9

Computer Vision

SIGHTED HUMANS GET MUCH OF THEIR INFORMATION THROUGH VISION. THAT PART of AI called "computer vision" (or, sometimes, "machine vision") deals with giving computers this ability. Most computer vision work is based on processing two-dimensional images gathered from a three-dimensional world – images gathered by one or more television cameras, for example. Because the images are two-dimensional projections of a three-dimensional scene, the imaging process loses information. That is, different three-dimensional scenes might produce the same two-dimensional image. Thus, the problem of reconstructing the scene faithfully from an image is impossible in principle.

Yet, people and other animals manage very well in a three-dimensional world. They seem to be able to interpret the two-dimensional images formed on their retinas in a way that gives them reasonably accurate and useful information about their environments.

Stereo vision, using two eyes, helps provide depth information. Computer vision too can use "stereopsis" by employing two or more differently located cameras looking at the same scene. (The same effect can be achieved by having one camera move to different positions.) When two cameras are used, for example, the images formed by them are slightly displaced with respect to each other, and this displacement can be used to calculate distances to various parts of the scene. The computation involves comparing the relative locations in the images that correspond to the objects in the scene for which depth measurements are desired. This "correspondence problem" has been solved in various ways, one of which is to seek high correlations between small areas in one image with small areas in the other. Once the "disparity" of the location of an image feature in the two images is known, the distance to that part of the scene giving rise to this image feature can be calculated by using trigonometric calculations (which I won't go into here.)[1]

Perhaps surprisingly, a lot of depth information can be obtained from other cues besides stereo vision. Some of these cues are inherent in a single image, and I'll be describing these in later chapters. Even more importantly, background knowledge about the kinds of objects one is likely to see accounts for much of our ability to interpret images. Consider the image shown in Fig. 9.1 for example.

Most people would describe this image as being of two tables, one long and narrow and the other more-or-less square. Yet, if you measure the actual table tops in the image itself, you might be surprised to find that they are exactly the same size and shape! (The illustration is based on an illusion called "turning the tables" by the psychologist Roger Shepherd and is adapted from Michael Bach's version

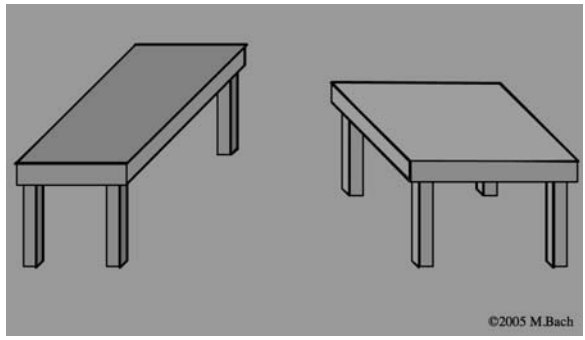

Figure 9.1. Two tables. (Illustration courtesy of Michael Bach.)

of Shepherd's diagram. If you visit Bach's Web site, http://www.michaelbach.de/ot/sze_shepardTables/, you can watch while one table top moves over to the other without changing shape.)

Something apart from the image provides us with information that induces us to make inferences about the shapes of the three-dimensional tables captured in the two-dimensional image shown in Fig. 9.1. As we shall see, that extra information consists of two things: knowledge about the image-forming process under various lighting conditions and knowledge about the kinds of things and their surfaces that occur in our three-dimensional world. If we could endow computers with this sort of knowledge, perhaps they too would be able to see.

9.1 Hints from Biology

There has been a steady flow of information back and forth between scientists attempting to understand how vision works in animals and engineers working on computer vision. An early example of work at the intersection of these two interests was described in an article titled "What the Frog's Eye Tells the Frog's Brain"[2] by four scientists at MIT. Guided by previous biological work, the four, Jerome Lettvin, H. R. Maturana, Warren McCulloch, and Walter Pitts, probed the parts of the frog's brain that processed images. They found that the frog's visual system consisted of "detectors" that responded only to certain kinds of things in its visual field. It had detectors for small, moving convex objects (such as flies) and for a sudden darkening of illumination (such as might be caused by a looming predator). These, together with a couple of other simple detectors, gave the frog information about food and danger. In particular, the frog's visual system did not, apparently, construct a complete three-dimensional model of its visual scene. As the authors wrote,

> The frog does not seem to see or, at any rate, is not concerned with the detail of stationary parts of the world around him. He will starve to death surrounded by food if it is not moving. His choice of food is determined only by size and movement. He will leap to capture any object the size of an insect or worm, providing it moves like one. He can be fooled easily not only by a bit of dangled meat but by any moving small object. His sex life is conducted by sound and touch. His choice of paths in escaping enemies does not seem to be governed by

anything more devious than leaping to where it is darker. Since he is equally at home in water and on land, why should it matter where he lights after jumping or what particular direction he takes?

