

Chapter 1

Introduction

This book is an introduction to the broad field of computer vision. Without a doubt, machines can be built to see; for example, machines inspect millions of light bulb filaments and miles of fabric each day. Automatic teller machines (ATMs) have been built to scan the human eye for user identification and cars have been driven by a computer using camera input. This chapter introduces several important problem areas where computer vision provides solutions. After reading this chapter, you should have a broad view of some problems and methods of computer vision.¹

1 DEFINITION *The goal of computer vision is to make useful decisions about real physical objects and scenes based on sensed images.*

In order to make decisions about real objects, it is almost always necessary to construct some description or model of them from the image. Because of this, many experts will say that *the goal of computer vision is the construction of scene descriptions from images*. Although our study of computer vision is problem-oriented, fundamental issues will be addressed. Critical issues raised in this chapter and studied in the remainder of the text include the following.

Sensing: How do sensors obtain images of the world? How do the images encode properties of the world, such as material, shape, illumination and spatial relationships?

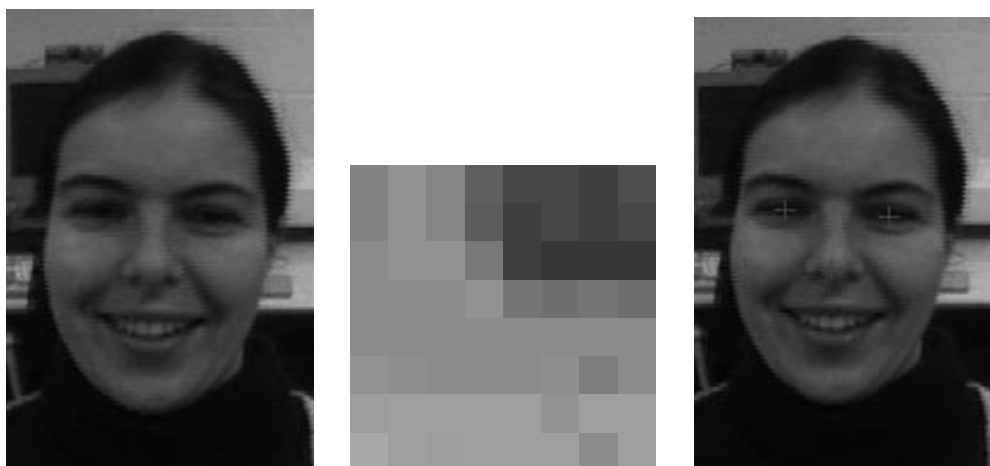
Encoded Information: How do images yield information for understanding the 3D world, including the geometry, texture, motion, and identity of objects in it?

Representations: What representations should be used for stored descriptions of objects, their parts, properties and relationships?

Algorithms: What methods are there to process image information and construct descriptions of the world and its objects?

These issues and others will be studied in the following chapters. We now introduce various applications and some important issues that arise in their context.

¹ In this book, we generally use the terms *machine vision* and *computer vision* to mean the same thing. However, we often use the term *machine vision* in the context of industrial applications and the term *computer vision* with the field in general.



	0	1	2	3	4	5	6	7
0	130	146	133	95	71	71	62	78
1	130	146	133	92	62	71	62	71
2	139	146	146	120	62	55	55	55
3	139	139	139	146	117	112	117	110
4	139	139	139	139	139	139	139	139
5	146	142	139	139	139	143	125	139
6	156	159	159	159	159	146	159	159
7	168	159	156	159	159	159	139	159

Figure 1.1: (Top left) Image of a face, (top center) subimage of 8x8 pixels from the right eye region, (top right) eye location detected by a computer program, and (bottom) intensity values from the 8x8 subimage. Images courtesy of Vera Bakic.

1.1 Machines that see?

Scientists and science fiction writers have been fascinated by the possibility of building intelligent machines, and the capability of understanding the visual world is a prerequisite that some would require of such a machine. Much of the human brain is dedicated to vision. Humans solve many visual problems effortlessly, yet most have little analytical understanding of visual cognition as a process. Allan Turing, one of the fathers of both the modern digital computer and the field of artificial intelligence, believed that a digital computer would achieve intelligence and the ability to understand scenes. Such lofty goals have proved difficult to achieve and the richness of human imagination is not yet matched by our engineering. However, there has been surprising progress along some lines of research. While building practical systems is a primary theme of this text and artificial intelligence is not, we will sometimes ponder the deeper questions, and, where we can, make some assessment of progress. Consider, for example, the following scenario, which could be realized within the next few years. A TV camera at your door provides images to your home computer which you have trained to recognize some faces of people important to you. When you call in to your home message center, your computer not only reports the phone messages, but

it also reports probable visits from your sister Eleanor and Chad the paper boy. We will discuss such current research ideas at various places in the book.

1.2 Application problems

The applications of computers in image analysis are virtually limitless. Only a small sample of applications can be included here, but these will serve us well for both motivation and orientation to the field of study.

