Can Computers See?

Can Computers Understand Visual Data?

Neelima Shrikhande

This article presents a brief introduction to the problem of computer vision. Computers, using cameras and other sensors as input devices record digital images in its memory. The computer programs must interpret these images to come up with a description of the scene. The computer vision problem can be stated simply as follows: Given a two-dimensional image, infer the objects that produced it, including their shapes, locations, colors and sizes. The concepts of low level, mid level image processing and high-level image understanding are presented. Various application areas including satellite imaging, assembly line manufacturing, handwriting and face recognition are discussed.

Neelima Shrikhande is Professor of Computer Science at Central Michigan University where she has been teaching since 1981. Her research interests are in the area of computer vision and artificial intelligence in which she has published numerous articles. She received her PhD in mathematics from University of Wisconsin in 1976.

Introduction

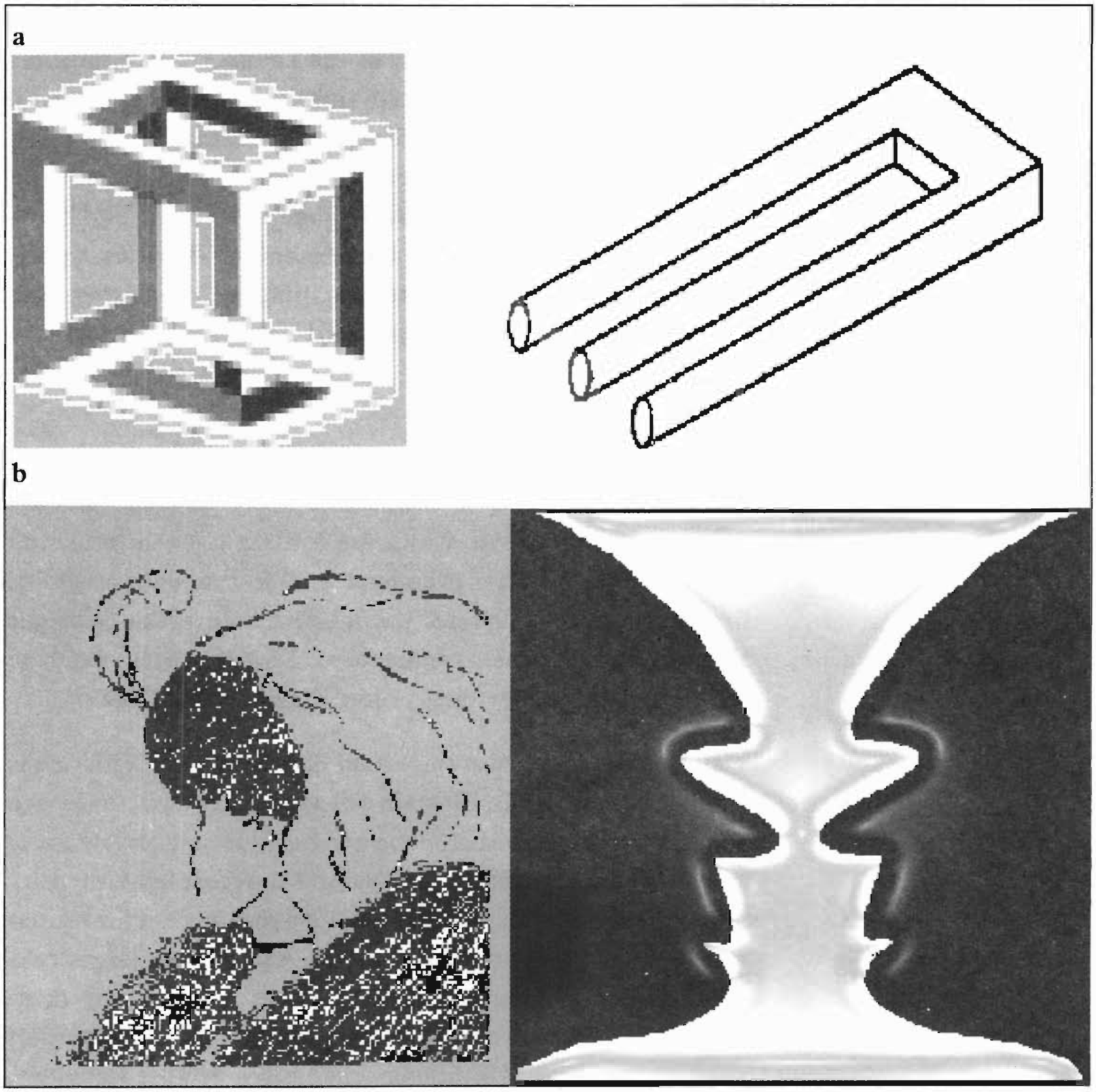
Imagine Rover roving on the planet Mars with some cameras, radar, and other sensors, which are sending visual information to its brain – a collection of computers. The computer programs first decide what they are 'seeing', and then make decisions about further movements of the robot. It would be reasonable to conclude that the rover has 'vision' in its limited domain.