Other experiments produced further information about how the brain processes visual images. Neurophysiologists David Hubel (1926–) and Torsten Wiesel (1924–) performed a series of experiments, beginning around 1958, which showed that certain neurons in the mammalian visual cortex responded selectively to images and parts of images of specific shapes. In 1959, they implanted microelectrodes in the primary visual cortex of an anesthetized cat. They found that certain neurons fired rapidly when the cat was shown images of small lines at one angle and that other neurons fired rapidly in response to small lines at another angle. In fact, they could make a "map" of this area of the cat's brain, relating neuron location to line angle. They called these neurons "simple cells" – to be distinguished from other cells, called "complex cells," that responded selectively to lines moving in a certain direction. Later work revealed that other neurons were specialized to respond to images containing more complex shapes such as corners, longer lines, and large edges.[3] They found that similar specialized neurons also existed in the brains of monkeys.[4] Hubel and Wiesel were awarded the Nobel Prize in Physiology or Medicine in 1981 (jointly with Roger Sperry for other work).[5]

As I'll describe in later sections, computer vision researchers were developing methods for extracting lines (both large and small) from images. Hubel and Wiesel's work helped to confirm their view that finding lines in images was an important part of the visual process. Yet, straight lines seldom occur in the natural environments in which cats (and humans) evolved, so why do they (and we) have neurons specialized for detecting them? In fact, in 1992 the neuroscientists Horace B. Barlow and David J. Tolhurst wrote a paper titled "Why Do You Have Edge Detectors?"[6] As a possible answer to this question, Anthony J. Bell and Terrence J. Sejnowski later showed mathematically that natural scenes can be analyzed as a weighted summation of small edges even though the scenes themselves do not have obvious edges.[7]

9.2 Recognizing Faces

In the early 1960s at his Palo Alto company, Panoramic Research, Woodrow (Woody) W. Bledsoe (who later did work on automatic theorem proving at the University of Texas), along with Charles Bisson and Helen Chan (later Helen Chan Wolf), developed techniques for face recognition supported by projects from the CIA.[8] Here is a description of their approach taken from a memorial article:[9]

This [face-recognition] project was labeled man-machine because the human extracted the coordinates of a set of features from the photographs, which were then used by the computer for recognition. Using a GRAFACON, or RAND TABLET, the operator would extract the coordinates of features such as the center of pupils, the inside corner of eyes, the outside corner of eyes, point of widows peak, and so on. From these coordinates, a list of 20 distances, such as width of mouth and width of eyes, pupil to pupil, were computed. These operators could process about 40 pictures an hour. When building the database, the name of the person in the photograph was associated with the list of computed distances and stored in the computer. In

the recognition phase, the set of distances was compared with the corresponding distance for each photograph, yielding a distance between the photograph and the database record. The closest records are returned.

Bledsoe continued this work with Peter Hart at SRI after leaving Panoramic in 1966.[10]

Then, in 1970, a Stanford Ph.D. student, Michael D. Kelly, wrote a computer program that was able automatically to detect facial features in pictures and use them to identify people.[11] The task for his program was, as he put it,

to choose, from a collection of pictures of people taken by a TV camera, those pictures that depict the same person. . . .

In brief, the program works by finding the location of features such as eyes, nose, or shoulders in the pictures. . . . The interesting and difficult part of the work reported in this thesis is the detection of these features in digital pictures. The nearest-neighbor method is used for identification of individuals once a set of measurements has been obtained.

Another person who did pioneering work in face recognition was vision researcher Takeo Kanade, now a professor at Carnegie Mellon University. In a 2007 speech at the Eleventh IEEE International Conference on Computer Vision, he reflected on his early work in this field:[12] "I wrote my face recognition program in an assembler language, and ran it on a machine with 10 microsecond cycle time and 20 kB of main memory. It was with pride that I tested the program with 1000 face images, a rare case at the time when testing with 10 images was called a 'large-scale' experiment." (By the way, Kanade has continued his face recognition work up to the present time. His face-recognition Web page is at http://www.ri.cmu.edu/labs/lab_51.html.)

Face recognition programs of the 1960s and 1970s had several limitations. They usually required that images be of faces of standard scale, pose, expression, and illumination. Toward the end of the book, I'll describe research leading to much more robust automatic face recognition.

9.3 Computer Vision of Three-Dimensional Solid Objects

9.3.1 *An Early Vision System*

Lawrence G. Roberts (1937–), an MIT Ph.D. student working at Lincoln Laboratory, was perhaps the first person to write a program that could identify objects in black-and-white (gray-scale) photographs and determine their orientation and position in space. (His program was also the first to use a "hidden-line" algorithm, so important in subsequent work in computer graphics. As chief scientist and later director of ARPA's Information Processing Techniques Office, Roberts later played an important role in the creation of the Arpanet, the forerunner of the Internet.)

In the introduction to his 1963 MIT Ph.D. dissertation,[13] Roberts wrote

The problem of machine recognition of pictorial data has long been a challenging goal, but has seldom been attempted with anything more complex than alphabetic characters. Many people have felt that research on character recognition would be a first step, leading the way to a more general pattern recognition system. However, the multitudinous attempts at character

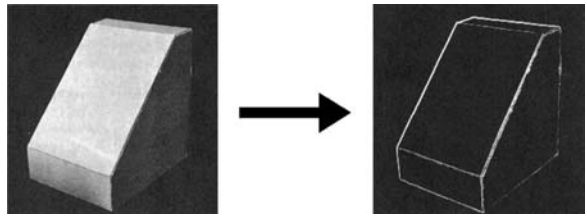

Figure 9.2. Detecting changes in intensity. (Photographs used with permission of Lawrence Roberts.)

recognition, including my own, have not led very far. The reason, I feel, is that the study of abstract, two-dimensional forms leads us away from, not toward, the techniques necessary for the recognition of three-dimensional objects. The perception of solid objects is a process which can be based on the properties of three-dimensional transformations and the laws of nature. By carefully utilizing these properties, a procedure has been developed which not only identifies objects, but also determines their orientation and position in space.