A preview of the digital image

A digital image might represent a cartoon, a page of text, a person's face, a map of Katmandu, or a product for purchase from a catalog. A digital image contains a fixed number of rows and columns of *pixels*, short for *picture elements*. Pixels are like little tiles holding quantized values – small numbers, often between 0 and 255, that represent the brightness at the points of the image. Depending on the coding scheme, 0 could be the darkest and 255 the brightest, or visa-versa. At the top left in Figure 1.1 is a printed digital image of a face that is 257 rows high by 172 columns wide. At the top center is an 8 x 8 subimage extracted from the right eye of the left image. At the bottom of the figure are the 64 numbers representing the brightness of the pixels in that subimage. The numbers below 100 in the upper right of the subimage represent the lower reflection from the dark of the eye (iris), while the higher numbers represent the brighter white of the eye. A color image would have three numbers for each pixel, perhaps one value for red, one for blue, and one for green. Digital images are most commonly displayed on a monitor, which is basically a television screen with a digital image memory. A color image that has 500 rows and 500 columns is roughly equivalent to what you see at one instant of time on your TV. A pixel is displayed by energizing a small spot of luminescent material; displaying color requires energizing 3 neighboring spots of different materials. A high resolution computer display has roughly 1200 by 1000 pixels. The next chapter discusses digital images in more detail, while coding and interpretation of color in digital images is treated in Chapter 6.

Image Database Query

Huge digital memories, high bandwidth transmission and multimedia personal computers have facilitated the development of image databases. Good use of the many existing images requires good retrieval methods. Standard database techniques apply to images that have been augmented with text keys; however, *content-based* retrieval is needed and is a topic of much current research. Suppose that a newly formed company wants to design and protect a new logo and that an artist has created several candidates for the company to consider. A logo cannot be used if it is too similar to one of an existing company, so a database of existing logos must be searched. This operation is analagous to patent search and is done by humans, but could be greatly aided by machine vision methods. See Figure 1.2. There are many similar problems. Suppose an architect or an art historian wants to search for buildings with a particular kind of entryway. It would be desirable to just provide a picture, perhaps fetched from the database itself, and request the system to produce other similar pictures. In a later chapter, you will see how geometric, color, and texture features can be used to aid in answering such an image database query. Suppose that an advertising

agency wants to search for existing images of young children enjoying eating. This semantic requirement, which is simple for humans to understand, presents a very high level of difficulty for machine vision. Characterizing “children”, “enjoyment”, and “eating” would require complex use of color, texture, and geometric features. We note in passing that a computer algorithm has been devised that decides whether or not a color image contains a naked person. This could be useful for parents who want to screen images that their children retrieve from the web. Image database retrieval methods are treated in Chapter 8.



Figure 1.2: Image query by example: query image(left) and two most similar images produced by an image database system (Courtesy of Graphic-sha, Tokyo).

Inspecting crossbars for holes

In the late 1970’s an engineer in Milwaukee implemented a machine vision system that successfully counted the number of bolt holes in crossbars made for truck companies. The truck companies demanded that every crossbar be inspected before being shipped to them, because a missing bolt hole on a partly assembled truck was a very costly defect. Either the assembly line would have to be stopped while the needed hole was drilled, or worse, a worker might ignore placing a required bolt in order to keep the production line running. To create a digital image of the truck crossbar, lights were placed beneath the existing transfer line and a digital camera above it. When a crossbar came into the field of view, an image was taken. Dark pixels inside the shadow of the crossbar were represented as 1’s indicating steel, and pixels in the bright holes were represented as 0’s, indicating that the hole was drilled. The number of holes can be computed as the number of *external corners* minus the number of *internal corners* all divided by four. Figure 1.3 shows three bright holes (‘0’s) in a background of ‘1’s. An *external corner* is just a 2 x 2 set of neighboring pixels containing exactly 3 ones while an *internal corner* is a 2 x 2 set of neighboring pixels containing exactly 3 zeroes. Example processing of an image with 7 rows and 33 columns is shown in the figure and a skeleton algorithm is also shown. Holecounting is only one example of many simple, but powerful operations possible with digital images. (As the exercises below show, the holecounting algorithm is correct only if the holes are “4-connected” and “simply connected” — that is, they have no background pixels inside them. These concepts are discussed further in Chapter 3 and in more detail in the text by Rosenfeld.)

Examining the inside of a human head.

Magnetic resonance imaging (MRI) devices can sense materials in the interior of 3D objects. Figure 1.4 shows a section through a human head: brightness is related to movement of material, so this is actually a picture of blood *flow*. One can “see” important blood vessels.

Input a binary image and output the number of holes it contains.

\mathbf{M} is a binary image of \mathbf{R} rows of \mathbf{C} columns.

'1' represents material through which light has not passed;

'0' represents absence of material indicated by light passing.

Each region of '0's must be 4-connected and all image border pixels must be '1's.

\mathbf{E} is the count of *external corners* (3 ones and 1 zero)

\mathbf{I} is the count of *internal corners* (3 zeros and 1 one)

```
integer procedure Count_Holes( $\mathbf{M}$ )
{
  examine entire image, 2 rows at a time;
  count external corners  $\mathbf{E}$ ;
  count internal corners  $\mathbf{I}$ ;
  return(number_of_holes = ( $\mathbf{E}$  -  $\mathbf{I}$ )/4);
}
```

Algorithm 1: Skeleton of algorithm for counting holes in a binary image.

Exercise 1 Hole counting

Consider the following three images, which are 4x5, 4x4, and 4x5 respectively.

1	1	1	1	1
1	0	1	0	1
1	0	1	0	1
1	1	1	1	1

1	1	1	1
1	1	0	1
1	0	1	1
1	1	1	1

1	1	1	1	1
1	0	1	0	1
1	0	0	0	1
1	1	1	1	1

In scanning for corner patterns, 12, 9, and 12 2x2 neighbors are checked by Algorithm 1 for the three images above. Each 2x2 neighborhood matches one of these patterns 'e', 'i', 'n', for external corner, internal corner, and neither. (a) For each of the three images, how many of each 2x2 pattern are there? (b) Does the holecounting formula work for all three images?

