KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)

(Established Under Section 3 of UGC Act, 1956)

Coimbatore-21

Department of Computer Science, Applications and Information Technology

M.Sc. Computer Science

Semester-III

18CSP303	**DIGITAL IMAGE PROCESSING**	**4H – 4C**

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Int : **40** Ext : **60** Total: **100**

SCOPE

The objectives of this course are tomake the students learn the fundamental theories and techniques of digital image processing cover the fundamental concepts of visual perception and image acquisition basic techniques of image manipulation segmentation and coding and a preliminary understanding of Computer Vision.

OBJECTIVES

- To perform image manipulations and analysis in many different fields.
- To provide students with the ability to apply knowledge of computing mathematics science and engineering to solve problems in multidisciplinary research.

UNIT-I

Introduction: Digital image processing – Origins of digital image processing- Examples of fields that use digital image processing-Fundamental steps in digital image processing-Components of an image processing system-Representing digital image.

UNIT-II

Some Basic relationships between Pixels-Basic gray level transformations- Histogram processing - Basic spatial filtering- Smoothing special filtering- Image Degradation/ Restoration process- Noise Models.

UNIT-III

Image Segmentation: Thresholding - Edge Based Segmentation – Region Based Segmentation – Matching. Image Compression: Error Criterion - Lossy Compression - Lossless Compression.

UNIT-IV

Shape Representation and Description: Region Identification - Contour Based Representation And Description – Region Based Shape Representation And Description

UNIT-V

Image Recognition: Introduction – Statistical Pattern Recognition - Neural Net- Syntactic Pattern Recognition - Graph Matching - Clustering

SUGGESTED READINGS

TEXT BOOK

1. Rafael, C. Gonzalez ., & Richard, E. Woods. (2008). Digital Image Processing (3rd Ed.). New Delhi:Pearson Education.

REFERENCES

1. Chanda, B., & Dutta Majumder, D. (2000). Digital Image Processing and Analysis (1st ed.). New Delhi: Prentice Hall of India.
2. Milan Sonka., Vaclav Hlavac.,& Roger Boyle. (2004). Image Processing Analysis and Machine Vision (2nd ed.). New Delhi: Vikas Publishing House.
3. Nick Efford. (2000). Digital Image Processing – A Practical introduction using JAVA (1st ed.). New Delhi: Pearson Education Limited.

WEB SITES

http://www.cs.dartmouth.edu/farid/tutorials/fip.pdf
http://www.imageprocessingbasics.com/
http://www.astropix.com/HTML/J_DIGIT/TOC_DIG.HTM

Signature of the Faculty HOD

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC ACT 1956)
M.sc Computer Science
DEPARTMENT OF CS, CA & IT
Third Semester

LECTURE PLAN

STAFF NAME:Dr. Mohankumar.M
SUBJECT NAME: Digital Image Processing **SUBJECT CODE: 17CSP301**
SEMESTER: III **CLASS: II MSc CS**
Subject code: 18CSP303 **Batch: 2018-2020**

DIGITAL IMAGE PROCESSING

S.NO	LECTURE DURATION /PERIOD	TOPICS TO BE COVERVED	SUPPORT MATERIALS/TEXT BOOKS
		UNIT-1	
1	1	Introduction: Digital image processing	T1:PG.NO:23-25
2	1	Origins of digital image processing-	T1:PG.NO:25-29
3	1	Examples of fields that use digital image processing	W1+T1:PG.NO:30-33
4	1	Imaging in the ultra-violet band Imaging in the visible and Infrared band	T1:PG.NO:33-39
5	1	Imaging in micro wave band, Imaging in radio wave band, Examples in which other imaging modalities are used	T1:PG.NO:34-47
6	1	Fundamental steps in digital image processing	T1:PG.NO:47-50
7	1	Components of an image processing system, Representing digital image	T1:PG.NO:50-54 T1:PG.NO:76-79
8	1	Recapitulation and discussion of possible questions	
Total no of periods planned for unit 1: 8			
		UNIT-2	
1	1	Some Basic relationships between Pixels: Neighbors of a pixel	T1:PG.NO:88
2	1	Some Basic relationships between Pixels: Adjacency,connectivity, regions and Boundaries	T1:PG.NO:88-90
3	1	Some Basic relationships between Pixels: Distance Measures, Image operations on a pixel basis	T1:PG.NO:90-91
4	1	Some Basic gray level transformations: Image Negatives, Log Transformations	T1:PG.NO:98-101

5	1	Some Basic gray level transformations: Power Law Transformations Piece-wise linear transformations	T1:PG.NO:102-110
6	1	Histogram Functions: -Histogram Equalization,Histogram matching,Histogram specifications	T1:PG.NO:110-124
7	1	Local Enhancement, Use of histogram statistics for image enhancement	T1:PG.NO:125-137
8	1	Basic spatial filtering, Smoothing spatial filtering,-smoothing Linear filters,-order-statistics filters	T1:PG.NO:138-147
9	1	Image Degradation/ Restoration process, Noise Models	T1:PG.NO:242-252
10	1	Recapitulation and discussion of Possible questions	
Total no of Hours planned for unit 2: 10			
UNIT-3			
1	1	Image Segmentation: Introduction, Thresholding:- Threshold detection methods	R2:PG.NO:123-125
2	1	Optimal Thresholding , Multi-spectral Thresholding, Thresholding in Hierarchical data structures	R2:PG.NO:125-132
3	1	Edge Based Segmentation,Edge Image Thresholding,Edge Relaxation, Border Tracing	R2:PG.NO:134-146
4	1	Border Detection as graph searching Border detection as dynamic programming	R2:PG.NO:147-162
5	1	Houghs Transform,Border detection using location information, Region construction from borders	R2:PG.NO:163-175
6	1	Region Based Segmentation,Region merging, Region splitting	R2:PG.NO:176-180
7	1	Splitting and Merging, water shed segmentation	R2:PG.NO:181-187
8	1	Region growing post-Processing, Matching-Matching criteria	R2:PG.NO:188-189 R2:PG.NO:190
9	1	control strategies for matching, Image Compression: Error Criterion - Lossy Compression - Lossless Compression	R2:PG.NO:190-195 W3
10	1	Recapitulation and discussion of Possible questions	
Total no of periods planned for unit 3: 10			
UNIT-IV			

1	1	Shape Representation And Description - Region Identification	R2:PG.NO:228-232
2	1	Contour Based Representation And Description, chain Codes	R2:PG.NO:232-236
3	1	Simple geometric border representation	R2:PG.NO:237-239
4	1	Fourier Transforms of boundaries	R2:PG.NO:240-242
5	1	Boundary description using segment sequences	R2:PG.NO:242-244
6	1	B-Spline representation Other contour based shape description approaches- Shape invariants	R2:PG.NO:246-249 R2:PG.NO:249-254
7	1	Region Based Shape Representation And Description simple scalar region descriptors	R2:PG.NO:254-258
8	1	Moments, convex hull Graph representation based on region skeleton	R2:PG.NO:259-266 R2:PG.NO:267-270
9	1	Region decomposition, region neighborhood graphs	R2:PG.NO:271-273
10	1	Recapitulation and discussion of important questions	
Total no of periods planned for unit 4: 10			
UNIT-V			
1	1	Image Recognition: Introduction, knowledge representation	R2:PG.NO:290-296
2	1	Statistical Pattern Recognition,classification principles- Classifier setting,classifier learning	R2:PG.NO:297-298 R2:PG.NO:298-302
3	1	Cluster Analysis Neural Nets-Feed forward networks	R2:PG.NO:303-308 R2:PG.NO:308-312
4	1	Unsupervised learning-Hop field neural nets	R2:PG.NO:312-315
5	1	Syntactic pattern recognition,grammars and languages	R2:PG.NO:315-318
6	1	Syntactic analysis, syntactic classifier, Syntactic classifier learning, grammar Inference	R2:PG.NO:319-320
7	1	Graph Matching-Isomorphism of graphs and subgraphs,similarity of graphs- Clustering	R2:PG.NO:321-323
8	1	Recapitulation and discussion of Possible questions	R2:PG.NO:324-330 W4
9	1	Revision of previous end semester question paper	
10	1	Revision of previous end semester question paper	
Total no of periods planned for unit 5: 10			

TOTAL HOURS PLANNED FOR ALL THE FIVE UNITS = 48 HRS

TEXT BOOK:

1. T1- Rafael C. Gonzalez, Richard E. Woods. 2004. Digital Image Processing, 2nd Edition, Pearson Education, New Delhi.

REFERENCES:

1. **R1**:Chanda. B and Dutta Majumder .D. 2000. Digital Image Processing and Analysis, 1st Edition, Prentice Hall of India, New Delhi.
2. **R2**:Milan Sonka and Vaclav Hlavac and Roger Boyle. 2004. Image Processing, Analysis and Machine Vision, 2nd Edition, Vikas Publishing House, New Delhi.
3. **R3**:Nick Efford. 2000. Digital Image Processing – A Practical introduction using JAVA, 1st Edition , Pearson Education Limited, New Delhi.

WEB SITES:

1. **W1-** http://www.cs.dartmouth.edu/farid/tutorials/fip.pdf
2. **W2-** http://www.imageprocessingbasics.com/
3. **W3-** http://www.astropix.com/HTML/J_DIGIT/TOC_DIG.HTM

Faculty Signature HOD Signature

UNIT I

SYLLABUS:

Introduction: Digital image processing – Origins of digital image processing- Examples of fields that use digital image processing-Fundamental steps in digital image processing- Components of an image processing system- representing digital image.

Digital Image Basics

When using digital equipment to capture, store, modify and view photographic images, they must first be converted to a set of numbers in a process called digitization or scanning. Computers are very good at storing and manipulating numbers, so once your image has been digitized you can use your computer to archive, examine, alter, display, transmit, or print your photographs in an incredible variety of ways.

Pixels and Bitmaps

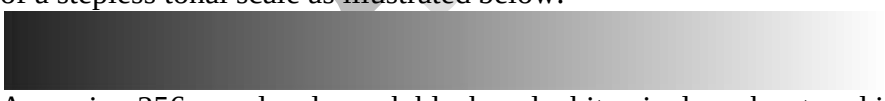
Digital images are composed of *pixels* (short for picture elements). Each pixel represents the color (or gray level for black and white photos) at a single point in the image, so a pixel is like a tiny dot of a particular color. By measuring the color of an image at a large number of points, we can create a digital approximation of the image from which a copy of the original can be reconstructed. Pixels are a little like grain particles in a conventional photographic image, but arranged in a regular pattern of rows and columns and store information somewhat differently. A digital image is a rectangular array of pixels sometimes called a *bitmap*.

Types of Digital Images

For photographic purposes, there are two important types of digital images—color and black and white. Color images are made up of colored pixels while black and white images are made of pixels in different shades of gray.

Black and White Images

A black and white image is made up of pixels each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different grays. Since the eye can barely distinguish about 200 different gray levels, this is enough to give the illusion of a stepless tonal scale as illustrated below:



Assuming 256 gray levels, each black and white pixel can be stored in a single byte (8 bits) of memory.

Color Images

A color image is made up of pixels each of which holds three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue

(sometimes referred to as RGB) are the primary colors for mixing light—these so-called additive primary colors are different from the subtractive primary colors used for mixing paints (cyan, magenta, and yellow). Any color can be created by mixing the correct amounts of red, green, and blue light. Assuming 256 levels for each primary, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors.
Note that for images of the same size, a black and white version will use three times less memory than a color version.

Binary or Bilevel Images

Binary images use only a single bit to represent each pixel. Since a bit can only exist in two states—on or off, every pixel in a binary image must be one of two colors, usually black or white. This inability to represent intermediate shades of gray is what limits their usefulness in dealing with photographic images.

Indexed Color Images

Some color images are created using a limited palette of colors, typically 256 different colors. These images are referred to as indexed color images because the data for each pixel consists of a palette index indicating which of the colors in the palette applies to that pixel. There are several problems with using indexed color to represent photographic images. First, if the image contains more different colors than are
in the palette, techniques such as dithering must be applied to represent the missing colors and this degrades the image. Second, combining two indexed color images that use different palettes or even retouching part of a single indexed color image creates problems because of the limited number of available colors.

Resolution

The more points at which we sample the image by measuring its color, the more detail we can capture. The density of pixels in an image is referred to as its *resolution*.
The higher the resolution, the more information the image contains. If we keep the image size the same and increase the resolution, the image gets sharper and more detailed. Alternatively, with a higher resolution image, we can produce a larger image with the same amount of detail.

For example, the following images illustrate what happens as we reduce the resolution of an image while keeping its size the same—the pixels get larger and larger and there is less and less detail in the image: (a)Original (400x262)(b)Half Size (200x131) Quarter Size (100x65) Eighth Size (50x32)

As we reduce the resolution of an image while keeping its pixels the same size—the image gets smaller and smaller while the amount of detail (per square inch) stays the same:

Color Terminology

While pixels are normally stored within the computer according to their red, green, and blue levels, this method of specifying colors (sometimes called the RGB *color space*) does not correspond to the way we normally perceive and categorize colors.

There are many different ways to specify colors, but the most useful ones work by separating out the hue, saturation, and brightness components of a color.

Primary and Secondary Colors and Additive and Subtractive Color Mixing

Primary colors are those that cannot be created by mixing other colors. Because of the way we perceive colors using three different sets of wavelengths, there are three primary colors. Any color can be represented as some mixture of these three primary colors.

There are two different ways to combine colors—*additive* and *subtractive* color mixing.

Subtractive color mixing is the one most of us learned in school, and it describes how two colored paints or inks combine on a piece of paper. The three subtractive primaries are Cyan (blue-green), Magenta (purple-red), and Yellow (not Blue, Red, and Yellow as we were taught).

Additive color mixing refers to combining lights of two different colors, for example by shining two colored spotlights on the same white wall. The additive color model is the one used in computer displays as the image is formed on the face of the monitor by combining beams of red, green, and blue light in different proportions.

Color printers use the subtractive color model and use cyan, magenta, and yellow inks. To compensate for the impure nature of most printing inks, a fourth color, black is also used since the black obtained by combining cyan, magenta, and yellow inks is often a murky dark

green rather than a deep, rich black. For this and other reasons, commercial color printing presses use a 4-color process to reproduce color
images in magazines.

A color created by mixing equal amounts of two primary colors is called a *secondary*.
In the additive color system, the primary colors are: and the secondary colors are:
In the subtractive color system, the roles of the primaries and secondaries are reversed.

Color Gamut

In the real world, the ideal of creating any visible color by mixing three primary colors is never actually achieved. The dyes, pigments, and phosphors used to create colors on paper or computer screens are imperfect and cannot recreate the full range of visible colors. The actual range of colors achievable by a particular device or medium is called its color gamut and this is mostly but not entirely determined by the characteristics of its primary colors. Since different devices such as computer monitors, printers, scanners, and photographic film all have different color gamuts, the problem of achieving consistent color is quite challenging. Different media also differ in their total dynamic range—how dark is the darkest achievable black and how light is the brightest white.

Color Management

The process of getting an image to look the same between two or more different media or devices is called *color management*, and there are many different color management systems available today. Unfortunately, most are complex, expensive, and not available for a full range of devices.

Hue

The hue of a color identifies what is commonly called "color". For example, all reds have a similar hue value whether they are light, dark, intense, or pastel.

Saturation : The saturation of a color identifies how pure or intense the color is. A fully saturated color is deep and brilliant—as the saturation decreases, the color gets paler and more washed out until it eventually fades to neutral.

Brightness

The brightness of a color identifies how light or dark the color is. Any color whose brightness is zero is black, regardless of its hue or saturation. There are different schemes for specifying a color's brightness and depending on which one is used, the results of lightening a color can vary considerably.

Luminance

The luminance of a color is a measure of its perceived brightness. The computation of luminance takes into account the fact that the human eye is far more sensitive to certain colors (like yellow-green) than to others (like blue).

Chrominance

Chrominance is a complementary concept to luminance. If you think of how a television signal works, there are two components—a black and white image which represents the luminance and a color signal which contains the chrominance information. Chrominance is a 2-dimensional color space that represents hue and saturation, independent of brightness.

Viewing Images on a Monitor

Before you can get consistent results working with digital images, you must create a standard viewing environment.

Most computer monitors use a standard set of *phosphors* (the chemical compounds on the face of the screen that are responsible for creating its primary colors). These phosphors, combined with the electronic circuitry that activates them create color images on your computer screen. To get an accurate match between what you see on your monitor and prints or other output requires that a number of variables be controlled:

Viewing Conditions

First, it is very important for critical evaluation of photographic images on a computer monitor to work in an environment with subdued lighting. When strong room light falls on the face of the monitor, it makes it almost impossible to discern shadow detail and seriously alters the appearance of many of the colors.

Monitor Brightness Control

Next, you must set the brightness control on your monitor correctly. If the level is too high, blacks will start to become grays and colors will be washed out. If the level is too low, there will be loss of shadow detail and colors will appear murky.

Gamma

Since the human eye has a nonlinear response curve (it is more sensitive to variations in low light than to an equal variation in bright light), it is important to track this nonlinearity with the display of different gray levels. The curve that relates the amount of light emitted from the monitor to the 256 different gray levels in the digital image is called the *gamma* curve. There are several different standard gamma curves in common use. They **all follow a power law of the following form:b = v**

where b is the brightness of the light emitted from the screen, v is the gray level value in the image and is the monitor gamma. The standard value for that most closely approximates the response of the human eye is 2.22. This is the value used by the television industry for storing and displaying video images. The prepress industry has standardized on a value of 1.8 which more closely corresponds to the characteristics of printing presses. Macintosh computers often use the value 1.4 for some unknown reason, while most uncorrected PC monitors have a gamma of about 2.3.

If an image was designed to be displayed on a monitor with a gamma of 2.2 and the monitor's actual gamma is set to 1.8, the image will appear too light and washed out. Conversely, if an image was adjusted to look good using a computer with a monitor gamma of 1.4 or 1.8 and viewed at 2.2, it will look too dark. For this reason, it is very important to set your computer's monitor to the same gamma setting as will be used to print it. If gamma is ignored, there is no way to guarantee that prints will not be too light or too dark.

White Point

Finally, there is an important characteristic of computer monitors called *white point*. This is usually given as a color temperature and for most uncorrected monitors, it run about 9300 degrees. What this means is that the white that is displayed on the monitor when you send it

maximum values of red, green, and blue has a distinctly bluish cast. If you were going to view your prints under a light of this same color temperature, you would get a good match (all other variables being set properly). Different lights like incandescent, quartz halogen, fluorescent, and daylight (morning, evening, indirect, etc.) all have different color temperatures and prints viewed using these different lighting conditions will look radically different. The standard white point used in the graphic arts industry is a 5000 degree color temperature and most light boxes are set to roughly this color. Viewing prints under fluorescent lighting can be particularly deceptive as mercury vapor lamps has several spikes in the green part of the spectrum whereas daylight has a relatively smooth spectrum. This can give rise to considerable variation in certain colors.

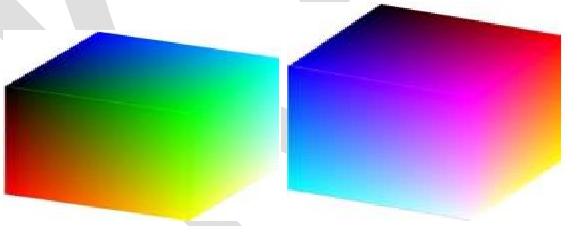
Two monitors which use the same phosphors, are both viewed in subdued light, have their gamma set the same, have their brightness control set properly, and have the same white point, should produce nearly identical displays.

Color Spaces

A color space is a mathematical system for representing colors. Since it takes at least three independent measurements to determine a color, most color spaces are three-dimensional. Many different color spaces have been created over the years in an effort to categorize the full gamut of possible colors according to different characteristics. Picture Window uses three different color spaces:

RGB

Most computer monitors work by specifying colors according to their red, green, and blue components. These three values define a 3-dimensional color space call the RGB color space. The RGB color space can be visualized as a cube with red varying along one axis, green varying along the second, and blue varying along the third. Every color that can be created by mixing red, green, and blue light is located somewhere within the cube. The following images show the outside of the RGB

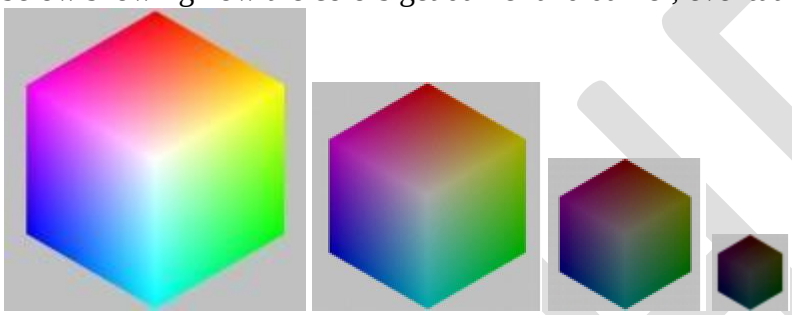


Cube viewed from two different directions:

The eight corners of the cube correspond to the three primary colors (Red, Green, Blue), the three secondary colors (Cyan, Magenta, Yellow) and black and white. All the different neutral grays are located on the diagonal of the cube that connects the black and the white vertices.

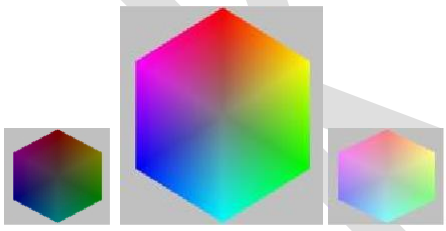
HSV (Hue Saturation Value)

The HSV color space attempts to characterize colors according to their hue, saturation, and value (brightness). This color space is based on a so-called hexcone model which can be visualized as a prism with a hexagon on one end that tapers down to a single point at the other. The hexagonal face of the prism is derived by looking at the RGB cube centered on its white corner. The cube, when viewed from this angle,
looks like a hexagon with white in the center and the primary and secondary colors making up the six vertices of the hexagon. This color hexagon is the one Picture Window uses in its color picker to display the brightest possible versions of all possible colors based on their hue and saturation. Successive crossections of the HSV hexcone as it narrows to its vertex are illustrated below showing how the colors get darker and darker, eventually reaching black.



HSL (Hue Saturation Lightness)

The HSL color space (also sometimes called HSB) attempts to characterize colors according to their hue, saturation, and lightness (brightness). This color space is based on a double hexcone model which consists of a hexagon in the middle that converges down to a point at each end. Like the HSV color space, the HSL spacegoes to black at one end, but unlike HSV, it tends toward white at the opposite end. The most saturated colors appear in the middle. Note that unlike in the HSL color space, this central crossection has 50% gray in the center and not white.

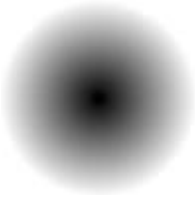


L = 25% L = 50% L = 75%

Continuous Tone vs. Halftone Images

There are many different technologies used to reproduce photographic images. Some printing technologies such as laser and ink jet printers work by creating individual dots, each of which is one of a small number of solid colors. For example, a black and white laser printer can only print black or white pixels -- it cannot produce gray dots.

To create the illusion of a photographic image, these tiny dots must be clustered in different proportions to reproduce the different colors and gray levels in the image. This dot clustering process is called *halftoning*or *dithering* and it can be done in many different ways.



continuous tone image | ordered dither halftone | error diffusion halftone

Devices such as computer monitors, dye sublimation printers, and film recorders can create dots of any color without halftoning, and these are called *continuous tone* printers because they can reproduce a continuous range of gray levels or colors.

Because a continuous tone printer packs more information into each dot, to achieve comparable image quality, printers that use halftoning or dithering require higher resolution, by a factor of roughly 6-8. Thus a continuous tone printer that outputs images at 200 dpi is roughly similar in quality to a halftone printer that outputs pixels

at roughly 1200-1600 dpi. For this reason, even a 600 dpi laser printer makes relatively coarse images equivalent to only about 75 dpi continuous tone output. For printed output to appear photographic when viewed from a standard distance of about 10 inches requires between 150 and 300 dpi. At 150 dpi, most images will appear noticeable soft, while between 200 and 300 dpi the incremental improvement is quite small. Beyond 300 dpi there is still some improvement, but for most images and most observers, it will be virtually unnoticeable.

How images are digitized

The process of converting an image to pixels is called *digitizing* or *scanning* and this function can be performed in many different ways.

Film Scanners

This type of scanner is sometimes called a slide or transparency scanner. They are specifically designed for scanning film, usually 35mm slides or negatives, but some of the more expensive ones can also scan medium and large format film. These scanners work by passing a narrowly focused beam of light through the film and reading the intensity and color of the light that emerges.

Flatbed Scanners

This type of scanner is sometimes called a reflective scanner. They are designed for scanning prints or other flat, opaque materials. These scanners work by shining white light onto the object and reading the intensity and color of the light that is reflected from it, usually a line at a time. Some flatbed scanners have available transparency adapters, but for a number of reasons, in most cases these are not very well suited to scanning film. On the other hand, flatbed scanners can be used as a sort of lensless camera to directly digitize flat objects like leaves.

Digital Cameras

One of the most direct ways to capture an image is a digital camera which uses a special semiconductor chip called a CCD (charge coupled device) to convert light to electrical signals right at the image plane. The quality of the images created in this manner is closely related to the number of pixels the CCD can capture. Affordable digital cameras suffer from relatively low resolution, limited dynamic range, and low ISO film speed equivalent, and consequently do not always produce high quality digital images. To get images with quality comparable to film photography currently requires very expensive digital cameras.

Video Frame Grabbers

This type of scanner uses a video camera to capture a scene or object and then converts the video signal that comes out of the camera to a digital image in your computer memory. A video camera can be used to digitize scenes containing 3- dimensional objects, but they usually have much lower image quality than film or flatbed scanners.

Scanning Services

Photo CD is a service started by Kodak a number of years ago whereby your film can be scanned using a high quality scanner and written to a compact disk you computer can access. Using Photo CD service is an inexpensive way to get high quality scans of your images without purchasing a scanner.

Many other scanning services are available which can scan prints or film to floppy disks or removable disk cartridges. These vary from low resolution snapshot quality images to professional drum scans at very high resolution.

