, a surface patch, and a viewer. In general the intcnsity in the image corresponding to the surface patch is a complex function of illumination position, surface material, surface position. surface orientation, vicwcr position, ctc. Let us assume. for simplicity, a Lambcrtian surface (ie, perfectly uniform diffuse reflection) and a parallcl incident light (ie, a distancc point source). The obscrvcd intcnsity I which corresponds to the surfacc patch is given by,

$$I = I_0 \rho \cos(i) \tag{4}$$

where I_0 is the intensity of the incident light, ρ the reflectance of the surface, and i the incident anglc which is made by the incident light and the surface normal.

In ordcr for this cquation to be usable in vision. we nccd to convert it into the imaging coordinate framc. This can be done by using the gradicnt spacc. The surfacc normal of the patch is $\mathbf{n} = (p,q,1)$ and the dircction vector of the illumination is cxprcssed as $\mathbf{s} = (p_s, q_s, 1)$. Since $\cos(i) = \mathbf{n} \cdot \mathbf{s}/|\mathbf{n}||\mathbf{s}|$,

$$I(x,y) = I_0 \rho (p\,p_s + q\,q_s + 1)/\sqrt{p^2+q^2+1}\,\sqrt{p_s^2+q_s^2+1} = R(p,q) \tag{5}$$

In this way. the obscrvcd intcnsity $I(x,y)$ at an imagc point (x,y) is cxprcssed as a function $R(p,q)$ of the surface oricntation (p,q) of the corrcsponding surface point. Figurc **5** shows the loci of equal-intensity gradicnts $R(p,q) = I$, for the case $(p_s,q_s,1) = (0.7,0.3,1)$; i.e., the light comes from slightly above and right of the viewing anglc. Figurc 5 is callcd a rcflcctancc map.

It is worthwhile to note the significance of the rcprcscntation like the rcflcctancc map (or cquation (5)) as comparcd with cquation **(4)** . Equation **(4)** rcprcscnts the physical rulc in imaging as vicwcd from a third person who obscrvcs the imaging proccss. In contrast. cquation *(5)*. though rcprcscnting the samc rulc. is in the viewer's coordinate framc. This is essential for using the rulc to interpret the imagcs bccausc the imagcs arc rccordcd in that coordinate framc.

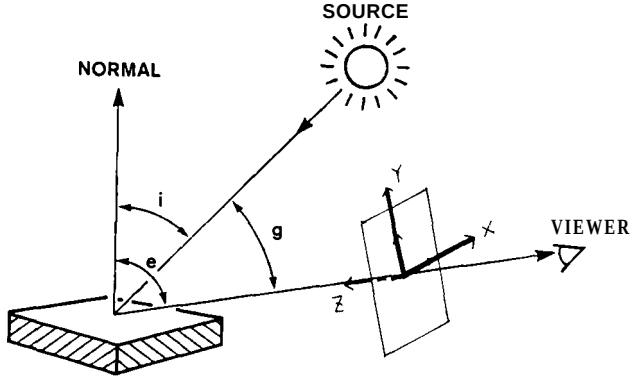


Figure 4: A simple reflection and imaging model

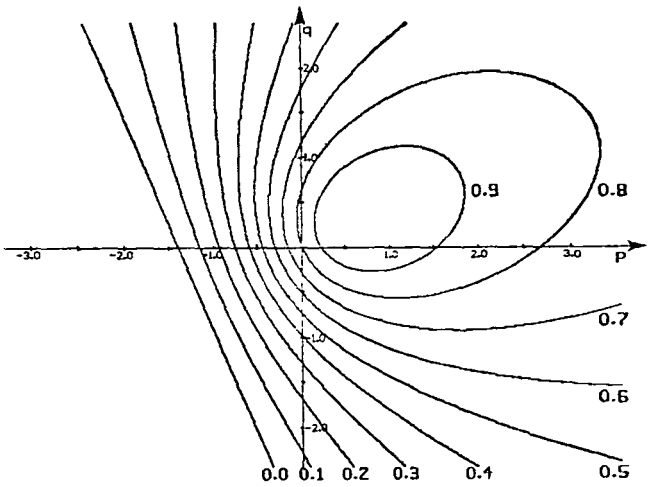


Figure 5: A reflectance map for a Lambertian surface illuminated by parallel flux of light from the direction (0.7.0.3, 1).

The reflectance maps can be used in a few ways. One is for shape-from-shading: given an image of an object. compute the shape of the object using the shading information. This can be accomplished by assigning to each pixel (i,j) in the image a gradient (p_{ij},q_{ij}). The reflectance map alone does not give enough constraints to uniquely determine the gradient at each point: i.e., R^{-1}: $I \rightarrow (p,q)$ is not a unique function. Here we need additional constraints. Ikeuchi and Horn [12] demonstrated a numerical shape from shading by assuming surface smoothness. They obtained the solution by minimizing the following.

$$E = \sum_i \sum_j E_{ij} = \sum_i \sum_j [\{ I(i,j) - R(p_{ij},q_{ij}) \}^2 + \lambda^2 \{ (p_{ij} - \bar{p}_{ij})^2 + (q_{ij} - \bar{q}_{ij})^2 \}] \quad (6)$$

The first term in E_{ij} corresponds to the difference between the observed and expected intensities. The second term approximates the Laplacian of the surface orientation and thus provides index of non-smoothness. In order to solve this minimization we further need boundary conditions: certain places in the image where the surface orientations are known. Occluding contours provide such boundary conditions. The surface orientations of those pixels at the occluding contours are known because the surface normal there is perpendicular to the line of sight (z axis) and the occluding contour projected onto the image. Using all these constraints 3D shape recovery from a grey-level image was demonstrated.

Another use of the reflectance map is photometric stereo [35]. Multiple images of an object are taken by the camera at one position. but with different lighting directions. At each pixel position. each image provides constraints on surface orientation according to the reflectance map. which corresponds to that lighting condition. Three or more images can determine the surface orientations uniquely. This method eliminates the so called correspondence problem in stereo. because the images are taken from a single position and thus are registered. It suggests use of dynamic clever control in lighting to inspect surface texture and anomalies in such applications as soldering and casting inspection.

The reflection properties of materials are complicated. and rather than analytical forms. they are often measured in forms such as the bidirectional reflectance distribution function (BRDF) or goniophotometric measurement. There have been developed computational procedures to convert such data into reflectance maps [10] [30]. These techniques make it possible to apply the reflectance map method to robotics tasks, because the environments are usually constrained and surface properties are known or measurable.

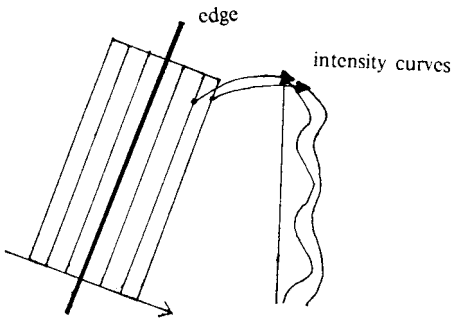


Figure 6: A stripe region around an edge. in which columns of intensity curves are taken for correlation.

3.1.2. Classification of Edge Types

We have noted that so called image edges (discontinuity of image intensity) can correspond to many types of scene edges. The most representative types are:

Type	Discontinuous Scene Feature
Convex/Concave	Surface orientation (Continuous distance)
Occluding	Distance (Different surfaces)
Cast Shadow	Illumination (Same surface)
Self Shadow	Illumination derivative
Pattern	Reflectance

If we can classify edges in the image into these types. it will make it far easier to analyze the 3-dimensional shapes of objects from the image. For example, the object shape can be found from occluding boundaries without being confused by non-shape related edges: and stereo matching would be much easier because the edge types tell what kind of different appearance should be expected on either side of an edge in making correspondence.

Initially. use of edge profiles [9] or intensity derivatives near the edge [3] was suggested for edge classification based on consideration **of** micro structures near the edge. Recently Fischler and Witkin [7] studied an interesting method to detect occluding boundaries and cast shadow edges in the image. Consider strip regions along an edge as shown in Figure 6 and take columns of intensity curves taken parallel with the edge along its length. **A** sequence of linear regressions is performed between each pair of consecutive intensity curves. As a result a normalized correlation coefficient is obtained. together with additive and multiplicative regression terms, each as a function **of** the location across the edge. An occluding boundary is indicated by a sharp notch in an otherwise high correlation at the nominal edge location. A cast shadow boundary is indicated by high correlation maintained across the edge with a sharp spike or notch in the additive and multiplicative regression term due to transition between shadow and nonshadow sides.

Justification of this method is given by the following simplified arguments. **At** the occluding boundary we are looking at different surfaces on each side. Thus the image properties across it should not be correlated. At the shadow boundary. illumination changes but the surface is the same. Thus image intensities across it should be correlated with change only in either magnitude or scale. This classification method rests on the principle that the coherence in the image intensity reflects the real coherence in the scene.

3.2. Geometry

Geometry is another important physical level of knowledge which governs the vision process and from which constraints can be extracted.