1	1	1	0	0	1	1	1
1	0	1	1	1	1	0	1

(a) 2 x 2 "external corner" patterns

0	0	0	1	1	0	0	0
0	1	0	0	0	0	1	0

(b) 2 x 2 "internal corner" patterns



(c) Three bright holes in dark background

	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	e	i
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	0	1		
2	1	0	0	0	1	1	1	1	1	1	0	1	1	0	0	1		
3	1	1	1	1	1	0	0	1	1	1	0	0	1	1	0	1		
4	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	1		
5	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1		
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

(d) Binary input image 7 rows high and 16 columns wide

	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	e	i
0	e			e					e		e		e		e		6	0
1									e	i							1	1
2	e			e	e		e				i	e	e	i			6	2
3				e	i		i	e				i		i			2	4
4				e	i		i	e		e					e		4	2
5					e		e										2	0
6																	0	0

(e) External corners patterns marked with 'e'; internal corners marked 'i'

Figure 1.3: Counting the number of holes in a binary image: 21 external corner patterns ('e') minus 9 internal corner patterns ('i') divided by 4 yields a count of 3 holes. Why?

The wispy comet-like structures are associated with the eyes. MRI images are used by doctors to check for tumors or blood flow problems such as abnormal vessel constrictions or expansions. The image at the right in Figure 1.4 was made from a copy of the one on the left by making every pixel of value 208 or more bright (255) and those below 208 dark (0). Most pixels correctly show blood vessels versus background, but there are many incorrectly “colored” pixels of both types. Machine vision techniques are often used in medical image analysis, although usually to aid in data presentation and measurement rather than diagnosis itself. Wouldn’t it be great if we could “see” thoughts occurring in the brain! Well, it turns out that MRI can sense organic activity related to thought processes and this is a very exciting current area of research.



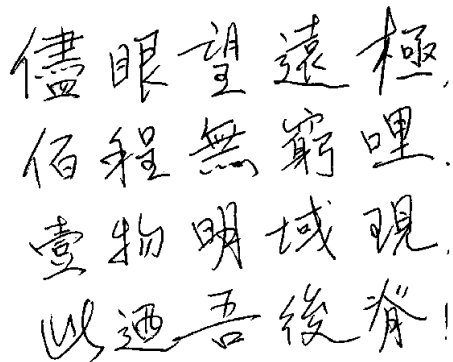
Figure 1.4: Magnetic resonance image (left) where brightness relates to blood flow and binary image (right) resulting from changing all pixels with value 208 or above to 255 and those below 208 to 0. Image courtesy of James Siebert, Michigan State Radiology.

Exercise 2 How many pixels per hole?

Consider the application of counting holes in truck crossbars at a more detailed level. Suppose that the area of the crossbar that is imaged is 50 inches long and 10 inches wide and suppose that this area almost fills up a digital image of 100 rows and 500 columns of pixels. Suppose a particular bolt hole in the crossbar is $1/2$ inch in diameter. What would you expect the radius and area of its image to be in terms of pixels?

Exercise 3 Imaging coins as pixels.

This problem is related to the one above. Obtain some graph paper (0.25 inch squares would be good) and a quarter. Randomly place the quarter on the graph paper and trace its circumference: do this five times. For each of the five placements, estimate the area of the image of the quarter in pixel units (a) by deciding whether a pixel is part of the quarter or not (no fractions) and (b) for each pixel cut by the circumference, estimate to the nearest tenth of a pixel how much of the pixel is part of the image of the quarter. After doing these measurements, compute the mean and standard deviation of the image area separately for methods (a) and (b).



儘眼望遠極,
佰程無窮哩。
壹物明域現,
此迺吾後脊!

I looked as hard as I could see,
beyond 100 plus infinity
an object of bright intensity
– it was the back of me!

Figure 1.5: (Left) Chinese characters and (right) English equivalent. Is it possible that a machine could automatically translate one into the other? Chinese characters and poem courtesy of John Weng.

Processing scanned text pages

A common problem is to convert information from paper documents into digital form for information systems. For example, we might want to make an old book available on the Internet, or we might need to convert a blueprint of some object into a geometry file so that the part can be made by a numerically controlled machine tool.

Figure 1.5 shows the same message in both Chinese and English. The Chinese characters were written on paper and scanned into an image of 482 rows and 405 columns. The postscript file encoding the graphics and printed in the figure has a size of 68,464 bytes. The English version is stored in a file of 115 bytes, each holding one ASCII character. There is an entire range of important applications in processing documents. Recognizing individual characters from the dots of the scanner or FAX files is one such application that is done fairly well today, provided that the characters conform to standard patterns. Providing a semantic interpretation of the information, possibly to be used for indexing in a large database, is a harder problem.

Accounting for snow cover using a satellite image

Much of the earth's surface is scanned regularly from satellites, which transmit their images to earth in digital form. These images can then be processed to extract a wealth of information. For example, inventory of the amount of snow in the watershed of a river may be critical for regulating a dam for flood control, water supply, or wildlife habitat. Estimates of snow mass can be made by accounting for the number of pixels in the image that appear as snow. A pixel from a satellite image might result from sensing a 10 meter by 10 meter spot of earth, but some satellites reportedly can see much smaller spots than that. Often, the satellite image must be compared to a map or other image to determine which pixels are in a particular area or watershed. This operation is usually manually-aided by a human user interacting with the image processing software and will be discussed more in Chapter 11 where image matching is covered.