Digitizing or **digitization**

The representation of an object, image, sound, document or a signal (usually an analog signal) by a discrete set of its points or samples. The result is called *digital representation* or, more specifically, a *digital image*, for the object, and *digital form*, for the signal. Strictly speaking, digitizing means simply capturing an analog signal in digital form. For a document the term means to trace the document image or capture the "corners" where the lines end or change direction

Process

The term digitization is often used when diverse forms of information, such as text, sound, image or voice, are converted into a single binary code. Digital information exists as one of two digits, either 0 or 1. These are known as bits (a contraction of *binary digits*) and the sequences of 0s and 1s that constitute information are called bytes. Analog signals are continuously variable, both in the number of possible values of the signal *at* a given time, as well as in the number of points in the signal *in* a given period of time. However, digital signals are discrete in both of those respects – generally a finite sequence of integers – therefore a digitization can, in practical terms, only ever be an approximation of the signal it represents.

Digitization occurs in two parts:

Discretization

The reading of an analog signal *A*, and, at regular time intervals (frequency), sampling the value of the signal at the point. Each such reading is called a *sample* and may be considered to have infinite precision at this stage;

Quantization

Samples are rounded to a fixed set of numbers (such as integers), a process known as quantization.

In general, these can occur at the same time, though they are conceptually distinct:

A series of digital integers can be transformed into an analog output that approximates the original analog signal. Such a transformation is called a DA conversion. The sampling rate and the number of bits used to represent the integers combine to determine how close such an approximation to the analog signal a digitization will be.

Analog signals to digital

Analog signals are continuous electrical signals. Digital signals are non-continuous. Nearly all recorded music has been digitized. About 12 percent of the 500,000+ movies listed on the Internet Movie Database are digitized on DVD. Digitization of personal multimedia such as home movies, slides, and photographs is a popular method of preserving and sharing older repositories. Slides and photographs may be scanned using an image scanner, but videos are more difficult. Many companies offer personal video digitization services

Image Acquisition:

- An image is captured by a sensor (such as a monochrome or color TV camera) and digitized.
- If the output of the camera or sensor is not already in digital form, an analog-to digital converter digitizes it.

Camera

Camera consists of 2 parts

- A lens that collects the appropriate type of radiation emitted from the object of interest and that forms an image of the real object
- a semiconductor device – so called charged coupled device or CCD which converts the irradiance at the image plan into an electrical signal.

Frame Grabber

Frame grabber only needs circuits to digitize the electrical signal from the imaging sensor to store the image in the memory (RAM) of the computer.

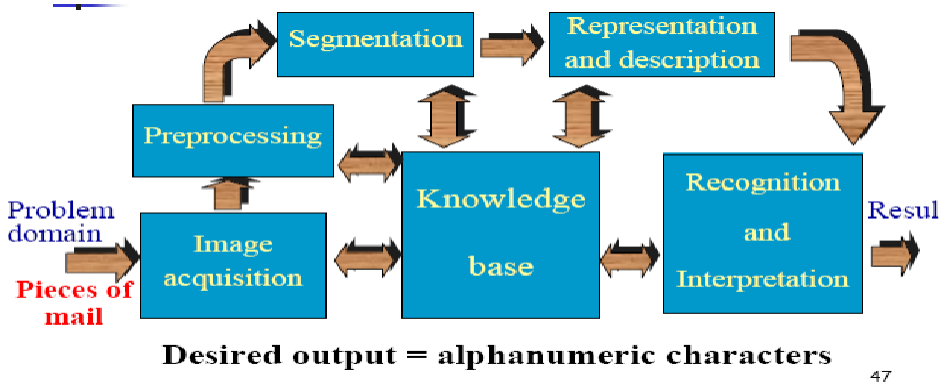
Representation & Description

Representation

make a decision whether the data should be represented as a boundary or as a complete region.

Boundary representation focus on external shape characteristics, such as corners and inflections.

Region representation focus on internal properties, such as texture or skeleton shape.



Recognition & Interpretation

Recognition

the process that assigns a label to an object based on the information provided by its descriptors.

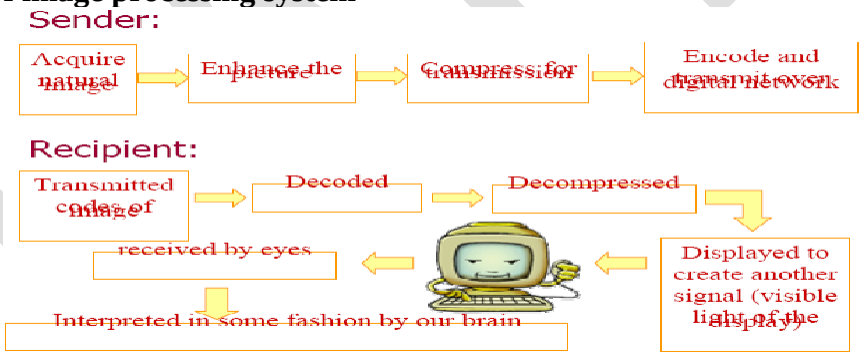
Interpretation

assigning meaning to an ensemble of recognized objects.

Knowledge base

- a problem domain-> detailing regions of an image where the information of interest is known to be located.
- Help to limit the search

Components of image processing system



From acquisition to interpretation, the initial signal may be transformed, modified, and retransmitted numerous times.

Applications

- Improvement of pictorial information for human interpretation
- Processing of image data for storage, transmission, and representation for autonomous machine perception
- Radiation from the Electromagnetic spectrum
  - Acoustic
  - Ultrasonic
  - Electronic (in the form of electron beams

- used in electron microscopy)
- Computer (synthetic images used for modeling and visualization)
- Remote sensing : Nighttime Lights of the World, Weather observation and prediction
- Astronomy
- Light microscopy
    - pharmaceuticals
        (a). taxol (anticancer agent)
        (b). cholesterol
    - Microinspection to materials characterization
        (c). Microprocessor
        (d). Nickel oxide thin film
        (e). Surface of audio CD
    - Industry : visual spectrum
    - (automated visual inspection of manufactured goods)

Image representation and image analysis tasks

Image understanding by a machine can be seen as an attempt to find a relation between input image(s) and previously established models of the observed world. Transition from the input image(s) to the model reduces the information contained in the image to relevant information for the application domain. This process is usually divided into several steps and several levels representing the image are used. The bottom layer contains raw image data and the higher levels interpret the data. Computer vision designs these intermediate representations and algorithms serving to establish and maintain relations between entities within and between layers.

Image representation can be roughly divided according to data organization into four Levels. The boundaries between individual levels arc inexact, and more detailed divisions are also proposed in the literature. This hierarchy of image representation and related algorithms is frequently categorized in an even simpler way. Two levels are often distinguished: Low-level image processing and high-level image understanding.

Low-level processing methods usually use very little knowledge about the content of images. In the case of the computer knowing image content, it is usually provided by high- level algorithms or directly by a human who understands the problem domain. Low-level methods often include image compression, pre-processing methods for noise filtering, edge extraction, and image sharpening, all of which we shall discuss in this book. Low-level image processing uses data which resemble the input image; for example, an input image captured by a TV camera is 2D in nature, being described by an image function /(x, y) whose value, at simplest, is usually brightness depending on two parameters x,y, the co-ordinates of the location in the image.

If the image is to be processed using a computer it will be digitized first, after which it may be represented by a rectangular matrix with elements corresponding to the brightness at appropriate image locations. Nowadays, this is usually an over-simplification since the image will be presented in color, implying (usually) three channels: red, green and blue. Very often, such a data set will be part of a video stream with an associated frame rate. Nevertheless, the raw

material will be a set or sequence of matrices which represent the inputs and outputs of low-level image processing.

High-level processing is based on knowledge, goals, and plans of how to achieve those goals, and artificial intelligence methods are widely applicable. High-level computer vision tries to imitate human cognition (although be mindful of the health warning given in the very first paragraph of this chapter) and the ability to make decisions according to the information contained in the image. In the example described, high-level knowledge would be related to the 'shape' of a cow and the subtle interrelationships between the different parts of that shape, and their (inter-)dynamics.

High-level vision begins with some form of formal model of the world, and then the 'reality' perceived in the form of digitized images is compared to the model. A match is attempted, and when differences emerge, partial matches (or sub-goals) are sought that overcome the mismatches; the computer switches to low-level image processing to find information needed to update the model. This process is then repeated iteratively, and 'understanding' an image thereby becomes a co-operation between top-down and bottom-up processes

Internal image representations are not directly understandable—while the computer is able to process local parts of the image, it is difficult for it to locate global knowledge. General knowledge, domain-specific knowledge, and information extracted from the image will be essential in attempting to 'understand' these arrays of numbers.

Low-level computer vision techniques overlap almost completely with digital image processing, which has been practiced for decades. The following sequence of processing- steps is commonly seen: An image is captured by a sensor (such as a TV camera) and digitized; then the computer suppresses noise (image pre-processing) and maybe enhances some object features which are relevant to understanding the image. **Edge extraction is an example of processing carried out at this stage.**

Image segmentation is the next step, in which the computer tries to separate objects from the image background and from each other. Total and partial segmentation may be distinguished; total segmentation is possible only for very simple tasks, an example being the recognition of dark non-touching objects from a light background. For example, in analyzing images of printed text (an early step in optical character recognition, OCR) even this superficially simple problem is very hard to solve without error. In more complicated problems (the general case), low-level image processing techniques handle the partial segmentation tasks, in which only the cues which will aid further high-level processing are extracted. Often, finding parts of object boundaries is an example of low-level partial segmentation.

Object description and classification in a totally segmented image are also understood as part of low-level image processing. Other low-level operations are image compression, and techniques to extract information from (but not understand) moving scenes. Low-level image processing and high-level computer vision differ in the data used. Low-level data are comprised of original images represented by matrices composed of brightness (or similar) values, while high-level data originate in images as well, but only those data which are relevant to high-level goals are extracted, reducing the data quantity considerably. High-level data represent

knowledge about the image content—for example, object size, shape, and mutual relations between objects in the image. High-level data are usually expressed in symbolic form.

High-level vision tries to extract and order image processing steps using all available knowledge—image understanding is the heart of the method, in which feedback from high-level to low-level is used.

Image representations:

Mathematical models are often used to describe images and other signals. A signal is a function depending on some variable with physical meaning; it can be one-dimensional (e.g., dependent on time), two-dimensional (e.g., an image dependent on two co-ordinates in a plane), three-dimensional (e.g., describing a volumetric object in space), or higher- dimensional. A scalar function might be sufficient to describe a monochromatic image, while vector functions are used in image processing to represent, for example, color images consisting of three component colors.

Functions we shall work with may be categorized as continuous, discrete, or digital. A continuous function has continuous domain and range; if the domain set is discrete, then we have a discrete function; if the range set is also discrete, then we have a digital function. Many of these functions will be linear, and correspondingly simple to deal with.

We shall take the usual intuitive definition of image—an example might be the image on the human retina, or captured by a TV camera. This can be modeled by a continuous (image) function of two variables f(x,y) where (x.y) are co-ordinates in a plane, or perhaps three variables f(x,y,t), where t is time. This model is reasonable in the great majority of applications that we encounter in day-to-day life, and whichare presented in this book. Nevertheless, it is worth realizing that an 'image' may be acquired in many ways. We shall note often that color is the norm, even when we present algorithms from the point of view of monochromatic images, but we do not need either to constrain ourselves to the visible spectrum. Cameras that operate in the infra-red part of the spectrum are now very common (for example, for night-time surveillance).

Other parts of the electro-magnetic [EM] spectrum may also be used; terahertz imaging, for example, is becoming widely available. Further, image acquisition outside the EM spectrum (that is, 'light') is also common: in the medical domain, datasets are generated via magnetic resonance (MR), computed tomography (CT), ultrasound etc. All of these approaches generate large arrays of data requiring analysis and understanding and with increasing frequency these arrays are of 3 or more dimensions.

The continuous image function

The (gray-scale) image function values correspond to brightness at image points. The function value can express other physical quantities as well (temperature, pressure distribution, distance from the observer, etc.). Brightness integrates different optical quantities—using brightness as a basic quantity allows us to avoid the complicated process of image formation .

The image on the human eye retina or on a TV camera sensor is intrinsically two-dimensional BD). We shall call such a 2D image bearing information about brightness points an intensity image. The 2D image on the imaging sensor is commonly the result of projection of a three-dimensional CD) scene

A non-linear perspective projection is often approximated by a linear parallel (or orthographic) projection, where / —>oo. Implicitly, z —>oc says that the orthographic projection is a limiting case of the perspective projection for faraway objects.

When 3D objects are mapped into the camera plane by perspective projection, a lot of information disappears because such a transform is not one-to-one. Recognizing or reconstructing objects in a 3D scene from one image is an ill-posed problem.

Recovering information lost by perspective projection is only one, mainly geometric, problem of computer vision—a second problem is understanding image brightness. The only information available in an intensity image is the brightness of the appropriate pixel (picture element, image element), which is dependent on a number of independent factors such as object surface reflectance properties (given by the surface material, microstructure, and marking), illumination properties, and object surface orientation with respect to a viewer and light source. It is a non-trivial and again ill-posed problem to separate these components when trying to recover the 3D geometry of an object from the intensity image. A monochromatic static image is represented by a continuous image function f(x,y) whose arguments are two co-ordinates in the plane.

Computerized image processing uses digital image functions which are usually represented by matrices, so co-ordinates are natural numbers. The domain of the image
function is a region R in the plane R={(x,y), l<x<xm,l<y<yn}, B.2)
wherexm, yn represent the maximal image co-ordinates. The image function has a limited domain—infinite summation or integration limits can be used, as it is assumed that the image function value is zero outside the domain R. The customary orientation of
co-ordinates in an image is in the normal Cartesian fashion (horizontal a:-axis, vertically-axis, origin bottom-left), although the (row. column, origin top-left) orientation used in matrices is also often used in digital image processing.

The range of image function values is also limited; by convention, in monochromatic images the lowest value corresponds to black and the highest to white. Brightness values bounded by these limits are gray-levels. The quality of a digital image grows in proportion to the spatial, spectral, radiometric, and time resolutions. The spatial resolution is given by the proximity of image samples in the image plane; spectral resolution is given by the bandwidth of the light frequencies captured by the sensor; radiometric resolution corresponds to the number of distinguishable gray-levels; and time resolution is given by the interval between time samples at which images are captured. The question of time resolution is important in dynamic image analysis, where time sequences of images are processed.

Images f(x. y) can be treated as deterministic functions or as realizations of stochastic processes. Mathematical tools used in image description have roots in linear system theory, integral transforms, discrete mathematics, and the theory of stochastic processes. Mathematical transforms usually assume that the image function f{x,y) is 'well-behaved', meaning that the function is integrable, has an invertible Fourier transform, etc.

Image digitization

An image to be processed by computer must be represented using an appropriate discrete data structure, for example, a matrix. An image captured by a sensor is expressed as a continuous function f(x,y) of two co-ordinates in the plane. Image digitization means that the function f(x,y) is sampled into a matrix with M rows and N columns. Image quantization assigns to each continuous sample an integer value—the continuous range of the image function f(x,y) is split into K intervals. The finer the sampling (i.e., the larger M and N) and quantization (the larger K), the better the approximation of the continuous image function f(x.y) achieved. Image function sampling poses two questions. First, the sampling period should be determined—this is the distance between two neighboring sampling points in the image. Second, the geometric arrangement of sampling points (sampling grid) should be set.

1 .Sampling

A continuous image is digitized at sampling points. These sampling points are ordered in the plane, and their geometric relation is called the grid. The digital image is then a data structure, usually a matrix. Grids used in practice are usually square or hexagonal It is important to distinguish the grid from the raster; the raster is the grid on which a neighborhood relation between points is defined1. One infinitely small sampling point in the grid corresponds to one picture element also called a pixel or image element in the digital image; in a three-dimensional image, an image element is called a voxel (volume element). The set of pixels together covers the entire image; however, the pixel captured by a real digitization device has finite.

2. Quantization

A value of the sampled image fs(jAx.kAy) is expressed as a digital value in image processing. The transition between continuous values of the image function (brightness)
and its digital equivalent is called quantization. The number of quantization levels should be high enough to permit human perception of fine shading details in the image.

Most digital image processing devices use quantization into k equal intervals. If b bits are used to express the values of the pixel brightness then the number of brightness levels is k = 2b'. Eight bits per pixel per channel (one each for red, green, blue) are commonly used although systems using other numbers D, 6, 12, ...) can be found.

The main problem in images quantized with insufficient brightness levels is the occurrence of false contours. This effect arises when the number of brightness levels is lower than that which humans can easily distinguish. This number is dependent on many factors—for example, the average local brightness—but displays which avoid this effect will normally provide a range of at least 100 intensity levels. This problem can be reduced when quantization into intervals of unequal length is used; the size of intervals corresponding to less probable brightness in the image is enlarged.

An efficient computer representation of brightness values in digital images requires that eight bits, four bits, or one bit are used per pixel, meaning that one, two, or eight pixel brightness can be stored in one byte.

Digital image properties

A digital image has several properties, both metric and topological, which are somewhat different from those of continuous two-dimensional functions with which we are familiar.

Another feature of difference is human perception of images, since judgment of image quality is also important.

1. Metric and topological properties of digital images

A digital image consists of picture elements with finite size—these pixels carry information about the brightness of a particular location in the image. Usually pixels are arranged into a rectangular sampling grid. Such a digital image is represented by a two-dimensional matrix whose elements are natural numbers corresponding to the quantization levels in the brightness scale. Some intuitively clear properties of continuous images have no straightforward analogy in the domain of digital images. Distance is an important example..

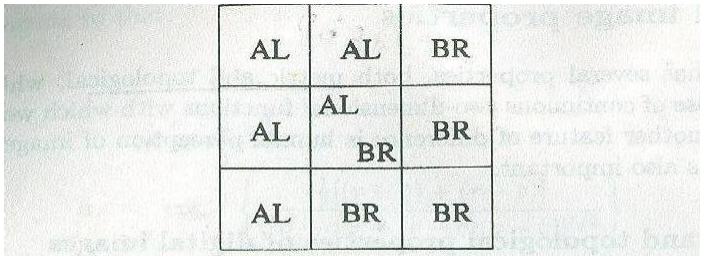
$$D_E[(i,j),(h,k)] = \sqrt{(i-h)^2 + (j-k)^2}$$

If there is a path between two pixels in the set of pixels in the image, these pixels are called contiguous. Alternatively, we can say that a region is a set of pixels in which each pair of pixels is contiguous. The relation 'to be contiguous' is reflexive, symmetric, and transitive and therefore defines a decomposition of the set (the image in our case) into equivalence classes (regions).

$$D_4[(i,j),(h,k)] = |i-h| + |j-k|$$

$$D_8[(i,j),(h,k)] = \max\{|i-h|, |j-k|\}$$

Assume that R:L are disjoint regions in the image which were created by the relation 'to be contiguous', and further assume (to avoid special cases) that these regions do not touch the image bounds. Let R be the union of all regions R,i\ Rc be the set complement of R. with respect to the image. The subset of Rc which is contiguous with the image bounds is called the background, and the remainder of the complement Rc is called holes.3 A region is called simple contiguous if it has no holes. Equivalently, the complement of a simply contiguous region is contiguous. A region with holes is called multiple contiguous.



: *Pixel neighborhoods used in chamfering—pixel p is the*

The brightness of a pixel is a very simple property which can be used to find objects in some images; if, for example, a pixel is darker than some predefined value (threshold), then it belongs to the object. All such points which are also contiguous constitute one object.

These neighborhood and contiguity definitions on the square grid create interesting paradoxes. Figure 2.7a shows two digital line segments with 45° slope. If 4-connectivity is used, the lines are not contiguous at each of their points. An even worse conflict with intuitive understanding of line properties is also illustrated; two perpendicular lines do intersect in one case (upper right intersection) and do not intersect in another case (lower left), as they do not have any common point (i.e., their set intersection is empty). It is known from Euclidean geometry that each closed curve (e.g., a circle) divides the plane into two non-contiguous regions. If images are digitized in a square grid using 8-connectivity, we can draw a line from the inner part of a closed curve into the outer part which does not intersect the curve. This implies that the inner and outer parts of the curve constitute only one region because all pixels of the line belong to only one region. This is another paradox. One possible solution to contiguity paradoxes is to treat objects using 4-neighborhoods and background using 8-neighborhoods (or vice versa).

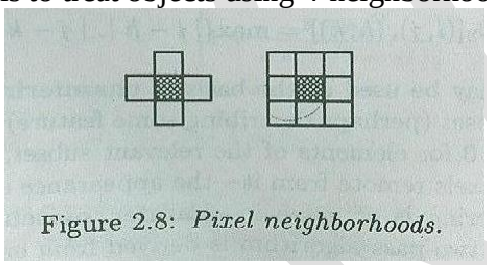


Figure 2.8: *Pixel neighborhoods.*

The distance transform is also called the distance function or chamfering algorithm or simply chamfering to build an analogy to a woodcarving operation. The idea of the distance transform is important as it provides the basis of several fast algorithms as will be shown more times in this book. The distance transform provides the distance of pixels from some image subset (perhaps describing objects or some features). The resulting 'image' has pixel values of 0 for elements of the relevant subset, low values for close pixels, and then high values for pixels remote from it—the appearance of this array gives the name to the technique. In other words, the distance transform of a binary image provides the distance from each pixel to the nearest non-zero pixel. Hypothetical pixels outside the image limits are also treated as non-zero pixels.

The direction of the crack edge is that of increasing brightness, and is a multiple of 90°, while its magnitude is the absolute difference between the brightness of the relevant pair of pixels.

The border (boundary) of a region is another important concept in image analysis. The border of a region R is the set of pixels within the region that have one or more neighbors outside R. The definition corresponds to an intuitive understanding of the border as a set of points at the bound of the region. This definition of border is sometimes referred to as the inner border to distinguish it from the outer border, that is. the border of the background (i.e., its complement) of the region. Due to the discrete nature of the image, some inner border elements which would be distinct in the continuous case coincide in the discrete case, as can be seen with the one-pixel-wide line at the right of Figure 2.13.

The border is a global concept related to a region, while edge expresses local properties of an image function. Borders and edges are related as well. One possibility for finding boundaries is chaining the significant edges (points with high gradient of the image function). If any two points within a region are connected by a straight line segment, and the whole line lies within the region, then this region is convex—see Figure 2.14. The property of convexity decomposes all regions into two equivalence classes: convex and non-convex. A convex hull is a concept used to describe qualitative properties of objects. The convex hull is the smallest convex region containing the input region, possibly non-convex.

For example, consider an object whose shape resembles the letter TT (see Figure 2.15). Imagine a thin rubber band pulled around the object; the shape of the rubber band provides the convex hull of the object.

Topological properties are not based on the distance concept. These properties are invariant to the homeomorphic transform which can be illustrated for images as rubber sheet transform. Imagine a small rubber balloon with an object painted on it; topological properties of the object are those which are invariant to arbitrary stretching of the rubber sheet. Stretching does not change contiguity of the object parts and does not change the number of holes in regions. One such image property is the Euler—Poincare characteristic, defined as the difference between the number of regions and the number of holes in them. We use the term 'topological properties' of the region to describe its qualitative properties invariant to small changes, e.g., the property of being convex. Strictly speaking, an arbitrary homeomorphic transformation can change a convex region to a non-convex one and vice versa. Following the rubber sheet transformation analogy, it means that the stretching of the sheet is only gentle.

An object with non-regular shape can be represented by a collection of its topological components, Figure. The set inside the convex hull which does not belong to an object is called the deficit of convexity. This can be split into two subsets: lakes (dark gray) are fully surrounded by the object; and bays (light gray) are contiguous with the border of the convex hull of the object.

The convex hull, lakes, and bays are sometimes used for object description; these features are used in Chapter 8 (object description) and in Chapter 13 (mathematical morphology).

2 Histograms

The brightness histogram hj(z) of an image provides the frequency of the brightness value z in the image—the histogram of an image with L gray-levels is represented by a one-dimensional array with L elements.

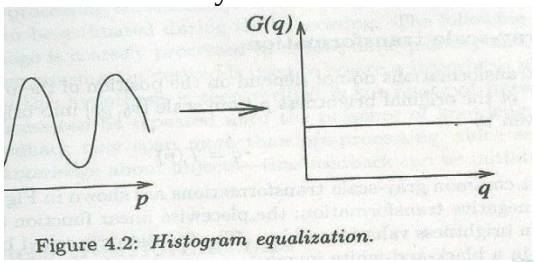


Figure 4.2: *Histogram equalization.*

$$\sum_{i=0}^{k} G(q_i) = \sum_{i=0}^{k} H(p_i) \qquad f = \frac{N^2}{q_k - q_0}$$

the equalized histogram for the continous probability density

$$N^2 \int_{q_0}^{q} \frac{1}{q_k - q_0}\, ds = \frac{N^2(q - q_0)}{q_k - q_0} = \int_{p_0}^{p} H(s)\, ds$$

Brightness transformation

$$q = T(p) = \frac{q_k - q_0}{N^2} \int_{p_0}^{p} H(s)\, ds + q_0$$

$$q = T(p) = \frac{q_k - q_0}{N^2} \sum_{i=p_0}^{p} H(i) + q_0$$

Algorithm 2.2: Computing the brightness histogram

1. Assign zero values to all elements of the array hf.
2. For all pixels (x,y) of the image /, increment hf(f(x,y)) by 1.

$$H[g_p] = H[g_p] + 1$$

3. Form the cumulative image histogram H_c:

$$H_c[0] = H[0]$$
$$H_c[p] = H_c[p-1] + H[p] \qquad p = 1, 2, \ldots, G-1$$

4. Set

$$T[p] = \text{round}\left(\frac{G-1}{NM} H_c[p]\right)$$

(This step obviously lends itself to more efficient implementation by look-up table of the multiples of NM, and making comparisons with th which are monotonic increasing).

5. Rescan the image and write an output image with gray-levels g_q, settin

$$g_q = T[g_p]$$

The histogram provides a natural bridge between images and a probabilistic description. Wc might want to find a first-order probability function p\(z;x,y) to indicate the probability that pixel (x,y) has brightness z. Dependence on the position of the pixel is not of interest in the histogram; a density function pi (z) is of interest and the brightness histogram is its estimate. The histogram is often displayed as a bar graph. An image and its brightness histogram are given in Figure 2.16.

The histogram is usually the only global information about the image which is available. It is used when finding optimal illumination conditions for capturing an image, gray-scale transformations, and image segmentation to objects and background. Note that one histogram may correspond to several images; for instance, a change of the object position on a constant background does not affect the histogram.