Roberts's system first processed a photograph of a scene to produce a representation of a line drawing. It then transformed the line drawing into a three-dimensional representation. Matching this representation against a stored list of representations of solid objects allowed it to classify the object it was viewing. It could also produce a computer-graphics image of the object as it might be seen from any point of view.

Our main interest here is in how Roberts processed the photographic image. After scanning the photograph and representing it as an array of numbers (pixels) representing intensity values, Roberts used a special calculation, later called the "Roberts Cross," to determine whether or not each small 2×2 square in the array corresponded to a part of the image having an abrupt change in image intensity. (The Roberts Cross was the first example of what were later called "gradient operators.") He then rerepresented the image "lighting up" only those parts of the image where the intensity changed abruptly and leaving "dark" those parts of the image with more-or-less uniform intensity. The result of this process is illustrated in Fig. 9.2 for a typical image used in Roberts's dissertation. As can be seen in that figure, large changes in image intensity are usually associated with the edges of objects. Thus, gradient operators, such as the Roberts Cross, are often called "edge detectors."

Further processing of the image on the right attempted to connect the dots representing abrupt intensity changes by small straight-line segments, then by longer line segments. Finally, a line drawing of the image was produced. This final step is shown in Fig. 9.3.

Roberts's program was able to analyze many different photographs of solid objects. He commented that "The entire picture-to-line-drawing process is not optimal but works for simple pictures." Roberts's success stimulated further work on programs for finding lines in images and for assembling these lines into representations of objects. Perhaps primed by Roberts's choice of solid objects, much of the subsequent work dealt with toy blocks (or "bricks" as they are called in Britain).

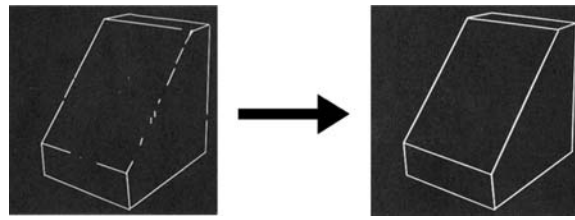

Figure 9.3. Producing the final line drawing. (Photographs used with permission of Lawrence Roberts.)

9.3.2 *The "Summer Vision Project"*

Interestingly, Larry Roberts was a student of MIT information theory professor Peter Elias, not of Marvin Minsky. But Minsky's group soon began to work on computer vision also. In the summer of 1966, the mathematician and psychologist Seymour Papert, a recent arrival at MIT's Artificial Intelligence Group, launched a "summer vision project." Its goal was to develop a suite of programs that would analyze a picture from a "videsector" (a kind of scanner) to "actually name objects [such as balls, cylinders, and blocks] by matching them with a vocabulary of known objects." One motivation for the project was "to use our summer workers effectively in the construction of a significant part of a visual system."[14]

Of course, the problem of constructing "a significant part of a visual system" was much more difficult than Papert expected. Nevertheless, the project was successful in that it began a sustained effort in computer vision research at MIT, which continues to this day.

After these early forays at MIT (and similar ones at Stanford and SRI to be described shortly), computer vision research focused on two areas. The first was what might be called "low-level" vision – those first stages of image processing that were aimed at constructing a representation of the image as a line drawing, given an image that was of a scene containing rather simple objects. The second area was concerned with how to analyze the line drawing as an assemblage of separate objects that could be located and identified. An important part of low-level vision was "image filtering," to be described next.

9.3.3 *Image Filtering*

The idea of filtering an image to simplify it, to correct for noise, and to enhance certain image features had been around for a decade or more. I have already mentioned, for example, that in 1955 Gerald P. Dinneen processed images to remove noise and enhance edges. Russell Kirsch and colleagues had also experimented with image processing.[15] (Readers who have manipulated their digital photography pictures on a computer have used some of these image filters.) Filtering two-dimensional images is not so very different from filtering one-dimensional electronic signals – a commonplace operation. Perhaps the simplest operation to describe is "averaging," which blurs fine detail and removes random noise specks. As in all averaging operations, image averaging takes into account adjacent values and combines them. Consider,

0	0	0	0	0	10	10	10	10	10
0	0	0	0	0	10	10	10	10	10
0	0	0	0	0	10	10	10	10	10
0	0	0	0	0	10	10	10	10	10
0	0	0	0	0	10	10	10	10	10
0	0	0	0	0	10	10	10	10	10
0	0	0	0	0	10	10	10	10	10
0	0	0	0	0	10	10	10	10	10
0	0	0	0	0	10	10	10	10	10
0	0	0	0	0	10	10	10	10	10

Figure 9.4. An array of image intensity values and an averaging window.

for example, the image array of intensity values shown in Fig. 9.4 containing a 3×3 "averaging window" outlined in bold. These intensity values correspond to an image whose right side is bright and whose left side is dark with a sharp edge between. (I adopt the convention that large numbers, such as 10 correspond to brightly illuminated parts of the image, and the number 0 corresponds to black.)