Computers are known for their ability to handle large amounts of data; certainly the earth scanning satellites produce a tremendous amount of data useful for many purposes. For example, counts and locations of snow pixels might be input to a computer program that simulates the hydrology for that region. (Temperature information for the region must be input to the program as well.) Another related application is taking inventory of crops and predicting harvests. Yet another is taking inventory of buildings for tax purposes: this is usually done manually with pictures taken from airplanes.



Figure 1.6: Satellite image of Wenatchie River watershed in Washington State.

Understanding a scene of parts

At many points of manufacturing processes, parts are transferred on conveyors or in boxes. Parts must be individually placed in machines, packed, inspected, etc. If the operation is dull or dangerous, a vision-guided robot might provide a solution. The underlying image of Figure 1.7 shows three workpieces in a robot's workspace. By recognizing edges and holes, the robot vision system is able to guess at both the identity of a part and its position in the workspace. Using a 3D model made by computer-aided-design (CAD) for each guessed part and its guessed position, the vision system then compares the sensed image data with a computer graphic generated from the model and its position in space. Bad matches are rejected while good matches cause the guess to be refined. The bright lines

in Figure 1.7 show three such refined matches between the image and models of the objects it contains. Finally, the robot eye-brain can tell the robot arm how to pick up a part and where to put it. The problems and techniques of 3D vision are covered in the later chapters of this text.



Figure 1.7: An inspection or assembly robot matches stored 3D models to a sensed 2D image (Courtesy of Mauro Costa).

Exercise 4 Other problem areas.

Describe a problem different from those already discussed for which machine vision might provide a solution. If you do not have a special application area in mind, choose one for the moment. What kind of scenes would be sensed? What would an image be like? What output would be produced?

Exercise 5 Examining problem context.

Problems can be solved in different ways and a problem solver should not get trapped early in a specific approach. Consider the problem of identifying cars in various situations: (a) entering a parking lot or secured area, (b) passing through a toll gate, (c) exceeding the speed limit. Several groups are developing or have developed machine vision methods to read a car's license plate. Suggest an alternative to machine vision. How do the economic and social costs compare to the machine vision approach?

1.3 Operations on Images

This book presents a large variety of image operations. Operations can be grouped into different categories depending on their structure, level, or purpose. Some operations are for the purpose of improving the image solely for human consumption, while others are for extracting information for downstream automatic processing. Some operations create new output images, while others output non-image descriptions. A few important categories of image operations follow.

Changing pixels in small neighborhoods

Pixel values can be changed according to how they relate to a small number of neighboring pixels, for example, neighbors in adjacent rows or columns. Frequently, isolated 1's or 0's in a binary image will be reversed in order to make them the same as their neighbors. The purpose of this operation could be to remove likely noise from the digitization process. Or, it could be just to simplify image content; for example, to ignore tiny islands in a lake or imperfections in a sheet of paper. Figure 1.8 shows a binary image of some red blood cells that has been cleaned by removal of tiny regions within a larger uniform background. These operations are treated in Chapter 3.

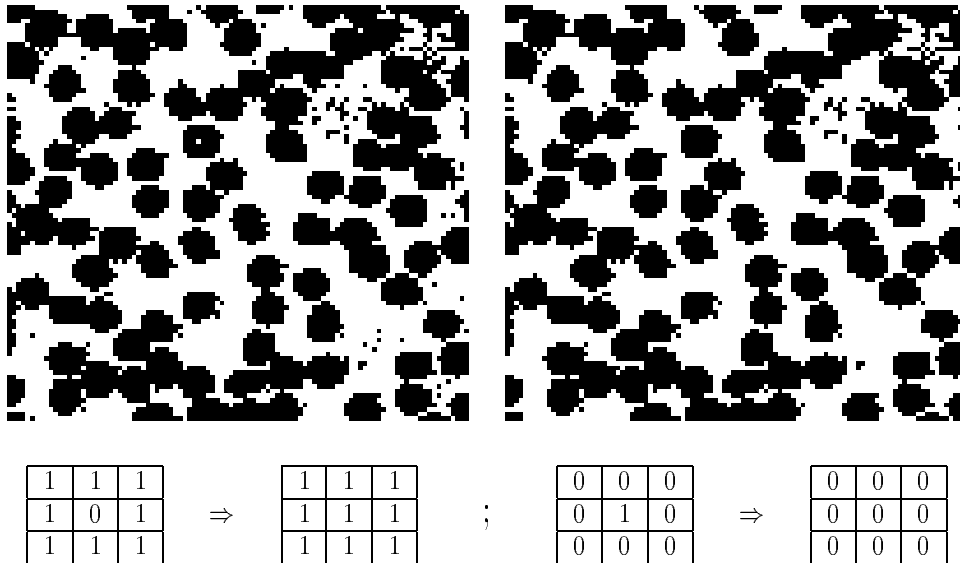


Figure 1.8: (Top left) Binary microscope image of red blood cells; (top right) cleaner image resulting from removal of tiny dark regions inside light regions or *visa versa*; (bottom) templates showing how pixel neighborhoods can be cleaned.