The histogram of a digital image typically has many local minima and maxima, which may complicate its further processing. This problem can be avoided by local smoothing of the histogram; this may be done, for example, using local averaging of neighboring histogram elements as the base, so that a new histogram h!f(z) is calculated according to where K is a constant representing the size of the neighborhood used for smoothing. This algorithm would need some boundary adjustment, and carries no guarantee of removing all local minima. Other techniques for smoothing exist, notably Gaussian blurring; in the case of a histogram, this would be a one-dimensional simplification of the 2D Gaussian blur.

3. Entropy

If a probability density p is known then image information content can be estimated regardless of its interpretation using entropy H. The concept of entropy has roots in thermodynamics and statistical mechanics but it took many years before entropy was related to information. The information-theoretic formulation of entropy comes from Shannon [Shannon, 1948] and is often called information entropy.

An intuitive understanding of information entropy relates to the amount of uncertainty about an event associated with a given probability distribution. The entropy can serve as an measure of 'disorder'. As the level of disorder rises, entropy increases and events are less predictable.

4. Visual perception of the image

Anyone who creates or uses algorithms or devices for digital image processing should take into account the principles of human image perception. If an image is to be analyzed by a human the information should be expressed using variables which are easy to perceive; these are psycho-physical parameters such as contrast, border, shape, texture, color, etc. Humans will find objects in images only if they may be distinguished effortlessly from the background.

A. Contrast

Contrast is the local change in brightness and is defined as the ratio between average brightness of an object and the background. Strictly speaking, we should talk about luminance4 instead of brightness if our aim is to be physically precise. The human eye is logarithmically sensitive to brightness, implying that for the same perception, higher brightness requires higher contrast.

Apparent brightness depends very much on the brightness of the local surroundings; this effect is called conditional contrast. Figure 2.17 illustrates this with five circles of the same size surrounded by squares of different brightness. Humans perceive the brightness of the small circles as different.

Figure 2.17: Conditional contrast effect. Circles inside squares have the same brightness and are perceived as having different brightness values.

B. Acuity

Acuity is the ability to detect details in an image. The human eye is less sensitive to slow and fast changes in brightness in the image plane but is more sensitive to intermediate changes. Acuity also decreases with increasing distance from the optical axis. Resolution in an image is firmly bounded by the resolution ability of the human eye; there is no sense in representing visual information with higher resolution than that of the viewer..

Some visual illusions

Human perception of images is prone to many illusions. There are many other visual illusions caused by phenomena such as color or motion; an Internet search will produce examples easily. Object borders carry a lot of information for humans. The Ebbinghaus illusion is a well-known example—two circles of the same diameter in the center of images appear to have different diameters . Perception of one dominant shape can be fooled by nearby shapes. Figure 2.19 shows parallel diagonal line segments which are not perceived as parallel. Figure 2.20 contains rows of black and white squares which are all parallel. However, the vertical zigzag squares disrupt our horizontal perception.

Perceptual grouping

A perceptual grouping [Palmer, 1999] is a principle used in computer vision to aggregate elements provided by low-level .Patterns take precedence over elements and have properties that

are not inherent in the elements themselves. The human ability to group items according to various properties is illustrated in direction properties of elements.

5. Image quality

An image might be degraded during capture, transmission, or processing, and measures of image quality can be used to assess the degree of degradation. The quality required naturally depends on the purpose for which an image is used. Methods for assessing image quality can be divided into two categories: subjective and objective. Subjective methods are often used in television technology, where the ultimate criterion is the perception of a selected group of professional and lay viewers.

Another class measures the resolution of small or proximate objects in the image. An image consisting of parallel black and white stripes is used for this purpose; then the number of black and white pairs per millimeter gives the resolution. Measures of image similarity are also important since they are used in assisting retrieval from image databases.

6. Noise in images

Real images are often degraded by some random errors—this degradation is usually called noise. Noise can occur during image capture, transmission, or processing, and may be dependent on, or independent of, the image content. Noise is usually described by its probabilistic characteristics. Idealized noise, called white noise is often used. White noise has a constant power spectrum, meaning that all noise frequencies are present and have the same intensity.

For example, the intensity of white noise does not decrease with increasing frequency as is typical in real-world signals. White noise is frequently employed to model the worst approximation of degradation, the advantage being that the use of white noise simplifies calculations.

A special case of white noise is a Gaussian noise. A random variable with a Gaussian (normal) distribution has its probability density function given by the Gaussian curve. In the ID case the density function is 1 -(x-mJ p(x) = —= est- , B.9) <7V27T whereji is the mean and a the standard deviation of the random variable. Gaussian noise is a very good approximation to noise that occurs in many practical cases.

When an image is transmitted through some channel, noise which is usually independent of the image signal occurs. Similar noise arises in old-fashioned vidicon cameras. This signal-independent degradation is called additive noise and can be described by the model f(x,y) = g{x,y) + v(x,y), B.10) where the noise v and the input image g are independent variables. Algorithm 2.3 generate zero mean additive Gaussian noise in an image—this can often be of use in testing or demonstrating many other algorithms in this book which are designed to remove noise, or to be noise resistant.

Algorithm 6: Generation of additive, zero mean Gaussian noise

1. Suppose an image has gray-level range [0, G — 1]. Select a > 0; low values generate less noise effect.generate zero mean additive Gaussian noise in an image—this can often be of use in testing or demonstrating many other algorithms in this book which are designed to remove noise, or to be noise resistant. Testing.

2. For each pair of horizontally neighboring pixels (x, y), (x, y + 1) generate a pair of independent random numbers r, </> in the range [0,1].
3. Calculate z\ = acosBn6)V—2 lnr , V ^ B.11) Z2 = asinB7T(j))V~2 lnr .
(This is the Box-Muller transform which assumes that zi, Z2 are independently normally distributed with zero mean and variance a2.)
4. Set /'(#, y) = #(x, y) + zi and /'(x, y + 1) = g{x, y + 1) + Z25 where ^ is the input image.
5. Set f(x,y) f(x,y + l)=t
6. Go to 3 until all pixels have been scanned.
0 if/'(a;,y)<0, G-l itf'(x,y)>G-l, B.12) I /'(a;, y) otherwise,
0 if/'(a;,y+l)<0,
G-\ if/'(s,y+l)>G-l, B.13)
I /7 (^> y + 1) otherwise.

The truncation performed by equations B.12) and B.13) will attenuate the Gaussian nature of the noise; this will become more marked for values of a that are high relative to G. Other algorithms for noise generation may be found in [Pitas, 1993].
Equation B.10) leads to a definition of signal-to-noise ratio (SNR); computing the total square value of the noise contribution (s.2/) we compare this with the total square value of the observed signal (*.2/)

The signal-to-noise ratio is then SNR=^ B.14) Ej (strictly, we are comparing the mean observation with the mean error—the computation is obviously the same). SNR represents a measure of image quality, with high values being 'good'.
Signal-to-noise is often expressed in the logarithmic scale, in decibels SNRdB = 10 log10 SNR. B.15)The noise magnitude depends in many cases on the signal magnitude itself f = 9". B-16) .

This model describes multiplicative noise. An example of multiplicative noise is television raster degradation, which depends on TV lines; in the area of a line this noise is maximal, and between two lines it is minimal. Another example of multiplicative noise is the degradation of film material caused by the finite size of silver grains used in photosensitive emulsion.

Impulse noise means that an image is corrupted with individual noisy pixels whose brightness differs significantly from that of the neighborhood. The term salt-and-pepper noise is used to describe saturated impulsive noise—an image corrupted with white and/or black pixels is an example. Salt-and-pepper noise can corrupt binary images. The problem of suppressing noise in images is addressed in Chapter 5. If nothing is known a priori about noise properties, local pre-processing methods are appropriate. If the noise parameters are known in advance, image restoration techniques can be used.

Color images

Human color perception adds a subjective layer on top of underlying objective physical properties—the wavelength of electromagnetic radiation. Consequently, color may be considered a psychophysical phenomenon. Color has long been used in painting, photography and films to display the surrounding- world to humans in a similar way to that in which it is perceived in

reality. There is considerable literature on the variants in the naming of colors across languages, which is a very subtle affair [Kay, 2005]. The human visual system is not very precise in perceiving- color in absolute terms; if we wish to express our notion of color precisely we would describe it relative to some widely used color which is used as a standard: recall, e.g., the red of a British public telephone box. There are whole industries which present images to humans— the press, films, displays, and hence a desire for color constancy. In computer vision, we have the advantage of using a camera as a measuring device, which yields measurements in absolute quantities.

Physics of color

The electromagnetic spectrum is illustrated in Figure 2.23.

Only a narrow section of the electromagnetic spectrum is visible to a human, with wavelength A from approximately 380 nm to 740 rim. Visible colors with the wavelengths shown in Figure 2.24 are called spectral colors and are those which humans see when white light is decomposed using a Newtonian prism, or which are observed in a

Figure 2.23: Division of the whole electromagnetic spectrum (ELF means Extremely Low Frequencies). on the sky. Colors can be represented as combinations of the primary colors, e.g., red, green, and blue, which for the purposes of standardization have been defined as 700 nm, 546.1 nm, and 435.8 nm, respectively [Pratt, 1978], although this standardization does not imply that all colors can be synthesized as combinations of these three.

The intensity of irradiation for different wavelengths A usually changes. This variation is expressed by a power spectrum (called also power spectrum distribution) S(X). Why do we see the world in color? There are two predominant physical mechanisms describing what happens when a surface is irradiated. First, the surface reflection rebounds incoming energy in a similar way to a mirror. The spectrum of the reflected light remains the same as that of the illuminant and it is independent of the surface—recall that shiny metals 'do not have a color'. Second, the energy diffuses into the material and reflects randomly from the internal pigment in the matter. This mechanism is called body reflection and is predominant in dielectrics as plastic or paints. Figure 2.25 illustrates both surface reflection (mirroring along surface normal n) and body reflection. Colors are caused by the properties of pigment particles which absorb certain wavelengths from the incoming illuminant wavelength spectrum.

Most sensors used for color capture, e.g., in cameras, do not have direct access to color; the exception is a spectrophotometer which in principle resembles Newton's prism. Incoming irradiation is decomposed into spectral colors and intensity along the spectrum with changing wavelength A is measured in a narrow wavelength band, for instance, by a mechanically moved point sensor. Actual spectrophotometers use diffraction gratings instead of a glass prism.

Sometimes, intensities measured in several narrow bands of wavelengths are collected in a vector describing each pixel. Each spectral band is digitized independently and is represented by an individual digital image function as if it were a monochromatic image. In this way, multi spectral images are created. Multi spectral images are commonly used in remote sensing from satellites, airborne sensors and in industry. Wavelength usually span from ultraviolet through the visible section to infrared

Color perceived by humans

Evolution has developed a mechanism of indirect color sensing in humans and some animals. Three types of sensors receptive to the wavelength of incoming irradiation have been established in humans, thus the term trichromacy. Color sensitive receptors on the human retina are the cones. The other light sensitive receptors on the retina are the rods which are dedicated to sensing monochromatically in low ambient light conditions. Cones are categorized into three types based on the sensed wavelength range: S (short) with maximum sensitivity at w 430 nm, M (medium) at « 560 nm, and L (long) at « 610 nm. Cones S, M, L are occasionally called cones B, G and R, respectively, but that is slightly misleading. We do not see red solely because an L cone is activated. Light with equally distributed wavelength spectrum looks white to a human, and an unbalanced spectrum appears as some shade of color. The reaction of a photoreceptor or output from a sensor in a camera can be modeled mathematically.

A phenomenon called color metamer is relevant. A metamer, in general, means two things that are physically different but perceived as the same. Red and green adding to produce yellow is a color metamer, because yellow could have also been produced by a spectral color. The human visual system is fooled into perceiving that red and green is the same as yellow.

Human vision is prone to various illusions. Perceived color is influenced, besides the spectrum of the illuminant, by the colors and scene interpretation surrounding the observed color. In addition, eye adaptation to changing light conditions is not very fast and perception is influenced by adaptation. Nevertheless, we assume for simplicity that the spectrum of light coming to a point on the retina fully determines the color. Since color can be defined by almost any set of primaries, the world community agreed on primaries and color matching functions which are widely used. The color model was introduced as a mathematical abstraction allowing us to express colors as tuples

The XYZ color standard fulfills three requirements:

• Unlike the color matching experiment yielding negative lobes of color matching functions, the color matching functions of XYZ color space are required to be non- negative;
• The value of Y(X) should coincide with the brightness (luminance);
• Normalization is performed to assure that the power corresponding to the three color matching functions is equal (i.e., the area under all three curves is equal).

The resulting color matching functions are shown in Figure 2.28. The actual color is a mixture (more precisely a convex combination) of cx X + cY Y + cz Z ,B.18)
where 0 < ex, cy, cz< 1 are weights (intensities) in the mixture. The subspace of colors perceivable by humans is called the color gamut and is demonstrated in Figure 2.29.

Display and printing devices use three selected real primary colors (as opposed to three syntactic primary colors of XYZ color space). All possible mixtures of these primary colors fail to cover the whole interior of the horseshoe in CIE chromaticity diagram. This situation is demonstrated qualitatively for three particular devices. Several different primary colors and corresponding color spaces are used in practice, andthese spaces can be transformed into each other. If the absolute color space is used then the transformation is the one-to-one mapping and

does not lose information (except for rounding errors). Because color spaces have their own gamuts, information is lost if the transformed value appears out of the gamut. See [Burger and Burge, 2006] for a full explanation and for algorithms: here, we list several frequently used color spaces. The RGB color space has its origin in color television where Cathode Ray Tubes (CRT) were used. RGB color space is an example of a relative color standard (as opposed to the absolute one, e.g., CIE 1931). The primary colors (R-red, G-green and B-blue) mimicked phosphor in CRT luminophore. The RGB model uses additive color mixing to inform what kind of light needs to be emitted to produce a given color.

The value of a particular color is expressed as a vector of three elements—intensities of three primary colors, recall equation B.18). A transformation to a different color space is expressed by a transformation by a 3 x 3 matrix. Assume that values for each primary are quantized to m = 2n values; let the highest intensity value be k = m — 1; then @,0,0) is black, (k, k, k) is (television) white, (k. 0,0) is 'pure' red, and so on. The value k = 255 = 28 — 1 is common, i.e., 8 bits per color channel. There are 2563 = 224 = 16,777,216 possible colors in such a discretized space.

. The CMY—for Cyan, Magenta, Yellow—color model uses subtractive color mixing which is used in printing processes. It describes what kind of inks need to be applied so the light reflected from the white substrate (paper, painter's canvas) and passing through the inks produces a given color. CMYK stores ink values for black in addition. Black color can be generated from C, M, Y components but as it is abundant in printed documents, it is of advantage to have a special black ink. Many CMYK colors spaces are used for different sets of inks, substrates, and press characteristics (which change the color transfer function for each ink and thus change the appearance).

HSV - Hue, Saturation, and Value (also known as HSB, hue, saturation, brightness) is often used by painters because it is closer to their thinking and technique. Artists commonly use three to four dozen colors (characterized by the hue; technically, the dominant wavelength). If another color is to be obtained then it is mixed from the given ones, for example, 'purple' or 'orange'. The painter also wants colors of different saturation, e.g., to change 'fire brigade red' to pink. She will mix the 'fire brigade red' with white (and/or black) to obtain the desired lower saturation.

HSV decouples intensity information from color, while hue and saturation correspond to human perception, thus making this representation very useful for developing image processing algorithms. This will become clearer as we proceed to describe image enhancement algorithms (for example, equalization Algorithm 5.1), which if applied to each component of an RGB model would corrupt the human sense of color, but which would work more or less as expected if applied to the intensity component of HSV (leaving the color information unaffected). HSL (hue, saturation, lightness/luminance), also known as HLS or HSI (hue, saturation, intensity) is similar to HSV. 'Lightness' replaces 'brightness'. The difference is that the brightness of a pure color is equal to the brightness of white, while the lightness of a pure color is equal to the lightness of a medium gray. 2.4.4 Palette images

Palette images (called also indexed images) provide a simple way to reduce the amount of data needed to represent an image. The pixel values constitute a link to a lookup table (also

called a color table, color map, index register, palette). The lookup table contains as many entries as the range of possible values in the pixel, which is typically 8 bits = 256 values. Each entry of the table maps the pixel value to the color, so there are three values, one for each of three color components. In the typical case of the RGB color model, values for red, green and blue are provided. It is easy to see that this approach would reduce data consumption to one-third if each of the RGB channels had been using 8 bits (plus size of the look up table). Many widely used image formats for raster images such as TIFF, PNG and GIF can store palette images.

If the number of colors in the input image is less than or equal to the number of entries in the lookup table then all colors can be selected and no loss of information occurs. Such images may be cartoon movies, or program outputs.

In the more common case, the number of colors in the image exceeds the number of entries in the lookup table and a subset of colors has to be chosen, and a loss of information occurs. This color selection may be done many ways. The simplest is to quantize color space regularly into cubes of the same size. In the 8 bit example, there would be 8 x 8 x 8 = 256 such cubes. If there is, e.g., a green frog in green grass in the picture then there will not be enough shades of green available in the lookup table to display the image well. In such a case, it is better to check which colors appear in the image by creating histograms for all three color components and quantize then to provide more shades for colors which occur in the image frequently. If an image is converted to a palette representation then the nearest color (in some metric sense) in the lookup table is used to represent the color. This is an instance of vector quantization (see Section 14.4) which is widely used in analyzing large multi-dimensional datasets. It is also possible to view the occupation by the pixels of RGB space as a cluster analysis problem, susceptible to algorithms such as k-rneans.

The term pseudocolor is usually used when an original image is gray-level and is displayed in color; this is often done to exploit the color discriminatory power of human vision. The same palette machinery as described above is used for this purpose; a palette is loaded into the lookup table which visualizes the particular gray-scale image the best. It could either enhance local changes, or might provide various views of the image. Which palette to choose depends on the semantics of the image and cannot be derived from image statistics alone. This selection is an interactive process. Almost all computer graphics cards work with palette images directly in hardware.

Image Functions

1.1. Linearity

Linearity also concerns more general elements of vector spaces, for instance, functions. The linear combination is a key concept in linear mathematics, permitting the expression of a new element of a vector space as a sum of known elements multiplied by coefficients (scalars, usually real numbers).

A general linear combination of two vectors x. y can be written as ax + by, where a, b are scalars. Consider a mapping C between two linear spaces. It is called additive if C(x + y) Cx + Cy and homogeneous if C (ax) = aCx for any scalar a. From a practical point of view, this means that the sum of inputs (respectively, multiple) results in the sum of the respective outputs

(respectively, multiple). This property is also called a superposition principle. We call the mapping C linear if it is additive and homogeneous (i.e., satisfies the superposition principle). Equivalently, a linear mapping satisfies C(ax + by) = aCx + bCy for all vectors x, y and scalars a, 6, i.e., it preserves linear combinations.

1.2 The Dirac distribution and convolution

Some formal background on moving from the continuous to discrete domains will be of help, as will a definition of convolution. These are fundamental motivators for appreciating the use of linear mathematical theory in considering image functions. An ideal impulse is an important input signal:

$\delta(x, y),$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(x, y)\, dx\, dy = 1$$

The folslowing equation is called the 'sifting property' of the Dirac distribution; it provides the value of the function f(x,y) at the point (A, j.l)

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\, \delta(x - \lambda, y - \mu)\, dx\, dy = f(\lambda, \mu)$$

The 'sifting equation' C.2) can be used to describe the sampling process of a continuous image function f(x,y). We may express the image function as a linear combination of Dirac pulses located at the points a, b that cover the whole image plane; samples are weighted by the image function f(x.y)

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b)\, \delta(a - x, b - y)\, da\, db = f(x, y)$$

Convolution is an important operation in the linear approach to image analysis. The convolution is an integral which expresses the amount of overlap of one function f(t) as it is shifted over another function h(t). A ID convolution / * ft of functions /, ft over a finite range [0, t] is given by

$$\begin{aligned} g(x, y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b)\, h(x - a, y - b)\, da\, db \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - a, y - b)\, h(a, b)\, da\, db \\ &= (f * h)(x, y) = (h * f)(x, y) \end{aligned}$$

To be precise, the convolution integral has bounds — oc, oc. Here we can restrict to the interval [0,?], because we assume zero values for negative co-ordinates. Let /, g. ft be functions and aa scalar constant. Convolution can be generalized to higher dimensions. Convolution of 2D functions / and ft is denoted by / * ft, and is defined by the integral .In digital image analysis, the discrete convolution is expressed using sums instead of integrals. A digital image has a limited domain on the image plane. However, the limited domain does not prevent us from the use of convolutions as their results outside the image domain are zero. The convolution expresses a linear filtering process using the filter h\ linear filtering is often used in local image pre-processing and image restoration. Linear operations calculate the resulting value in the output image pixel g(i,j) as a linear combination of image intensities in a local neighborhood O of the pixel f{i,j) in the input image.

The contribution of the pixels in the neighborhood O is weighted by coefficients is equivalent to discrete convolution with the kernel ft, which is called a convolution mask. Rectangular neighborhoods O are often used with an odd number of pixels in rows and columns, enabling specification of the central pixel of the neighborhood.

Linear integral transforms are frequently employed in image processing. Using such transforms, images are treated as linear (vector) spaces. Same as when dealing with ID signals, there are two basic and commonly used representations of image functions: the spatial domain (pixels) and the frequency domain (frequency spectra). In the latter case, the image is expressed as a linear combination of some basis functions of some linear integral transform. For instance, the Fourier transform uses sines and cosines as basis functions. If linear operations are involved in the spatial domain (an important example of such linear operation is convolution) then there is a one-to-one mapping between the spatial and frequency representations of the image. Advanced signal/image processing goes beyond linear operations, and these non-linear image processing techniques are mainly used in the spatial domain.

2.1 Images as linear systems

Images and their processing can be modeled as superpositions of point spread functions which are represented by Dirac pulses 5 (equation 3.1). If this image representation is used, well-developed linear system theory can be employed. Anoperator is a mapping from one vector space to another. A linear operator C (also called linear system) has the property An image / can be expressed as a linear combination of point spread functions represented by Dirac pulses 5. Assume that the input image / is given by equation C.3). The response g of the linear system C to the input image / is given by where h is the impulse response of the linear system ?. In other words the output of the linear system C is expressed as the convolution of the input image / with an impulse response h of the linear system C. If the Fourier transform (explained in Sections 3.2.3 and 3.2.4) is applied to equation C.14) and the Fourier images are denoted by the respective capital letters then the following equation is obtained G{u, v) = F(u, v) H{u, v). C.15)Equation C.15) is often used in image pre-processing to express the behavior of smoothing or sharpening operations, It is important to remember that operations on real images are not in fact linear—-both the image co-ordinates and values of the image function (brightness) are limited. Real

images always have limited size, and the number of brightness levels is also finite. Nevertheless, image processing can be approximated by linear systems in many cases.

Sampling and the Shannon constraint

Equipped with understanding of the Fourier transform, we can now discuss more fully the issues surrounding sampling. A continuous image function /(#, y) can be sampled using a discrete grid of sampling points in the plane, but a second possibility is to expand the image function using some orthonormal functions as a basis—the Fourier transform is an example—and the coefficients of this expansion then represent the digitized image. The image is sampled at points x = j Ax, y = kAy, for j = 1,... ,M and k — 1, AT. Two neighboring sampling points are separated by distance Ax along the x axis and Ay along the y axis. Distances Ax and Ay are called the sampling intervals (on the x or y axis), and the matrix of samples f(j Ax. k Ay) constitutes the discrete image.

The ideal sampling s(x,y) in the regular grid can be represented using a collection of Dirac distributions S M N s(x, y) = ^2J2 s(x ~ iAx% y ~k Ay"> • C.39)

The sampled image fs(x,y) is the product of the continuous image f{xyy) and the sampling function s(x, y)

fs(x,y) = f(x,y)s(x,y)

M N = ffay) 5Z 5Z 8^x~iAx'y~kAy}•C.40)

We may consider an infinite sampling grid which is periodic with periods Ax. Ay and expand the sampling into a Fourier series. We obtain (see [Oppenheim et al., 1997])

•M E E S(x-j^x,y-kAy)

j=—oo k=—oo

Ax Ay ^—' ^ V Ax

m——oo n= —oo

? ?«(-?•« "

Equation C.40) can be expressed in the frequency domain using equation C.41):

Ay, C.41)

Fa(u,v) = Ax Ay m=—oo n=—ooAt5 A?/ At/] C.42) Thus the Fourier transform of the sampled image is the sum of periodically repeated Fourier transforms F(u, v) of the image. We can demonstrate this effect in ID case: assume that the maximal frequency of the signal is fm, so the signal is band-limited, meaning that its Fourier transform F is zero outside a certain interval of frequencies I/I >fm- The spectra will be repeated as a consequence of discretization—see Figure 3.10.

In the case of 2D images, band-limited means that the spectrum F(u, v) = 0 for | u \> U, | v |> V, where U, V are maximal frequencies. -« -V. -I 0 /. .A- 2JI 3j: /[Hz]

Figure 3.10: Repeated spectra of the ID signal due to sampling. Non-overlapped case when /. >Vrr,- Periodic repetition of the Fourier transform result F(u, v) may under certain conditions cause distortion of the image, which is called aliasing; this happens when individual digitized components F(u, v) overlap. Overlapping of the periodically repeated results of the Fourier transform F{u. v) of an image with band-limited spectrum can be prevented if the sampling interval is chosen such that Ax < w' Ay<w- C-43)

This is the Shannon sampling theorem, known from signal processing theory or control theory. The theorem has a simple physical interpretation in image analysis: The sampling interval should be chosen in size such that it is less than half of the smallest interesting detail in the image.