3.2.1. 2D Shape

Two dimensional shapes of regions (i.e., projection of surfaces) in the image convey information about the three-dimensional shape of an object in the scene. This is illustrated by such simple drawings in Figure 7. Figures 7(a) and (b) are topologically the same and the slight difference in **the** 2D shape of the lower two regions results in different perceptions "cube" and "trapezoidal block". In fact, even such simple figures as Figures 7 (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 aspects. but until recently very little has been formulated in a manner usable for machine vision.

Kanade [15] demonstrated a systematic method *to* recover 3-dimensional shapes from a single view by mapping image properties into shape constraints. The 2D shape property of Figure 7 (c) and (d) is one of the properties he studied. It was named ***skewed symmetry***, because they are interpreted as symmetric figures viewed obliquely. In other word, whereas in the usual symmetry reflective correspondences are found along the direction perpendicular ***to*** the symmetry axis, in the skewed symmetry it is is found along the direction not necessarily perpendicular *to* the axis, but at a fixed angle to it. Formally. such shapes can be defined as 2-D Affine transforms of real symmetrics.

There are 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.

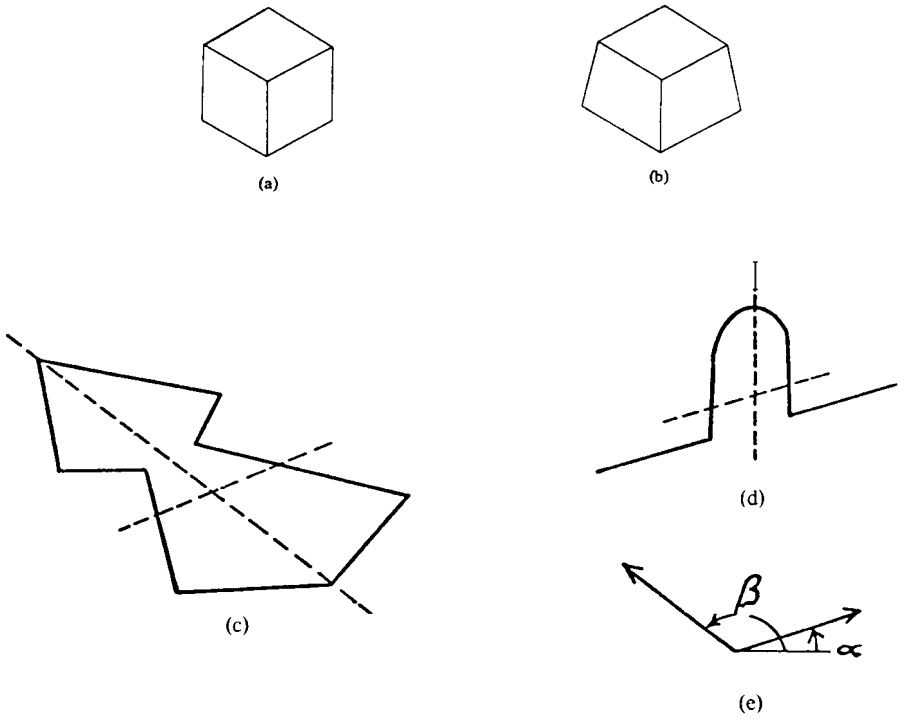


Figure 7: (a)(b) Simple line drawings: (a) "cube": (b) "trapezoidal block"; (c)(d) skewed symmetry: (e) **axes** of skewed symmetry *of* (c)

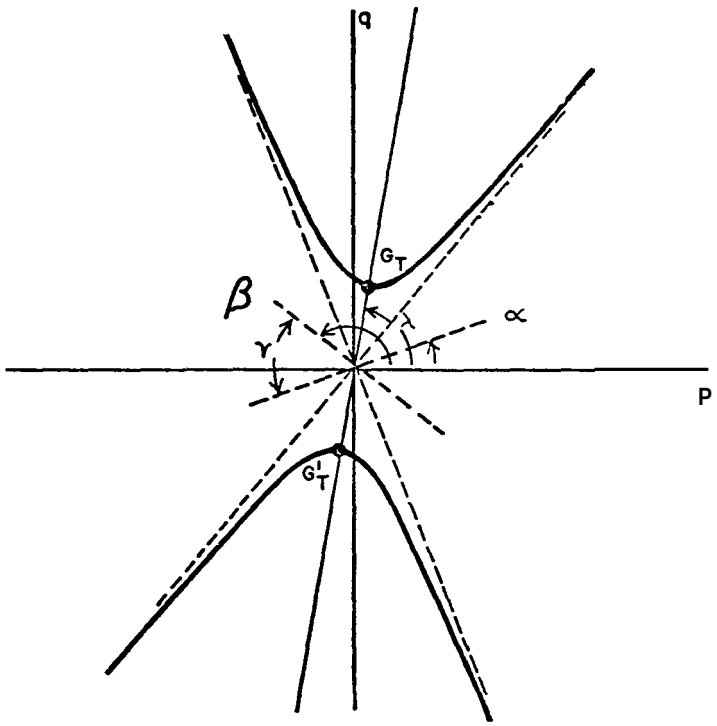


Figure 8: The hyperbola of eq. (8) corresponding to Figure 7(c). The axis of the hyperbola is the bisector of the obtuse angle made by α and β. The asymptotes make the same angle as the acute angle made by α and β. The tips or vertices G_T and $G_{T'}$ of the hyperbola represent special orientations with interesting properties. Especially, since they are closest to the origin of the gradient space. and since in general the distance from the origin to a gradient (p,q) 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.

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. As shown in Figure 7(c), a skewed symmetry detines two directions: let us call them the skewed-symmetry axis and the skewed-transverse axis, and denote their directional angles in the picture by α and β, respectively. Let $G=(p,q)$ be the gradient of the plane which includes the skewed symmetry. In general the 2-D unit vector e in the direction γ is $e=(\cos\gamma,\sin\gamma)$. From (3), the 3-D vector corresponding to c on the plane in the space is given as:

$$u_\gamma=(\cos\gamma.\ \sin\gamma.\ -(p\cos\gamma+q\sin\gamma)). \tag{7}$$

The assumption about the skewed symmetry demands that the two 3D vectors u_α and u_β be perpendicular in the 3D space: i.e., their inner product vanishes, $u_\alpha \cdot u_\beta = 0$, or:

$$\cos(\alpha-\beta)+(p\cos\alpha+q\sin\alpha)(p\cos\beta+q\sin\beta)=0. \tag{8}$$

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 bisects of *the* anglc made by the skewed-symmetry and skewed- the skewed-symmetry and skcwcd-transverse axes, it is easy to show that (8) represents a hypcrbola in the gradient spacc shown in Figurc 8. That is. the skewed symmetry defined by α and β in the picture can be a projection of a real symmetry *if and only if* its surfacc gradient (p,q) is on this hyperbola. The skcwcd symmetry thus imposes a one-dimensional family of constraints on the underlying surface orientation. Figure 9 illustrates how this skewed symmetry constraint can be used to rccovcr the shapc of "cube" from the image.

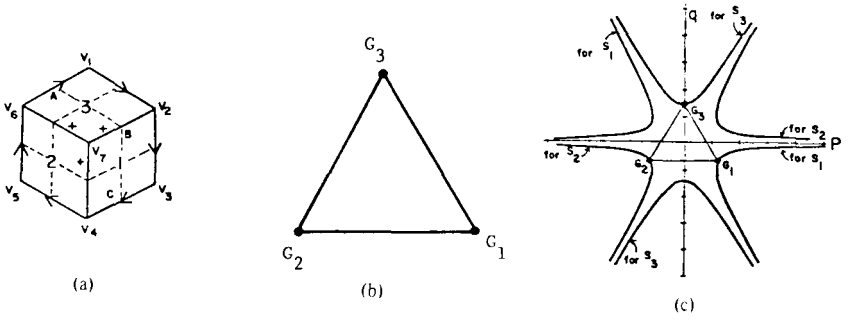
The same approach was extended to other properties: parallel lines, affine-transformable patterns. and textures [17]. We can summarize the assumptions used in these cases: ***regular properties observable in the picture are not by accident, but are projections of some preferred corresponding*** 3-0 ***properties.*** Figure 10 lists instances of this principle **of** *non-accidental image regularity.* Note also that the principle used in classifying the edge types in the previous section is also the same effect: the cohcrcnce in the image reflects the real cohcrcnce. Since the mapping from the picturc domain to the Scene domain is one-to-many (ambiguous), we nccd to rely on this type of general assumptions or task-specific constraints to rcsolvc the ambiguity.

3.2.2 Shadow

Shadow gives good clues on spatial relationship between objccts and surfaces. Aerial photo interpreters makc much use of it in figuring out. for cxamplc. the hcight of the objccts. Lowe and Binford [19] dcmonstratcd the reconstructing of the shape of an airplanc by using shadow information. They first paired shadow-making cdgcs and castcd-shadow cdgcs on the ground by assuming the ground planc and the sun anglc. Then the hcight of the airplane along the contour made of shadow-making cdgcs was estimated, which gave the shapc of the airplanc.