Enhancing an entire image

Some operations treat the entire image in a uniform manner. The image might be too dark – say its maximum brightness value is 120 – so all brightness values can be scaled up by a factor of 2 to improve its displayed appearance. Noise or unnecessary detail can be

Exercise 6

Assume that '1' represents light background and '0' represents red blood cell in the images of Figure 1.8. Identify cases where pixels of the left image have been changed in the right image using the templates shown at the bottom of the figure. For each of the two templates, find two cases where it was applied.

removed by replacing the value of every input pixel with the average of all nine pixels in its immediate neighborhood. Alternatively, details can be enhanced by replacing each pixel value by the contrast between it and its neighbors. Figure 1.9 shows a simple contrast computation applied at all pixels of an input image. Note how the boundaries of most objects are well detected. The output image results from computations made only on the local 3x3 neighborhoods of the input image. Chapter 5 describes several of these kinds of operations. Perhaps an image is taken using a fish eye lens and we want to create an output image with less distortion: in this case, we have to “move” the pixel values to other locations in the image to move them closer to the image center. Such an operation is called *image warping* and is covered in Chapter 11.

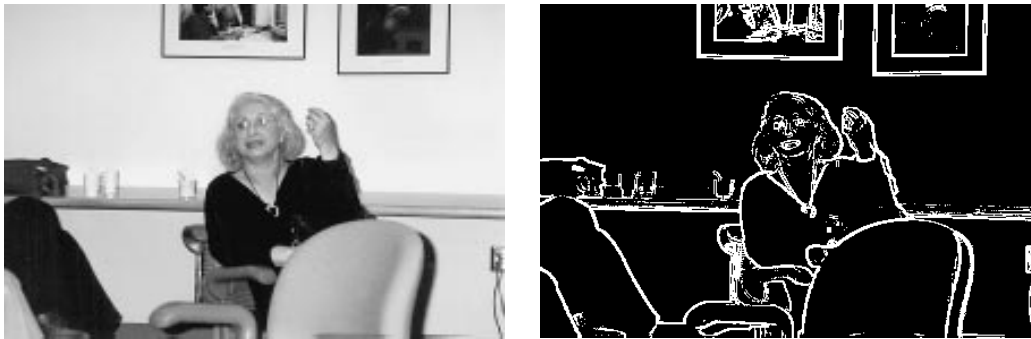


Figure 1.9: Contrast in the left image is shown in the right image. The top 10% of the pixels in terms of contrast are made bright while the lower 90% are made dark. Contrast is computed from the 3x3 neighborhood of each pixel.

Combining multiple images

An image can be created by adding or subtracting two input images. Image subtraction is commonly used to detect change over time. Figure 1.10 shows two images of a moving part and the difference image resulting from subtracting the corresponding pixel values of the second image from those of the first image. Image subtraction captures the boundary of the moving object, but not perfectly. (Since negative pixel values were not used, not all changes were saved in the output image.) In another application, urban development might be more easily seen by subtracting an aerial image of a city taken five years ago from a current image of the city. Image addition is also useful. Figure 1.11 shows an image of Thomas Jefferson “added” to an image of the great arch opening onto the lands of the Louisiana Purchase; more work is needed in this case to *blend* the images better.

Computing features from an image

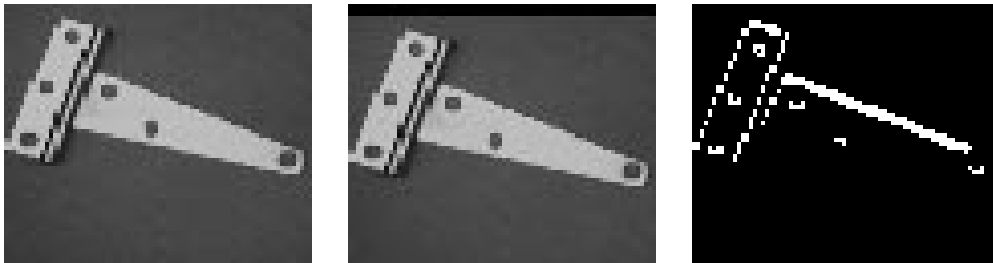


Figure 1.10: Images of a moving part (left and center) and a difference image (right) that captures the boundary of the part.

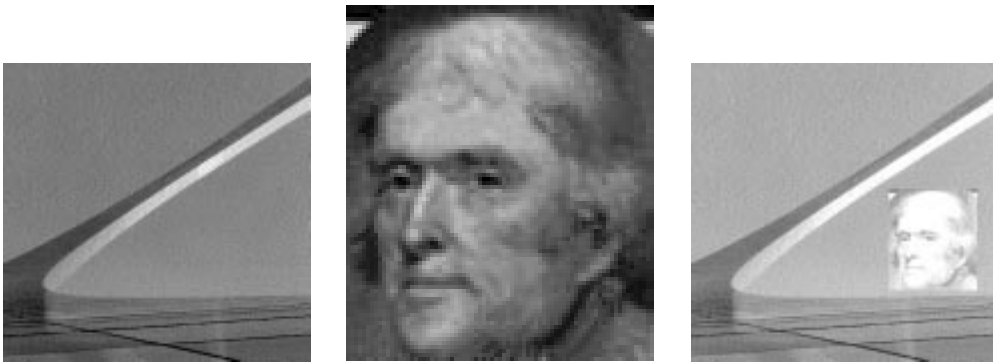


Figure 1.11: Image of the great archway at St. Louis (left); face of Jefferson (center); and, combination of the two (right).

We have already seen the example of counting holes. More generally, the regions of 0's corresponding to holes in the crossbar inspection problem could be images of objects, often called *blobs* – perhaps these are microbes in a water sample. Important features might be average object area, perimeter, direction, etc. We might want to output these important features separately for every detected object. Chapter 3 describes such processing. Chapters 6 and 7 discuss means of quantitatively summarizing the color or texture content of regions of an image. Chapter 4 shows how to classify objects according to these features; for example, is the extracted region the image of microbe A or B? Figure 1.12 shows output from a well-known algorithm applied to the blood cell image of Figure 1.8 giving features of separate regions identified in the image, including the region area and location. Regions with area of about 50 pixels correspond to isolated blood cells while the larger regions are due to several touching cells.