The sampling function is not the Dirac distribution in real digitizers—limited impulses (quite narrow ones with limited amplitude) are used instead. Assume a rectangular sampling grid which consists of M x TV such equal and non-overlapping impulses hs(x,y) with sampling period Ax, Ay; this function realistically simulates real image sensors. Outside the sensitive area of the sensor, hs(x,y) — 0. Values of image samples are obtained by integration of the product fhs—in reality this integration is done on the sensitive surface of the sensor element. The sampled image is then given by

M N f8(x,y) =]T ^/(x,y) ha(x-jAx,y - k Ay) . C.44) j=l fc=l The sampled image /A. is distorted by the convolution of the original image / and the limited impulse hs. The distortion of the frequency spectrum of the function Fs can be expressed using the Fourier transform y m=-oo n=-oo x «> / \ o / where HH = P{hs}. In real image digitizers, a sampling interval about ten times smaller than that indicated by the Shannon sampling theorem, equation C.43), is used—this is because algorithms which reconstruct the continuous image on a display from the digitized image function use only a step function, i.e., a line in the image is created from pixels represented by

individual squares:

A demonstration with an image of 256 gray-levels will illustrate the effect of sparse sampling. Figure 3.11a shows a monochromatic image with 256 x 256 pixels; Figure 3.11b shows the same scene digitized into a reduced grid of 128 x 128 pixels, Figure 3.11c into 64 x 64 pixels, and Figure 3.lid into 32 x 32 pixels. Decline in image quality is clear from Figures .3.11a-d. Quality may be improved by viewing from a distance and with screwed-up eyes, implying that the under-sampled images still hold substantial information. Much of this visual degradation is caused by aliasing in the reconstruction of the continuous image function for display. This can be improved by the reconstruction algorithm interpolating brightness values in neighboring pixels and this technique is called anti-aliasing; this is often used in computer graphics [Rogers, 1985]. If anti-aliasing is used, the sampling interval can be brought near to the theoretical value of **Shannon'sDiscrete cosine transform** The discrete cosine transform (DCT) is a linear integral transformation similar to the discrete Fourier transform (DFT) [Rao and Yip, 1990]. In ID, cosines with growing frequencies constitute the basis functions used for function expansion: the expansion is a linear combination of these basis cosines, and real numbers suffice for such an expansion (the Fourier transform required complex numbers). The DCT expansion corresponds to a DFT of approximately double length operating on a function with even symmetry. Similarly to the DFT, the DCT operates on function samples of finite length, and a periodic extension of this function is needed to be able to perform DCT (or DFT) expansion. The DCT requires a stricter periodic extension (a more strict boundary condition) than the DFT—it requires that the extension is an even function.

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-2\pi i(xu+yv)}\, dx\, dy$$

Two options arise in relation to boundary conditions for a discrete finite sequence. The first one is whether the function is even or odd at both the left and right boundaries of the domain, and the second is about which point the function is even or odd. As illustration, consider an example sequence wxyz. If the data are even about sample w, the even extension is zyxwxyz. If the sequence is even about the point halfway between w and the previous point, the extension sequence is that in which w is repeated.

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, e^{2\pi i(xu+yv)}\, du\, dv$$

Consider the general case which covers both the discrete cosine transform (with even symmetry) and the discrete sine transform (with odd symmetry). The first choice has to be made about the symmetry at both left and right bounds of the signal, i.e., 2x2=4 possible options. The second choice is about which point the extension is performed, also at both left and right bounds of the signal, i.e., an additional 2x2=4 possible options. Altogether 4 x 4=16 possibilities are obtained. If we do not allow odd periodic extensions
then the sine transforms are ruled out and 8 possible choices remain yielding 8 different
types of DCT. If the same type of point is used for extension at left and right bounds then only half the options remain, i.e., 8/2=4. This yields four basic types of DCT—they are usually denoted by suffixing Roman numbers as DCT-I, DCT-II, DCT-III, DCT-IV.

$$\mathcal{F}\{f(x,y)\} = F(u,v)$$

$$\mathcal{F}\{a\, f_1(x,y) + b\, f_2(x,y)\} = a\, F_1(u,v) + b\, F_2(u,v)$$

$$\mathcal{F}\{f(x-a, y-b)\} = F(u,v)\, e^{-2\pi i(au+bv)}$$

$$\mathcal{F}\{f(x,y)\, e^{2\pi i(u_0 x + v_0 y)}\} = F(u-u_0, v-v_0)$$

Wavelet transform

The Fourier transform (Section 3.2.3) expands a signal as a possibly infinite linear combination of sines and cosines. The disadvantage is that only information about the frequency spectrum is provided, and no information is available on the time at which events occur. In another words, the Fourier spectrum provides all the frequencies present
in an image but does not tell where they are present. We also know that the relation between the frequency and spatial resolutions is given by the uncertainty principle, Equation C.24).

One solution to the problem of localizing changes in the signal (image) is to use the short time Fourier transform, where the signal is divided into small windows and treated locally as it were periodic (as was explained in Section 3.2.3). The uncertainty principle provides guidance on how to select the windows to minimize negative effects, i.e., windows have to join neighboring windows smoothly. The window dilemma remains—a narrow window yields poor frequency resolution, while a wide window provides poor localization.

The wavelet transform goes further than the short time Fourier transform. It also analyzes the signal (image) by multiplying it by a window function and performing an orthogonal expansion, analogously to other linear integral transformations. There are two directions in which the analysis is extended. In the first direction, the basis functions (called wavelets, meaning a small wave, or mother wavelets) are more complicated than sines and cosines. They provide localization. In n the second direction, the analysis is performed at multiple scales. To understand this, note that modeling a spike in a function (a noise dot, for example) with a sum of a huge number of functions will be hard because of the spike's strict locality Functions that are already local will be naturally suited to the task. This means that such functions lend themselves to more compact representation via wavelets—sharp spikes and discontinuities normally take fewer wavelet bases to represent as compared to the sine-cosine basis functions. Localization in the spatial domain together with the wavelet's localization in frequency yields a sparse representation of many practical signals (images). This sparseness opens the door to successful applications in data/image compression, noise filtering and detecting features in images.

The wavelet transform was defined generally in equations C.48)-C.49) without the need to specify a particular mother wavelet: the user can select or design the basis of the expansion according to application needs. The scheme is actually a variant of the classical scheme two-channel sub-band coder—known in signal processing. This method yields a fast wavelet transform which can be imagined as a box into which a signal (an image) enters and the wavelet coefficients c appear quickly at its output.

Eigen-analysis

Many disciplines, including image analysis, seek to represent observations, signals, images and general data in a form that enhances the mutual independence of contributory components. Linear algebra provides very appropriate tools for such representations. One observation or measurement is assumed to be a point in a linear space; this space will have some 'natural' basis vectors which allow data to be expressed as a linear combination in a new coordinate system consisting of orthogonal basis vectors. These basis vectors are the eigen-vectors, and the inherent orthogonality of eigen-vectors secures mutual independence. For annxn square regular matrix A, eigen-vectors are solutions of the equation 4x = Ax, C.55) where A is called an eigen-value (which may be complex). We will review eigen-analysis from linear algebra which approaches the task from a deterministic standpoint, and in a later section, we will develop a statistical view based on covariance matrices and principal component analysis.

A system of linear equations can be expressed in a matrix form as As. = b, where A is the matrix of the system. The extended matrix of the system is created by concatenating a column vector b to the matrix A, i.e., [A\b]. There is another class of matrix transformation called similar transformations.

Let A be a regular matrix: matrices A and B with real or complex entries are called similar if there exists an invertible square matrix P such that P_1AP = B. Similar matrices share many useful properties—they have the same rank, determinant, trace, characteristic polynomial, minimal polynomial and eigen-values (but not necessarily the same eigen-vectors). Similarity transformations allow us to express regular matrices in several useful forms. Let / be the unitary

matrix having values 1 only on the main diagonal and zeros elsewhere. The polynomial of degree n given as det(^4 — XI) is called the characteristic polynomial. Then the eigen-equation C.55) holds if det(A — XI) — 0. The roots of the characteristic polynomial are the eigen-values A. Consequently, A has n eigen-values which are not necessarily distinct—multiple eigen-values arise from multiple roots of the polynomial. Seeking roots of the characteristic polynomial is usually rather poor computationally, and more effective methods such as singular value decomposition (SVD) are used.

Singular value decomposition

Eigen-values and eigen-vectors are defined on square matrices; a generalization—singular values—operates on rectangular matrices, and is approached via the singular value decomposition (SVD). SVD is a powerful linear algebra factorization technique of a rectangular real or complex matrix; it works even for singular or numerically near-singular matrices. SVD is used with many applications for solving linear equations in the least-square sense, e.g., in signal processing and statistics. It can be viewed as a generalization of the 2For a matrix A, A* will be the conjugate transpose (also called the adjoint) which is the transpose of the complex conjugate of A.

SVD proceeds by noting that any m x n matrix A, rn> n, (with real or complex entries) can be decomposed into a product of three matrices, A = UDVT , C.57) where U is m x m with orthonormal columns, D is a non-negative diagonal matrix, and VT has orthonormal rows. SVD can be understood as decoupling input of size m into output of size n. The matrix V contains a set of orthonormal 'input' or basis vector directions (left-singular vectors) for the matrix A, and the matrix U contains a set of orthonormal 'output' basis vector directions (right-singular vectors) for A. The matrix D contains the singular values, which can be understood as scalar 'gains' by which each corresponding input is multiplied to give the corresponding output.

Principal component analysis

Principal component analysis (PCA) is a powerful and widely used linear technique in statistics, signal processing, image processing, and elsewhere. It appears under several names: it is also called the (discrete) In statistics, PCA is a method for simplifying a multidimensional dataset to lower dimensions for analysis or visualization. PCA is a linear transform that represents the data in a new coordinate system in which basis vectors follow modes of greatest variance in the data: it is the optimal linear transformation which divides an observed space into orthogonal subspaces with the largest variance. Thus, new basis vectors are calculated for the particular data set. One price to be paid for PCA's flexibility is in higher computational requirements as compared to, e.g., the fast Fourier transform.

As an example, consider the use of PCA on images—this approach was popularized by the application of PCA to face recognition. The image is considered as a very long ID vector by concatenating image pixels column by column (or alternatively row by row). If we have fewer observations than unknowns, the system of linear equations is not over- The significance of image reconstruction from projections can be seen in computer tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), astronomy, holography, etc. In

these applications the image formation can be modeled using the Radon transform. In image reconstruction, projections in different directions are acquired by sensors and the two-dimensional image must be reconstructed. The inverse Radon transform is of particular interest.

3.3 Images as stochastic processes

Images are statistical in nature due to random changes and noise, and it is sometimes of advantage to treat image functions as realizations of a stochastic In such an approach, questions regarding image information content and redundancy can be answered using probability distributions, and simplifying probabilistic characterizations as the mean, dispersion, correlation functions, etc.

$$H = -\sum_{k=1}^{n} p(a_k) \log_2 p(a_k)$$

A stochastic process (random process, random field) is a generalization of the random variable concept. We will constrain ourselves to stochastic processes of with two independent variables x. y which are the coordinates in the image. We denote a stochastic
process by 0 and (j)(x,y) is a random variable representing the gray-level at pixel (x,y).

$$P_k(z_1, \ldots, z_k; x_1, y_1, \ldots, x_k, y_k) = \mathcal{P}\{f(x_1, y_1, \omega_{i_1}) < z_1, f(x_2, y_2, \omega_{i_2}) < z_2, \ldots, f(x_k, y_k, \omega_{i_k}) < z_k\}$$

DEPARTMENT OF COMPUTER SCIENCE

Digital Image Processing

Unit-1 Possible Questions

2 Marks Questions:

1. Define an Image
2. Define digital image processing
3. Mention the two principal application of image processing
4. Mention the 3 main primitive operations in image processing
5. Abbreviate the following CAT , PET
6. Draw and explain the spectral bands in electro magnetic waves
7. Mention any two applications of x-ray imaging
8. Mention any two applications of UV-ray imaging
9. Mention any two applications of gamma ray imaging
10. Which band is used in detecting fault in manufactured goods
11. Which EM waves can penetrate through clouds and vegetation
12. Define fractals
13. Mention the steps adopted in digital image processing
14. Mention the different categories of digital storage.
15. How is image represented digitally.

8 marks

1. explain in detail the fundamental steps in digital image processing with suitable diagram
2. Illustrate with neat diagram the components of an image processing system
3. explain the following:
   i). gamma ray imaging ii) x-ray imaging iii) ultraviolet band imaging
4. discuss in detail the examples of fields that use digital image processing system

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC ACT 1956)
M.sc Computer Science
DEPARTMENT OF CS, CA & IT
Third Semester(Digital Image processing)

S.no	Questions	OPT 1	OPT 2	OPT 3	OPT 4	Answers
1	An image may be defined as a 2D function *f(x,y)* , where x and y are spatial co-ordinates and the amplitude of *f* at any point is called _____	mask	contrast	intensity	histogram	intensity
2	A digital image is composed of a finite number of elements, each of which has a particular location and value called _____	intensity	objects	attributes	pixels	pixels
3	_____ is the first process in image processing, which involves preprocessing.	Image enhancement	Image acquisition	Image restoration	Image segmentation	Image acquisition
4	_____ technique brings out the details that is obscured or simply to highlight certain features of interest in an image.	Image enhancement	Image acquisition	Image restoration	Image segmentation	Image enhancement
5	_____ is a objective process that tend to based on mathematical and probabilistic models of image degradation.	Image enhancement	Image acquisition	Image restoration	Image segmentation	Image restoration
6	_____ is a process of representing images in various degrees of resolution	Color image processing	Wavelets	Image restoration	Image segmentation	Wavelets
7	_____ deals with techniques for reducing the storage required to save an image , or the bandwidth required to transmit it.	Image restoration	Image segmentation	Image acquisition	Image compression	Image compression
8	_____ deals with tools for extracting image components that are useful in the representation and description of shape	Morphological processing	Image description	Image recognition	Image representation	Morphological processing
9	_____ procedures partitions an image into its constituent parts or objects	Image enhancement	Image acquisition	Image restoration	Image segmentation	Image segmentation
10	_____ is appropriate when the focus is on external properties such as corners and inflections	Feature selection	Boundary representation	Regional representation	Image analysis	Boundary representation
11	_____ is appropriate when the focus is on internal properties such as texture or skeletal shape.	Feature selection	Boundary representation	Regional representation	Image analysis	Regional representation
12	_____ deals with extracting attributes that result in some quantitative information of interest.	Feature selection	Boundary representation	Regional representation	Image analysis	Feature selection
13	_____ is a process that assign a label based on its descriptors.	Image recognition	Image analysis	Image description	Image representation	Image recognition

14	The mechanical digitizers that move in two linear directions and use a flat bed arrangement are called as	photodiode	filters	CCD	micro densitometers	micro densitometers
15	_____ sensors are used in digital cameras and other light sensing instruments.	photodiode	filters	CCD	micro densitometers	CCD
16	_____ method combine image formation and recording on a single density called photographic flim.	Photoelectrical	Photochemical	Multispectral	Iconoscope	Photochemical
17	_____ method separate the recording process from image formation and detection	Photoelectrical	Photochemical	Multispectral	Iconoscope	Photoelectrical
18	A _____ is required input images to a digital computer, produces digital image composed of discrete intensity values at discrete positions	sensor	hard copier	processor	digitizer	digitizer
19	In general purpose computer the entire image is kept in a memory commonly called _____	spool	video buffer	frame buffer	raster memory	frame buffer
20	_____ are also called non-erasable display devices.	sensor	hard copier	processor	digitizer	hard copier
21	_____ produces image by blurring the surface of a special type of paper.	Ink jet	Laser	Photodiode	Thermal copier	Thermal copier
22	_____ is concerned with brightness measurement	Photometry	frequencies	raster diode	x-ray	Photometry
23	_____ studies light reflectance or emission depending on wavelength.	brightness	spatial resolution	Colorimetry	frequencies	Colorimetry
24	The _____ is given by the proximity of image samples in the image plane.	spatial solution	spatial resolution	spatial combination	spacial decomposition	spatial resolution
25	The _____is given by the bandwidth of the light frequencies captured by the sensor	spatial solution	spatial resolution	spatial combination	spectral resolution	spectral resolution
26	The _____corresponds to the number of distinguishable gray levels.	radiometric resolution	video resolution	audio resolution	dvd	radiometric resolution
27	The _____is given by the interval between time samples at which images are captured.	hour	date	time resolution	session	time resolution
28	_____ means that the function f(x,y) is sampled into a matrix with M rows and N columns.	image fragramentation	Image digitization	image digital	image vertix	Image digitization
29	The image _____assigns to each continuous sample an integer value.	quality	coordinates	sample	quantitation	quantitation
30	Two neighboring sampling points are separated by distance Delta_x along the x axis and Delta_y along the y axis. Distances Delta_x and Delta_y are called the _____	sampling interval	sampling time	sampling code	sample value	sampling interval

31	The ideal sampling s(x,y) in the regular grid can be represented using a collection of _____	image vertix	video resolution	Dirac distributions	furier transform	Dirac distributions
32	Periodic repetition of the Fourier transform result F(u,v) may under certain conditions cause distortion of the image which is called _____	video resolution	aliasing	image vertix	sampling code	aliasing
33	These sampling points are ordered in the plane and their geometric relation is called the _____	latency	hexagonal	vertency	grid	grid
34	Grids used in practice are mainly square or _____	hexagonal	image	image vertix	video resolution	hexagonal
35	One infinitely small sampling point in the grid corresponds to one _____ in the digital image.	transfer	aliasing	pixel	latency	pixel
36	A _____ of the sampled image is expressed as a digital value in image processing.	aliasing	magnitude	latency	hexagonal	magnitude
37	The transition between continuous values of the image function (brightness) and its digital equivalent is called _____	quantam	transfer	aliasing	quantitation	quantitation
38	_____ is one of the important metric properties of an image	Distance	quantam	latency	transfer	Distance
39	The _____ distance is defined by $D_8((i,j),(h,k)) = sqrt((i - h)^2 + (j - k)^2)$	hexagonal	Euclidean	aliasing	distance	Euclidean
40	The _____ distance is defined by $D_4((i,j), (h.k))) = \|i - h\| + \| j - k \|$	random	quantam	city block	both a & b	city block
41	The _____ distance is defined by $D_8((i,j), (h.k))) = \max\{\|i - h\|, \| j - k \|\}$	chessgraph	chesssquare	chessbox	chessboard	chessboard
42	_____ is another important metric in digital images.	Pixel adjacency	auxilary	pixel point	pixel adjent	Pixel adjacency
43	The pixel adjacency include _____	4 X 4	4 & 8-neighborhood	4 * 8-neighborhood	4 $ 8-neighborhood	4 & 8-neighborhood
44	_____ is a contiguous set, which consists of several adjacent pixels	division	point	Region	zone	Region
45	_____ of a region R is the set of pixels within the region that have one or more neighbors outside R	line	margin	end	Border	Border
46	_____is a local property of a pixel and its immediate neighborhood --it is a vector given by a magnitude and direction.	Edge	end	endpoint	boundry	Edge
47	The _____is perpendicular to the gradient direction which points in the direction of image function growth.	edge correction	edge direction	edge collection	edge division	edge direction

48	The _____ is a global concept related to a region, while _____ expresses local properties of an image function.	edge, boundary	line, edge	Border , edge	line, region	Border , edge
49	The _____attaches to each pixel, whose direction is that of increasing brightness, and is a multiple of 90 degrees, while its magnitude is the absolute difference between the brightness of the relevant pair of pixels.	cut edges	rounded edges	broken edges	Crack edges	Crack edges
50	Topological properties of images are invariant to _____.	rubber sheet transformations	plastic sheet transformations	plastic sheet transforms	metal sheet transforms	rubber sheet transformations
51	_____ finds the difference between the number of regions and the number of holes in them	latency	Euler--Poincare	edge, boundary	Eler--Pointcare	Euler--Poincare
52	_____ is used to describe topological properties of objects.	concave hull	histogram hull	Convex hull	opaque hull	Convex hull
53	The _____ is the smallest region which contains the object, such that any two points of the region can be connected by a straight line, all points of which belong to the region	histogram hull	opaque hull	concave hull	convex hull	convex hull
54	_____ provides the frequency of the brightness value z in the image.	Brightness histogram	hue histogram	contrast histrogram	saturation histogram	Brightness histogram
55	_____ has constant power spectrum (its intensity does not decrease with increasing frequency); very crude approximation of image noise	black noise	White noise	gray noise	spatial noise	White noise
56	_____ are used for description of object borders	lines	circle	Chains	sqare	Chains
57	_____ are examples of hierarchical data structures	pyramids	n pyramids	bi-pyramids	Pyramids and quadtrees	Pyramids and quadtrees
58	_____ consists of images containing original data; integer matrices with data about pixel brightness.	Iconic images	latency images	dual images	single image	Iconic images
59	_____ is an example of topological data structures	latency	Graph	chart	view	Graph
60	_____ often used to represent strings of symbols in an image matrix	encoding	half length	Run length encoding	both a & b	Run length encoding

UNIT II

SYLLABUS:

Some Basic relationships between Pixels-Basic gray level transformations- Histogram processing - Basic spatial filtering- Smoothing special filtering- Image Degradation/ Restoration process- Noise Models.

Image pre-processing

Pre-processing is the name used for operations on images at the lowest level of abstraction— both input and output are intensity images. These iconic images are usually of the same kind as the original data captured by the sensor, with an intensity image usually represented by a matrix or matrices of image function values (brightnesses). Pre-processing does not increase image information content. If information is measured using entropy, then pre-processing typically decreases image information content. From the information-theoretic viewpoint it can thus be concluded that the best pre-processing is no pre-processing, and without question, the best way to avoid (elaborate) pre-processing is to concentrate on high-quality image acquisition. Nevertheless, preprocessing is very useful in a variety of situations since it helps to suppress information that is not relevant to the specific image processing or analysis task. Therefore, the aim of pre-processing is an improvement of the image data that suppresses undesired distortions or enhances some image features important for further processing, although geometric transformations of images (e.g., rotation, scaling, translation) are also classified as pre-processing methods here since similar techniques are used.

Image pre-processing methods are classified here into four categories according to the size of the pixel neighborhood that is used for the calculation of a new pixel brightness. A considerable redundancy of information in most images allows image pre-processing methods to explore image data itself to learn image characteristics in a statistical sense.

These characteristics are used either to suppress unintended degradations such as noise or to enhance the image. Neighboring pixels corresponding to given object in real images have essentially the same or similar brightness value, so if a distorted pixel can be picked out from the image, it can usually be restored as an average value of neighboring pixels.

1.Gray-scale transformation

Gray-scale transformations do not depend on the position of the pixel in the image. A transformation T of the original brightness p from scale [po>Pfc] into brightness q from a new scale [<?o, Qk] is given by q = T(p) . E.3)

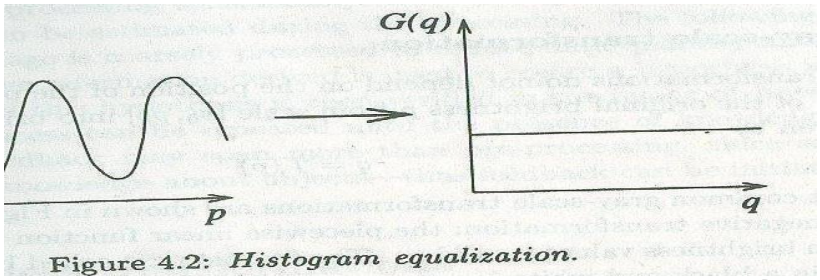


Figure 4.2: *Histogram equalization.*

The most common gray-scale transformations are shown in Figure 5.1; the piecewise linear function a enhances the image contrast between brightness values pi and p2. The function b is called brightness thresholding and results in a black-and-white image; the straight line c denotes the negative transformation.

$$\sum_{i=0}^{k} G(q_i) = \sum_{i=0}^{k} H(p_i)$$

Digital images have a very limited number of gray-levels, so gray-scale transformations are easy to realize both in hardware and software. Often only 256 bytes of memory called a look-up table) are needed. The original brightness is the index to the look-up, and the table content gives the new brightness. The image signal usually passes through a look-up table in image displays, enabling simple gray-scale transformation in real time. The same principle can be used for color displays. A color signal consists of three components—red, green, and blue; three look-up tables provide all possible color scale transformations. These tables are called the palette in personal computer terminology .

$$f = \frac{N^2}{q_k - q_0}$$

Gray-scale transformations are used mainly when an image is viewed by a human observer, and a transformed image might be more easily interpreted if the contrast is enhanced. For instance, an X-ray image can often be much clearer after transformation. A gray-scale transformation for contrast enhancement is usually found automatically using the histogram equalization technique. The aim is to create an image with equally distributed brightness levels over the whole brightness scale. Histogram equalization enhances contrast for brightness values close to histogram maxima, and decreases contrast near minima.

Denote the input histogram by H(p) and recall that the input gray-scale is \po,Pk]-
The intention is to find a monotonic pixel brightness transformation q = T(p) such that the desired output histogram G(q) is uniform over the whole output brightness scale The histogram can be treated as a discrete probability density function. The monotonic property of the transform T implies The sums in equation E.4) can be interpreted as discrete distribution functions. Assume that the image has N rows and columns; then the equalized histogram G(q) corresponds to the uniform probability density function / whose function value is a constant: The value from

equation E.5) replaces the left side of equation E.4). The equalized histogram can be obtained precisely only for the 'idealized' continuous probability density, in which case equation E.4) becomes

The desired pixel brightness transformation T can then be derived as rip)

The integral in equation E.7) is called the cumulative histogram, which is approximated by a sum in digital images, so the resulting histogram is not equalized ideally. The discrete approximation of the continuous pixel brightness transformation from equation

Formally, the algorithm to perform equalization is as follows.

Algorithm: Histogram equalization

1. For aniVxM image of G gray-levels (often 256), create an array H of length G initialized with 0 values.
2. **Form the image histogram:** Scan every pixel and increment the relevant member of H—if pixel p has intensity gp, perform

H[gp] = H[gp] + 1.

3. Form the cumulative image histogram Hc:

$$H_c[0] = H[0]$$
$$H_c[p] = H_c[p-1] + H[p] \qquad p = 1, 2, \ldots, G-1$$

4. Set

$$g_q = T[g_p]$$

(This step obviously lends itself to more efficient implementation by constructing a look-up table of the multiples of (G — l)/NM, and making comparisons with the values in jffc, which are monotonically increasing.) gray-scale transformation is also used to compensate for exponential 7-correction used in cameras.

Pseudo-color is yet another kind of gray-scale transform. The individual brightnesses in the input monochromatic image are coded to some color. Since the human eye is much more sensitive to change in color than to change in brightness, much more detail can be perceived in pseudo-colored images.