Shafcr and Kanadc [31] invcstigatcd a general and compact gradient-space rcprcscntation of geometrical constraints givcn by shadow. Figurc 11 shows a basic shadow prohlcm. It consists of the parallel-light illuminator I, the shadow making (occluding) surfacc S_o and the shadow surfacc S_s. 'l'hc problems includes six parameters to bc cornputcd: the gradicnt $\mathbf{G}_o = (p_o, q_o)$ of S_o, the gradicnt of $\mathbf{G}_s = (p_s, q_s)$ of S_s and the dirccticn of illumination (p_I, q_I).

This problem can bc studicd by considcring two othcr surfaccs S_{I1} and S_{I2} (and thcir gradients $\mathbf{G}_{I1}$ and $\mathbf{G}_{I2}$), cach of which includcs a pair of shadow making and castcd shadow cdgcs: for cxarnplc, E_{o1} and E_{s1} define S_{I1}. Notc that S_{I1} and S_s makc a concave cdgc along E_{s1}, and so do S_{I2} and S_s along E_{s2}. Now thrcc constraints are provided from the basic shadow problcm gcomctry: 1) 'l'hc anglc $\mathbf{G}_o$-$\mathbf{G}_{I1}$-$\mathbf{G}_s$, which comcs from the anglc E_{o1} - E_{s1}; *2)* 'l'hc anglc



(a) (b) (c)

Figure 9: (a) A labeled linc drawing: the dotted lincs show the axes of skcwcd symrnctry. (b) The constraints on the gradients of the thrcc surfaccs due to thcir intcrconnccction: the gradients form a triangle in the gradicnt spacc whose shapc and oricntation should as shown. but the location and the size arc arbitrary. (c) 'l'hc hypcrholas shown corrcspond io the skcwcd symmetries of the thrcc rcgions. 'l'hc prohlcm is thus how *to* placc the trianglc of (b), by translation and scalc changc. so that so that cach vcrtcx is on the corrcsponding hypcrbola. The locations shown is proven to bc the only position. and the resultant shapc is a cubc.

$\mathbf{G}_o$-$\mathbf{G}_{12}$-$\mathbf{G}_s$, which comes from the angle between F_{o2} and F_{s2}; 3) The direction of the line l_{illum} (containing $\mathbf{G}_{11}$ and $\mathbf{G}_{12}$), which comes from the direction of F_{11} (line containing the two vertices 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 geometry. Figure 12 shows a construction for the case that the direction of illumination (actually the relative depth component of illumination vector - one parameter - is given) and the the orientation of shadow casted surface ($\mathbf{G}_s$) are known.

The basic shadow geometry provides three constraints, and thus three parameters have to given by other means to solve the problem. It is interesting to compare this situation with the situation without shadows: an image which only depicts S_o and S_s intersecting along F_{os}. Here. there are four parameters ($\mathbf{G}_o$ and $\mathbf{G}_s$) to compute, and one constraint from the image (E,). so three pieces of information are still needed in advance. With shadows, the Same number of a priori parameters are needed, but one of them can be a description of the light source position instead of a description of a surface orientation. 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.

3.3. Texture

Perception of depth and surface from texture gradient has been studied by psychologists. Recently, Kender [18] developed a very powerful computational paradigm *for* shape from texture. His central idea is a normalized texture property map (NTPM), which is again for mapping image features (texture in this case) into scene properties.

Let us show an example of NTPM for length. Suppose we have a texture pattern shown in Figure 13. which is made *of* line segments with two orientations: the horizontal ones with length L_2 and the diagonal ones **(45")** 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 3-dimensional line in the scene? From (7). $l\,\mathbf{u}$ is the corresponding **3D** vector, and its **3D** length is

$$L(\gamma,l) = l|\mathbf{u}| = l\sqrt{1 + (p\cos\gamma + q\sin\gamma)^2} \tag{9}$$

$L(\gamma,l)$ is the NTPM for a line segment with slope γ and length l. As with the reflectance map. we can represent (9) as a set of contours in the gradient space, $L(\gamma,l) = L$, 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 A are *of* length L. Figure 14 shows the NTPM for $\gamma = 45^0$. 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 (eg, 2D line length) comes from it. In other word, it represents *deprojected* scene properties from image.

Picture	Scene
Parallel in **2D**	Parallel in **3D**
Skewed symmetry	Symmetry
Similar color edge profile	Similar physical edge property
Matched **T**	Interception
Affine-transformable patterns	Similar or congruent patterns
Gradient in	Regularity in
--Spacing	Spacing--
--Length	Length--
--Density	Density--

Figure 10: Instances of the principle of non-accidental image regularity. Notice that the rules are mostly true when going from the scene to the picture. but the other direction is heuristic.

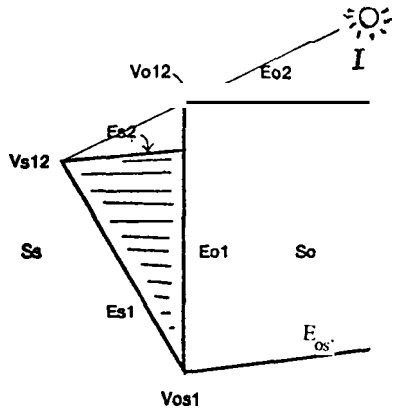


Figure 11: Basic shadow problem: the suffixes are given to show the correspondences; for example, E_{o1} is a shadow making edge and E_{s1} is a corresponding shadow edge.

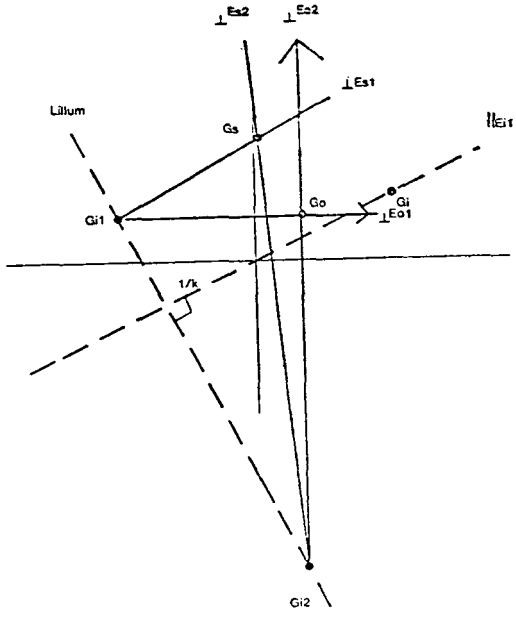


Figure 12: The construction in the gradient space for computing G_o proceeds as follows:

1. Draw the line parallel to E_{s1} through the origin. Plot the given G_1 (it should be on this line). Let k be the distance from the origin to G_1. Draw the line L_{illum} so that it is perpendicular to E_{11}, opposite to G_1 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_{11}. Through G_{11}, draw a line per 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_{12}. 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 be G_o.

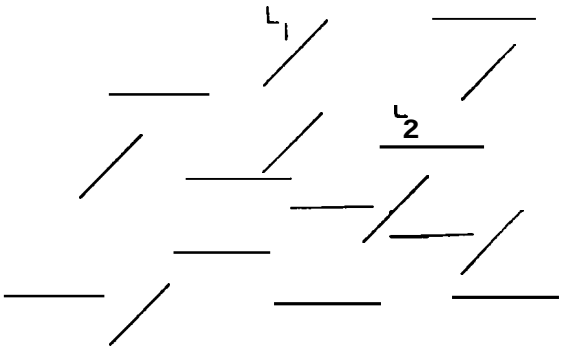


Figure 13: A texture image of line segments. It is assumed that all line segments are on a plane surface and the their real length are the same. What orientations are possible?

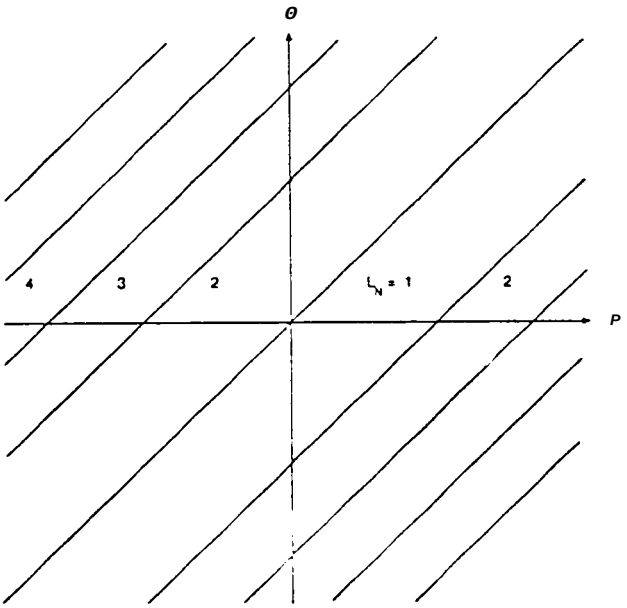


Figure 14: A normalized texture property map for length $L(45^o,1)$.