Extracting non-iconic representations

Higher-level operations usually extract representations of the image that are non-iconic, that is, data structures that are not like an image. (Recall that extraction of such descriptions is often defined to be the goal of computer vision.) Figure 1.12 shows a non-iconic description derived from the blood cell image. In addition to examples already mentioned,

Object	Area	Centroid	Bounding Box
2	383	(8.82 , 20)	[(1, 1) (22, 39)]
3	83	(5.81 , 49.7)	[(1, 42) (11, 55)]
6	1048	(18.7 , 75.4)	[(1, 35) (40, 100)]
c	48	(17.7 , 12.9)	[(14, 9) (21, 17)]
h	194	(30.6 , 18.4)	[(22, 9) (40, 29)]
m	53	(33.4 , 2.6)	[(25, 1) (39, 6)]
p	52	(30 , 42.6)	[(27, 39) (33, 47)]
t	50	(35.1 , 30.7)	[(32, 27) (38, 35)]
z	45	(43 , 31.6)	[(40, 28) (46, 35)]
B	52	(45 , 87.5)	[(42, 83) (48, 91)]
C	54	(47.9 , 53.1)	[(44, 49) (52, 57)]
D	304	(54.4 , 82.1)	[(46, 63) (66, 100)]
#	44	(87.6 , 77.8)	[(85, 74) (90, 82)]
...			

Figure 1.12: Some components automatically identified in the blood cell image in Figure 1.8 (output has been abbreviated). Single blood cells have an area of about 50 pixels: larger components consist of several touching cells.

consider a report of the count of microbes of type A and B in a slide from a microscope or the volume of traffic flow between two intersections of a city computed from a video taken from a utility pole. In another important application, the (iconic) input might be a scanned magazine article and the output a hypertext structure containing sections of recognized ASCII text and sections of raw images for the figures. As a final example, in the application illustrated in Figure 1.7, the machine vision system would output a set of three detections, each encoding a part number, three parameters of part position and three parameters of the orientation of the part. This scene description could then be turned over to the motion-planning system, which would decide on how to manipulate the three parts.

1.4 The Good, the Bad, and the Ugly

Having cited many applications of machine vision, we cannot proceed without saying that success is usually hard won. Often, implementors have to accept environmental constraints that compromise system flexibility. For example, scene lighting might have to be carefully controlled, or objects might have to be mechanically separated or positioned before imaging. This is because the real world yields exorbitant variations in the input image, challenging the best computer algorithms in their task of extracting the “essence” or *invariant features* of objects. Appearance of an object can vary significantly due to changes in illumination or presence of other objects, which might be unexpected. Consider, for example, the shadows in Figure 1.7 and Figure 1.9. Moreover, decisions about object structure must often be made by integrating a variety of information from many pixels of the image. For example, the brightness of the tops of the glasses on the counter in Figure 1.9 is the same as that of the wall, so no glass-wall boundary is evident at the pixel level. In order to recognize each glass as a separate object, pixels from a wider area must be grouped and organized.

Humans are quite good at this, but developing flexible grouping processes for machine vision has proved difficult. Problems of occlusion hamper recognition of 3D objects. Can a vision system recognize the person or the chair in Figure 1.9, even though neither appears to have legs? At a higher level yet, what model of a *dog* could empower a machine to recognize the diverse individuals that could be imaged? These difficulties, and others, will be discussed further throughout this book.

Exercise 7

What invariant features of the following objects enable you to recognize them in rain or sunshine, alone or alongside other objects, from the front or side: (a) your tennis shoes, (b) your front door, (c) your mother, (d) your favorite make of automobile?

1.5 Use of Computers and Software

Computers are legendary for accurate accounting of quantitative information. Computing with images has gone on for over 30 years – initially mostly in research labs with mainframe computers or in production shops with special-purpose computers. Recently, large inexpensive memories and high speed general-purpose processors have brought image computing potential to every multimedia personal computer user, including the hobbyist working in her dining room.

One can compute with images in different ways. The easiest is to acquire an existing program that can perform many of the needed image operations. Some programs are free to the public; others must be purchased: some options are given in the Appendices. Many free images are available from the World-Wide-Web. To control your own image input, you can buy a flatbed scanner or a digital camera, each available for a few hundred dollars. Software libraries are available which contain many subroutines for processing images: the user writes an application program which calls the library routines to perform the required operations on the user's image data. Most companies selling input devices for machine vision also provide libraries for image operations and even driver programs with nice graphical user interfaces (GUI). Special-purpose hardware is available for speeding up image operations that can take many seconds, or even minutes, on a general purpose processor. Many of the early parallel computers costing millions of dollars were designed with image processing as a primary task; however, today most of the critical operations can be provided by sets of boards costing a few thousand dollars. Usually, special hardware is only needed for high production rates or real-time response. Special programming languages with images and image operations as language primitives have been defined; sometimes, these have been combined with operations for controlling an industrial robot. Today, it is apparent that much good image processing can and will be done using a general purpose language, such as C, and a general purpose computer available via mail order or the local computer store. This bodes exceedingly well for the machine vision field, since challenging problems will now be attacked from all directions possible! The reader is invited to join in.