2 Geometric transformations

Geometric transforms are common in computer graphics, and are often used in image analysis as well. They permit elimination of the geometric distortion that occurs when an image is captured. If one attempts to match two different images of the same object, a geometric transformation may be needed. We consider geometric transformations only in 2D, as this is sufficient for most digital images. One example is an attempt to match remotely sensed images of the same area taken after one year, when the more recent image was probably not taken from precisely the same position. To inspect changes over the year, it is necessary first to execute a geometric transformation, and then subtract

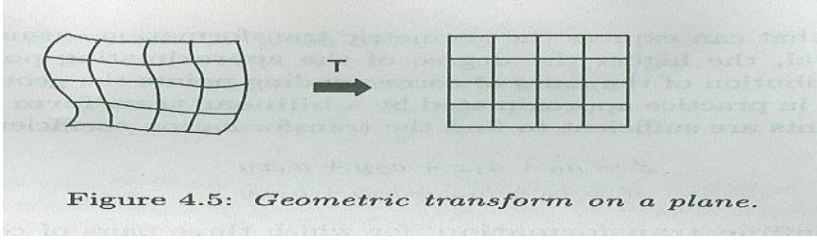


Figure 4.5: *Geometric transform on a plane.*

one image from the other. Another example, commonly encountered in document image processing applications, is correcting for document skew, which occurs when an image with an obvious orientation (for example, a printed page) is scanned, or otherwise captured, at a different orientation. This difference may be very small, but can be critical if the orientation is exploited in subsequent processing—this is usually the case in optical
character recognition (OCR).

A geometric transform is a vector function T that maps the pixel (x, y) to a new position {pc' ,yf)—an illustration of the whole region transformed on a point-to-point basis is in Figure. T is defined by its two component equations

$$x' = T_x(x, y) \qquad y' = T_y(x, y)$$

E.9) The transformation equations Tx and Ty are either known in advance—for example, in the case of rotation, translation, scaling—or can be determined from known original and transformed images. Several pixels in both images with known correspondences are used to derive the unknown transformation.

A geometric transform consists of two basic steps. First is the pixel co-ordinate transformation, which maps the co-ordinates of the input image pixel to the point in the output image. The output point co-ordinates should be computed as continuous values (real numbers), as the position does not necessarily match the digital grid after the transform. The second step is to find the point in the digital raster which matches the transformed point and determine its brightness value. The brightness is usually computed
as an interpolation of the brightnesses of several points in the neighborhood. This idea enables the classification of geometric transforms among other pre-processing techniques, the criterion being that only the neighborhood of a processed pixel is needed for the calculation. Geometric transforms are on the boundary between point and local operations.

5.2.1 Pixel co-ordinate transformations

Equation E.9) shows the general case of finding the co-ordinates of a point in the output image after a geometric transform. It is usually approximated by a polynomial equation .This transform is linear with respect to the coefficients ark, Wk and so if pairs of corresponding points (x, y), {x\yf) in both images are known, it is possible to determine arfc, brk by solving a set of linear equations. More points than coefficients are usually used to provide robustness; the mean square method is often used.

In the case where the geometric transform does not change rapidly depending on position in the image, low-order approximating polynomials, m = 2 or m = 3, are used, needing at least 6 or 10 pairs of corresponding points. The corresponding points should be distributed in the image

in a way that can express the geometric transformation— usually they are spread uniformly. In general, the higher the degree of the approximating polynomial, the more sensitive to the distribution of the pairs of corresponding points the

Geometric transform is. $$x' = \sum_{r=0}^{m} \sum_{k=0}^{m-r} a_{rk}\, x^r\, y^k \qquad y' = \sum_{r=0}^{m} \sum_{k=0}^{m-r} b_{rk}\, x^r\, y^k$$

Equation E.9) is in practice approximated by a bilinear transform for which four pairs of corresponding points are sufficient to find the transformation coefficients

x = ao + ai x + a2 y + as x y ,

y' = b0 + b1x + b2y + b3 x y .

E.11) Even simpler is the affine transformation, for which three pairs of corresponding points are sufficient to find the coefficients

x = clq + a-i x + a2 y ,

yf = 60 + 6i x + b2y . E.12)

The affine transformation includes typical geometric transformations such as rotation, translation, scaling, and skewing. If the transformation is singular (has no inverse), then J= 0. If the area of the image is invariant under the transformation, then J = 1.

It is possible to approximate complex geometric transformations (distortion) by partitioning an image into smaller rectangular subimages; for each subimage, a simple geometric transformation, such as the affine, is estimated using pairs of corresponding pixels. The geometric transformation (distortion) is then repaired separately in each subimage. Even simpler is the affine transformation, for which three pairs of corresponding points are sufficient to find the coefficients

x = clq + a-i x + a2 y , yf = 60 + 6i x + b2y . E.12)

The affine transformation includes typical geometric transformations such as rotation, translation, scaling, and skewing. It is possible to approximate complex geometric transformations (distortion) by partitioning an image into smaller rectangular subimages; for each subimage, a simple geometric transformation, such as the affine, is estimated using pairs of corresponding pixels. The geometric transformation (distortion) is then repaired separately in each subimage.

Brightness interpolation

Assume that the planar transformation given by equation E.9 has been accomplished, and new point co-ordinates (xf,yf) obtained. The position of the point does not in general fit the discrete raster of the output image, and the collection of transformed points gives the samples of the output image with non-integer co-ordinates. Values on the integer grid are needed, and each pixel value in the output image raster can be obtained by brightness interpolation of some neighboring non-integer samples [Moik, 1980].

$$f(i,j) = e(i,j)\; g(i,j)$$

Brightness interpolation influences image quality. The simpler the interpolation, the greater is the loss in geometric and photometric accuracy, but the interpolation neighborhood is often reasonably small due to computational load. The three most common interpolation methods

are nearest neighbor, linear, and bi-cubic. The brightness interpolation problem is usually expressed in a dual way by determining the brightness of the original point in the input image that corresponds to the point in the output image lying on the discrete raster. Assume that we wish to compute the brightness value of the pixel (#', yf) in the output image where x' and yf lie on the discrete raster (integer numbers, illustrated by solid lines in figures). The co-ordinates of the point (x, y) in the original image can be obtained by inverting the planar transformation in equation E.9):

(x,y) = T-1(x',y'). E.19)

In general, the real co-ordinates after inverse transformation (dashed lines in figures) do not fit the input image discrete raster (solid lines), and so the brightness is not known. The only information available about the originally continuous image function /'(&*, y) is its sampled version gs(l Ax,kAy). To get the brightness value of the point (x,y), the input image is resampled. Denote the result of the brightness interpolation by fn(x,y), where n distinguishes different interpolation methods. The brightness can be expressed by the convolution equation

fn{x,y)= ^2 H gs(l&x,kAy)hn(x-lAx,y-kAy). E.20) -co k= — oo

The function hn is called the interpolation kernel. Usually, only a small neighborhood is used, outside which hn is zero. (The same idea was used in continuous image sampling— recall that in equation C.44) the function h8 represented the limited impulse.) The discrete raster of the original image is depicted by the solid line.

Nearest-neighborhood interpolation assigns to the point (x,y) the brightness value of the nearest point g in the discrete raster; this is demonstrated in Figure 5.7. On the right side is the interpolation kernel h\ in the ID case. The left side of Figure 5.7 shows how the new brightness is assigned. Dashed lines show how the inverse planar transformation maps the raster of the output image into the input image; full lines show the raster of the input image. Nearest-neighborhood interpolation is given by

fi(x,y) =0s(round(a;),roundB/)) . E.21)

The position error of the nearest-neighborhood interpolation is at most half a pixel. This error is perceptible on objects with straight-line boundaries that may appear step-like after the transformation. The discrete raster of the original image is depicted by the solid line. Linear interpolation explores four points neighboring the point (.x, y), and assumes that the brightness function is linear in this neighborhood. Linear interpolation is demonstrated in Figure 5.8, the left-hand side of which shows which points are used for

interpolation. **Linear interpolation is given by the equation**

f2(x,y) = (l-a)(l-b)gs(l,k) + a A - b) gs{l + 1, k) + b A - a) gs(l, k + 1) + abgs(l + 1, k + 1)

E.22)

Local pre-processing

Local pre-processing methods can be divided into two groups according to the goal of the processing. First, smoothing aims to suppress noise or other small fluctuations in the image; it is equivalent to the suppression of high frequencies in the Fourier transform domain. Unfortunately, smoothing also blurs all sharp edges that bear important information about the image. Second, gradient operators are based on local derivatives of the image function.

Derivatives are bigger at locations of the image where the image function undergoes rapid changes, and the aim of gradient operators is to indicate such locations in the image. Gradient operators have a similar effect to suppressing low frequencies in the Fourier transform domain. Noise is often high frequency in nature; unfortunately, if a gradient operator is applied to an image, the noise level increases simultaneously. Clearly, smoothing and gradient operators have conflicting aims. Some pre-processing algorithms solve this problem and permit smoothing and edge enhancement simultaneously.

Another classification of local pre-processing methods is according to the transformation properties; linear and non-linear transformations can be distinguished. Linear operations calculate the resulting value in the output image pixel g{i,j) as a linear combination of brightnesses in a local neighborhood O of the pixel f{i,j) in the input image. The contribution of the pixels in the neighborhood O is weighted by coefficients ft: is equivalent to discrete convolution with the kernel ft, which is called a convolution mask. Rectangular neighborhoods O are often used with an odd number of pixels in rows and columns, enabling specification of the central pixel of the neighborhood.

Local pre-processing methods typically use very little a priori knowledge about the image contents. It is very difficult to infer this knowledge while an image is being processed, as the known neighborhood O of the processed pixel is small. Smoothing operations will benefit if some general knowledge about image degradation is available; this might, for instance, be statistical parameters of the noise. The choice of the local transformation, size, and shape of the neighborhood O depends strongly on the size of objects in the processed image. If objects are rather large, an image can be enhanced by smoothing of small degradations.

1 Image smoothing

Image smoothing is the set of local pre-processing methods whose predominant use is the suppression of image noise—it uses redundancy in the image data. Calculation of the new value is based on the averaging of brightness values in some neighborhood O. Smoothing poses the problem of blurring sharp edges in the image, and so we shall concentrate on smoothing methods which are edge preserving. They are based on the general idea that the average is computed only from those points in the neighborhood which have similar properties to the point being processed.

Local image smoothing can effectively eliminate impulse noise or degradations appearing as thin stripes, but does not work if degradations are large blobs or thick stripes

Averaging, statistical principles of noise suppression :

Assume that the noise value v at each pixel is an independent random variable with zero mean and standard deviation a. We might capture the same static scene under the same conditions n times. From each captured image a particular pixel value gi, i = 1,..., n is selected; the index i indicates to which image the pixel value gi belongs. An estimate of the correct value can be obtained as an average of these values, with corresponding noise values

The second term here describes the effect of the noise, which is again a random value with zero mean and standard deviation a/y/n. Thus, if n images of the same scene are available, smoothing can be accomplished without blurring the image by This reasoning has roots in

statistics which aims at estimating the most probable value from some population: a random sample is taken from the population and the corresponding sample mean value is calculated. If random samples were repeatedly selected and their sample mean values were calculated, we would obtain a distribution of sample mean values. **This distribution of sample means has some useful properties:**

• The mean value of the distribution of sample mean values is equal to the mean value of the population.

• The distribution of sample mean values has smaller variance than the original population. Suppose n samples are selected, and the standard deviation of the population is cr. Then the standard deviation of the distribution of sample mean values is cr/y/n.

• If the original distribution is normal (Gaussian) then the distribution of sample mean values is also normal. In fact, no matter what the shape o the original distribution, the distribution of sample means converges to a normal distribution. This is the central limit theorem, which is one reason why the normal distribution plays such a critical role in statistics.

• From the practical point of view, it is important that not too many random selections have to be made. The central limit theorem tell us the distribution of sample mean values without the need to create them. In statistics, usually about 30 samples are considered the lowest limit of the necessary number of observations. The degree of belief in the parameter describing the population (we have only considered the mean so far) is expressed by a confidence interval.

In many cases only one image corrupted by noise is available, and averaging is then realized in a local neighborhood. Results are acceptable if the noise is smaller in size than the smallest objects of interest in the image, but blurring of edges is a serious disadvantage. In the case of smoothing within a single image, one has to assume that there are no changes in the gray-levels of the underlying image data. This assumption is clearly violated at locations of image edges, and edge blurring is a direct consequence of violating the assumptions. Averaging is a special case of discrete convolution

The significance of the pixel in the center of the convolution mask h or its 4-neighbors is sometimes increased, as it better approximates the properties of noise with a Gaussian probability distribution . There are two commonly used smoothing filters whose coefficients gradually decrease to have the values close to zero at the limit of the and the mask coefficients are normalized to have a unit sum. The Butterworth filter will be explained in Section 5.3.8 dealing with the local pre-processing in the frequency domain.

$$h(i,j) = 0.5\,\frac{\delta(i,j)}{\sum_{(m,n)\in\mathcal{O}}\delta(i,j)}$$

There is an important special case of convolution filters called separable filters. Separability in 2D means that the convolution kernel can be factorized as a product of two one-dimensional vectors, and theory provides a clue as to which convolution masks are separable. Every convolution mask with rank one is separable.

Averaging using a rotating mask

Averaging using a rotating mask is a non-linear smoothing method that avoids edge blurring by searching for the homogeneous part of the current pixel neighborhood and the resulting image is

in fact sharpened [Nagao and Matsuyama, 1980]. The brightness average is calculated only within this region; a brightness dispersion σ^2 is used as the region homogeneity measure.

Brightness dispersion σ^2 is calculated as:

$$\sigma^2 = \frac{1}{n}\left\{\sum_{(i,j)\in R}\left[g(i,j) - \frac{1}{n}\sum_{(i,j)\in R} g(i,j)\right]^2\right\}$$

Let n be the number of pixels in a region R and g be the Median filtering.

In probability theory, the median divides the higher half of a probability distribution from the lower half. For a random variable x, the median M is the value for which the probability of the outcome x < M is 0.5.

$$\sigma^2 = \frac{1}{n}\sum_{(i,j)\in R}\left\{[g(i,j)]^2 - 2\,g(i,j)\frac{\sum_{(i,j)\in R} g(i,j)}{n} + \left[\frac{\sum_{(i,j)\in R} g(i,j)}{n}\right]^2\right\}$$

$$= \frac{1}{n}\left\{\sum_{(i,j)\in R}[g(i,j)]^2 - 2\frac{\left[\sum_{(i,j)\in R} g(i,j)\right]^2}{n} + n\left[\frac{\sum_{(i,j)\in R} g(i,j)}{n}\right]^2\right\}$$

$$= \frac{1}{n}\left\{\sum_{(i,j)\in R}[g(i,j)]^2 - \frac{\left[\sum_{(i,j)\in R} g(i,j)\right]^2}{n}\right\}$$

The median of a finite list of real numbers can be found by ordering the values and selecting the middle one. Lists are often constructed to be odd in length to secure uniqueness.

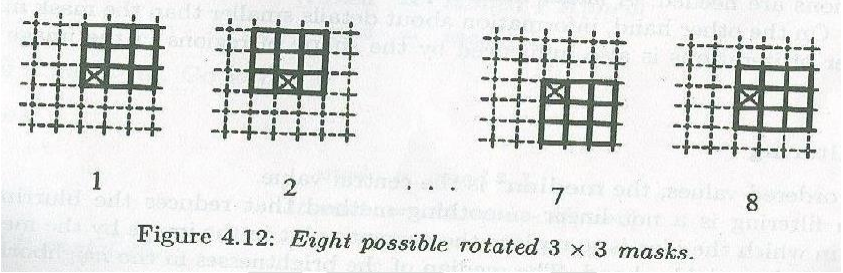


Figure 4.12: *Eight possible rotated 3 × 3 masks.*

Median filtering is a non-linear smoothing method that reduces the blurring of edges [Tyan, 1981], in which the idea is to replace the current point in the image by the median of the brightnesses in its neighborhood. The median of the brightnesses in the neighborhood is not affected by individual noise spikes and so median smoothing eliminates impulse noise quite well.

Algorithm: Efficient median filtering

1. Set ra n (We would always avoid unnecessary floating point operations: if ra and n are both odd, round t.)
2. Position the window at the beginning of a new row, and sort its contents. Construct a histogram H of the window pixels, determine the median ra, and record nm, the number of pixels with intensity less than or equal to ra.
3. For each pixel p in the leftmost column of intensity pg, perform
4. Move the window one column right. For each pixel p in the rightmost column of

intensity pg, perform
5. If nm = t then go to (8).
6. If nm > t then go to G).
Repeat
m = m 4- 1,
nm = nm + H[m] ,
until nm > t. Go to (8).
7. (We have nm > t, if here). Repeat
rim = rim- H[m] ,
m = m — 1,
until nm < ?.
8. If the right-hand column of the window is not at the right-hand edge of the image, go to C).
9. If the bottom row of the window is not at the bottom of the image, go to B).
The main disadvantage of median filtering in a rectangular neighborhood is its damaging of thin lines and sharp corners in the image—this can be avoided if another shape of neighborhood is used.

Median smoothing is a special instance of more general rank filtering techniques, the idea of which is to order pixels in some neighborhood into a sequence. The results of pre-processing are some statistics over this sequence, of which the median is one possibility. Another variant is the maximum or the minimum values of the sequence. This defines generalizations of dilation and erosion operators (Chapter 13) in images with more brightness values.

A similar generalization of median techniques is their method is called order statistics (OS) filtering. Values in the neighborhood are again ordered into sequence, and a new value is given as a linear combination of the values of Non-linear mean filter .
If the weights o,(i,j) are constant, the filter is called homomorphic. Some homo- morphic filters used in image processing are
• Arithmetic mean, u(g) = #,
• Harmonic mean, u(g) = l/#,
• Geometric mean, u(g) = log#.

Edge detectors

Edge detectors are a collection of very important local image pre-processing methods used to locate changes in the intensity function; edges are pixels where this function (brightness) changes abruptly. Neurological and psychophysical research suggests that locations in the image in which the function value changes abruptly are important for image perception. Edges are to a certain degree invariant to changes of illumination and viewpoint.

If only edge elements with strong magnitude (edgels) are considered, such information often suffices for image understanding. The positive effect of such a process is that it leads to significant reduction of image data. Nevertheless such a data reduction does not undermine understanding the content of the image (interpretation) in many cases. Edge detection provides

appropriate generalization of the image data. For instance, painters of line drawings perform such a generalization.

An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of that pixel. It is a vector variable with two components, magnitude and direction. The edge magnitude is the magnitude of the gradient, and the edge direction <j> is rotated with respect to the gradient direction '0 by —90°. The gradient direction gives the direction of maximum growth of the function. e.g., from black f(i,j) = 0 to white f(i,j) = 255. The orientation 0° points east.

Edges are often used in image analysis for finding region boundaries. Provided that the region has homogeneous brightness, its boundary is at the pixels where the image function varies and so in the ideal case without noise consists of pixels with high edge magnitude. It can be seen that the boundary and its parts (edges) are perpendicular to the direction of the gradient. Image sharpening has the objective of making edges steeper- the sharpened image is intended to be observed by a human.

sharpening using a Laplacian.

Image sharpening can be interpreted in the frequency domain as well. The derivative of the harmonic function sm(nx) is n cos(nx); thus the higher the frequency, the higher the magnitude of its derivative. This is another explanation of why gradient operators enhance edges. A similar image sharpening technique to that given in equation E.35), called unsharp masking, is often used in printing industry applications [Jain, 1989]. A signal proportional to an unsharp image (e.g., heavily blurred by a smoothing operator) is subtracted from the original image.

Gradient operators as a measure of edge sheerness can be divided into three categories:

1. Operators approximating derivatives of the image function using differences. Some of them are rotationally invariant (e.g., Laplacian) and thus are computed from one convolution mask only. Others, which approximate first derivatives, use several masks. The orientation is estimated on the basis of the best matching of several simple patterns.
2. Operators based on the zero-crossings of the image function second derivative (e.g., Marr-Hildreth or Canny edge detectors).
3. Operators which attempt to match an image function to a parametric model of edges.

The remainder of this section will consider some of the many operators which fall into the first category, and the next section will consider the second.

Edge detection represents an extremely important step facilitating higher-level image analysis and therefore remains an area of active research, with new approaches continually being developed.

Roberts operator

The Roberts operator is one of the oldest operators [Roberts, 1965]. It is very easy to compute as it uses only a 2 x 2 neighborhood of the current pixel. Its convolution masks are

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

hi = L A , h2 = _! 0 ' E-38)
so the magnitude of the edge is computed as

$$|g(i,j) - g(i+1,j+1)| + |g(i,j+1) - g(i+1,j)|$$

The primary disadvantage of the Roberts operator is its high sensitivity to noise, because very few pixels are used to approximate the gradient.

Laplace operator

The Laplace operator ∇^2 is a very popular operator approximating the second derivative which gives the gradient magnitude only. The Laplacian, equation E.34), is approximated in digital images by a convolution sum. A 3 x 3 mask h is often used; for 4-neighborhoods and 8-neighborhoods it is defined as E.40)

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

A Laplacian operator with stressed significance of the central pixel or its neighborhood is sometimes used. In this approximation it loses invariance to rotation

$$h = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix} \qquad h = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

The Laplacian operator has a disadvantage—it responds doubly to some edges in the image.

Prewitt operator

The Prewitt operator, similarly to the Sobel, Kirsch, Robinson (as discussed later), and some other operators, approximates the first derivative. The gradient is estimated in eight (for a 3 x 3 convolution mask) possible directions, and the convolution result of greatest magnitude indicates the gradient direction. Larger masks are possible.

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Operators approximating the first derivative of an image function are sometimes called compass operators because of their ability to determine gradient direction. We present only the first three 3x3 masks for each operator; the others can be created by simple rotation.

H1= $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

$$h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

edge strength is calculated as

$$\sqrt{x^2+y^2} \quad \text{or} \quad |x|+|y|$$

The direction of the gradient is given by the mask giving maximal response. This is also the case for all the following operators approximating the first derivative.

Sobel operator

The Sobel operator is often used as a simple detector of horizontality and verticality of edges, in which case only masks hi and h% are used. If the hi response is y and the h% response x. we might then derive edge strength (magnitude) as yjx2 + y2 or \x\ + \y\ E.44) and direction as arctan(y/a;).

Robinson operator :

$$h_1=\begin{bmatrix}1&1&1\\1&-2&1\\-1&-1&-1\end{bmatrix}\quad h_2=\begin{bmatrix}1&1&1\\-1&-2&1\\-1&-1&1\end{bmatrix}\quad h_3=\begin{bmatrix}-1&1&1\\-1&-2&1\\-1&1&1\end{bmatrix}$$

Kirsch Operator

$$h_1=\begin{bmatrix}3&3&3\\3&0&3\\-5&-5&-5\end{bmatrix}\quad h_2=\begin{bmatrix}3&3&3\\-5&0&3\\-5&-5&3\end{bmatrix}\quad h_3=\begin{bmatrix}-5&3&3\\-5&0&3\\-5&3&3\end{bmatrix}$$

Zero-crossings of the second derivative

Edge detection techniques existing at that time (e.g., the Kirsch, Sobel, and Pratt operators) were based on convolution in very small neighborhoods and worked well only for specific images. The main disadvantage of these edge detectors is their dependence on the size of the object and sensitivity to noise.

An edge detection technique based on the zero-crossings of the second derivative (in its original form, the Marr-Hildreth edge detector [Marr and Hildreth, 1980] or the same paper in a more recent collection, [Marr and Hildreth, 1991]) explores the fact that a step edge corresponds to an abrupt change in the image function. The first derivative of the image function should have an extremum at the position corresponding to the edge in the image, and so the second derivative should be zero at the same position; however, it is much easier and more precise to find a zero-crossing position than an extremum. In Figure 5.22 this principle is illustrated in ID for the sake of simplicity. Figure 5.22a shows step edge profiles of the original image function with two different slopes. Figure 5.22b depicts the first derivative of the image function, and Figure 5.22c illustrates the second derivative; notice that this crosses the zero level at the same position as the edge.

One possibility is to smooth an image first (to reduce noise) and then compute second derivatives. When choosing a smoothing filter, there are two criteria that should be fulfilled .First, the filter should be smooth and roughly band limited in the frequency domain to reduce the possible number of frequencies at which function changes can take place. Second, the constraint of spatial localization requires the response of a filter to be from nearby points in the image. These two criteria are conflicting, but they can be optimized simultaneously using a Gaussian distribution. In practice, one has to be more precise about what is meant by the

localization performance of an operator, and the Gaussian may turn out to be sub-optimal. We shall consider this in the next section.

The 2D Gaussian smoothing operator G(x, y) (also called a Gaussian filter, or simply a Gaussian) is given by

G(x,y) = e-(x2+y2V2°\ E.47)

where x, y are the image co-ordinates and a is a standard deviation of the associated probability distribution. Sometimes this is presented with a normalizing factor G(x, y) = -J_c-<*2+va>/2*a or G(x. y) = -L-e-l*2+v*V*°2 .

The standard deviation a is the only parameter of the Gaussian filter—it is proportional to the size of the neighborhood on which the filter operates. Pixels more distant from the center of the operator have smaller influence, and pixels farther than 3<r from the center have negligible influence. Our goal is to obtain a second derivative of a smoothed 2D function f(x,y). We have already seen that the Laplace operator V2 gives the second derivative, and is non- directional (isotropic). Consider then the Laplacian of an image f(x,y) smoothed by a Gaussian (expressed using a convolution *). The operation is abbreviated by some authors as LoG, from Laplacian of Gaussian

Canny edge detection

Canny proposed an approach to edge detection [Canny, 1983; Brady, 1984; Canny, 1986] that is optimal for step edges corrupted by white noise. The optimality of the detector is related to three criteria.

• The detection criterion expresses the fact that important edges should not be missed and that there should be no spurious responses.

• The localization criterion says that the distance between the actual and located position of the edge should be minimal.

• The one response criterion minimizes multiple responses to a single edge. This is partly covered by the first criterion, since when there are two responses to a single edge, one of them should be considered as false. This third criterion solves the problem of an edge corrupted by noise and works against non-smooth edge operators .

Canny's derivation of a new edge detector is based on several ideas.

1. The edge detector was expressed for a ID signal and the first two optimality criteria. A closed-form solution was found using the calculus of variations.

2. If the third criterion (multiple responses) is added, the best solution may be found by numerical optimization. The resulting filter can be approximated effectively with error less than 20% by the first derivative of a Gaussian smoothing filter with standard deviation a [Canny, 1986]; the reason for doing this is the existence of an effective implementation.

3. The detector is then generalized to two dimensions. A step edge is given by its position, orientation, and possibly magnitude (strength). It can be shown that convolving an image with a symmetric 2D Gaussian and then differentiating in the direction of the gradient (perpendicular to the edge direction) forms a simple and effective directional operator (recall that

the Marr-Hildreth zero-crossing operator does not give information about edge direction, as it uses a Laplacian filter).