In Figure 13. if we assumc that all the linc scgmcnts are on the same planc and arc of the samc lcngth in the space, the possihlc oricntations of the surfacc can be obtaincd by

$$I_1(0, L_2) = I_1(45^o, L_1) \tag{10}$$

or hy graphically intcrsccting the NTPM surfaccs. This is hypcrbola in the gradient spacc. Though this is a very simple example, the same tcchniquc can hc uscd to estimatc the surfacc orientations of walls **of** buildings in outdoor scenes or of mechanical parts with textures.

Wc can observe a striking similarity bctwccn the casc of tcxturc and shading. Wc can makc the correspondence bctwccn

Shading	Texture
pixcl	texel
imaging set-up	sccnc constituent
reflectance map	NTPM

In fact, as in shape from shading, shape recovery of a curved surface which has a textured pattern on it can bc performed by assuming surface smoothness and regularity in the texture. [17] [11]

3.4. Remarks

'I hc rcchniqiics and approaches in obtaining and representing constraints from the physical level of knowlcdgc arc suggestive to othcr arcas of robotics which dcal with sensing and interpreting natural environments. I'hc key idca is to modcl thc projcction process and represent it in a form that can be uscd for invcrsc projcction in conjunction with other constraints involved in interpretation. It is noteworthy that thc forward projection rules used in Section 2 (reflection or projective geometry) arc simple and fairly well understood. But they arc oftcn so local or microscopic that thcir direct application may rcsult in gigantic unmanipulative cqiiations. hppropriatc representational spaces, such as gradient spacc. enable them to be applied in a macroscopic manner.

4. 3D Model-Based Vision

'I hc thcorics presented in the previous section extract natural constraints undcr reasonable physical, mostly task independent assumptions. In this sense, those methods are modcl weak. Model-based vision attcmpts to usc **task**-dcpcndcnt semantic constraints. This idea is not new. In fact. *ad* hoc methods can be regarded as modcl based. **Also** two dimensional relationship among parts of objccts, rcprcscntcd as cithcr graphs or procedures. have been uscd to do direct two-dimensional pattern analysis such as face rccognition. chest X-ray analysis and region segmentation **of** aerial photos. Hut xhat wc aim at is 3D model-based vision which can cope with difficulties due to variations **of** object shapes and vicw angles. Although such pcrfcct systems are probably still far away, we can see important progress.

An early atrcmpt in model-based vision [6] used a prcdctcrmincd set of models with fixed shapes to validate the hypothesized recognition result. Given an impcrfcct linc drawing. the systcm cxtractcd features and dccidcd what objects appeared in what anglcs in the scene, then gcncratcd a linc drawing anticipdtcd if the recognition result was correct. It was comparcd with the input. and if thcy wcrc close the rccognition result was correct. otherwise other possibilities wcrc pursued.

3-dimensional shapc modcls can be also uscd to predict 2-dimensional appcaranccs **of** objects beforehand under various lighting conditions. Then intcrprctation consists of mostly verification (*verification vision*). An cxamplc chown in Figure 15 is especially interesting bccausc thc appcarances **of** *stable* postures of a part arc computed from its 3-dimensional geometric model. [34] First. thc cnnvcx hull of thc ohjcct **is** obtaincd, and cach of its surfaces is tcstcd whether it provides a stable support of the object.

A general model-based vision systcm should have variable or gcncric modcls of objccts and interpret an input image as *instances* of the generic model (recall Figure 1) to rccognizc the spccific objccts (e.g. sizc, shape) which appear

in a specific manner (e.g. orientation, distance). The basic difficulty here is the difference in the coordinate frames used in the generic models and in the representation obtained from images. The generic models are (and should be) described by viewer-independent representations, whereas the image features and surface representation obtained from images are viewer-dependent. Somehow the two types of representations must be made compatible. Most of the previous systems did this either by quickly hypothesizing the parameters from a very small set of image features or by generating 2-dimensional appearances so that they could work totally in the picture domain. Both were possible because the shapes were fixed and limited.

What is needed is a more systematic way to access the model and to reason about the parameters. The ACRONYM system at Stanford [4] uses invariant and pseudo-invariant features which are predicted from the given object models. In ACRONYM. an object is modeled by its subparts and their affixment (spatial relationships). Volume primitives to represent subparts are generalized cones: A generalized cone is a volume swept by moving a cross section along a 3-dimensional curve (called a spine). When a cross section is a round disc and it is moved along a straight line with its size varying linearly, a ordinary cone is generated. Parameters that specify the generalized cones and their affixment are given not only by specific values but also by free variables with which range and mutual constraints can be associated. In this way one can represent generic object models which allow variations in size and shape. The image features that ACRONYM uses are ribbons (two dimensional stripe) and ellipses, which can provide an approximation as the projection of the straight spine and the circular cross section. respectively, of the corresponding generalized cylinder.

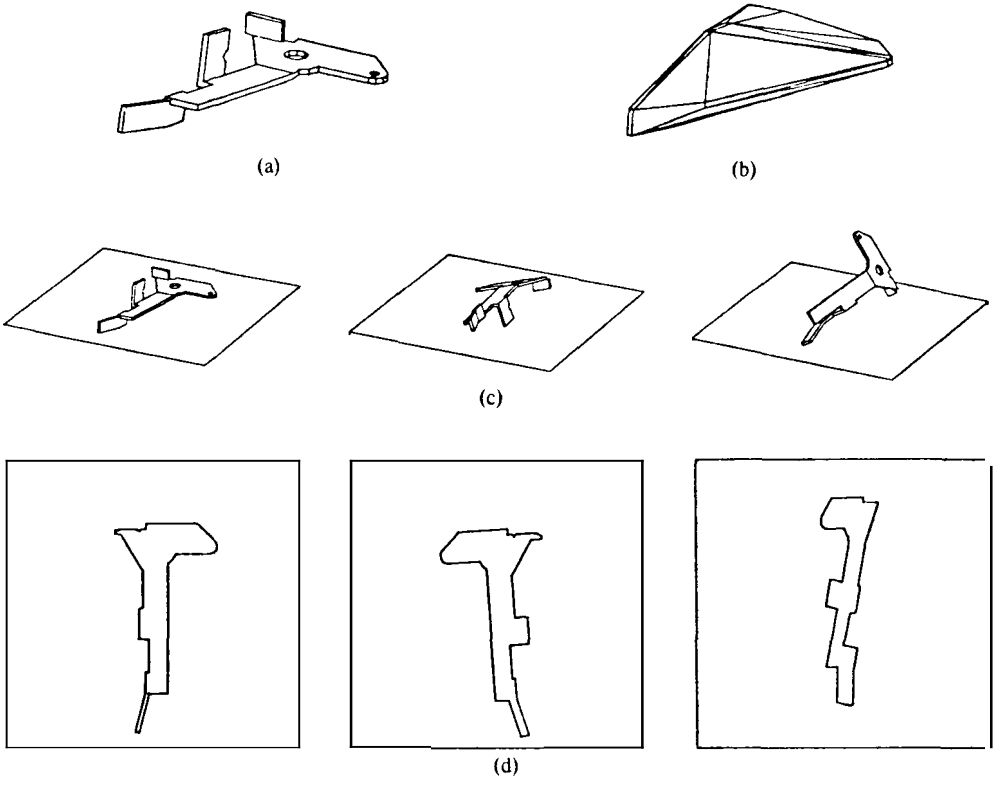
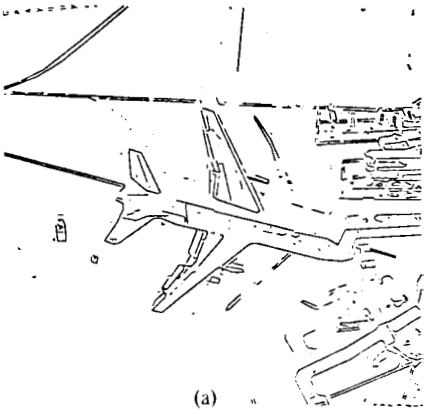
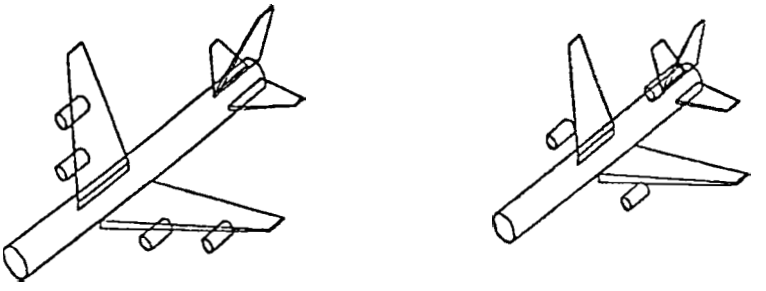


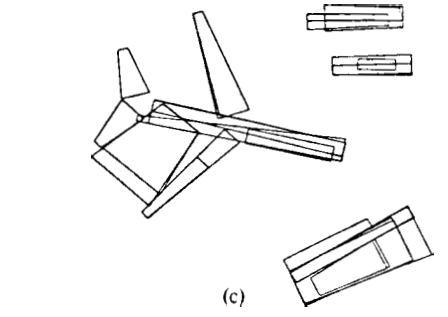
Figure 15: Anticipating the appearances of a parts in stable positions. (a) 3D model of a pan: (b) convex convex hull; (c) computed stable positions; (d) silhouettes of stable postures viewed from overhead.