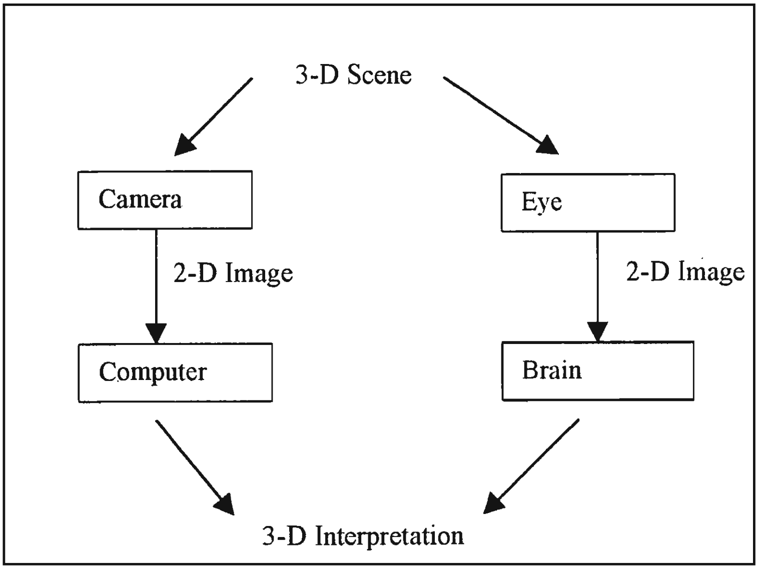
Vision is one of the most important human senses. A picture is worth thousand words, Einstein was a visionary man, to see is to believe: we use the visual metaphor freely as an equivalence of knowledge. Human brain uses two eyes to accept light intensity information and then processes the data recorded on the retinas to make an interpretation of the scene. A prior knowledge of the scene we are looking at is essentially used in making these conclusions. Human brain is remarkable in that it can interpret a vast variety of scenes in different circumstances quite accurately.

There are exceptions, well known optical illusions tell us that human visual inferring system can be fooled: see *Box* 1. In *Figure* 1a perfect 2-D line drawings are seen that are impossible as 3-D objects. In *Figure* 1b each drawing is ambiguous; i.e. each of these line drawings can be interpreted correctly in two different ways.

Computers, using cameras and other sensors as input devices record digital images in its memory. The computer programs

Figure 1a. Two images for which there are no three dimensional analogs.
Figure 1b. Two images for which there are two unambiguous interpretations.





must interpret these images to come up with a description of the scene. *The computer vision problem can be stated simply as follows: Given a two-dimensional image, infer the objects that produced it, including their shapes, locations, colors and sizes.*

Unfortunately, many different objects with different physical and geometrical properties are capable of producing the same image. The image processing algorithms must use additional knowledge about the context to solve the problem. Decision making always requires knowledge of the application or goal. For example, a program to drive a car needs information about roads. Without explicit use of knowledge, machine vision systems can be designed to work only in a very constrained environment for limited applications. Visual perception is thus based on selecting models that are relevant to the analysis of a sensed scene. To provide flexibility and robustness, knowledge is represented explicitly using artificial intelligence techniques.

Visual perception is thus based on selecting models that are relevant to the analysis of a sensed scene. To provide flexibility and robustness, knowledge is represented explicitly using artificial intelligence techniques.