Suppose G is a 2D Gaussian [equation E.47)] and assume we wish to convolve the image with an operator Gn which is a first derivative of G in the direction n

The direction n should be oriented perpendicular to the edge. Although this direction is not known in advance, a robust estimate of it based on the smoothed gradient direction is available. If / is the image, the normal to the edge n is estimated as

The edge location is then at the local maximum of the image / convolved with the operator Grl in the direction n ^G„*/ = 0. E.59) Substituting in equation E.59) for Gn from equation E.57), we get ^G*f = 0. E.60) This equation E.60) illustrates how to find local maxima in the direction perpendicular to the edge; this operation is often referred to as non-maximal suppression (see also Algorithm 6.4).

As the convolution and derivative are associative operations in equation E.60), we can first convolve an image /' with a symmetric Gaussian G and then compute the directional second-derivative using an estimate of the direction n computed according to equation E.58). The strength of the edge (magnitude of the gradient of the image intensity function /) is measured as |G„*/| = |V(G*/)|. E.61)

A different generalization of this optimal detector into two dimensions was proposed by Spacek [Spacek, 1986], and the problem of edge localization is revisited in [Tagare and deFigueiredo, 1990]. 4. Spurious responses to the single edge caused by noise usually create a 'streaking' problem that is very common in edge detection in general. The output of an edge detector is usually thresholded to decide which edges are significant, and streaking means the breaking up of the edge contour caused by the operator fluctuating above and below the threshold. Streaking can be eliminated by thresholding with hysteresis. If any edge response is above a high threshold, those pixels constitute definite edge output of the detector for a particular scale. Individual weak responses usually correspond to noise, but if these points are connected to any of the pixels with strong responses, they are more likely to be actual edges in the image. Such connected pixels are treated as edge pixels if their response is above a low threshold.

The low and high thresholds are set according to an estimated signal-to-noise ratio

5. The correct scale for the operator depends on the objects contained in the image. The solution to this unknown is to use multiple scales and aggregate information from them. Different scales for the Canny detector are represented by different standard deviations a of the Gaussians. There may be several scales of operators that give significant responses to edges (i.e., signal-to-noise ratio above the threshold); in this case the operator with the smallest scale is chosen, as it gives the best localization of the edge.

Canny proposed a feature synthesis approach. All significant edges from the operator with the smallest scale are marked first, and the edges of a hypothetical operator with larger a are synthesized from them (i.e., a prediction is made of how the large a should perform on the evidence gleaned from the smaller. Then the synthesized edge response is compared with the actual edge response for larger a. Additional edges are marked only if they have a significantly

stronger response than that predicted from synthetic output. This procedure may be repeated for a sequence of scales, a cumulative edge map being built by adding those edges that were not identified at smaller scales.

Algorithm 5.4: Canny edge detector

1. Convolve an image / with a Gaussian of scale a.
2. Estimate local edge normal directions n using equation E.58) for each pixel inthe image.
3. Find the location of the edges using equation E.60) (non-maximal suppression).
4. Compute the magnitude of the edge using equation E.61).
5. Threshold edges in the image with hysteresis (Algorithm 6.5) to eliminate spurious responses.
6. Repeat steps A) through E) for ascending values of the standard deviation a.
7. Aggregate the final information about edges at multiple scale using the 'feature synthesis' approach.

Parametric edge models

Parametric models are based on the idea that the discrete image intensity function can be considered a sampled and noisy approximation of the underlying continuous or piecewise continuous image intensity function [Nevatia, 1977].

While the continuous image intensity function is not known, it can be estimated from the available discrete image intensity function and image properties can be determined from this continuous estimate, possibly with sub-pixel precision. Since modeling the image as a single continuous function leads to high-order intensity functions in x and y, it is practically impossible to represent image intensities using a single continuous function. Instead, piecewise continuous function estimates called facets are used to represent (a neighborhood of) each image pixel. Such image representation is called the facet .

The intensity function in a pixel neighborhood can be estimated using models of different complexity. The simplest one is the flat facet model that uses piecewise constants and each pixel neighborhood is represented by a flat function of constant intensity. The sloped model uses piecewise linear functions forming a sloped plane fitted to the image intensities in the pixel neighborhood. Quadratic and bi-cubic facet models employ correspondingly more complex functions.

A thorough treatment of facet models and their modifications for peak noise removal, segmentation into constant-gray-level regions, determination of statistically significant edges, gradient edge detection, directional second-derivative zero-crossing edge detection.

To provide an edge detection example, consider a bi-cubic facet model the parameters of which are estimated from a pixel neighborhood [the co-ordinates of the central pixel are @,0)]. To determine the model parameters, a least-squares method with singular-value decomposition (see Section 3.2.9) may be used. Alternatively, when using a 5 x 5 pixel neighborhood, coefficients Ci can be computed directly using a set of ten 5x5 kernels.

Once the facet model parameters are available for each image pixel, edges can be detected as extrema of the first directional derivative and/or zero-crossings of the second

directional derivative of the local continuous facet model functions. Edge detectors based on parametric models describe edges more precisely than convolution-based edge detectors. Additionally, they carry the potential for sub-pixel edge localization.

Edges in multi-spectral images

One pixel in a multi-spectral image is described by an n-dimensional vector, and brightness values in n spectral bands are the vector components.

The first is to detect edges separately in individual image spectral components using the ordinary local gradient operators. Individual images of edges can be combined to get the resulting image, with the value corresponding to edge magnitude and direction being the maximal edge value from all spectral components. A linear combination of edge spectral components can also be used, and other combination techniques. A second possibility is to use the brightness difference of the same pixel in two different spectral components. This is a very informative feature for classification based on properties of the individual pixel.

A third possibility is to create a multi-spectral edge detector which uses brightness information from all n spectral bands; this approach is also applicable to multi-dimensional images forming three- or higher-dimensional data volumes. This multi-spectral edge detector gives very good results on remotely sensed images.

Some basic examples of spatial filtering are linear low-pass, high-pass, and bandpass frequency filters.

A low-pass filter is defined by a frequency transfer function H(u, v) with small values at points located far from the co-ordinate origin in the frequency domain (that is, small transfer values for high spatial frequencies) and large values at points close to the origin (large transfer values for low spatial frequencies). It preserves low spatial frequencies and suppresses high spatial frequencies, and has behavior similar to smoothing by standard averaging—it blurs sharp edges.

• A high-pass filter is defined by small transfer function values located around the frequency co-ordinate system origin, and larger values outside this area—larger transfer coefficients for higher frequencies (Figure 5.26b).

• Band-pass filters, which select frequencies in a certain range for enhancement, are constructed in a similar way, and also filters with directional response, etc.

The most common image enhancement problems include noise suppression, edge enhancement, and removal of noise which is structured in the frequency spectrum. Noise represents a high-frequency image component, and to suppress it, the magnitudes of image frequencies of the noise must be decreased. This can be achieved by applying a low-pass filter as shown in Figure 5.27, which demonstrates the principles of frequency filtering on Fourier image spectra; the original image spectrum is multiplied by the filter spectrum and a low-frequency image spectrum results. Unfortunately, as a result of noise suppression, all high-frequency phenomena are suppressed, including high frequencies that are not related to noise (sharp edges, lines, etc.). Low-pass filtering results in a blurred image.

To remove noise which is structured in the frequency domain, the filter design must include a priori knowledge about the noise properties. This knowledge may be acquired either

from the image data or from the corrupted image Fourier spectrum, where the structured noise usually causes notable peaks.

The vertical periodic noise lines in the original image are transformed into frequency spectrum peaks after the transform. To remove these frequencies from an image, a filter was designed which suppresses the periodic noise in the image, which is visible as white circular areas.

There are several filters which prove useful for filtering in the frequency domain: two important representatives of them are the Gaussian and Butterworth filters. Choose an isotropic filter for simplicity, D(u,v) — D(r) — y/u2 -\-v2. Let D0 be a parameter of the filter called the cut-off frequency. For the Gaussian Dq coincides with the dispersion a2. The Fourier spectrum of a low-pass Gaussian filter Cqow is

Line detection by local pre-processing operators

Several other local operations exist which are used for different purposes as, e.g., line finding, line thinning, and line filling operators. The second group of operators finds 'interest points' or 'locations of interest' in the image. There is yet another class of local nonlinear operators, mathematical morphology techniques.

Recall that one of the reasons why edges are being detected is that they bear a lot of information about underlying objects in the scene. Taking just edge elements instead of all pixels reduces the amount of data which has to be processed. The edge detector is a general tool which does not depend on the content of the particular image. The detected edges are to some degree robust as they do not depend much on small changes in illumination, viewpoint change, etc. It is interesting to seek yet richer features which can be reliably detected in the image and which can outperform simple edge detectors in some classes of applications. Line detectors and corner detectors are some such. Line detectors are used to detect linear objects such as dimension lines in engineering drawings or railways or roads in satellite images. Corner detectors and other interest point-like detectors are used mainly to register two or more images one to the other (e.g, in stereo vision, motion analysis, panorama stitching, object recognition from images) or to index the image or dominant objects in it to an image database.

Line finding operators aim to find very thin curves in the image; it is assumed that curves do not bend sharply. Such curves and straight lines are called lines for the purpose of describing this technique. If a cross section perpendicular in direction to the tangent of a line is examined, we get a roof profile (see Figure 5.19) when examining edges. We assume that the width of the lines is approximately one or two pixels. The presence of a line may be detected by local convolution of the image with convolution kernels which serve as line patterns [Vernon, 1991; Petrou, 1993]. The simplest collection of four such patterns of size 3 x 3 is able to detect lines rotated modulo the angle 45°.

A similar principle can be applied to bigger masks. The case of 5 x 5 masks is common. Such line detectors sometimes produce more lines than needed, and other non-linear constraints may be added to reduce this number. More sophisticated approaches determine lines in images as ridges and ravines using the facet model. Line detection is frequently used in remote sensing and in document processing.

Local information about edges is the basis of a class of image segmentation techniques that are discussed in Chapter 6. Edges which are likely to belong to object boundaries are usually found by simple thresholding of the edge magnitude. Thresholded edges are usually wider than one pixel, and line thinning techniques may give a better result. One line thinning method uses knowledge about edge orientation and in this case edges are thinned before thresholding. Edge magnitudes and directions provided by some gradient operator are used as input, and the edge magnitudes of two neighboring pixels perpendicular to the edge direction are examined for each pixel in the image. If at leastone of these pixels has edge magnitude higher than the edge magnitude of the examined pixel, then the edge magnitude of the examined pixel is assigned a zero value. This technique is called non-maximal suppression and is similar to the idea mentioned in conjunction with the Canny edge detector.

Detection of corners:

In many cases it is advantageous to find pairs of corresponding points in two similar images; when considering geometric transforms. Knowing the position of corresponding points enables the estimation of parameters describing geometric transforms from live data. The same transformation usually holds for almost all pixels of the image. The necessary number of corresponding pairs of points is usually rather small and is equal to the number of parameters of the transform. In general, all possible pairs of pixels should be examined to solve this correspondence problem, and this is very computationally expensive. If two images have n pixels each, the complexity is 0(n2). This process might be simplified if the The first are corner detectors and the second one are the more rich structures called MSERs (Maximally Stable Extremal Regions).

Corners in images can be located using local detectors; input to the corner detector is the gray-level image, and output is the image in which values are proportional to the likelihood that the pixel is a corner. Interest points are obtained by thresholding the result of the corner detector. Corners serve better than lines when the correspondence problem is to be solved.

This is due to the aperture problem. Assume a moving line is seen through a small aperture. In such a case, only the motion vector perpendicular to the line can be observed.
The component collinear with the line remains invisible. The situation is better with corners. Edge detectors themselves are not stable at corners. This is natural as the gradient at the tip of the corner is ambiguous. This is illustrated in Figure 5.35 in which a triangle with sharp corner is shown. However, near the corner there is a discontinuity in the gradient direction. This observation is used in corner detectors.

The corner in the image can also be defined as a pixel in its small neighborhood where are two dominant and different edge directions. This definition is not precise as an isolated point of local intensity maximum or minimum, line endings, or an abrupt change in the curvature of a curve gives a response similar to a corner. Nevertheless, such detectors are named corner detectors in the literature and are widely used. If corners have to be detected then some additional constraints have to be applied.

Corner detectors are not usually very robust. This deficiency is overcome either by manual expert supervision or large redundancies introduced to prevent the effect of individual

errors from dominating the task. The latter means that many more corners are detected in two or more images than are needed for estimating a transformation sought between these images.

Image restoration

Pre-processing methods that aim to suppress degradation using knowledge about its nature are called image restoration. Most image restoration methods are based on convolution applied globally to the whole image. Degradation of images can have many causes: defects of optical lenses, non-linearity of the electro-optical sensor, graininess of the film material, relative motion between an object and camera, wrong focus, atmospheric turbulence in remote sensing or astronomy, scanning of photographs, etc. The objective of image restoration is to reconstruct the original image from its degraded version.

Image restoration techniques can be classified into two groups: deterministic and stochastic. Deterministic methods are applicable to images with little noise and a known degradation function. The original image is obtained from the degraded one by a transformation inverse to the degradation. Stochastic techniques try to find the best restoration according to a particular stochastic criterion, e.g., a least-squares method. In some cases the degradation transformation must be estimated first. It is advantageous to know the degradation function explicitly. The better this knowledge is, the better are the results of the restoration. There are three typical degradations with a simple function: relative constant speed movement of the object with respect to the camera, wrong lens focus, and atmospheric turbulence. In most practical cases, there is insufficient knowledge about the degradation, and it must be estimated and modeled. The estimation can be classified into two groups according to the information available: a priori and a posteriori. If degradation type and/or parameters need to be estimated, this step is the most crucial one, being responsible for image restoration success or failure. It is also the most difficult part of image restoration. A priori knowledge about degradation is either known in advance or can be obtained before restoration. For example, if it is known in advance that the image was degraded by relative motion of an object with respect to the sensor, then the modeling determines only the speed and direction of the motion. An example of the second case is an attempt to estimate parameters of a capturing device such as a TV camera or digitizer, whose degradation remains unchanged over a period of time and can be modeled by studying a known sample image and its degraded version.

A posteriori knowledge is that obtained by analyzing the degraded image.

Inverse filteration

$$G(u,v) = F(u,v)\,H(u,v) + N(u,v)$$

$$F(u,v) = G(u,v)\,H^{-1}(u,v) - N(u,v)\,H^{-1}(u,v)$$

this equation works well for images which are corrupted by noise.

A degraded image g can arise from the original image / by a process which can be expressed as

$$e^2 = \mathcal{E}\left\{[f(i,j) - \hat{f}(i,j)]^2\right\}$$

$$\hat{F}(u,v) = H_W(u,v)\, G(u,v)$$

$$H_W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + [S_{vv}(u,v)/S_{ff}(u,v)]}$$

where s is some non-linear function and v describes the noise. The degradation is very often simplified by neglecting the non-linearity and by assuming that the function h is invariant with respect to position in the image. Degradation can be then expressed as
convolution: and the noise is not significant, then image restoration equates to inverse convolution (also called deconvolution). If noise is not negligible, then the inverse convolution is solved as an overdetermined system of linear equations. Methods based on minimization of the least square error such as Wiener filtering (off-line) or Kalman filtering (recursive, on-line)

1 Degradations that are easy to restore

hese degradations can be expressed by convolution, the Fourier transform H of the convolution function is used. In the absence of noise, the relationship between the Fourier representations F, G, H of the undegraded image, the degraded image g, and the degradation convolution kernel h, respectively, is G = HF. E.79)

Therefore, not considering image noise v. knowledge of the degradation function fully facilitates image restoration by inverse convolution (Section 5.4.2). We first discuss several degradation functions.

Relative motion of the camera and object

Assume an image is acquired with a camera with a mechanical shutter. Relative motion of the camera and the photographed object during the shutter open time T causes smoothing of the object in the image. Suppose V is the constant speed in the direction of the x axis;the Fourier transform H(u>v) of the degradation caused in time T is given by

Wrong lens focus

Image smoothing caused by imperfect focus of a thin lens can be described by the following function:

H{UjV) = ilMi E.81)

where J\ is the Bessel function of the first order, r2 = u2 + v2, and a is the displacement—
the model is not space invariant.

Atmospheric turbulence

Atmospheric turbulence is degradation that needs to be restored in remote sensing and astronomy. It is caused by temperature non-homogeneity in the atmosphere that deviates passing light rays. The mathematical model is derived in [Hufnagel and Stanley, 1964] and is expressed as (u,v) = e-«u2+v2f'\ E.82)

where c is a constant that depends on the type of turbulence which is usually found experimentally. The power 5/6 is sometimes replaced by 1. 5.4.2 Inverse filtration An obvious approach to image restoration is inverse filtration based on properties of the Fourier transforms [Sondhi, 1972; Andrews and Hunt, 1977; Rosenfeld and Kak, 1982]. Inverse filtering uses the assumption that degradation was caused by a linear function h(i.j) (cf. equation E.78)) and considers the additive noise v as another source of degradation. It is further assumed that v is independent of the signal. After applying the Fourier transform to equation E.78), we get G(u, v) = F(u, v) H(u, v) + N(u, v). E.83)

The degradation can be eliminated using the restoration filter with a transfer function that is inverse to the degradation h. The Fourier transform of the inverse filter is then expressed as H~1(u,v). We derive the original undegraded image F (its Fourier transform to be exact) from its degraded version G [equation E.83)], as follows

F(u,v) = G(u,v)H-1(u,v) - N{u,v)H-\u1v). E.84)

This equation shows that inverse filtration works well for images that are not corrupted by noise [not considering possible computational problems if H(u, v) gets close to zero at some location of the u, v space--fortunately, such locations can be neglected without perceivable effect on the restoration result]. However, if noise is present, several problems arise. First, the noise influence may become significant for frequencies where H(u, v) has small magnitude. This situation usually corresponds to high frequencies u,v. In reality, H{u, v) usually decreases in magnitude much more rapidly than N(u, v) and thus the noise effect may dominate the entire restoration result. Limiting the restoration to a small neighborhood of the u,v origin in which H(u,v) is sufficiently large overcomes this problem, and the results are usually quite acceptable. The second problem deals with the spectrum of the noise itself—we usually do not have enough information about the noise to determine N(u,v) sufficiently well.

3 Wiener filtration

Restoration by the Wiener filter gives an estimate / of the original uncorrupted image f with minimal mean square error

e2 = s{(f(i,j)-f(hj)J} , E-85)

where ? denotes the mean operator.

If no constraints are applied to the solution of equation E.85), then an optimal estimate is the conditional mean value of the ideal image / under the condition g. This approach is complicated from the computational point of view. Moreover, the conditional probability density between the optimal image /and the corrupted image g is not usually known. The optimal estimate is in general a non-linear function of the image g. Minimization of equation E.85) is easy if the estimate is a linear combination of the values in the image g the estimate is then close (but not necessarily equal) to the theoretical optimum. The estimate is equal to the theoretical optimum only if the stochastic processes describing images, g, and the noise v are homogeneous, and their probability density is Gaussian. These conditions are not usually fulfilled for typical images.

Denote the Fourier transform of the Wiener filter by Hw- Then, the estimate F of the Fourier transform F of the original image / can be obtained as

$$\hat{F}(u,v) = H_W(u,v)\, G(u,v)$$

If Wiener filtration is used, the nature of degradation H and statistical parameters of the noise need to be known. Wiener filtration theory solves the problem of optimal a posteriori linear mean square estimates—all statistics (for example, power spectrum) should be available in advance. Note the term Sff(u.v) in equation E.87), which represents the spectrum of the undegraded image. This information may be difficult to obtain considering the goal of image restoration, to determine the undegraded image.

Note that the ideal inverse filter is a special case of the Wiener filter in which noise is absent, i.e., Svv = 0.

$$H_W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + [S_{vv}(u,v)/S_{ff}(u,v)]}$$

Restoration is illustrated in Figures 5.39 and 5.40. Figure 5.39a shows an image that was degraded by 5 pixels motion in the direction of the x axis, and Figure 5.39b shows the result of restoration where Wiener filtration was used. Figure 5.40a shows an image degraded by wrong focus and Figure 5.40b is the result of restoration using Wiener filtration. Despite its unquestionable power, Wiener filtration suffers several substantial limitations. First, the criterion of optimality is based on minimum mean square error and weights all errors equally, a mathematically fully acceptable criterion that unfortunately does not perform well if an image is restored for human viewing. The reason is that humans perceive the restoration errors more seriously in constant-gray-level areas and in bright regions, while they are much less sensitive to errors located in dark regions and in high-gradient areas. Second, spatially variant degradations cannot be restored using the standard Wiener filtration approach, and these degradations are common. Third, most images are highly non-stationary, containing large homogeneous areas separated by high-contrast edges. Wiener filtration cannot handle non-stationary signals and noise.

Digital Image Processing

Unit-2 Possible Questions

2 Marks Questions:

1. Define sampling with diagram
2. Define quantization with diagram
3. Write short notes on neighbors of a pixel
4. Mention the types of adjacency that can be adopted for any pixel in an image
5. Define a path and explain the formula for identifying length of a path and closed path
6. Define a region of a image with example
7. Define the boundary of the region with example
8. Write short notes on distance measures
9. Write notes on linear and nonlinear operations
10. Write short notes on image negatives with diagram
11. Write notes on power law transformation with diagram
12. Define histogram processing
13. Write short notes on histogram equalization
14. Write short notes on histogram matching
15. Write short notes on model for image degradation and restoration process with suitable diagram.

8 marks

1. Illustrate in detail some basic relation ships between pixels with example
2. Discuss in detail about piece wise linear transformation functions with suitable diagram
3. Discuss in detail about smoothing spatial filters with suitable example
4. Explain in detail about some important noise probability functions
5. Explain in detail about the various noise models available in digital image processing

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC ACT 1956)
M.sc Computer Science
DEPARTMENT OF CS, CA & IT
Third Semester(Digital Image processing)

S.no	Questions	OPT1	OPT2	OPT3	OPT4	Answer
1	_____ is a common name for operations with images at the lowest level of abstraction -- both input and output are intensity images.	Pre-processing	dual processing	post -processing	semi processing	Pre-processing
2	The aim of _____ is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.	post	pre-processing	pocessing	both a & c	pre-processing
3	_____ is one of the four categories of image pre-processing methods that is used for the calculation of a new pixel brightness	brightness transformations	pixel transformations	Pixel brightness transformations	image brightness transformation	Pixel brightness transformations
4	_____ permit the elimination of geometric distortion that occurs when an image is captured.	co-axial transformations	brightness transformations	pixel transformations	Geometric transformations	Geometric transformations
5	Several pixels in both images with known correspondence are used to derive the _____transformation	known	unknown	not known	well known	unknown

6	_____ is one of the image pre-processing methods that is used for the calculation of a new pixel contrast	Local neighborhood method	logical neighborhood method	programmable neighborhood method	neighborhood method	Local neighborhood method
7	_____ suppressing image degradation using knowledge about its nature	image	Image restoration	image data	image segmentation	Image restoration
8	_____ modify pixel brightness with regard to pixel position and the properties of a pixel itself.	image data	degradation	Brightness corrections	transformation	Brightness corrections
9	_____ modify pixel brightness without regard to pixel position.	transform	coefficients	Brightness	Gray scale transformations	Gray scale transformations
10	Uneven sensitivity of light sensors and _____ are the sources of degradation.	Uneven object illumination	image data	transformation	Even object illumination	Uneven object illumination
11	_____ can be suppressed by brightness correction	image data	trigger	Systematic degradation	system trigger	Systematic degradation
12	A _____ is a vector function T that maps the pixel (x,y) to a new position (x',y').	arbitrary point	transform	geometric transform	fourier transform	geometric transform
13	A geometric transform consists of two basic steps namely pixel co-ordinate transformation and _____	vector	Brightness	arbitrary point	interpolation	interpolation

14	The geometric transform is often approximated by the _____ for which 4 pairs of corresponding points are sufficient to find transformation coefficients $x' = a_0+a_1x+a_2y+a_3xy$ and $y' = b_0+b_1x + b_2y + b_3xy$	bilinear transformation	corresponding points	transform	interpolation	bilinear transformation
15	The geometric transform is often approximated by the _____ for which three pairs of corresponding points are sufficient to find the coefficients $x' = a_0+a_1x+a_2y$ and $y' = b_0+b_1x + b_2y$	geometric transformation	vector	pre-processing method	affine transformation	affine transformation
16	16. The _____ includes typical geometric transformations such as rotation, translation, scaling and skewing.	arbitrary point	Brightness	affine transformation	interpolation	affine transformation
17	Each pixel value in the output image raster can be obtained by _____ of some neighboring noninteger samples.	Brightness	vector	corresponding points	brightness interpolation	brightness interpolation
18	_____ assigns to the point (x,y) the brightness value of the nearest point g in the discrete raster $f_1(x,y) = g_8(\text{round}(x), \text{round}(y))$	pre-processing method	Nearest neighbor interpolation	arbitrary point	suppressing	Nearest neighbor interpolation
19	_____ explores four points neighboring the point (x,y), and assumes that the brightness function is linear in this neighborhood.	arbitrary point	Linear interpolation	discrete raster	all the above	Linear interpolation

20	_____ can cause a small decrease in resolution and blurring due to its averaging nature. But the problem of step like straight boundaries with the nearest neighborhood interpolation is reduced.	discrete raster	geometric transformation	suppressing	Linear interpolation	Linear interpolation
21	_____ improves the model of the brightness function by approximating it locally by a bicubic polynomial surface; sixteen neighboring points are used for interpolation.	geometric transformation	arbitrary point	Bi-cubic interpolation	interpolation	Bi-cubic interpolation
22	_____ does not suffer from the step-like boundary problem of nearest neighborhood interpolation, and copes with linear interpolation blurring as well.	pre-processing interpolation	Bi-cubic interpolation	discrete raster	Dual-cubic interpolation	Bi-cubic interpolation
23	_____ is often used in raster displays that enable zooming with respect to an arbitrary point and preserves fine details in the image very well.	interpolation	suppressing	Bi-cubic interpolation	brightness function	Bi-cubic interpolation
24	_____ is a pre-processing method that use a small neighborhood of a pixel in an input image to get a new brightness value in the output image.	global pre-processing	private pre-processing	public pre-processing	Local pre-processing	Local pre-processing
25	Pre-processing methods use a small neighborhood of a pixel in an input image to get a new brightness value in the output image. Such pre-processing operations are called also _____.	laser	transformation	filtration	interpolation	filtration