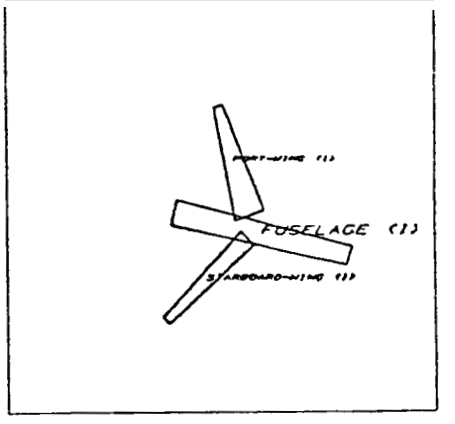
(a)



(b)



(c)



(d)

Figure 16: An example of analysis by ACRONYM: (a) Line segments for the input image; (b) instances of class models of Boeing-747a and L-1011s;(c) ribbon description: (d) an L-1011located.

One important idea is to predict from the model invariant and pseudo-invariant features of an object. which will be invariantly observable in the image over the modclcd range of variations. They include parallclism, collinearity, and the length ratio and anglcs between spines. (Hcrc we see an application of the principle of non-accidental regularity.) They provide a coarse filter for hypothesizing possible objects. For hypothesized objects, the unknown parameters (variations of object shape and distance and orientations of the object) are estimated by the algebraic constraints obtained by associating the observation (such as length and angle in the image) with the anticipated range of its value.

Figure 16 shows an example scene: (a) is the line segmented image extracted from input. (b) shows the instances of modcls of a Boeing-747 and an L-1011; after extracting ribbons (shown in (c)) from the line segment image, an L-1011 was identified as shown in (d). This deduction is not based on the size of the image, but on the relationships between the subparts, such as wingspan-to-fuselage-length.

5. Construction of Scene Description

The goal of the visual process is to construct the description of the environment which is sensed by visual methods: it must be recognized that this is different from classifying parts of an image, detecting the objcct existence or scgmcnting the imagcs. The diffcrcncc 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.

At CMU. Kanade and Hcrman [16] arc dcvcloping a system callcd Incrcmcntal **3D** Mosaic. which builds a scene description from a scqucncc of (stcrco) imagcs. It is currently applied to an urban scene for building a 3-dimcnsional model of the target area from low-altitude aerial photos. **A** single pair of stereo photos can give only partial information on the scene: only limited portions are visible in both images and the stcrco system cannot be perfect anyway. The information must be incrcmcntally accumulated into a consistent dcscription as new imagcs arc available. The information on the portions of the sccnc which become visible should be added, and the information on the previously known portion should be uscd to improve the description. either by correcting errors or by increasing the confidcncc and precision.

Figurc 17 shows a typical sequcncc. Wc havc a stcrco pair of imagcs (Figurc 17(a)). Lines and junctions are extracted from thcm (Figurcs 17(b)). Then, junctions and lines are matchcd. Thcy arc mostly corners and cdgcs of buildings. Here we use the fact that buildings arc mostly block-shaped, and thcy havc (gravitational) vertical cdgcs. This allows one to hypothesize the changes of junction appcaranccs along the cpipolar linc of stcrco. and to cope with the difficulty in matching wide-angle stcrco imagcs with large disparity jumps. such as urban scenes. Wc can compute the 3-dimensional locations of the matched junctions and lincs, thus forming wircframcs. Figurc 17(c) shows their perspective vicw.

The wireframe representation docs not yet identify surfaccs. The ncxt stcp is to perform reasoning on surfaces. This is done by assigning planar surfaces so that an enclosed object is obtained with the wireframes as cdgcs. The proccss is similar to obtaining solid objects from wircframcs [21], but we assume that our wircframcs arc not always complete. Figurc 17(d) shows a pcrspcctive vicw of the constructed blocks objccn. Once we have such a dcscription, we can crop imagc patches from original imagcs to know the normalized appcaranccs of surfaccs (e.g. window patterns). **A** natural looking display can bc generated for the scene viewed from any angle by appropriately transforming such appcaranccs according to surface oricntations. Figure 17(e) is an cxamplc. Notice that parts of surfaccs which wcrc not visible in the original imagcs are displayed distinctively as such (i.e. red color).

This kind of description is useful, for cxamplc. for planning the angles that the ncxt imagcs should take: it is generally better to cover as much of the "red" portions as possible to increase the knowledge of the target area. This scenario of 3DMosaic is applicable to robot navigation and to change detection in a scene.

6. Organization and Control of Vision Systems

A key attribute of an image-understanding system is the interaction between high-level knowlcdgc---object models---and low-level knowledgc--- image or scene features. While the general flow of information is *bottom-up*, from pixcls to image features. to scene fcatures. to object labeling, many systems also have some it *top-down* information flow from objcct modcls to image fcaturcs. In the face rccognition program by Kanadc [13], for cxamplc. a model of the arrangement and intcnsity characteristics of face components guides all of the low-level processing. After finding

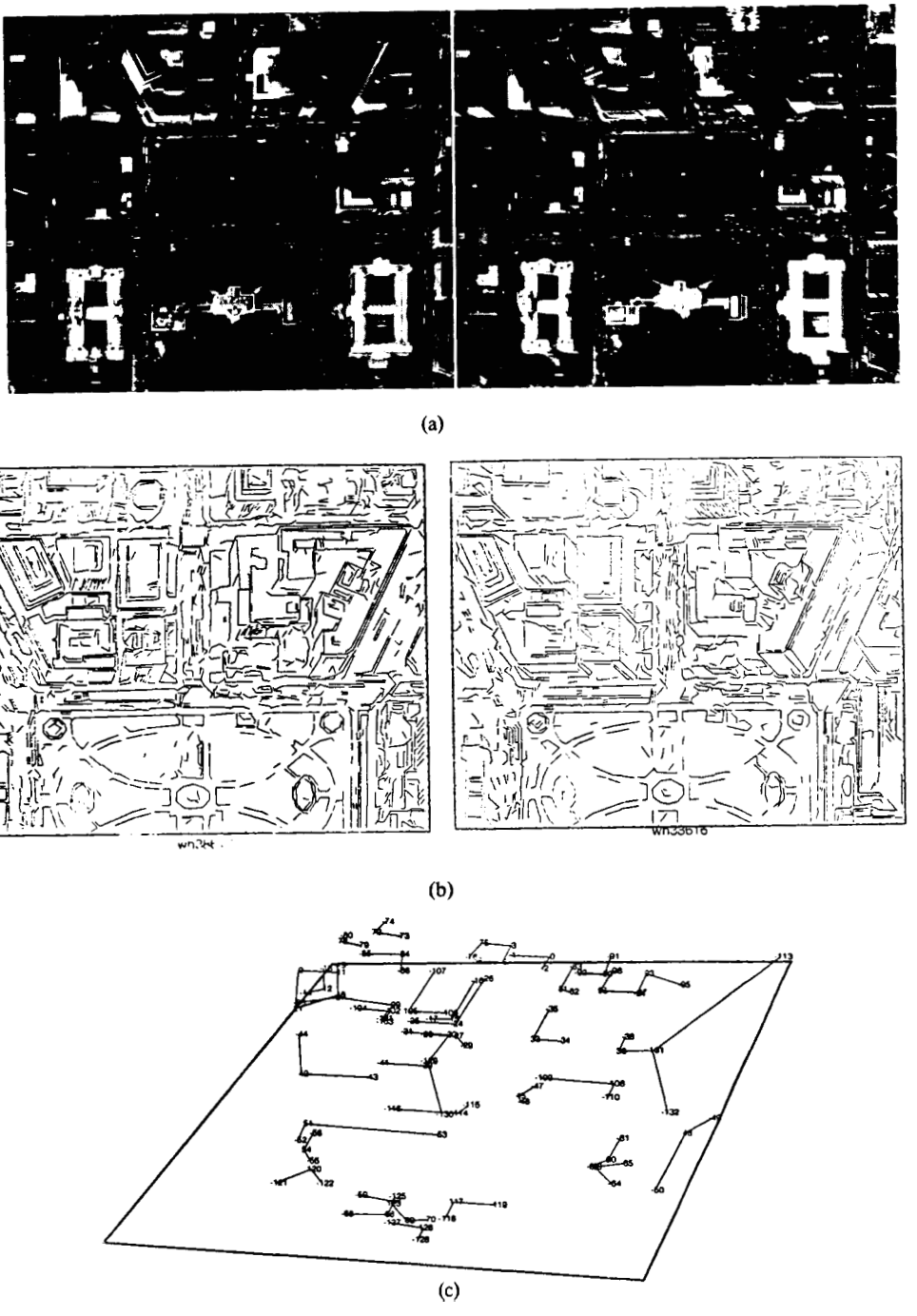
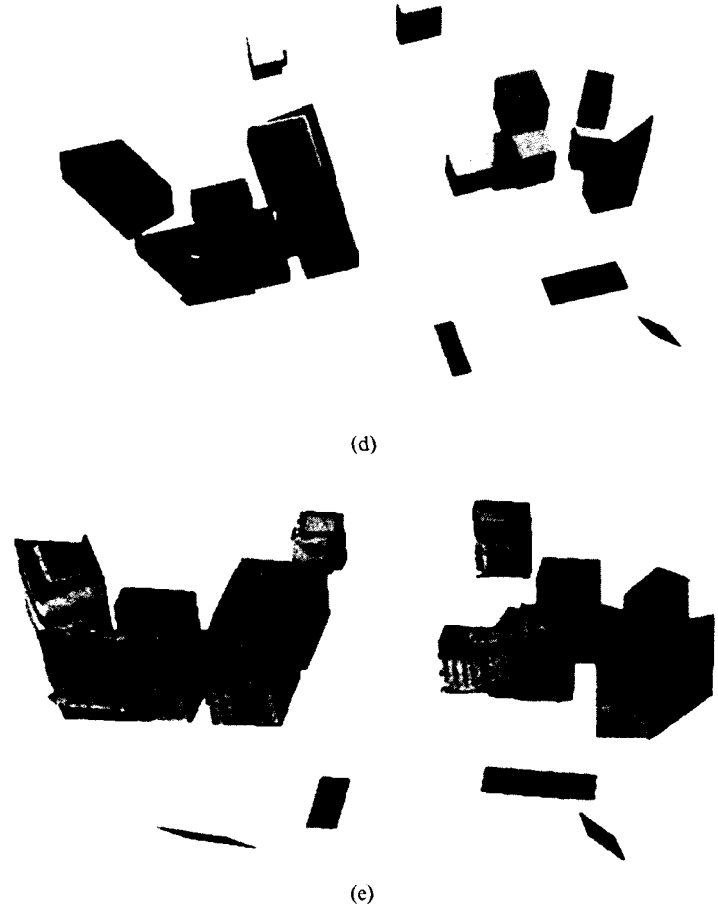