The averaging operation moves the averaging window over the entire image so that its center lies over each pixel in turn. For each placement of the window, the value of the intensity at its center is replaced (in the filtered version) by the average intensity of the values within the window. (The process of moving a window around the image and doing calculations based on the numbers in the window is called *convolution*.) In this example, the 0 at the center of the window would be replaced by 3.33 (perhaps rounded down to 3). One can see that averaging blurs the sharp edge – with the 10 fading to (a rounded) 7 fading to 3 fading to 0 as one moves from right to left. However, intensities well within evenly illuminated regions are not changed.

I have already mentioned another important filtering operation, the Roberts Cross, for detecting abrupt brightness changes in an image. Another one was developed in 1968 by a Ph.D. student at Stanford, Irwin Sobel. It was dubbed the "Sobel Operator" by Raj Reddy who described it in a Computer Vision course at Stanford.[16] The operator uses two filtering windows – one sensitive to large gradients (intensity changes) in the vertical direction and one to large gradients in the horizontal direction. These are shown in Fig. 9.5.

Each of the Sobel filters works the same way as the averaging filter, except that the image intensity at each point is multiplied by the number in the corresponding cell of the filtering window before adding all of the numbers. The sum would be 0

-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

Figure 9.5. Sobel's vertical (left) and horizontal (right) filters.

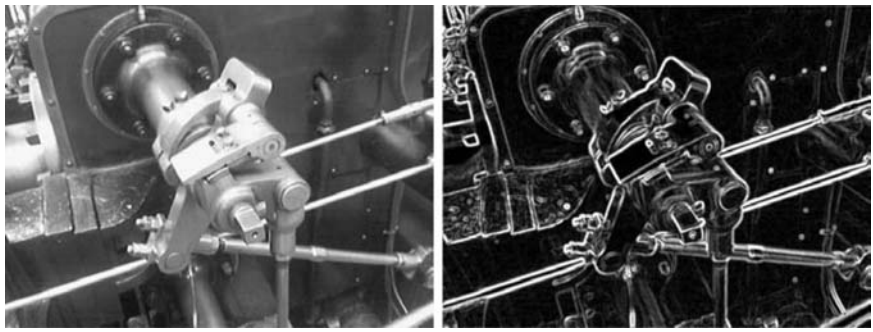

Figure 9.6. Finding abrupt changes in image brightness with the Sobel Operator. (Photographs taken by George Miller. Used under the terms of the GNU Free Documentation License.)

inside regions of uniform illumination. If the vertical filter is centered over a vertical edge (with the right side brighter than the left), the sum would be positive. (I'll let you think about the other possibilities.) Results from the two filtering windows are combined mathematically to detect abrupt changes in any direction.

The images in Fig. 9.6 illustrate the Sobel Operator. The image on the right is the result of applying the Sobel Operator to the image on the left.

A number of other more complex and robust image processing operations have been proposed and used for finding edges, lines, and vertices of objects in images.[17] A particularly interesting one for finding edges was proposed by the British neuroscientist and psychologist David Marr (1945–1980) and Ellen Hildreth.[18] The Marr–Hildreth edge detector uses a filtering window called a "Laplacian of Gaussian (LoG)." (The name arises because a mathematical operator called a "Laplacian" is used on a bell-shaped curve called a "Gaussian," commemorating two famous mathematicians, namely, Pierre-Simon Laplace and Carl Friedrich Gauss.) In Fig. 9.7, I show an example of LoG numbers in a 9×9 filtering window. This window is moved around an image, multiplying image numbers and adding them up, in the same way as the other filtering windows I have already mentioned.

If LoG numbers are plotted as "heights" above (and below) a plane, an interesting-looking surface results. An example is shown in Fig. 9.8. This LoG function is often called, not surprisingly, a Mexican hat or sombrero function.

0	0	3	2	2	2	3	0	0
0	2	3	5	5	5	3	2	0
3	3	5	3	0	3	5	3	3
2	5	3	−12	−23	−12	3	5	2
2	5	0	−23	−40	−23	0	5	2
2	5	3	−12	−23	−12	3	5	2
3	3	5	3	0	3	5	3	3
0	2	3	5	5	5	3	2	0
0	0	3	2	2	2	3	0	0

Figure 9.7. A Laplacian of Gaussian filtering window.

https://doi.org/10.1017/CBO9780511819346.012 Published online by Cambridge University Press

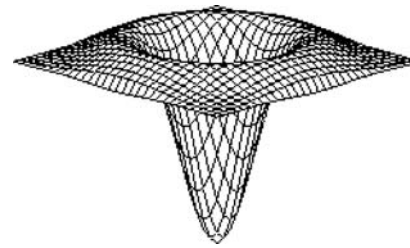

Figure 9.8. A Laplacian of Gaussian surface.