1.6 Related Areas

Computer vision is related to many other disciplines: we are not able to pursue all of these relations in depth in this text. First, it is important to distinguish between *image processing* and *image understanding*. Image processing is primarily concerned with the transformation of images into images, whereas, image understanding is concerned with making decisions based on images and explicitly constructing the scene descriptions needed to do so. Image processing is quite often used in support of image understanding and thus will be treated to some extent in this book. Books concerned with image processing typically are based on the model of an image as a continuous function $f(x, y)$ of two spatial parameters x and y , whereas this text will concentrate on the model of an image as a discrete 2D array $I[r, c]$ of integer brightness samples. In this book, we use the terms *computer vision*, *machine vision*, and *image understanding* interchangeably; however, experts would certainly debate their nuances.

The psychology of human perception is very important for two reasons; first, the creator of images for human consumption must be aware of the characteristics of the client, and secondly, study of the tremendous human capability in image understanding can guide our development of algorithms. While this text includes some discussion of human perception and cognition, its approach is primarily hands-on problem solving. The physics of light, including optics and color science, is important to our study. We will present the basic material necessary; however, readers who want to be experts on illumination, sensing, or lenses will need to access the related literature. A variety of mathematical models are used throughout the text; for mastery, the reader must be comfortable with the notions of functions, probability, calculus and analytical geometry. The intuitive concepts of image processing often strengthen the mathematical concepts. Finally, any book about computer vision must be strongly related to computer graphics. Both fields are concerned with how objects are viewed and how objects are modeled; the prime distinction is one of direction – computer vision is concerned with description and recognition of objects from images, while computer graphics is concerned with generation of images from object descriptions. Recently, there has been a great deal of integration of these two areas: computer graphics is needed to display computer vision results and computer vision is needed to make object models. Digital images are commonly used as input for computer graphics products.

1.7 The rest of the book

The previous sections informally introduced many of the concepts in the book and indicated the chapters in which they are treated. The reader should now appreciate the range of problems attacked by machine vision and a few of its methods. The chapters that immediately follow describe 2D machine vision. In those chapters, the image is analyzed in self-referencing terms of pixels, rows, intersections, colors, textures, etc. To be sure, knowledge about how the image was taken from the real 3D world is present, but the relationship between image pixels and real-world elements is obvious – only the scale is different. For example, a radiologist can readily tell from an image if a blood vessel is constricted without knowing much about the physics of the sensor or about what portion of the body a pixel represents. So can a machine vision program. Similarly, the essence of a character recognition algorithm has nothing to do with the real font size being scanned. Consequently, the material in Part II has a 2D character and is more generic and simpler than material in Part III. In the Part III chapters, the 3D nature of objects and the viewpoints used to

image them are crucial. The analysis cannot be done with the coordinates of a single image because we need to relate multiple images or images and models; or, we need to relate a sensor's view to a robot's view. In Part III, we are analyzing 3D scenes, not 2D images, and the most important tool for the analysis is 3D analytical geometry. As in computer graphics, the step from 2D to 3D is a large one in terms of both modeling abstraction and computational effort.

1.8 References

The machine vision literature is highly specialized according to application area. For example, the paper by Fleck *et al* addresses how pornographic images might be detected so they could be screened from a child's computer. Our discussion of the hole-counting algorithm was derived from the work of Kopydlowski (1983), which described its use in the inspection of truck crossbars. The design of sensors for satellites is very different from the design of medical instruments, for example. Manufacturing systems are different still. Some references to the special areas are Nagy (1972) and Hord (1982) for remote sensing, Glasby (1995) for biological sciences, Ollus (1987) and QCAV (1999) for industrial applications, and ASEE (1983) for agricultural applications. One of several early articles on the early development of color CCD cameras is by Dillon *et al* (1978). Problems, methods, and theory shared among several application areas are, of course, topics for textbooks, this one included. Perhaps the first textbook on picture processing using a computer by Rosenfeld (1969) contains material on image processing without use of higher-level models. The book by Ballard and Brown (1982), perhaps the first *Computer Vision* text, concentrated on image analysis using higher-level models. Levine's text (1985) is noteworthy due to its inclusion of significant material on the human visual system. The two-volume set by Haralick and Shapiro (1992) is a modern resource for algorithms and their mathematical justification. The text by Jain, Kasturi, and Schunk (1995) is a modern introduction to machine vision with primarily an engineering viewpoint.

1. ASAE (1983), *Robotics and intelligent machines in agriculture: Proceedings of the 1st International Conf. on Robotics and Intelligent Machines in Agriculture*, 2-4 Oct 1983, Tampa, FL, American Society of Agricultural Engineers, St. Joseph, MI (1984).
2. D. H. Ballard and C. M. Brown (1982), **Computer Vision**, Prentice-Hall, Englewood Cliffs, NJ.
3. M. Fleck, D. Forsyth and C. Pregler (1996), *Finding naked people [in images]*, in **Proc. of the European Conference on Computer Vision**, Springer Verlag (1996)593-602.
4. C. A. Glasby and G. W. Horgan (1995), **Image Analysis for the Biological Sciences**, John Wiley and Sons, Chichester, England.
5. R. Haralick and L. Shapiro (1992/3), **Computer and Robot Vision, Volumes I and II**, Addison-Wesley
6. R. M. Hord (1982), **Digital image processing of remotely sensed data**, Academic Press, New York, 1982, 256p.