Figure 17: 3D Mosaic: (a) lrnagcs of right and left views; (b) linc segments cxtractcd; (c) perspective vicw of 3-1) wircfrarncs which arc madc by edges obtaincd by stcrco matching: (d) planc-surfaced modcls of buildings: (e) synthesized imagc of the scene from the anglc diffcrcnt from original angle.



(d)

(e)

Figure 17: (continued)

the outline of the head, the program estimates the probable position of the eyes, and look for the dark spots that characterize the pupils in and around the predicted locations. When it fails in finding components with such characteristics, it goes back to the previous steps. assuming a certain error there, and try another possibilities. This ***model-driven*** processing can be both efficient and effective. However, programs that depend very much on high-level control of low-level processing tend to be too domain-dependent and respond poorly when viewing conditions change even slightly. In this section we will examine the organization and control of three different types of vision systems. The following descriptions focus on mechanisms for achieving cooperation and flow of control between low-level and high-level processing stages.

6.1. Production System Organization for Outdoor Scene Analysis

Ohta, Kanade and Sakai [27, 26] developed a semantic region analysis system for outdoor scenes. Given a color image of outdoor scene, the system assigns semantic labels. such as tree and buildings. to regions. As shown in Figure 18 the system consists of bottom-up part and top-down part. Initially the bottom-up part first segments the image into

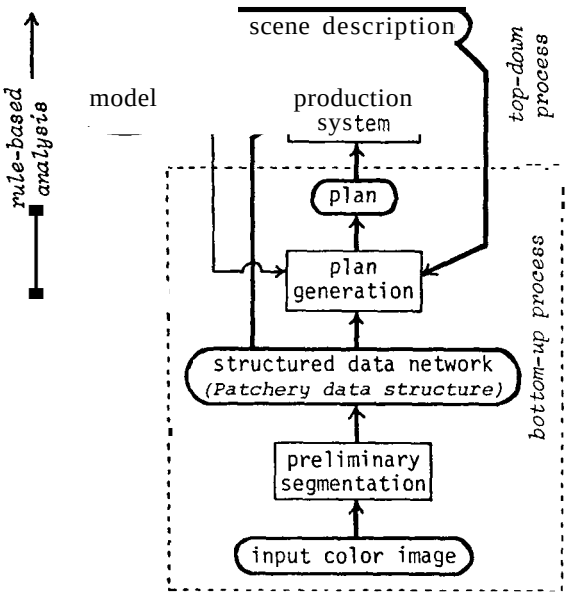


Figure 18: Organization of the region-based scene analysis program by Ohta, Kanade and Sakai

homogeneous regions by an Ohlander-type region segmentation method [28]. The segmentation result is described as a patchery data network which stores properties of all the patches extracted.

Then a *plan* is generated from it by selecting regions by large area, called keypatches. After merging small patches adjacent to keypatches, the system tentatively assigns a set **of** object labels with corresponding estimates of correctness which are computed first from the unary properties of the keypatches, such as size, shape, and color, and then from the binary relations between them, such as relative positions.

Figure 19 shows an example scene: (a) is an input color image, (b) is the result of preliminary segmentation, and (c) is the plan image. The first plan obtained by using only the unary property rules is shown in Figure 20(a), and Figure 20(b) shows a revised one by by using the binary relations.

The top-down process then starts symbolic interpretation of the image by analyzing detailed structure of the scene in the context given by the plan. (The system still can change the interpretations in the plan, in which case the bottom-up process is re-activated.) The analysis uses a production system organization with knowledge of the world (outdoor scene) represented as a set of rules. Each production rule has a condition part and an action part. The condition part is made of fuzzy predicates on properties of and relations between regions. The action part is a set of actions to manipulate the database (patchery network, plan, and scene description) to build the scene description. Each action is described **as** a form in Lisp. There are TO-DO and IF-DONE rules, corresponding to the antecedent and consequent theorems of **PLANNER.**

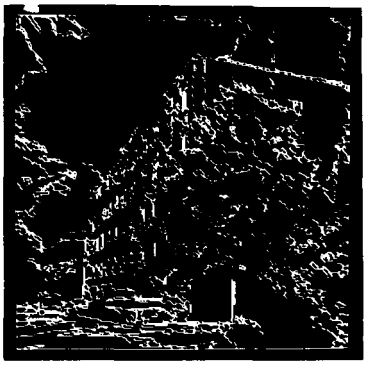
The world model is described as a network of *knowledge* ***blocks*** *(KB)* which define the objects, materials, and concepts in the given task world. The production rules are divided into subsets according to the roles they play in the analysis: For example, the subset for the scene phase analysis is stored in the KB *SCENE* and the subset to analyze the "**sky**" is stored in the KB ***SKY.***

Control of the production system is handled by an agenda which registers all the executable rules (i.e., whose condition parts are satisfied). The analysis iterates the following three steps:

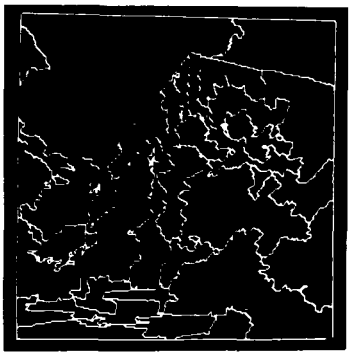
1. An executable action with the highest Score on the agenda is executed. A patch or a set of patches is interpreted and the database is modified.
2. If a keypatch is interpreted in step 1. the control enters into the Scene phase. The production rules included in the KB ***SCENE*** are activated to (re-)examine the keypatches not yet interpreted. The scene phase. in general. considers the overall structure of the Scene, such as the location of horizon and relationships among objects.
3. Otherwise the control enters into the object phase Corresponding to the object as which the patch has been just interpreted in step 1. The production rules in the corresponding KB are activated. This phase mainly analyze the local structure related to the particular object.

Figure 21 shows how the plan is modified as the analysis proceeds: (a) when the horizon is determined, and (b) when the outline of the building is extracted. Figure 21(c) shows the final labeled interpretation.