26	Local pre-processing methods can be divided into the two groups according to the goal of the processing namely smoothing and _____.	interpolation	Gradient operators	brightness function	interpolation	Gradient operators
27	_____ aims to suppress noise or other small fluctuations in the image and it is equivalent to the suppression of high frequencies in the frequency domain	discrete function	Smoothing	suppressing	Local pre-processing	Smoothing
28	_____ are based on local derivatives of the image function and have a similar effect as suppressing low frequencies in the frequency domain.	frequency operators	rational operators	linear operators	Gradient operators	Gradient operators
29	Noise is often high frequency in nature; unfortunately, if a _____ is applied to an image the noise level increases simultaneously.	combination	linear combination	gradient operator	Local pre-processing	gradient operator
30	Some pre-processing algorithms solve this problem and permit smoothing and _____ simultaneously.	suppressing	edge enhancement	combination	interpolation	edge enhancement
31	_____ calculate the resulting value in the output image pixel g(i,j) as a linear combination of brightnesses in a local neighborhood of the pixel f(i,j) in the input image.	Linear operations	pixel neighborhood	brightnesses	Nonlinear operations	Linear operations
32	Smoothing poses the problem of blurring sharp edges in the image, and so we have to concentrate on smoothing methods which are _____.	edge preserving	output image	linear combination	Nonlinear operations	edge preserving

33	_____ can effectively eliminate impulsive noise or degradations appearing as thin stripes, but does not work if degradations are large blobs or thick stripes.	pixel neighborhood	image smoothing	Local image smoothing	Local pre-processing	Local image smoothing
34	Averaging with _____ try to avoid blurring by averaging only those pixels which satisfy some criterion, the aim being to prevent involving pixels that are part of a separate feature.	output image	current pixel	limited data validity	resulting image	limited data validity
35	Averaging according _____ to is applied when brightness change within a region is usually smaller than between neighboring regions. The convolution mask is calculated at each pixel according to the inverse gradient.	reverse gradient	resulting image	inverse gradient	all the above	inverse gradient
36	Averaging using a _____ avoids edge blurring by searching for the homogeneous part of the current pixel neighborhood and the resulting image is in fact sharpened	pixel mask	filtering mask	edge mask	rotating mask	rotating mask
37	_____ reduces blurring of edges , not affected by individual noise spikes, eliminates impulsive noise quite well and can be applied iteratively.	Median filtering	discrete function	both a & b	Mean filtering	Median filtering

38	The main disadvantage of _____ in a rectangular neighborhood is its damaging of thin lines and sharp corners in the image and this can be avoided if another shape of neighborhood is used.	individual pixel	median filtering	resulting image	both a & c	median filtering
39	Edges are pixels where brightness changes abruptly and _____ are used to locate sharp changes in the intensity function	Edge detectors	masking	filter	all the above	Edge detectors
40	An _____is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of the pixel. It is a vector variable with two components, magnitude of the gradient and direction.	node	edge	both a & b	pixel	edge
41	The Marr-Hildreth edge detection technique, based on the _____of the second derivative explores the fact that a step edge corresponds to an abrupt change in the image function.	zero crossings	zebra crossing	both a & b	Mean filtering	zero crossings
42	The _____ produces edge magnitudes without regard to their orientations and have the same properties in all directions and are therefore invariant to rotation in the image.	laptop	Laplacian	Lapleaciean	Prewitt	Laplacian

43	Another image sharpening approach that is often used in printing industry applications is _____	linear masking	masking	Unsharp masking	all the above	Unsharp masking
44	Edge detection techniques like the Kirsch, Sobel and _____ operators are based on convolution in very small neighborhoods and work well for specific images only. The main disadvantage of these edge detectors is their dependence on the size of objects and sensitivity to noise.	rotating mask	degradation function	both a & d	Prewitt	Prewitt
45	_____ optimal for step edges corrupted by white noise and optimality is related to three criteria detection criterion, localization criterion and response criterion	Aanny edge detection	Canny edge detection	Vaclav edge detection	Simple edge detection	Canny edge detection
46	Edge points after thresholding do not create contiguous boundaries. _____ tries to recover edge pixels on the potential object boundary which are missing.	center filling	Edge filling	node filing	circle filling	Edge filling
47	_____ explores a priori knowledge about the noise and solves the problem of a posteriori linear mean square estimate	Wiener filtration	Inverse filtration	degradation function	interpolation	Wiener filtration
48	_____ is based on the assumption that degradation was caused by a linear function h(i,j) and the additive noise *nu* that is independent of the signal.	Wiener filtration	Inverse filtration	Non-Wiener filtration	Non-Inverse filtration	Inverse filtration

49	_____ suppresses image degradation using knowledge about its nature	Image gradient	both a & C	Image restoration	Image gradient	Image restoration
50	Degradation occurs due to I) defects of optical lenses II) nonlinearity of the electro-optical sensor III) graininess of the film material IV) wrong focus	I, II & III	I, II & IV	II, III & IV	I, II, III & IV	I, II, III & IV
51	Image restoration techniques falls under two groups namely _____	deterministic and stochastic	both a & c	rotating mask	circular mask	deterministic and stochastic
52	_____ methods are applicable to images with little noise and a known degradation function. The original image is obtained from the degraded one by a transformation inverse to the degradation.	stochastic	Deterministic	both a & b	statistic	Deterministic
53	_____ is an example of stochastic techniques that best restoration according to some stochastic criterion.	Least squares method	stochastic	degraded image	improved image	Least squares method
54	A _____ about degradation is either known in advance or obtained before restoration.	degraded image	image restoration	both a & b	priori knowledge	priori knowledge
55	A _____ is obtained by analyzing the degraded image	posteriori knowledge	image restoration	noise image	prior knowledge	posteriori knowledge
56	The degradation is very often simplified by neglecting the _____	stochastic	rotating mask	nonlinearity	linear	nonlinearity
57	Degradation can be then expressed as convolution and if noise is not significant then image restoration equates to _____	pixels	thresholding	deconvolution	pels	deconvolution

58	_____ gives poor results in pixels suffering from noise since the noise is not taken into account.	fast segmentation	thresholding	both a & b	Inverse filtration	Inverse filtration
59	The method used to generate a processed image that has a specified histogram is called _____	histogram specification	histogram matching	histogram processing	both a&b	both a&b
60	A plot of $p_r(r_k)$ versus r_k is called _____	histogram	image negative	image log	positive image	histogram

UNIT III

SYLLABUS:

Some Basic relationships between Pixels-Basic gray level transformations- Histogram processing - Basic spatial filtering- Smoothing special filtering- Image Degradation/ Restoration process- Noise Models

Image segmentation is one of the most important steps leading to the analysis of processed image data—its main goal is to divide an image into parts that have a strong correlation with objects or areas of the real world contained in the image. We may aim for complete segmentation, which results in a set of disjoint regions corresponding uniquely with objects in the input image, or for partial segmentation, in which regions do not correspond directly with image objects. To achieve a complete segmentation, cooperation with higher processing levels which use specific knowledge of the problem domain is necessary. However, there is a whole class of segmentation problems that can be solved successfully using lower-level processing only. In this case, the image commonly consists of contrasted objects located on a uniform background—simple assembly tasks, blood cells, printed characters, etc. Here, a simple global approach can be used and the complete segmentation of an image into objects and background can be obtained. Such processing is context independent; no object-related model is used, and no knowledge about expected segmentation results contributes to the final segmentation.

If partial segmentation is the goal, an image is divided into separate regions that are homogeneous with respect to a chosen property such as brightness, color, reflectivity, texture, etc. If an image of a complex scene is processed, for example, an aerial photograph of an urban scene, a set of possibly overlapping homogeneous regions may result. The partially segmented image must then be subjected to further processing, and the final image segmentation may be found with the help of higher-level information.

Totally correct and complete segmentation of complex scenes usually cannot be achieved in this processing phase, although substantial reduction in data volume offers an immediate gain. A reasonable aim is to use partial segmentation as an input to higher-level processing.

Image data ambiguity is one of the main segmentation problems, often accompanied by information noise. Segmentation methods can be divided into three groups according to the dominant features they employ: First is global knowledge about an image or its part; the knowledge is usually represented by a histogram of image features. Edge-based segmentations form the second group, and region-based segmentations the third— many different characteristics may be used in edge detection or region growing, for example, brightness, texture, velocity field, etc. The second and the third groups solve a dual problem. Each region can be represented by its closed boundary, and each closed boundary describes a region. Because of the different natures of the various edge- and region-based algorithms, they may be expected to give somewhat different results and consequently different information. The segmentation results of these two approaches can therefore be combined in a single description structure. A common example of this is a region adjacency graph, in which regions are represented by nodes and graph arcs represent adjacency relations based on detected region borders.

3.1 Thresholding

Gray-level thresholding is the simplest segmentation process. Many objects or image regions are characterized by constant reflectivity or light absorption of their surfaces; a brightness constant or threshold can be determined to segment objects and background. Thresholding is computationally inexpensive and fast—it is the oldest segmentation method and is still widely used in simple applications; thresholding can easily be done in
real time using specialized hardware.

Thresholding is one of the most importantapproaches to image segmentation.

• Gray level histogram of an image f(x,y)composed of a light object on a dark background.

• To extract the object: select a threshold T that separates the gray levels of the background andthe object.

Complete segmentation can result from thresholding in simple scenes.

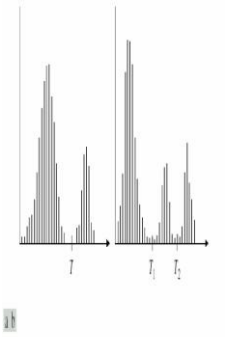
$$R = \bigcup_{i=1}^{S} R_i , \quad R_i \cap R_j = \emptyset , \quad i \neq j .$$

Thresholding is the transformation of an input image / to an output (segmented) binary image g as follows:

$$g(i,j) = 1 \quad \text{for } f(i,j) \geq T\,, \qquad = 0 \quad \text{for } f(i,j) < T\,, \tag{6.2}$$

Single threshold: points with f(x,y)>T belong to object; other points belong to background.

• Multiple thresholds: points with f(x,y)>T2 belong to object; points with f(x,y)<T1 belong to bakground.



neral can be calculated as:

T=T(x,y,p(x,y),f(x,y))

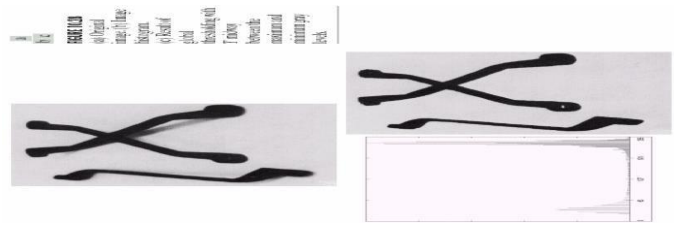
f(x,y): gray level at (x,y)

p(x,y): some local property of the point (x,y) (e.g., the average gray level of a neighborhood centered on (x,y).

T depends only on f(x,y): global threshold

T depends on f(x,y) and p(x,y): local threshold

T depends on f(x,y) and p(x,y) and x,y : dynamic threshold



Algorithm 6.1: Basic thresholding

1. Select an initial estimate for T

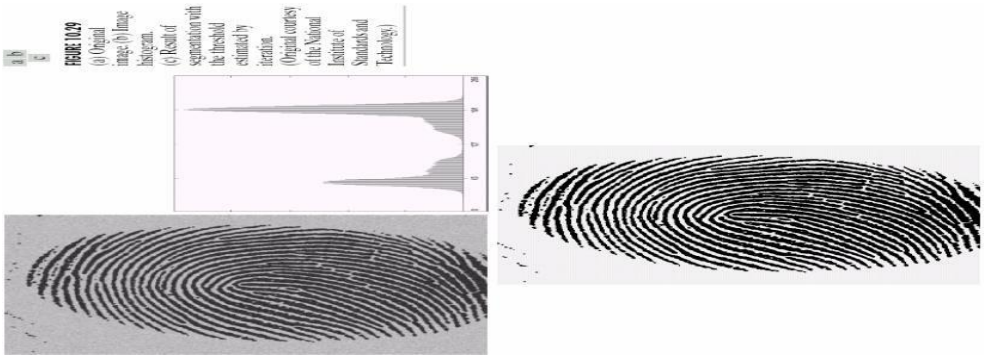
2. Segment the image using T. This will produce two group of pixels G1 (with gray level greater than T)

and G2 (with gray level less than T)

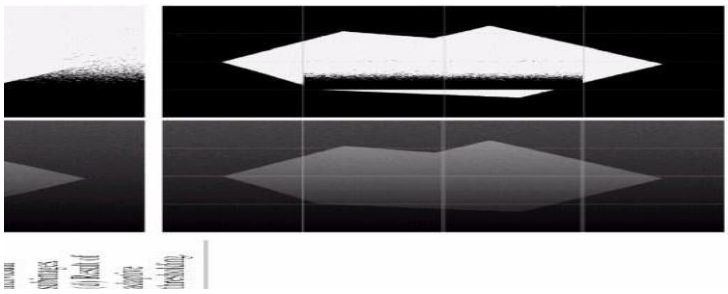
3. Compute average gray level values μ1 and μ2 for pixels in regions G1 and G2

4. Compute a new threshold: T=(μ1+ μ2)/2

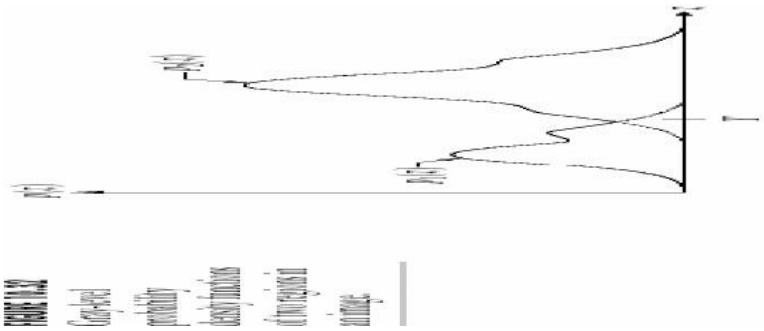
5. Repeat step 2 through 4 until the difference in T in successive iterations is smaller than a predefined value



Basic Adaptive (Local) Thresholding



Optimal Global Thresholding



$p(z) = P1 \cdot p1(z) + P2 \cdot p2(z)$

$1\ 2\ P + P = 1$

Problem: how to optimally determine T to minimize the segmentation error?

Error 1: $E_1(T) = \int_{-\infty}^{T} p_2(z)dz$

Error 2: $E_2(T) = \int_{T}^{\infty} p_1(z)dz$

Total error: $E(T) = P_1 E_2(T) + P_2 E_1(T)$

Goal: what's the value of T to minimize E(T)?

Result: $P_1 p_1(T) = P_2 p_2(T)$

Optimal Global Thresholding: Gaussian PDFs.

$$p(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1}\exp(-\frac{(z-\mu_1)^2}{2\sigma_1^2}) + \frac{P_2}{\sqrt{2\pi}\sigma_2}\exp(-\frac{(z-\mu_2)^2}{2\sigma_2^2})$$

Solution: $AT^2 + BT + C = 0$

$A = \sigma_1^2 - \sigma_2^2$

$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)$

$C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2 2\sigma_2^2 \ln(\sigma_2 P_1 / \sigma_1 P_2)$

Thresholding has been presented relying only on gray-level image properties. Note that this is just one of many possibilities; thresholding can be applied if the values f(i,j) do not represent gray-levels, but instead represent gradient, a local texture property , or the value of any other image decomposition criterion.

This method is called p-tile thresholding. Unfortunately, we do not usually have such definite prior information about area ratios. This information can sometimes be substituted by knowledge of

another property, for example, the average width of lines in drawings, etc. The threshold can be determined to provide the required line width in the segmented image.

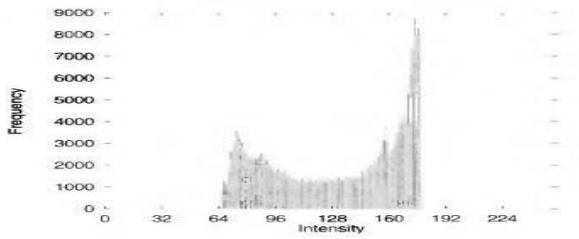


Figure 6.3: A bi-modal histogram.

More complex methods of threshold detection are based on histogram shape analysis. If an image consists of objects of approximately the same gray-level that differs from the gray-level of the background, the resulting histogram is bi-modal. Pixels of objects form one of its peaks, while pixels of the background form the second peak—Figure 6.3 shows a typical example. The histogram shape illustrates the fact that the gray values between the two peaks are not common in the image, and probably result from border pixels between objects and background.

To decide if a histogram is bi-modal or multi-modal may not be so simple in reality, it often being impossible to interpret the significance of local histogram maxima. Bi-modal histogram threshold detection algorithms usually find the highest local maxima first and detect the threshold as a minimum between them; this technique is called the mode method. To avoid detection of two local maxima belonging to the same global maximum, a minimum distance in gray-levels between these maxima is usually required, or techniques to smooth histograms (see Section 2.3.2) are applied. Note that histogram bi-modality itself does not guarantee correct threshold segmentation even if the histogram is bi-modal, correct segmentation may not occur with objects located on a background different gray-levels. A two-part image with one half white and the second half black actually has the same histogram as an image with randomly spread white and black pixels
(i.e.. a salt-and-pepper noise image, see Section 2.3.6). This is one example showing the need to check threshold segmentation results whenever the threshold has been determined
from a histogram only, using no other image characteristics.

A more general approach takes gray-level occurrences inside a local neighborhood into consideration when constructing a gray-level histogram, the goal being to build a histogram with a better peak-to-valley ratio. One option is to weight histogram contributions to suppress the

influence of pixels with a high image gradient. This means that a histogram will consist mostly of the gray values of objects and background, and that border gray-levels (with higher gradient) will not contribute.

This will produce a deeper histogram valley and allow an easier determination of the threshold. Another method based on the same idea uses only high-gradient pixels to form the gray-level histogram, meaning that the histogram will consist mostly of border gray-levels and should be unimodal in which the peak corresponds to the gray-level of borders between objects and background. The segmentation threshold can be determined as the gray value of this peak, or as a mean of a substantial part of the peak.

2 Optimal thresholding

Methods based on approximation of the histogram of an image using a weighted sum of two or more probability densities with normal distribution represent a different approach called optimal thresholding. The threshold is set as the closest gray-level corresponding to the minimum probability between the maxima of two or more normal distributions, which results in minimum error segmentation (the smallest number of pixels is mis-segmented). The difficulty with these methods is in estimating normal distribution parameters together with the uncertainty that the distribution may be considered normal.

These difficulties may be overcome if an optimal threshold is sought that maximizes gray-level variance between objects and background. Note that this approach can be applied even if more than one threshold is needed. Alternatively, minimization of variance of the histogram, sum of square errors, spatial entropy, average clustering, or other optimization approaches may be used.

Algorithm 6.2: Iterative (optimal) threshold selection

1. Assuming no knowledge about the exact location of objects, consider as a first approximation that the four corners of the image contain background pixels only and the remainder contains object pixels.
2. At step t, compute μ_B^t and μ_O^t as the mean background and object gray-level, respectively, where segmentation into background and objects at step t is defined by the threshold value T^t determined in the previous step [equation 6.9]

$$\mu_B^t = \frac{\sum_{(i,j)\in \text{background}} f(i,j)}{\#\text{background\_pixels}}, \qquad \mu_O^t = \frac{\sum_{(i,j)\in \text{objects}} f(i,j)}{\#\text{object\_pixels}}. \tag{6.8}$$

3. Set

$$T^{(t+1)} = \frac{\mu_B^t + \mu_O^t}{2}, \tag{6.9}$$

$T^{(t+1)}$ now provides an updated background—object distinction.

4. If $T^{(t+1)} = T^{(t)}$, halt; otherwise return to step 2.

A combination of optimal and adaptive thresholding (equation F.4)) was used for brain image segmentation from MR image data.

Many practical segmentation problems need more information than is contained in one spectral band. Color images are a natural example, in which information is coded in three spectral bands, for example, red, green, and blue; multi-spectral remote sensing images or meteorological satellite images may have even more spectral bands. One segmentation approach determines thresholds independently in each spectral band and combines them into a single segmented image.

Algorithm 3: Recursive multi-spectral thresholding

1. Initialize the whole image as a single region.
2. Compute a smoothed histogram (see Section 2.3.2) for each spectral band. Find the most significant peak in each histogram and determine two thresholds as local minima on either side of this maximum. Segment each region in each spectral band into sub-regions according to these thresholds. Each segmentation in each spectral band is projected into a multi-spectral segmentation—see Figure 6.7. Regions for the next processing steps are those in the multi-spectral image.
3. Repeat step 2 for each region of the image until each region's histogram contains only one significant peak.

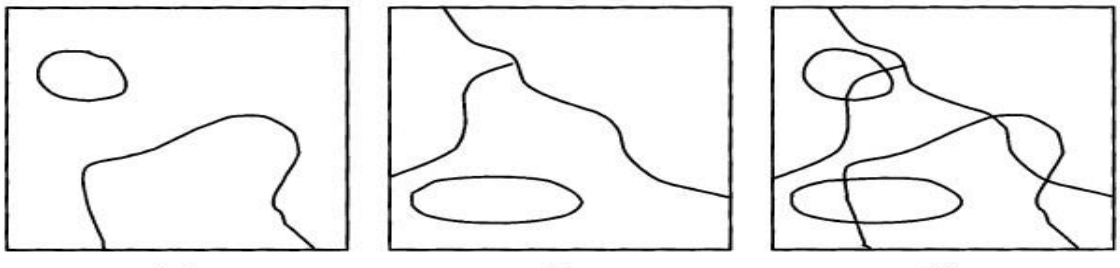


Figure 6.7: Recursive multi-spectral thresholding. (a) Band 1 thresholding. (b) Band 2 thresholding. (c) Multi-spectral segmentation.

stret onal histograms instead of histograms for each spectral band in step 2 of the previous algorithm. Multi-spectral segmentations are often based on n-dimensional vectors of gray-levels in n spectral bands for each pixel or small pixel neighborhood. This segmentation approach, widely used in remote sensing, results from a classification process which is applied to these n-dimensional vectors.

2 Edge-based segmentation

Edge-based segmentation represents a large group of methods based on information about

edges in the image; it is one of the earliest segmentation approaches and still remains very important. Edge-based segmentations rely on edges found in an image by edge detecting operators—these edges mark image locations of discontinuities in gray-level, color, texture,etc.

The most common problems of edge-based segmentation, caused by image noise or unsuitable information in an image, are an edge presence in locations where there is no border, and no edge presence where a real border exists. Clearly both these cases have a negative influence on segmentation results.

First, we will discuss simple edge-based methods requiring minimum prior information, and the necessity for prior knowledge will increase during the section. Construction of regions from edge-based partial segmentations is discussed at the end of the section.

Edge image thresholding

Almost no zero-value pixels are present in an edge image, but small edge values correspond to non-significant gray-level changes resulting from quantization noise, small lighting irregularities, etc. Simple thresholding of an edge image can be applied to

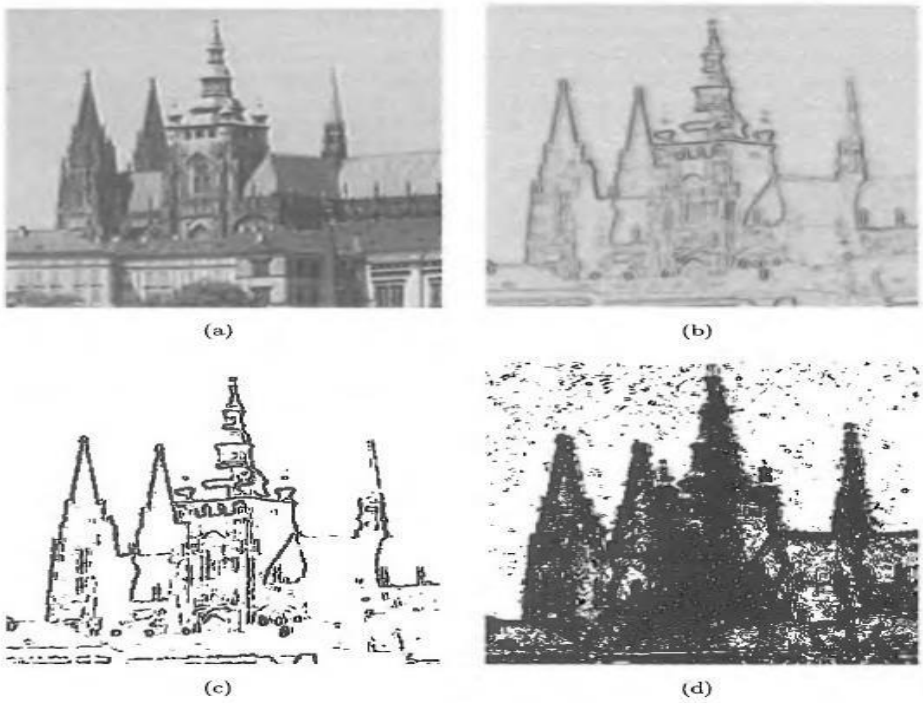


Figure 6.8: Edge image thresholding. (a) Original image. (b) Edge image (low contrast edges enhanced for display). (c) Edge image thresholded at 30. (d) Edge image thresholded at 10.

remove these small values. Figure 6.8a shows an original image, an edge image equation E.44)] is in Figure 6.8b, an 'over-thresholded' image is in Figure 6.8c, and an 'under-thresholded' image is in Figure 6.8d. Selection of an appropriate global threshold is often difficult and sometimes impossible; p-tile thresholding can be applied to define a threshold, and a more exact approach using orthogonal basis functions is described in [Flynn, 1972] which, if the original data has good contrast and is not noisy, gives good
results.

A problem with simple detectors is the thickening that is evident in Figure 6.8b where there should only be a simple boundary. This can be partially rectified if edges carry directional information (as they do with the Sobel) by performing some form of non-maximal suppression to suppress multiple responses in the neighborhood of single boundaries. The following algorithm generates Figure 6.9a from Figure 6.8b.

Algorithm4: Non-maximal suppression of directional edge data

1. Quantize edge directions eight ways according to 8-connectivity .For each pixel with non-zero edge magnitude, inspect the two adjacent pixels
indicated by the direction of its edge .