Marr and Hildreth used the LoG filtering window on several example images. One example, taken from their paper, is shown in Fig 9.9. Notice that the image on the right has whitish bands surrounding darker parts of the image. The Marr–Hildreth edge detector employs a second image-processing operation that looks for the transitions from light to dark (and vice versa) in the LoG-processed image to produce a final "line drawing," as shown in Fig. 9.10.

Further advances have been made in edge detection since Marr and Hildreth's work. Among the currently best detectors are those related to one proposed by John Canny called the Canny edge detector.[19]

As a neurophysiologist, Marr was particularly interested in how the human brain processes images. In a 1976 paper,[20] he proposed that the first stage of processing produces what he called a "primal sketch." As he puts it in his summary of that paper,

> It is argued that the first step of consequence is to compute a primitive but rich description of the grey-level changes present in an image. The description is expressed in a vocabulary of kinds of intensity change (EDGE, SHADING-EDGE, EXTENDED-EDGE, LINE, BLOB etc.). . . . This description is obtained from the intensity array by fixed techniques, and it is called the primal sketch.

Marr and Hildreth put forward their edge detector as one of the operations the brain uses in producing a primal sketch. They stated that their theory "explains

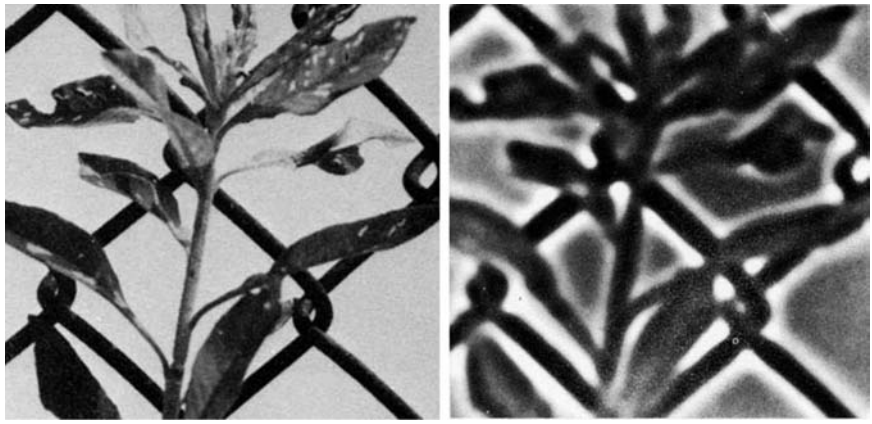

Figure 9.9. An image (left) and its LoG-processed version (right). (Images taken from David Marr and E. Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London*, Series B, Biological Sciences, Vol. 207, No. 1167, p. 198, February 1980.)

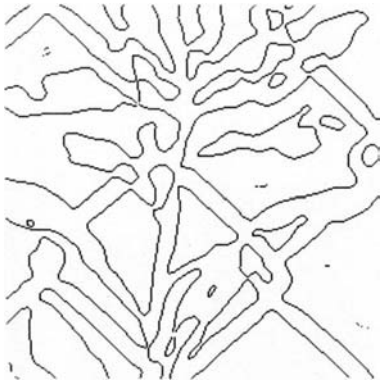

Figure 9.10. The final result of a Marr–Hildreth edge-detecting operation. (From David Marr and E. Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London*, Series B, Biological Sciences, Vol. 207, No. 1167, p. 198, February 1980.)

several basic psychophysical findings, and . . . forms the basis for a physiological model of simple [nerve] cells."

Marr's promising career in vision research ended when he succumbed to cancer in 1980. During the last years of his life he completed an important book detailing his theories of human vision.[21] I'll describe some of Marr's ideas about other visual processing steps in a subsequent chapter.

9.3.4 *Processing Line Drawings*

Assuming, maybe somewhat prematurely, that low-level vision routines could produce a line-drawing version of an image, many investigators moved on to develop methods for analyzing line drawings to find objects in images.

Adolfo Guzman-Arenas (1943–), a student in Minsky's AI Group, focused on how to segment a line drawing of a scene containing blocks into its constituents, which Guzman called "bodies." His LISP program for accomplishing this separation was called SEE and ran on the MIT AI Group's PDP-6 computer.[22] The input to SEE was a line-drawing representation of a scene in terms of its surfaces, lines (where two surfaces came together), and vertices (where lines came together).

SEE's analysis of a scene began by sorting its vertices into a number of different types. For each vertex, depending on its type, SEE connected adjacent planar surfaces with "links." The links between surfaces provide evidence that those surfaces belong to the same body. For example, some links for a scene analyzed by SEE are shown in Fig. 9.11.

SEE performed rather well on a wide variety of line drawings. For example, it correctly found all of the bodies in the scene shown in Fig. 9.12.

For most of his work, Guzman assumed that somehow other programs would produce his needed line drawings from actual images. As he wrote in a paper describing his research,[23]

> The scene itself is not obtained from a visual input device, or from an array of intensities of brightness. Rather, it is assumed that a preprocessing of some sort has taken place, and the scene to be analyzed is available in a symbolic format . . . in terms of points (vertices), lines (edges), and surfaces (regions)."