(a) digitized input scene



(b) result of preliminary segmentation



(c) plan image

Figure 19: An outdoor scene:

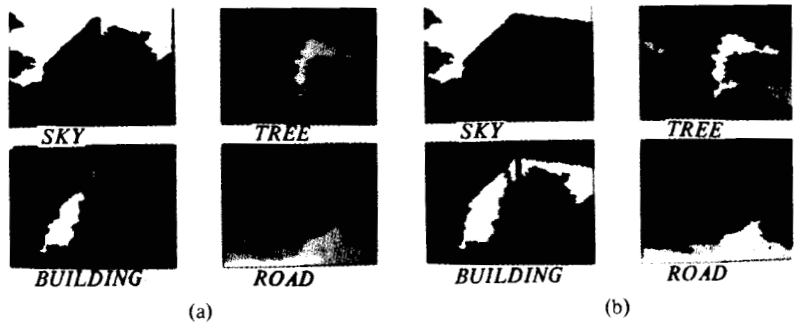
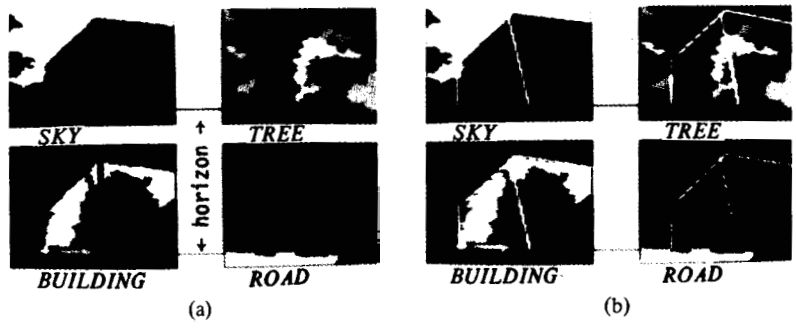


Figure 20: Plan The brightness shows the correctness of the assignments: (a) by use of unary properties of regions; (b) after using binary relations



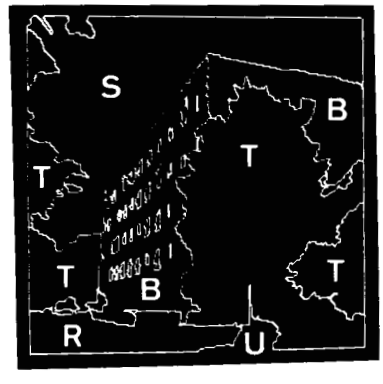


Figure 21: (a) Plan when the horizon is determined. (b) Plan when the outline of the building is determined. (c) final result of interpretation: S: *SKY*; T: *TREE*; R: *ROAD*; B: *BUILDING*; U: *UNKNOWN*

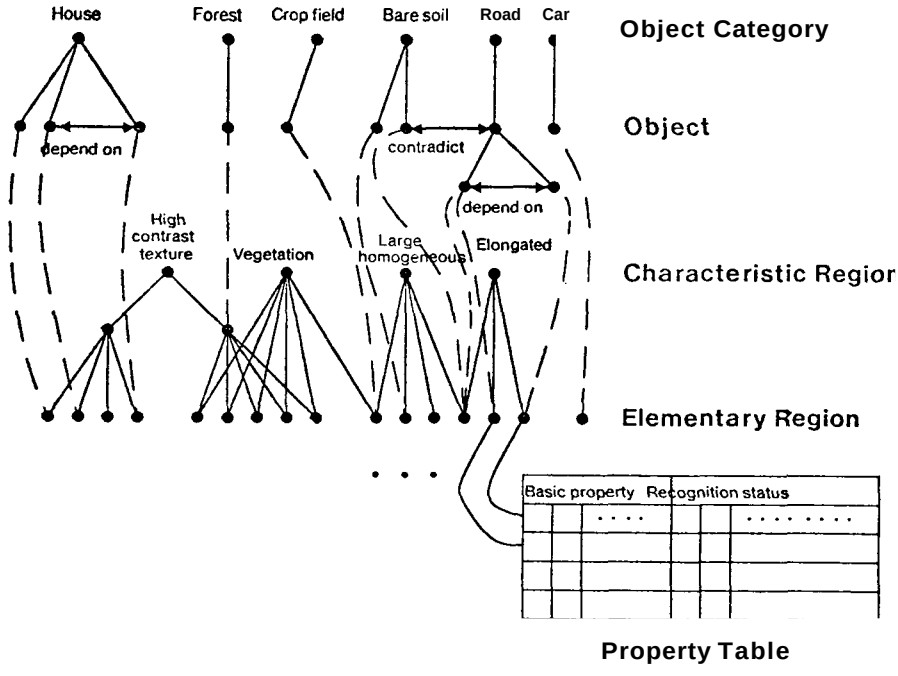


Figure 22: Structure of a blackboard of the system by Nagao, Matsuyama and Ikeda

6.2. Blackboard Architecture **for** Interpreting Multiband Aerial Photographs

Nagao, Matsuyama, and Ikeda [24, 25] developed a system that interpreted a class of multiband aerial photos. Their image-interpretation system employs multiple. independent knowledge sources that operate on a common, multilevel database. This database. or ***blackboard.*** is represented as shown in Figure 22. The abstraction levels of image information are ***elementary region. cue region. object,*** and ***object category***. Models are described in terms of two-dimensional features that can be observed in images. In general. it is not possible to do scene interpretation with two-dimensional models, but it is an acceptable technique for aerial photography because the view angle is so constrained that object shapes change little and occlusion is not much of a problem.

The first step of processing is **to** smooth the image. **A** nonsemantic segmenter defines a set of elementary regions---a set of patches that are homogeneous in multispectral properties.

The next step is to extract cue regions. The types of cue regions are large homogeneous regions. shadow and shadow-causing regions. elongated regions, vegetation regions. high-contrast regions. and high-contrast vegetation regions. Each type of cue region triggers one or more object recognizers. Different cue regions may overlap; for example. high-contrast vegetation regions are simply the intersection of high-contrast regions and vegetation regions.

Cue regions are extracted by screening elementary regions: for example. any patch with very low intensity. particularly in red and infrared. is classified as a shadow. An adjacent region with an appropriate boundary on the sunward side is a shadow-maker. Vegetation regions have a high ratio of infrared to red; high-contrast areas are aggregations of small elementary regions. Shadow-making regions trigger the house detector. while high-contrast vegetation regions are likely to be considered forest.

Each elementary region is represented by a node in the lowest level of the blackboard. Nodes at higher levels represent cue regions and objects; they are linked to the elementary regions they subsume. Furthermore. a node can

have a dcpendcncy link to another node, indicating that its intcrprctation was aided by the prior intcrprctation of the other node.

The property table shown in Figurc **22** stores the coordinate rangc. or *bounding rectangle*, of **a** rcgion and records whcthcr the rcgion is *unanalyzed, recognized, irregularly* ***shaped.*** or *it rejected.* Each rcgion has only one cntry. which means that there can be only one objcct hypothesis for a rcgion. The first intcrprctation of a rcgion is kept until a contradiction arises. To rcsolvc contradictions. the systcrn dclctcs the conflicting rcgion interprctation for xhich it is least confidcnt. It marks the rcgion as unanalyzed, restarting the intcrprctation of the rcgion; ohjcct hypothcscs that dcpcnd on the dclctcd node arc thcmsclvef deleted.

6.3.Schema-based System: VISIONS

'Thc Univcrsity of Massachusetts' VISIONS systcrn[8] is pattcrncd after the HEARSAY--II speech-understanding systcrn[5]. In VISIONS, hypothcscs are posted and accessed on a *blackboard* by indcpcndcnt procedural ***knowledge*** *sourceds; KS* Their activation and scheduling arc under the control **of** a central executive. The systcrn has bccn tcstcd with outdoor scenes. Figurc **23** outlines the structurc **of** VISIONS.

The blackboard in this systcm rcprcscnts a laycrcd description of the contents of an irnagc. The lowest levels rcprcscnt ***regions,*** *segments,* and *vertices;* thcy form a structure callcd an *RSV graph.*

Preprocessing stages arc shown in the lcft half of Figurc *23.* There arc thrcc stages of information representation. The first is the image itself, represented by a resolution pyramid (processing cone). 'Thc second stage comprises separate edge and region analysis. **The** third stage is a merged reprcscntation of the results of a corrclation hctwccn the edge analysis and region analysis. The representations at these **low** lcvcls arc of image characteristics, rather than of Scene characteristics.

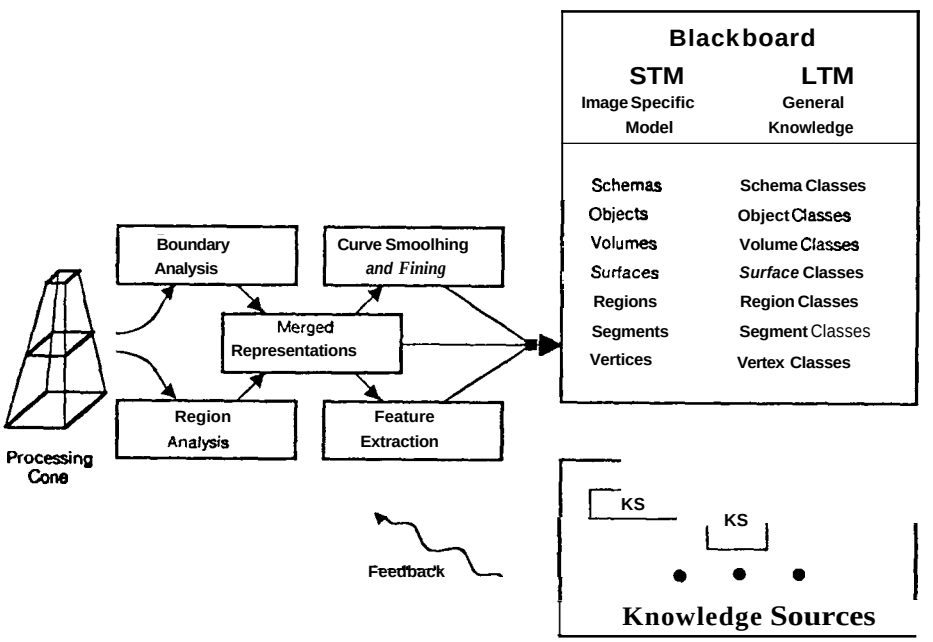


Figure 23: Overview of VISION systcm by Hanson and Riseman

The next two levels in the blackboard work with surfaces and volumes. At these levels, the system attempts to reconstruct the three-dimensional configuration of the scene. The top two levels work with representations of objects and schemas. At the object level, hypotheses are formed about what the objects in the scene must have been to result in the observed image. The schema level imposes constraints on the selection of object models. There may be office schemas, airport schemas, and so on. Schemas serve the same purpose as Minsky's frames [23].