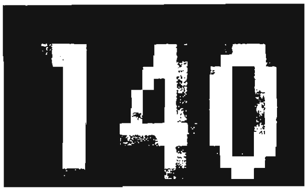
The Mechanics of Computer Vision

Imagine that we have a photograph such as one appearing in *Figure* 2, and that we want a computer program to make sense of it. The first step, called digitization, is to convert the image into digital data: binary numbers (series of 0's and l's) that the computer can understand. This is similar to the way compact discs transform music (sound waves) into binary numbers. Special hardware called frame grabbers transforms the visual image into digital form. Or the image may be taken directly by a digital camera. The process of digitization is well understood. The problem is that a single image usually produces a large amount of data. A single black and white picture might produce one megabyte of information. The result of digitizing a small portion of the photograph is shown in *Figure* 2. Each of the points in the digital image is called a pixel (for picture element); i.e. a pixel is a sample of the image intensity quantized to an integer value. *Figure* 2a shows the pixels in an image of the number 140. *Figure* 2b shows the portion of digital values corresponding to the digits 1 and 4.

Once the picture is digitized, there are several levels of processing that the computer program follows: Early processing that deals with the raw image and produces low level features, mid level processing that groups these together into high level features

Figure 2a(top). Image of the number 140.

Figure 2b(bottom). Digital values in the image corresponding to digits 1 and 4.