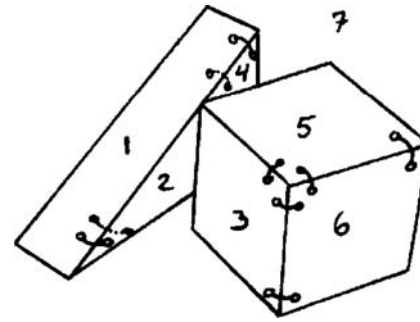

Figure 9.11. Links established by SEE for a sample scene. (Illustration used with permission of Adolpho Guzman.)

Additionally, Guzman did not concern himself with what might be done after the scene had been separated into bodies:

. . . it cannot find "cubes" or "houses" in a scene, since it does not know what a "house" is. Once SEE has partitioned a scene into bodies, some other program will work on them and decide which of those bodies are "houses."

Later extensions to SEE, reported in the final version of his thesis, involved some procedures for image capture. But the images were of specially prepared scenes, as he recently elaborated:[24]

Originally SEE worked on hand-drawn scenes, "perfect scenes" (drawings of lines). . .

Later, I constructed a bunch of wooden polyhedra (mostly irregular), painted them black, carefully painted their edges white, piled several of them together, and took pictures of the scenes. The pictures were scanned, edges found, and given to SEE. It worked quite well on them.

Although SEE was capable of finding bodies in rather complex scenes, it also could make mistakes, and it could not identify blocks that had holes in them.

The next person to work on the problem of scene articulation was David Huffman (1925–1999), a professor of Electrical Engineering at MIT. (Huffman was famous for

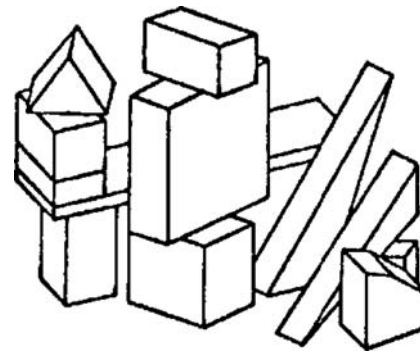

Figure 9.12. A scene analyzed by SEE. (Illustration used with permission of Adolpho Guzman.)

https://doi.org/10.1017/CBO9780511819346.012 Published online by Cambridge University Press

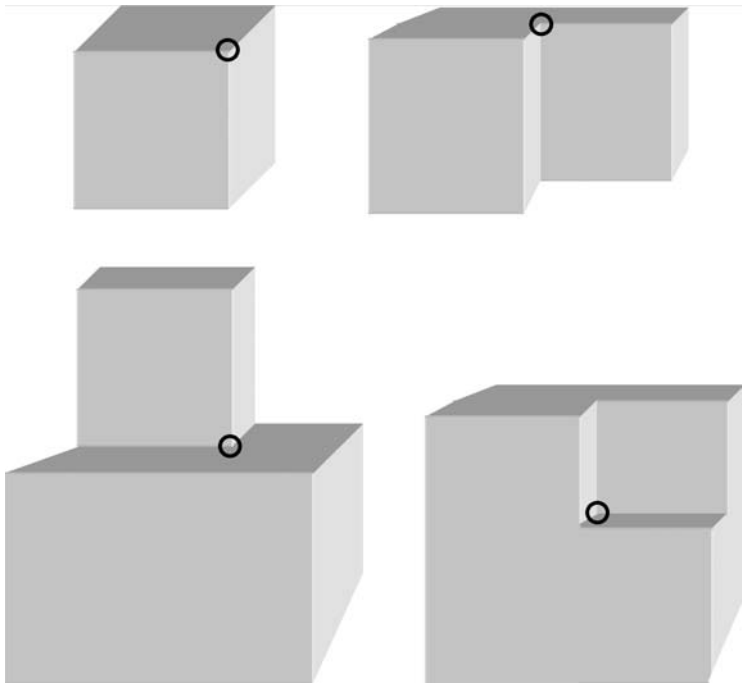

Figure 9.13. The four different kinds of vertices that can occur in trihedral solids.

his invention, while a graduate student at MIT, of what came to be called "Huffman coding," an efficient scheme that is used today in many applications involving the compression and transmission of digital data.) Huffman was bothered by what he considered Guzman's incomplete analysis of what kinds of objects could correspond to what kinds of line drawings. After leaving MIT in 1967 to become a professor of Information and Computer Science at the University of California at Santa Cruz, he completed a theory for assigning labels to the lines in drawings of trihedral solids – objects in which exactly three planar surfaces join at each vertex of the object. The labels depended on the ways in which planes could come together at a vertex. (I got to know Huffman well at that time because he consulted frequently at the Stanford Research Institute.)

Huffman pointed out that there are only four ways in which three plane surfaces can come together at a vertex.[25] These are shown in Fig. 9.13. In addition to these four kinds of vertices, a scene might contain what Huffman called "T-nodes" – line intersection types caused by one object in a scene occluding another. These all give rise to a number of different kinds of labels for the lines in the scene; these labels specify whether the lines correspond to convex, concave, or occluding edges.