7. T. Igarashi (Ed.) (1983), **World Trademarks and Logotypes**, Graphic-sha, Tokyo.
8. T. Igarashi (Ed.) (1987), **World Trademarks and Logotypes II: A Collection of International Symbols and their Applications**, Graphic-sha, Tokyo.
9. R. Jain, R. Kasturi and B. Schunk (1995), **Machine Vision**
10. D. Kopydlowski (1983), *100% Inspection of Crossbars using Machine Vision*, Publication **MS83-210**, Society of Manufacturing Engineers, Dearborn, MI (1983).
11. M.D. Levine (1985), **Vision in Man and Machine**, McGraw-Hill.
12. G. Nagy (1972), *Digital Image Processing Activities in Remote Sensing for Earth Resources*, Proceedings IEEE, Vol 60, No. 10 (Oct 1972)1177-1200.
13. M. Ollus (ed.) (1987), *Digital image processing in industrial applications: Proceedings of the 1st IFAC workshop, Espoo, Finland, 10-12 June 1986*, Pergamon Press (1987), 173p.
14. W. Pratt (1991), **Digital Image Processing, 2nd Ed.**, John Wiley, New York, 1991.
15. QCAV (1999), *Quality Control by Artificial Vision: Proceedings of the 5th International Conf. on Quality Control by Artificial Vision*, Trois-Rivieres, Canada, (18-21 May 1999)349p.
16. A. Rosenfeld (1969), **Picture Processing by Computer**, Academic Press, New York, 1969.

1.9 Additional Exercises

Below are several thought questions requiring essay-type answers. A few questions require quantitative thinking. Most of these questions will be revisited in more detail later in the text.

Exercise 8 The problem of selling produce.

Consider the problem a grocery store checker has in charging you for your purchases. Bar code technology makes some items easy; a soup can need only be dragged over the bar code reader until an acknowledging “beep” is heard. This system does not work for produce picked by the shopper in variable unpackaged quantities and the checker has to stop to do special actions. What actions? Do you think that a camera could be placed above or inside the combined scale/bar-code-reader to tell the cash register what kind of produce it is? Suppose using methods of this textbook a machine vision system could tell the difference between spinach greens and collard greens, Fuji apples versus MacIntosh, etc. Describe how this machine vision system would integrate with the other technology already in place to help the checker compute your bill.

Exercise 9 Counting blood cells

Examine the image of red blood cells in Figure 1.8 and the sample of automatically computed features in Figure 1.12. Is there potential for obtaining a count of red blood cells, say within 5% accuracy, shown in this example? Explain.

Exercise 10 3D model from video?

Suppose you are given a video of Notre Dame Cathedral in Paris. The video was taken by a person walking around the outside and inside of the cathedral; so many viewpoints are available. Do you think that you could make a reasonable 3D model of that cathedral using only the video? (If you lack confidence, assume you're an architect.) If not, why not? If so, how could you construct a 3D model when all you had were 2D images?

Exercise 11 On computing contrast

Think of a method that computes the contrast of a 3x3 image neighborhood similar to what is shown in Figure 1.9. Assume that the 9 input pixel values are brightness values between 0 and 255 and that the output pixel value is a single value between 0 and 255 measuring the amount of contrast. (The picture at the right in Figure 1.9 actually uses only the two pixel values 0 and 255; however, you may use the entire range.)

Exercise 12 On face interpretation

(a) Is it easy for you to decide the gender and approximate age of persons pictured in magazine ads? (b) Psychologists might tell us that humans have the ability to see a face and immediately decide on the age, sex, and degree of hostility of the person. Assume that this ability exists for humans. If you think it would be based on image features, then what are they? If you think that image features are not used, then explain how humans might make such decisions?

Exercise 13 Is a picture worth a thousand words?

Consider the following passage from **The Sound and the Fury** by William Faulkner (Vintage Books Edition, 1987, Copyright 1984 by Jill Faulkner Summers, page 195). Do you think that a machine could extract such a description from a video of the scenes discussed?

I could smell the curves of the river beyond the dusk and I saw the last light supine and tranquil upon tideflats like pieces of broken mirror, then beyond them lights began in the pale clear air, trembling a little like butterflies hovering a long way off.

Exercise 14 * Toward the Correctness of Holecounting

This question requires some thought and reading that extends beyond this Chapter and should be regarded as enrichment. (a) How many possible 2x2 neighborhood patterns are there in a binary image? List them all. (b) Which of the patterns of part (a) cannot occur in a binary image that is 4-connected? Define "border point" to be the center grid point of a 2x2 pixel neighborhood that contains both 0' and 1' pixels. (c) Argue that a single hole can be accounted for by just counting the number of 'e' and 'i' patterns along its border and that the formula $n = (e - i)/4$ is correct when one hole is present. (d) Argue that no two holes can have a common border point. (e) Argue that the formula is correct when an arbitrary number of holes is present.

Exercise 15 Are binary images suitable?

Consider imaging some scene and then processing it to produce a binary image as output, such as the red blood cell image, where target objects image as regions of 0's and the background, or non-objects, image as 1's. Think about whether or not this can be achieved for the following scenes. In your opinion, why can or can't we create such a binary image?

1. The input image is of a paper containing typing and it is scanned using a page scanner. Our overall objective is to recognize most of the typed characters and make an ASCII file so that we can edit the text using a word processor.
 2. The input image is an X-ray of someone's head. We would like to find bullets or tumors as regions of 0's in a background of 1's.
 3. The input image is a satellite image of Richmond, VA taken in spring. By tuning the sensor or by some simple computer algorithm, we would like to create an image where a 0' indicates the presence of an azalea bush and a 1' indicates no bush.
 4. We want to check the width of the stem of an auto engine valve by just counting the number of pixels across its shadow. Since we will be making hundreds of thousands of valves per day, detailed control of the environment and costly equipment can be justified.
-