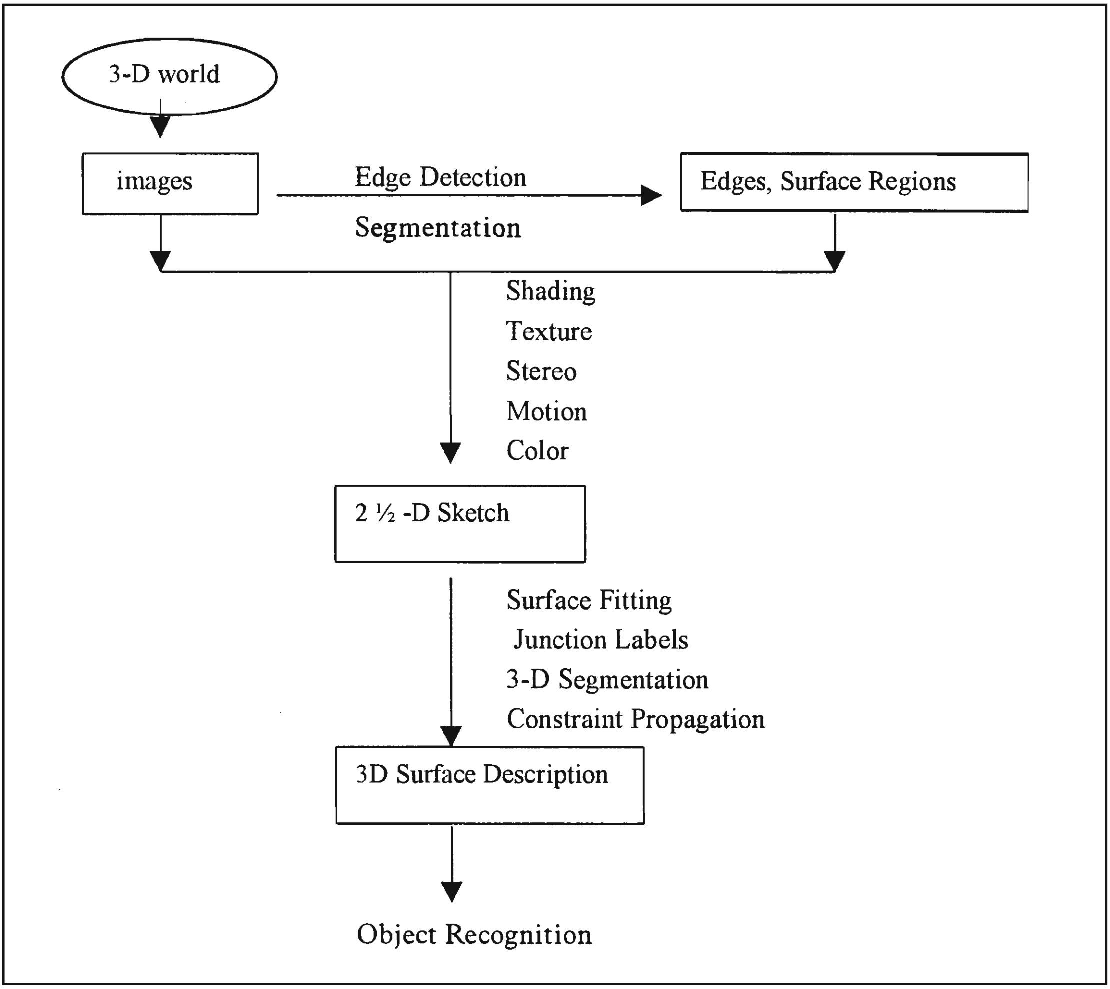


0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	32	64	64	64	0	0	0	0	0	0	48	64	48	0	0	0	0
0	128	255	255	255	0	0	0	0	0	32	255	255	191	0	0	16	0
0	96	207	255	255	0	0	0	0	0	143	255	255	191	0	0	128	0
0	0	64	255	255	0	0	0	0	16	239	239	255	191	0	0	207	0
0	0	64	255	255	0	0	0	0	96	255	143	255	191	0	0	255	0
0	0	64	255	255	0	0	0	0	223	159	128	255	191	0	64	255	0
0	0	64	255	255	0	0	0	80	255	80	128	255	191	0	64	255	0
0	0	64	255	255	0	0	0	191	223	0	128	255	191	0	64	255	0
0	0	64	255	255	0	0	0	255	255	255	255	255	255	191	32	255	0
0	0	64	255	255	0	0	0	191	191	191	223	255	239	143	0	239	0
0	0	64	255	255	0	0	0	0	0	0	128	255	191	0	0	175	0
0	0	64	255	255	0	0	0	0	0	0	128	255	191	0	0	64	0
0	0	48	191	191	0	0	0	0	0	0	96	191	143	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

and high level processing that makes 'sense' out of these features. The diagram below shows some examples of the kind of processing that can occur at each level. We discuss each of these levels in what follows.

Early Processing

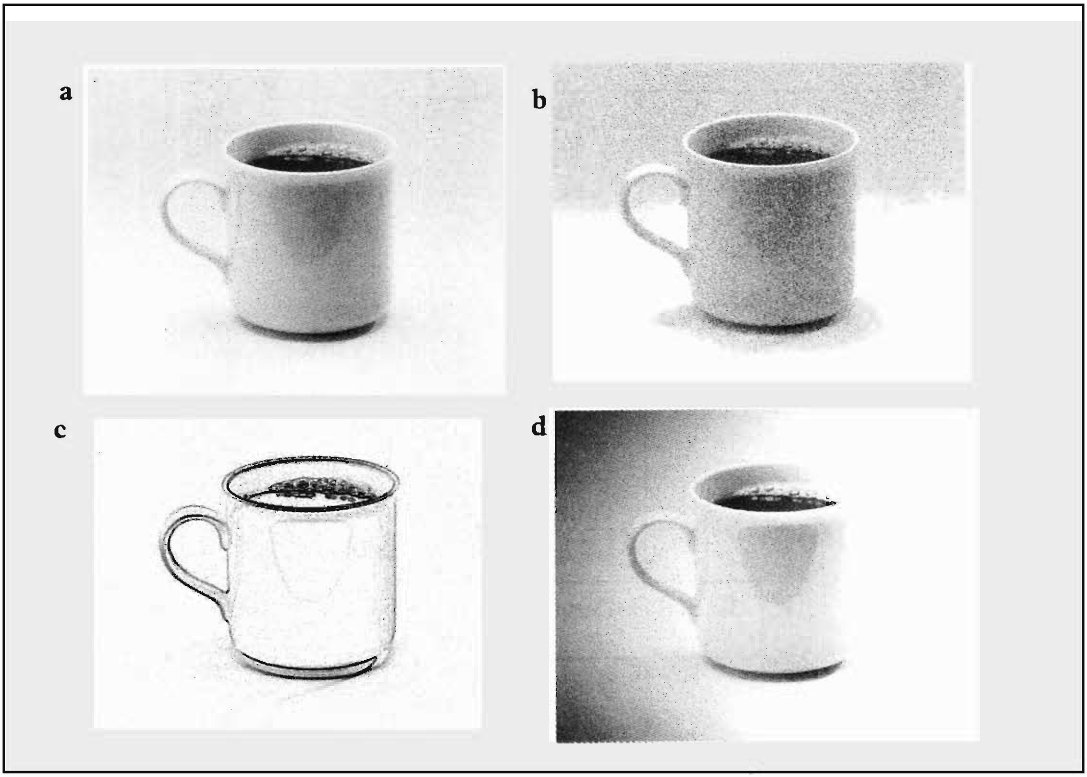
Noise Removal: The information in the digital version of the photo contains many small inaccuracies that arise in the digitization process. When an image is digitized, random variations can be introduced in the data depending on the sampling method. These variations are called noise. Local processing usually filter this noise out. For example, each pixel



can be replaced by an average of the pixels around it, thus reducing the disparity between it and its neighbors. Depending on the kind of noise that is present in the data, different weights may be attached to the neighboring intensities. This process smoothens the data and removes local discontinuities.