Huffman noted that the labels of the lines in a drawing might be locally consistent (around some vertices) but still be globally inconsistent (around all of the vertices). Consider, for example, Roger Penrose's famous line drawing of an "impossible

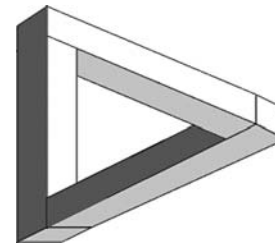

Figure 9.14. An impossible object.

object" shown in Fig. 9.14.[26] (It is impossible because no three-dimensional object, viewed in "general position," could produce this image.) No "real scene" can have a line with two different labels.

Max Clowes (circa 1944–1981) of Sussex University in Britain developed similar ideas independently,[27] and the labeling scheme is now generally known as Huffman–Clowes labeling.

Next comes David Waltz (1943–). In his 1972 MIT Ph.D. thesis, he extended the Huffman–Clowes line-labeling scheme to allow for line drawings of scenes with shadows and possible "cracks" between two adjoining objects.[28] Waltz's important contribution was to propose and implement an efficient computational method for satisfying the constraint that all of the lines must be assigned one and only one label. (For example, an edge can't be concave at one end and convex at the other.) In Fig. 9.15, I show an example of a line drawing that Waltz's program could correctly segment into its constituents.

Summarizing some of the work on processing line drawings at MIT, Patrick Winston says that "Guzman was the experimentalist, Huffman the theoretician, and Waltz the encyclopedist (because Waltz had to catalog thousands of junctions, in order to deal with cracks and shadows)."[29]

Meanwhile, similar work for finding, identifying, and describing objects in three-dimensional scenes was being done at Stanford. By 1972 Electrical Engineering Ph.D. student Gilbert Falk could segment scenes of line drawings into separate objects using techniques that were extensions of those of Guzman.[30] And by 1973, Computer Science Ph.D. student Gunnar Grape performed segmentation of scenes containing parallelepipeds and wedges using models of those objects.[31]

Other work on analysis of scenes containing polyhedra was done by Yoshiaki Shirai while he was visiting MIT's AI Lab[32] and by Alan Mackworth at the Laboratory of Experimental Psychology of the University of Sussex.[33]

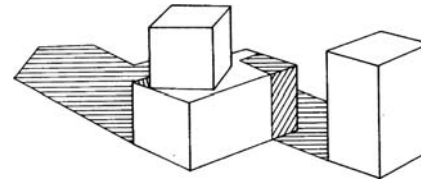

Figure 9.15. A scene with shadows analyzed by Waltz's program. (Illustration used with permission of David Waltz.)

Notes

1. For a thorough treatment, see David Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, Chapter 13, Upper Saddle River, NJ: Prentice Hall, 2003. [125]
2. Lettvin *et al.*, "What the Frog's Eye Tells the Frog's Brain," *Proceedings of the IRE*, Vol. 47, No. 11, pp. 1940–1951, 1959. [Reprinted as Chapter 7 in William C. Corning and Martin Balaban (eds.), *The Mind: Biological Approaches to Its Functions*, pp. 233–258, 1968.] [126]
3. David H. Hubel and Torsten N. Wiesel, "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex," *Journal of Physiology*, Vol. 160, pp. 106–154, 1962. [127]
4. David H. Hubel and Torsten N. Wiesel, "Receptive Fields and Functional Architecture of Monkey Striate Cortex," *Journal of Physiology*, Vol. 195, pp. 215–243, 1968. [127]
5. An interesting account of Hubel's and Wiesel's work and descriptions about how the brain processes visual images can be found in Hubel's online book *Eye, Brain, and Vision* at http://neuro.med.harvard.edu/site/dh/index.html. [127]
6. Horace B. Barlow and D. J. Tolhurst, "Why Do You Have Edge Detectors?," in *Proceedings of the 1992 Optical Society of America Annual Meeting*, Technical Digest Series, Vol. 23, pp. 172, Albuquerque, NM, Washington: Optical Society of America, 1992. [127]
7. Anthony J. Bell and Terrence J. Sejnowski, "Edges Are the 'Independent Components' of Natural Scenes," *Advances in Neural Information Processing Systems*, Vol. 9, Cambridge, MA: MIT Press, 1996. Available online at ftp://ftp.cnl.salk.edu/pub/tony/edge.ps.Z. [127]
8. Woodrow W. Bledsoe and Helen Chan, "A Man–Machine Facial Recognition System: Some Preliminary Results," Technical Report PRI 19A, Panoramic Research, Inc., Palo Alto, CA, 1965. [127]
9. Michael Ballantyne, Robert S. Boyer, and Larry Hines, "Woody Bledsoe: His Life and Legacy," *AI Magazine*, Vol. 17, No. 1, pp. 7–20, 1996. Also available online at http://www.utexas.edu/faculty/council/1998-1999/memorials/Bledsoe/bledsoe.html. [127]
10. Woodrow W. Bledsoe, "Semiautomatic Facial Recognition," Technical Report SRI Project 6693, Stanford Research Institute, Menlo Park, CA, 1968. [128]
11. Michael D. Kelly, "Visual Identification of People by Computer," Stanford AI Project, Stanford, CA, Technical Report AI-130, 1970. [128]
12. http://iccv2007.rutgers.edu/TakeoKanadeResponse.htm. [128]
13. Lawrence G. Roberts, "Machine Perception of Three-Dimensional Solids," MIT Ph.D. thesis, 1963; published as Lincoln Laboratory Technical Report #315, May 22, 1963; appears in J. T. Tippett *et al.* (eds.), *Optical and Electro-Optical Information Processing*, pp. 159–197, Cambridge, MA: MIT Press, 1965. Available online at http://www.packet.cc/files/mach-per-3D-solids.html. [128]
14. The project is described in MIT's Artificial Intelligence Group Vision Memo No. 100 available at ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-100.pdf. [130]
15. Russell A. Kirsch *et al.*, "Experiments in Processing Pictorial Information with a Digital Computer," *Proceedings of the Eastern Joint Computer Conference*, pp. 221–229, Institute of Radio Engineers and Association Association for Computing Machinery, December 1957. [130]
16. According to Sobel, he and a fellow student, Gary Feldman,Feldman, Gary first presented the operator in a Stanford AI seminar in 1968. It was later described in Karl K. Pingle,Pingle, Karl "Visual Perception by a Computer," in A. Grasselli (ed.), *Automatic*