The blackboard model in Figure 23 illustrates the distinction made in VISIONS between a priori models and image-specific models, though both may be represented in the same manner. The a priori models are stored in *long-term memory* (LTM), while the image-specific models are stored in *short-term memory* (STM). Recognizing that they did not have adequate KSs to make surface and volume hypotheses reliably, the designers of VISIONS compensated by relying heavily on top-down hypotheses represented by models in LTM. By projecting these models into two dimensions, they construct RSV-level models of objects, and these are matched to the actual image.

VISIONS chooses a KS by traversing a decision tree. Its *model builder* decides to expand or to develop a new hypothesis for a model. To expand a model, the *level focuser* first decides which level of the blackboard to work on. Then, that level is expanded under the control of the *node focuser*, the *node expander*, and the *node verifier*. The focuser selects a node from the blackboard to process further, the expander calls a KS to create new hypotheses, and the verifier checks the results for satisfaction of constraints.

6.4. Remarks

Some other interesting computer vision systems that interpret natural scenes include: Shirai [33] (procedural representation), Rubin [29] (constraint network representation) and Ballard, Brown and Feldman [2] (query-oriented analysis). The image/map database **MAPS**, being developed at CMU by McKeown [22], attempts to use multi-level multi-source knowledge (such as terrain map and cultural maps) for photo interpretation.

7. Conclusion

We have first discussed a general structure of vision, and then identified important areas to work on for realizing a more capable and general vision system than current robotics vision. Emphasis has been put on:

- Computational aspects of vision
- Obtaining constraints from physical and semantic knowledge
- Multi-level representations
- Use of generic models
- Scene descriptions
- Control and information flow in the vision system

References

1. Artificial Intelligence, *Special Issue on Vision* North-Holland, 1981.

2. Ballard, D. H., Brown, C. M. and Feldman, J. A. An Approach to Knowledge-Directed Image Analysis. Computer Vision Systems, New York, 1978. pp. 271-281.

3. Barrow, H. G. and Tenenbaum, J. M. Recovering Intrinsic Scene Characteristics from Images. Computer Vision Systems. New York, 1978, pp. 3-26.

4. Brooks, R. A. "Symbolic Reasoning among 3-D Models and 2-D Images." *Artificial Intelligence 17* (1981), 285-349.

5. Erman, L. D.. Hayes-Roth, F.. Lesser, V. R., and Reddy. D. R. "The HEARSAY-I1 Speech-Understanding System: lntcgrating Knowlcdgc to Resolvc Uncertainty." *Computing Surveys* 12.2 (1980).

6. **Falk.** G. "Interpretation of Imperfect Line Data as a 'Three-Dimensional Scene." *Artificial Intelligence 3* (1972). 101-144.

7. Fischlcr. M. **A.** and Witkin, A. P. Recovering lnstrinsic Scene characteristics from Images. Tech. Rept. Interim Technical Report, SKI International, Nov., 1981.

8. Hanson, A. R. and Riseman, E. M. Scgrncntation of Natural Scenes. Computer Vision Systems. New York. 1978, pp. 129-163.

9. Horn, B. K. P. "Understanding Image Intensities." *Artificial Intelligence 8* (1977), 201-231.

10. Horn, B. K. P. and Sjoberg. R. W. "Calculating the Reflectance Map." *Applied Optics 18* (1979). 1770-1779.

11. Ikcuchi. K. Shape from Regular Patterns (An Example of Constraint Propagation in Vision). Tech. Rept. **A.I.** Memo 567. Massachusetts Institute of Technology, March, 1980. Artificial Intelligcncc Laboratory

12. Ikcuchi, K. and Horn, B. K. P. "Numerical Shapc from Shading and Occluding Boundaries." *Artificial Intelligence 17* (1981). 141-185.

13. Kanadc. T.. *Interdisciplinary Systems Research* Volume 41: *Cornpurer Recognition of Human Faces.* Birkhauser Verlag. New York, 1977.

14. Kanadc. T. "Region Scgmcntation: Signal vs Scmantics." *Cornpurer Graphics and Image Processing 13* (1980). 279-297.

15. Kanadc, T. "Recovery of the Three-Dimensional Shape of an Object from a Single View." *Artificial Intelligence 17* (1981). 409-460.

16. Kanadc, **'I.** and Hcrman. M. lncrcmcntal 3D Mosaic Systcm. Tech. Rcpt. (in prcparatron), Carncgic-Mellon Univcrsity Computcr Scicncc Department, 1982.

17. Kanadc. T. and Kcndcr J. R. Mapping Image Properties into Shapc Constraints: Skcwcd Symmetry. Affine Transformable Patterns and the Shape-from-Tecture Paradigm. Tech. Rcpt. CMU-CS-80-133. Carncgic-Mcllon Univcrsity Computcr Scicncc Department, July, 1980.

18. Kender, J. R. *Shape from Texture.* Ph.D. Th., Carnegie-Mellon University Computer Scicncc Department, 1980.

19. Lowe. D. G. and Binford, T. **O.** The Interpretation of Three-Dimensional Structure from Image Curves. Proc. 7th International Joint Conference on Artificial lntclligcncc, 1981, pp. 613-624.

20. Mackworth. A. K. "Interpreting Pictures of Polyhedral Scenes." *Artificial Intelligence 4* (1973). 121-137.

21. Markowsky, G., Wesley, M.A. Fleshing Out Wire Frames. Tech. Rcpt. RC 8124. IBM Rcscarch Division. February. 1980. IBM Thomas J. Watson Rcscarch Center, Yorkiown Heights. NY 10598

22. McKeown, D. M. Jr. Maps: Integrating Imagery, Terrain and Map Databases. Proc. IEEE Workshop on Applied Imagery Pattern Rccognition. Scpt., 1982.

23. Minsky. M. L. **A** Framework for Rcprescnting Knowledge. The Phychology of Computer Vision, New York, 1915, pp. 211-277.

24. Nagao. M., Matsuyama, T. and Ikeda, *Y.* Region Extraction and Shape Analysis of Aerial Photographs. Proc. 4th International Joint Conference on Pattern Rccognition, 1978, pp. 620-628.

25. Nagao. M.. Matsuyama. T. and Ikeda. Y. Structural Analysis of Complex Aerial Photographs. Proc. 6th International Joint Conference on Artificial Intelligence. 1979, pp. 610-616.

26. Ohta, Y. ***A Region-Orienied Image-Analysis System by Computer.*** Ph.D. Th., Kyoto University, March 1980.

27. Ohta. Y.. Kanade. T. and Sakai. T. **A** Production System for Region Analysis. Proc. 6th International Joint Confcrencc on Artificial Intelligence. Aug, 1979, pp. 684-686.

28. Ohta. Y., Kanade. T. and Sakai, T. "Color Information for Region Segmentation." *Computer Graphics and* **Image** *Processing* ***13*** (Scpt. 1980). 222-241.

29. Rubin, S. M. "Natural Scene Recognition Using LOCUS Search." ***Cornpurer Graphics and Image Processing 13*** (1980), 298-333.

30. Shafcr. S. and Kanadc, T. Goniophotomcuy and the Reflectance Atlas. Tech. Rept. (in preparation), Carnegie-Mcllon University, 1982.

31. Shafcr. S. and Kanadc. T. Using Shadows in Finding Surface Orientations. Tech. Rept CMU-CS-82-100, Carncgic-Mcllon Univcrsity. Jan., 1982.

32. Shafcr, S., Kanadc. T. and Kcndcr. J. Gradient Space Under Orthography and Perspective. Tech. Kept CMU-CS-82-123, Carncgic-Mcllon University, May, 1982

33. Shirai, Y. Rccognition of Man-Madc Objects Using Edge **Cuts.** Computcr Vision Systcms, Sept., 1978.

34. Wesley, M.A., Lozano-Perez, T., Leiberman, L.I., Levin, M.A., Grossman, D.D. **"A** Gcomctric Modeling Systcm far Automated Mechanical Assembly." ***IBM J. Res Develop. 24.*** 1 (January 1980). **64-74.**

35. Woodham, R. J. "Analyzing Images **of** Curved Surfaces." *Artificial Intelligence 17* (1981), 117-141.