Feature detection: The next step is to identify small features such as edges that are usually present in an image. Edges typically occur on the boundary between two different regions in an image, where a significant change in the image intensity is present. Discontinuities in the image intensity correspond to the maxima and minima of the first derivative of the intensity function. In practice, because the image is digital, discrete approximations of the derivative operators are applied. Thus, many inaccuracies are present in the detected edges. *Figure* 3a shows the image of a cup. *Figure* 3b shows a noisy image of the cup. This kind of noise cannot be removed by averaging tech-

Figure 3. a) Image of a coffee cup. b) Noisy image of the cup. c) Edges detected in the cup. d) Cup image in different lighting conditions.



niques. *Figure* 3c shows the edges detected in the image. Note the small edges detected near the foam on the coffee. *Figure* 3d shows another image of the cup taken under different lighting conditions. Note that the light is projected from the right side, consequently, the right edge of the cup image is hard to detect. Other features such as surface regions corner points etc. may also be detected at this stage.

It is possible to obtain three-dimensional information from the 2-D image alone or two or more cameras may be used to obtain stereo information.

Mid Level Processing

Segmentation: The features obtained by the early processing are now grouped together into larger 'segments' of the image. This operation, which is so natural and easy for humans, is surprisingly difficult for computers. Ideally, a partition of an image into region segments represents objects or parts of objects in an image. One important technique used to achieve this is called thresholding. All pixels within a certain intensity range are identified as belonging to the same region. Different threshold levels may be applied to the image to obtain different regions. For example, in *Figure* 1b two black blobs and one white blob would be found along with the information about their boundaries.

Recovering 3-D information: Having identified edges and surfaces in our image, we next need to remove the ambiguities introduced by our conversion of three-dimensional reality into a two dimensional image. Which objects are in front of which others? How far away are the objects? What is the orientation of the various surfaces that have been identified? It is possible to obtain three-dimensional information from the 2-D image alone or two or more cameras may be used to obtain stereo information.

The 2 1/2-D sketch: Additional information about the objects depicted in the image can be found in a variety of ways. Properties such as stereo, shading, perspective, color, texture may be used to get more information. Active sensors such as laser beams or sonar signals may be used to obtain 3-D information. The aggregate of information such as orientation and depth of visible

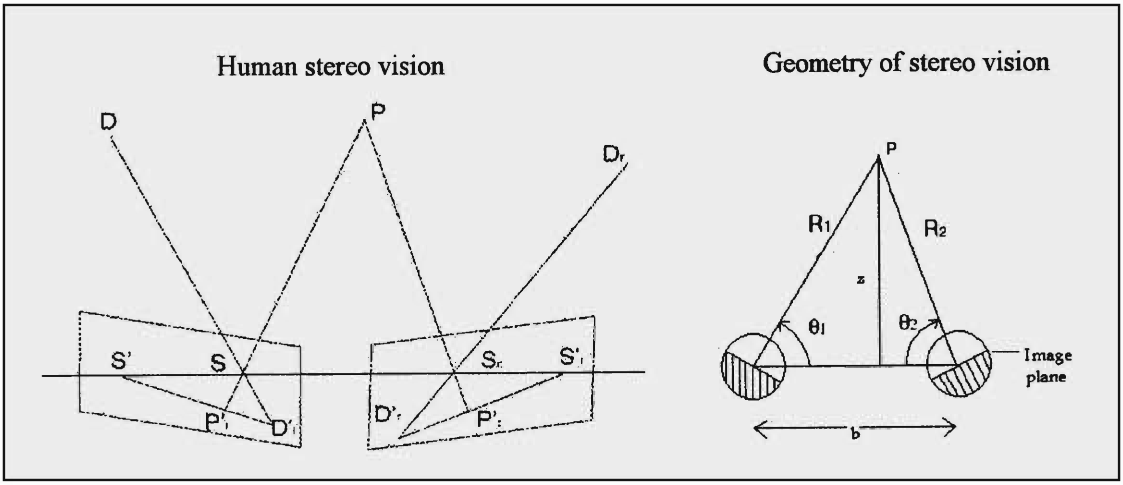


Figure 4. Depicting stereo vision.

surfaces, distance from viewer, contours of discontinuities comprise what is known as 2 ½ D sketch.