Interpretation and Classification of Images, pp. 277–284, New York: Academic Press, 1969. It was also mentioned in Richard O. Duda and Hart@Hart, Peter Peter E. Hart, *Pattern Classification and Scene Analysis*, pp. 271–272, New York: John Wiley & Sons, 1973. [131]

17. See, for example, M. H. Hueckel, "An Operator Which Locates Edges in Digitized Pictures," *Journal of the ACM*, Vol. 18, No. 1, pp. 113–125, January 1971, and Berthold K. P. Horn, "The Binford–Horn Line Finder," MIT AI Memo 285, MIT, July 1971 (revised December 1973 and available online at http://people.csail.mit.edu/bkph/AIM/AIM-285-OPT.pdf). [132]
18. David Marr and Ellen Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London*, Series B, Biological Sciences, Vol. 207, No. 1167, pp. 187–217, February 1980. [132]
19. John E. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 679–714, 1986. [133]
20. David Marr, "Early Processing of Visual Information," *Philosophical Transactions of the Royal Society of London*, Series B, Biological Sciences, Vol. 275, No. 942, pp. 483–519, October 1976. [133]
21. David Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, San Francisco: W.H. Freeman and Co., 1982. [134]
22. Guzman's 1968 Ph.D. thesis is titled "Computer Recognition of Three Dimensional Objects in a Visual Scene" and is available online at http://www.lcs.mit.edu/publications/pubs/pdf/MIT-LCS-TR-059.pdf. [134]
23. Adolfo Guzman, "Decomposition of a Visual Scene into Three-Dimensional Bodies," *AFIPS*, Vol. 33, pp. 291–304, Washington, DC: Thompson Book Co., 1968. Available online as an MIT AI Group memo at ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-171.pdf. [134]
24. Personal communication, September 14, 2006. [135]
25. David A. Huffman, "Impossible Objects as Nonsense Sentences," in B. Meltzer and D. Michie (eds.), *Machine Intelligence 6*, pp. 195–234, Edinburgh: Edinburgh University Press, 1971, and David A. Huffman, "Realizable Configurations of Lines in Pictures of Polyhedra," in E. W. Elcock and D. Michie (eds.), *Machine Intelligence 8*, pp. 493–509, Chicester: Ellis Horwood, 1977. [136]
26. According to Wikipedia, this impossible object was first drawn by the Swedish artist Oscar Reutersvärd in 1934. [137]
27. Max B. Clowes, "On Seeing Things," *Artificial Intelligence*, Vol. 2, pp. 79–116, 1971. [137]
28. David L. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," MIT AI Lab Technical Report No. AITR-271, November 1, 1972. Available online at https://dspace.mit.edu/handle/1721.1/6911. A condensed version appears in Patrick Winston (ed.), *The Psychology of Computer Vision*, pp. 19–91, New York: McGraw-Hill, 1975. [137]
29. Personal communication, September 20, 2006. [137]
30. Gilbert Falk, "Computer Interpretation of Imperfect Line Data as a Three-Dimensional Scene," Ph.D. thesis in Electrical Engineering, Stanford University, Artificial Intelligence Memo AIM-132, and Computer Science Report No. CS180, August 1970. Also see Gilbert Falk, "Interpretation of Imperfect Line Data as a Three-Dimensional Scene," *Artificial Intelligence*, Vol. 3, pp. 101–144, 1972. [137]
31. Gunnar Rutger Grape, "Model Based (Intermediate Level) Computer Vision," Stanford Computer Science Ph.D. thesis, Artificial Intelligence Memo AIM-204, and Computer Science Report No. 266, May 1973. [137]

32. Yoshiaki Shirai, "A Heterarchical Program for Recognition of Polyhedra," MIT AI Memo No. 263, June 1972. Available online at ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-263.pdf. [137]
33. Alan K. Mackworth, "Interpreting Pictures of Polyhedral Scenes," *Artificial Intelligence*, Vol. 4, No. 2, pp. 121–137, June 1973. [137]