Stereo: Humans use perspective information using two eyes to guess about the depth of an object. Two sensors, e.g., two cameras or one camera and one laser sensor can be used to gain depth information in images. (See *Figure* 4.) The major problem in this process is called the correspondence problem. The problem is to determine which point in the right image corresponds to the equivalent point in the left image. Usually an effective solution requires recognition of salient features in the scene, which is a difficult task. The stereo computations also require accurate camera calibration computations.

Texture plays a significant role in our perception of the shapes of surfaces and their layout in space.

Texture: As the pictures in *Figure* 5 confirm, texture plays a significant role in our perception of the shapes of surfaces and their layout in space. In some cases, texture alone is sufficient clue for recognition of an object. Conceptually, texture is repetition of a basic pattern. However, this pattern may not be deterministic. It may be statistically defined, or the repetitions may be subject to geometric distortions due to natural variations in the pattern or due to imaging conditions. The various approaches to texture determination may be classified as structural or statistical. Statistical methods are usually employed in aerial images: e.g., for LANDSAT images. Structural analysis

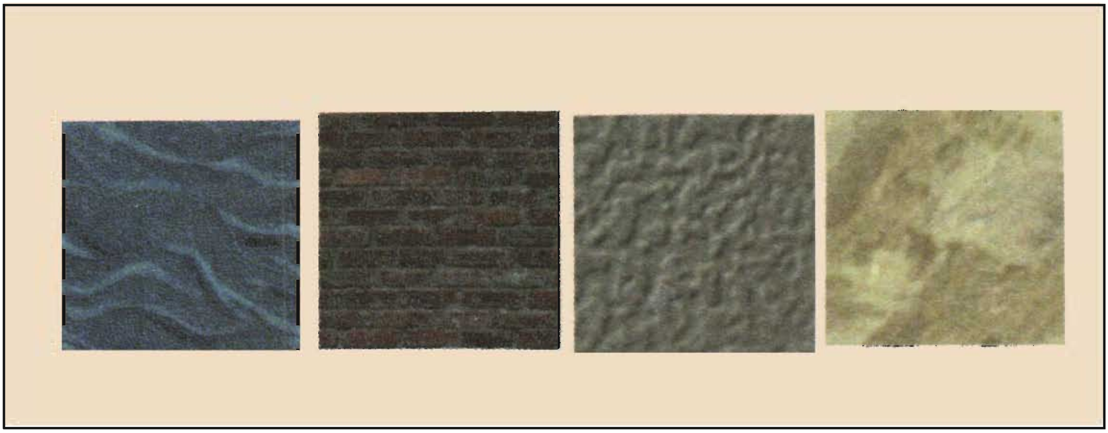


Figure 5. Some examples of textures.

is suited to images with textures that have deterministic underlying causes.

High Level Image Understanding

Object and scene recognition: What we have discussed so far is image processing without giving any meaning to the contents of the image. We have been concerned with trying to understand the geometric properties of the image. At some point we need to understand the image in a way that allows us to use what we see. The first step is to identify the objects in the image for what they are. When we decide what is in a particular image, we pursue the information about the world in general. This phase of vision is difficult and uses results and techniques of artificial intelligence to reason about the 2½ sketch. This involves making assumptions about the domain from which the image is obtained. For example, making sense of an X-ray requires that we know what part of the body is being depicted. Looking at *Figure* 5, humans can easily recognize the bricks in the wall. But is it extremely difficult for a computer program to do so.

Knowledge Bases

Recognition thus involves two things: a collection of stored 3-D model descriptions, and various indexes into the collection that allow a newly derived description to be associated with a description in the collection. This catalogue of model descriptions is called the knowledge base. The first problem is to decide which models to include in this knowledge base. Usually this phase is application dependent. For example, a program to assist a robot to walk on Mars would not contain models of elephants and tigers. The 3-D models that are included in the knowledge base have many different properties. Exactly which properties to include for indexing depends on a particular application. For example, if the image processing software can only extract black and white images, color properties of the 3-D model would not be included in the knowledge base. Construction of knowledge database general enough to include most related models yet restrictive enough to be efficient is very important to the object recognition problem. Geometric models such as lines and curves, illumination models that relate the light sources, surface reflectance, semantic models that store descriptions of objects that might appear in an image are some examples of model databases that are currently used in image processing applications.

A complex vision system thus consists of several different modules that interact with each other. They represent visual inputs, intermediate representations, knowledge bases and so on. It tries to find relationships between the internal representations and its knowledge base. These relationships 'match' entities at one level with those at another level. Ultimately, matching establishes an interpretation of the input data and generates output that gives symbolic information about the input scene. Now our Mars Rover knows that it is 'looking' at a rock that is three feet away from it and it is two feet high.

Machine vision systems have been used in quality control of products ranging from pizza to submicron structures on computer chips.

Application Areas

Machine vision systems have been used in quality control of products ranging from pizza to submicron structures on computer chips. Some important applications of computer vision include the following:

Automation and Inspection: Many factories now employ robots with vision software in assembly line applications. For example, General Motors has an assembly line set-up where thirteen cameras are used to check the gasket fitting of an auto windshield. Integrated circuit chips are examined for defects using image-processing software.

Satellite imagery can be processed to acquire data about remote territories on earth and in space.

Medical Imaging: Computed images such as MRI, tomography images are used by physicians to diagnose diseases. Machine vision systems help the physician to recover information by enhancing the images. Quantitative measurements including three-dimensional data on regions of interest can be made easily available. Such systems are being developed for all imaging models useful in different types of health care.

Multimedia: By the end of 1994, the number of personal computers exceeded 80 million in the United States and 200 million worldwide. A new driving force behind these astonishing numbers is the increasing availability of multimedia. The breakthrough in the use of multimedia is largely fueled by capabilities for handling large digital images.

Remote Sensing: Satellite imagery can be processed to acquire data about remote territories on earth and in space. For example, relief maps of Mars were generated using image-processing techniques.

Human-Computer Interaction: Face recognition, fingerprint recognition, signature recognition are some of the areas where image processing can be used to automate identification of human characteristics. Programs are being built for visually handicapped to assist them in ‘seeing’.

Programs are being built for visually handicapped to assist them in ‘seeing’

Summary and Conclusion: Can Computers See?

The question asked in the title has two answers: no and yes. The general problem of computer vision, designing a program that will recognize and interpret scenes in a general setting is not solved and is not likely to be solved in the near future. However, many special purpose programs that deal with specific applications do 'see' in their own domain as can be seen from the application areas described earlier.

Research in leading universities including Carnegie Mellon (NAVLAB: a car that is driven by a computer), MIT (multimedia lab, navigable robots), Stanford (Cortical Visual Processing) has provided powerful directions for progress in writing robust software. The current research efforts focus on specific areas of the vision problem as described in the above paragraphs and try to get efficient solutions for that particular subproblem.

The present article gives a brief introduction to the problem of computer vision and image processing. Techniques employed for different levels of image processing are described. A brief description of applications areas is given. The references provided in the suggested readings should be of further help.

Acknowledgments: Most of the figures are downloaded from the web. *Figure* 2 is based on figures in [2].

Suggested Reading

[1] D Ballard and C Brown, *Computer Vision*, Prentice Hall, 1982.

[2] E Charniak, and D McDermott, *Introduction to Artificial Intelligence*, Addison Wesley, 1984.

[3] R Gonzalez and R Woods, *Digital Image Processing*, Second Edition, Addison Wesley, 1993.

[4] R Haralick and L Shapiro, *Computer and Robot Vision*, Addison Wesley, Vol. 1, Second ed., 1993; Vol. 2, 1992.

[5] B K P Horn, *Robot Vision*, McGraw Hill, 1992.

[6] R Jain, R Kasturi and B Schunck, *Machine Vision*, McGraw Hill, 1996.

[7] D Marr, *Vision*, Freeman Press, 1982.

[8] V Nalwa, *A Guided Tour of Computer Vision*, Addison Wesley, 1993.

Useful website address:

To Computer Vision Links http://www-vision.ucsd.edu/links.html

Address for Correspondence
Neelima Shrikhande
Center for Computer Vision
Computer Science Department
Central Michigan University
Mount Pleasant
MI 48859, USA.
Email: neelima@cps.cmich.edu