3. If the edge magnitude of either of these two exceeds that of the pixel under inspection, mark it for deletion.

4. When all pixels have been inspected, re-scan the image and erase to zero all edge data marked for deletion.

This algorithm is based on 8-connectivity and may be simplified for 4-connectivity; it is also open to more sophisticated measurement of edge direction.

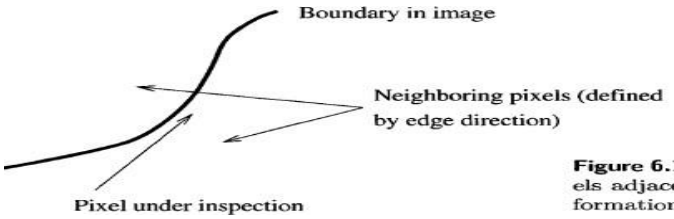


Figure 6.10: Non-maximal suppression; pixels adjacent with respect to local edge information are inspected.

Algorithm 5: Hysteresis to filter output of an edge detector

1. Mark all edges with magnitude greater than t\ as correct.

2. Scan all pixels with edge magnitude in the range [to,ti].

3. If such a pixel borders another already marked as an edge, then mark it too. 'Bordering' may be defined by 4- or 8-connectivity.

4. Repeat from step 2 until stability. Other forms of post-processing are also available, for example, to remove all border segments with length less than a specified value, or to consider the average edge strength of a (partial) border.

2 Edge relaxation

Borders resulting from the previous method are strongly affected by image noise, often with important parts missing. Considering edge properties in the context of their mutual neighbors can increase the quality of the resulting image. All the image properties, including those of further edge existence, are iteratively evaluated with more precision until the edge context is totally clear—based on the strength of edges in a specified local neighborhood, the confidence of each edge is either increased or decreased.

A weak edge positioned between two strong edges provides an example of context; it is highly probable that this inter-positioned weak edge should be a part of a resulting boundary. If, on the other hand, an edge (even a strong one) is positioned by itself with no supporting context, it is probably not a part of any border.

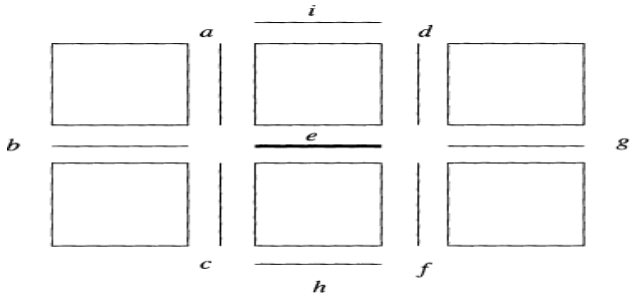


Figure 6.11: Crack edges surrounding central edge e.

Edge context is considered at both ends of an edge, giving the minimal edge neighborhood shown in Figure 6.11. All three possible edge positions at the end of the edge e must be included to cover all the possible ways the border can continue from both ends of e. Furthermore, two edge positions parallel with the edge e can be included in the local neighborhood—these parallel positions compete with the edge e in the placement of the border. Edge relaxation aims for

continuous border construction, so we discuss the edge patterns that can be found in the local neighborhood. The central edge e has a vertex at each of its ends, and three possible border continuations can be found from both of these vertices. Let each vertex be evaluated according to the number of edges emanating from the vertex, not counting the edge e; call this number the vertex type. The type of edge e can then be represented using a number pair i — j describing edge patterns at each vertex, where i and j are the vertex types of the edge e. For example, it assigns type 0 — 0 for the edges shown in Figure 6.12a, type 3 — 3 in for Figure 6.12d, etc. By symmetry, we need only

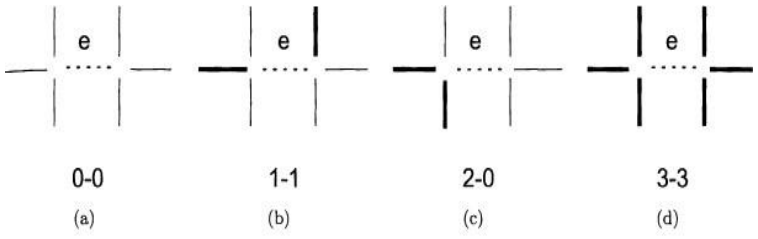


Figure 6.12: Edge patterns and corresponding edge types.

consider the cases where i < j. The following context situations are possible:

• 0-0 isolated edge—negative influence on the edge confidence,

• 0-1 uncertain—weak positive, or no influence on edge confidence,

• 0-2, 0-3 dead end—negative influence on edge confidence,

• 1-1 continuation—strong positive influence on edge confidence,

• 1-2, 1-3 continuation to border intersection—medium positive influence on edge confidence,

• 2-2, 2-3, 3-3 bridge between borders—not necessary for segmentation, no influence on edge confidence.

An edge relaxation can be defined from the given context rules that may be considered as a production system, Section 9.1. Edge relaxation is an iterative method, with edge confidences converging either to zero (edge termination) or one (the edge forms a border).

The confidence c^(e) of each edge e in the first iteration can be defined as a normalized magnitude of the crack edge, with normalization based on either the global maximum of crack edges in the whole image, or on a local maximum in some large neighborhood of the edge, thereby decreasing the influence of a few very high values of edge magnitude in

the image.

Algorithm 6: Edge relaxation

1. Evaluate a confidence $c^{(1)}(e)$ for all crack edges e in the image.
2. Find the edge type of each edge based on edge confidences $c^{(k)}(e)$ in its neighborhood.
3. Update the confidence $c^{(k+1)}(e)$ of each edge e according to its type and its previous confidence $c^{(k)}(e)$.
4. Stop if all edge confidences have converged either to 0 or 1. Repeat steps 2 and 3 otherwise.

Edge confidence modification rules can be simplified and just one value of δ can be used, not including the weak, moderate, or strong confidence increase/decrease options. Further, vertex types 2 and 3 can be considered the same in implementation because they result in the same production rules.

Edge relaxation, as described above, rapidly improves the initial edge labeling in the first few iterations. Unfortunately, it often slowly drifts, giving worse results than expected after larger numbers of iterations. A theoretical explanation, convergence proof, and practical solutions are given in [Levy, 1988]. The reason for this strange behavior is in searching for the global maximum of the edge consistency criterion over all the image, which may not give locally optimal results. A solution is found in setting edge confidences to zero under a certain threshold, and to one over another threshold which increases the influence of original image data. Therefore, one additional step must be added to the edge confidence computation [equations (6.13) and (6.14)]:

$$\text{If } c^{(k+1)}(e) > T_1 \text{ then assign } c^{(k+1)}(e) = 1, \tag{6.15}$$

$$\text{If } c^{(k+1)}(e) < T_2 \text{ then assign } c^{(k+1)}(e) = 0, \tag{6.16}$$

where T_1 and T_2 are parameters controlling the edge relaxation convergence speed and resulting border accuracy. Moreover, this method makes multiple labelings possible; the existence of two edges at different directions in one pixel may occur in corners, crosses, etc.

Edge relaxation results are shown in Figure 6.13, where edges parallel with the central edge were not considered in the relaxation process. The relaxation method can

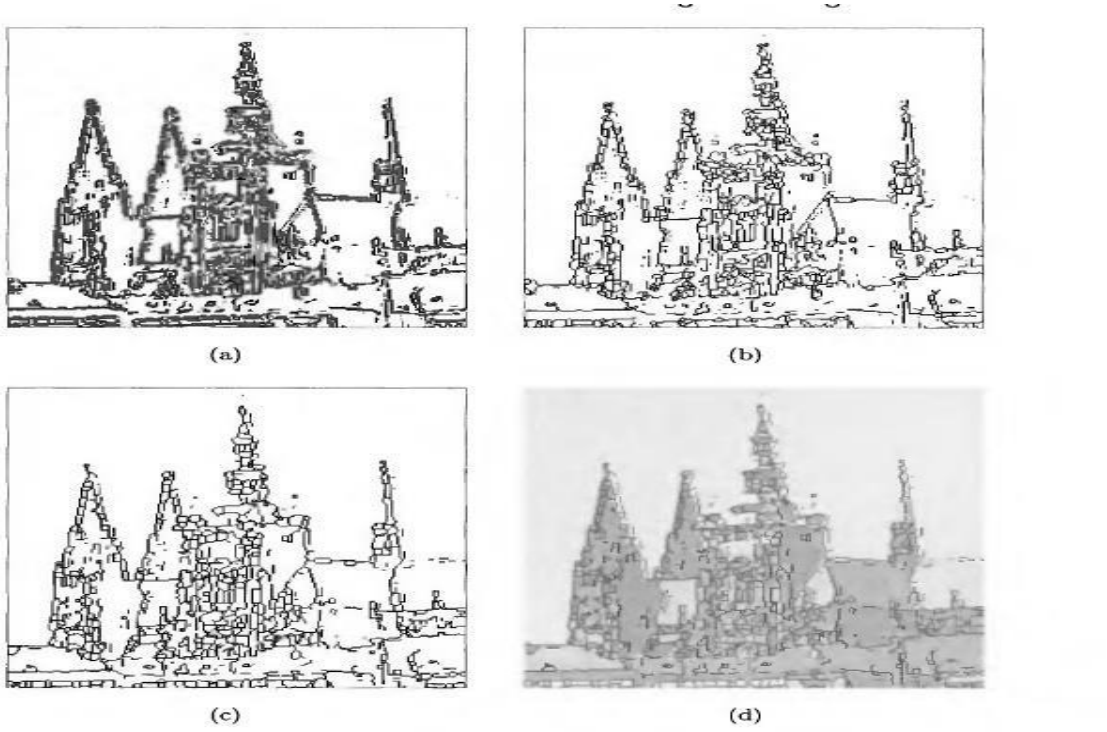


Figure 6.13: Edge relaxation, see Figure 3.11a for original. (a) Resulting borders after 10 iterations. (b) Borders after thinning. (c) Borders after 100 iterations, thinned. (d) Borders after 100 iterations overlaid over original.

Edge relaxation results are shown in Figure 6.13, where edges parallel with the central edge were not considered in the relaxation process. The relaxation method can easily be implemented in parallel, with surprisingly good speedup. Implementation on a 16-processor hypercube showed almost linear speedup if 2, 4, 8, or 16 processors were used; in other words, using 16 processors the processing was almost 16 times faster than using

3 Border tracing

If a region border is not known but regions have been defined in the image, borders can be uniquely detected. First, let us assume that the image with regions is either binary or that regions have been labeled (see Section 8.1). The first goal is to determine inner region borders. As defined earlier, an inner region border is a subset of the region—conversely, the outer border is not a subset of the region. The following algorithm covers inner boundary tracing in both 4-connectivity and 8-connectivity.

Algorithm7: Inner boundary tracing

1. Search the image from top left until a pixel of a new region is found; this pixel Po then

has the minimum column value of all pixels of that region having the minimum row value. Pixel Po is a starting pixel of the region border. Define a variable dir which stores the direction of the previous move along the border from the previous border element to the current border element. Assign

(a) dir = 3 if the border is detected in 4-connectivity (Figure 6.14a),

(b) dir = 7 if the border is detected in 8-connectivity (Figure 6.14b).

2. Search the 3x3 neighborhood of the current pixel in an anti-clockwise direction, beginning the neighborhood search in the pixel positioned in the direction (a) (dir+ 3) mod 4 (Figure 6.14c),

(b) (dir+ 7) mod 8 if dir is even (Figure 6.14d),

(dir + 6) mod 8 if dir is odd (Figure 6.14e).

The first pixel found with the same value as the current pixel is a new boundary element Pn. Update the dir value.

3. If the current boundary element Pn is equal to the second border element Pi, and if the previous border element Pn-i is equal to Po, stop. Otherwise repeat step 2.

4. The detected inner border is represented by pixels $P_0 \ldots P_{n-2}$.

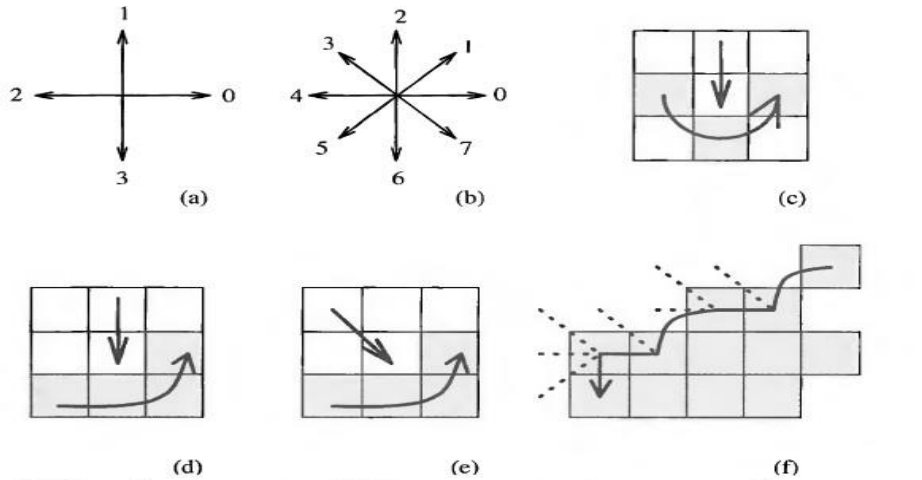


Figure 6.14: Inner boundary tracing. (a) Direction notation, 4-connectivity. (b) 8-connectivity. (c) Pixel neighborhood search sequence in 4-connectivity. (d), (e) Search sequence in 8-connectivity. (f) Boundary tracing in 8-connectivity (dotted lines show pixels tested during the border tracing).

Algorithm 7 works for all regions larger than one pixel. This algorithm is able to find region borders but does not find borders of region holes. To search for hole borders as well, the border must be traced starting in each region or hole border element if this element has never been a member of any

border previously traced. The search for border elements always starts after a currently traced border is closed, and the search for 'unused' border elements can continue in the same way as the search for the first border element was done.

Algorithm 8: Outer boundary tracing

1. Trace the inner region boundary in 4-connectivity until done.
2. The outer boundary consists of all non-region pixels that were tested during the search process; if some pixels were tested more than once, they are listed more than once in the outer boundary list. Note that some border elements may be repeated in the outer border up to three times—see Figure 6.15. The outer region border is useful for deriving properties such as perimeter, compactness, etc., and is consequently often used.

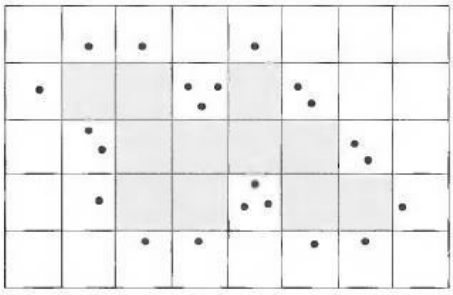


Figure 6.15: Outer boundary tracing; • denotes outer border elements. Note that some pixels may be listed several times.

The inner border is always part of a region but the outer border never is. Therefore, if two regions are adjacent, they never have a common border, which causes difficulties in higher processing levels with region description, region merging, etc. The inter-pixel boundary extracted, for instance, from crack edges is common to adjacent regions. The main advantage of the extended boundary definition is that it defines a single common border between adjacent regions, and it may be specified using standard pixel co-ordinates (see Figure 6.16).

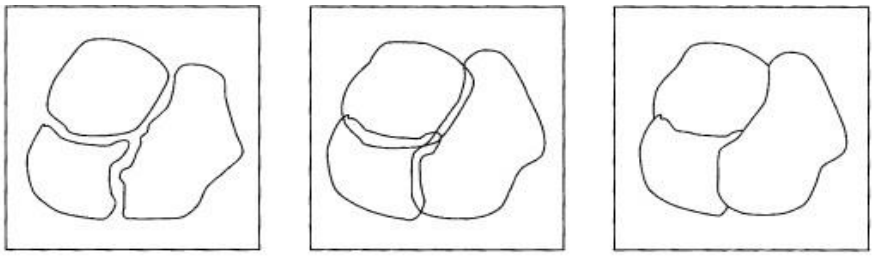


Figure 6.16: Boundary locations for inner, outer, and extended boundary definition. (a) Inner. (b) Outer. (c) Extended.

All the useful properties of the outer border still remain; in addition, the boundary shape is exactly equal to the inter-pixel shape but is shifted one half-pixel down and one half-pixel right. The existence of a common border between regions makes it possible to incorporate into the boundary tracing a boundary description process. An evaluated graph consisting of border segments and vertices may result directly from the boundary tracing process; also, the border between adjacent regions may be traced only once and not twice as in conventional approaches. The extended boundary is defined using 8-neighborhoods, and the pixels are coded according to Figure 6.17a,

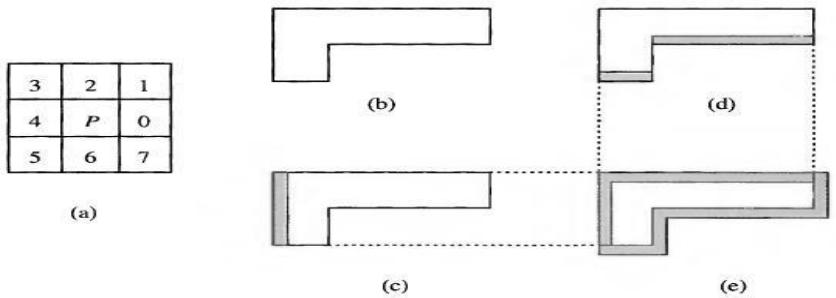


Figure 6.17: Extended boundary definition. (a) Pixel coding scheme. (b) Region R. (c) LEFT(R). (d) LOWER(R). (e) Extended boundary.

e.g., Pa(P) denotes the pixel immediately to the left of pixel P. Four kinds of inner boundary pixels of a region R are defined: if Q denotes pixels

outside the region R, then a pixel P G R is

a LEFT pixel of R if P4(P) G Q ,

a RIGHT pixel of R H PQ(P) ? Q ,

an UPPER pixel of R if P2(P) G Q ,

a LOWER pixel of R if P6(P) G Q .

Let LEFT(ii), RIGHT(iJ), UPPER(#), LOWER(i?) represent the corresponding subsets

of R. The extended boundary EB is defined as a set of points P, Pq^Pq. P7 satisfying the following conditions [Pavlidis, 1977; Liow, 1991]:

EB ={P : P G LEFT(fi)} U {P : P G UPPER(P)}

U {P6(P) : P e LOWER(fl)} U {PQ(P) : P G RIGHT(i?)}

U {P7(P) : P G RIGHT(#)} . F.17)

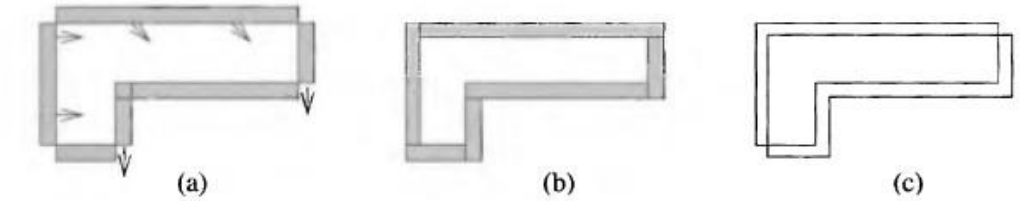


Figure 6.18: Constructing the extended boundary from outer boundary. (a) Outer boundary. (b) Extended boundary construction. (c) Extended boundary has the same shape and size as the natural object boundary.

The extended boundary can easily be constructed from the outer boundary. Using an intuitive definition of RIGHT, LEFT, UPPER, and LOWER outer boundary points, the extended boundary may be obtained by shifting all the UPPER outer boundary point some pixel down and right, shifting all the LEFT outer boundary points one pixel to the right, and shifting all the RIGHT outer boundary points one pixel down. The LOWER outer boundary point positions remain unchanged; see Figure 6.18.

Algorithm 9: Extended boundary tracing

1. Define a starting pixel of an extended boundary in a standard way (the first region pixel found in a left-to-right and top-to-bottom line-by-line image search).
2. The first move along the traced boundary from the starting pixel is in direction dir = 6 (down), corresponding to the situation (i) in Figure 6.19.
3. Trace the extended boundary using the look-up table in Figure 6.19 until a closed extended border results. Note that no hole-border tracing is included in the algorithm.

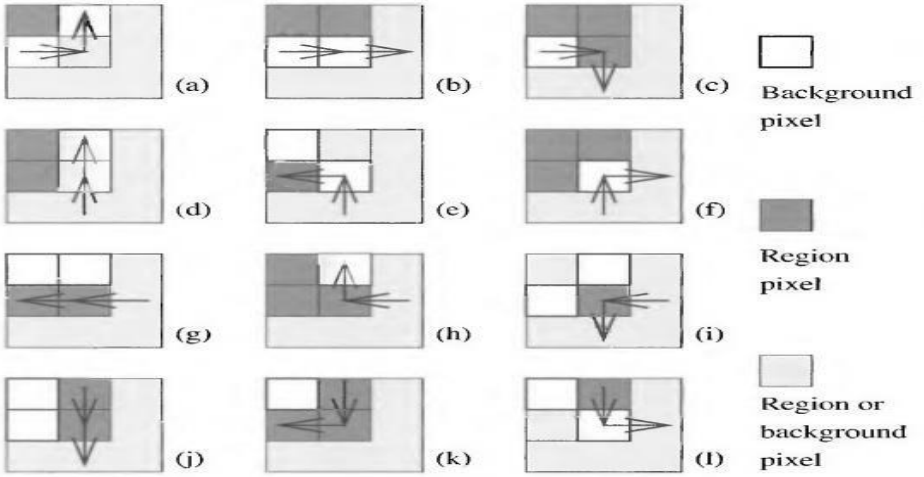
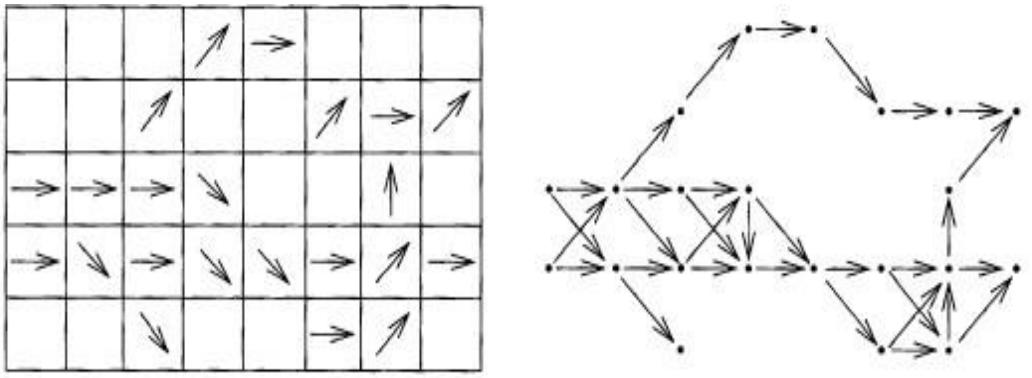


Figure 6.19: Look-up table defining all 12 possible situations that can appear during extended border tracing. Current position is in the central pixel. The direction of the next move depends on the local configuration of background and region points, and on the direction of approach to the current pixel. *Adapted from [Liow, 1991].*

The holes are considered separate regions and therefore the borders between the region and its hole are traced as a border of the hole. The look-up table approach makes the tracing more efficient than conventional methods and makes parallel implementation possible. This method is very suitable for representing borders in higher-level segmentation approaches including methods that integrate edge-based and region-based segmentation results. Moreover, in the conventional approaches, each border between two regions must be traced twice. The algorithm can trace each boundary segment only once, storing the information about what has already been done in double-linked lists.

These conditions can be modified in specific edge detection problems. Figure 6.20a shows an image of edge directions, with only significant edges according to their magnitudes listed. Figure 6.20b shows an oriented graph constructed in accordance with the presented principles.



(a) (b)

Figure 6.20: Graph representation of an edge image. (a) Edge directions of pixels with above-threshold edge magnitudes. (b) Corresponding graph.

The application of graph search to edge detection was first published in [Martelli, 1972], in which Nilsson's A-algorithm [Nilsson, 1982] applies. Let $\mathbf{x}_A$ be the starting border element, and $\mathbf{x}_B$ be the end border element. To use graph search for region border detection, a method of oriented weighted-graph expansion must first be defined (one possible method was described earlier). A cost function $f(\mathbf{x}_i)$ must also be defined that is a cost estimate of the path between nodes n_A and n_B (pixels $\mathbf{x}_A$ and $\mathbf{x}_B$) which goes through an intermediate node n_i (pixel $\mathbf{x}_i$). The cost function $f(\mathbf{x}_i)$ typically consists of two components; an estimate $\tilde{g}(\mathbf{x}_i)$ of the minimum path cost between the starting border element $\mathbf{x}_A$ and $\mathbf{x}_i$, and an estimate $\tilde{h}(\mathbf{x}_i)$ of the minimum path cost between $\mathbf{x}_i$ and the end border element $\mathbf{x}_B$. The cost $\tilde{g}(\mathbf{x}_i)$ of the path from the starting point to the node n_i is usually a sum of costs associated with the arcs or nodes that are in the path. The cost function must be separable and monotonic with respect to the path length, and therefore the local costs associated with arcs are required to be non-negative. A simple example of $\tilde{g}(\mathbf{x}_i)$ satisfying the given conditions is to consider the path length from $\mathbf{x}_A$ to $\mathbf{x}_i$. An estimate $\tilde{h}(\mathbf{x}_i)$ may be the length of the border from $\mathbf{x}_i$ to $\mathbf{x}_B$, it making sense to prefer shorter borders between $\mathbf{x}_A$ and $\mathbf{x}_B$ as the path with lower cost. This implies that the following graph search algorithm (Nilsson's A-algorithm) can be applied to the border detection.

Algorithm 6.11: A-algorithm graph search

1. Expand the starting node n_A and put all its successors into an OPEN list with pointers back to the starting node n_A. Evaluate the cost function f for each expanded node.
2. If the OPEN list is empty, fail. Determine the node n_i from the OPEN list with the lowest associated cost $f(n_i)$ and remove it. If $n_i = n_B$, then trace back through the pointers to find the optimum path and stop.
3. If the option to stop was not taken in step 2, expand the specified node n_i, and put its successors on the OPEN list with pointers back to n_i. Compute their costs f. Go to step 2.

An example of this algorithm is given in Figure 6.21. Here, nodes currently on the OPEN list are shaded and the minimum-cost node on the OPEN list is shaded and outlined. In Figure 6.21c, note that the node with a cumulative cost of 7 is also expanded; however no successors are found. In Figure 6.21e, since an expansion of a node in the final graph layer was attempted, the search is over.

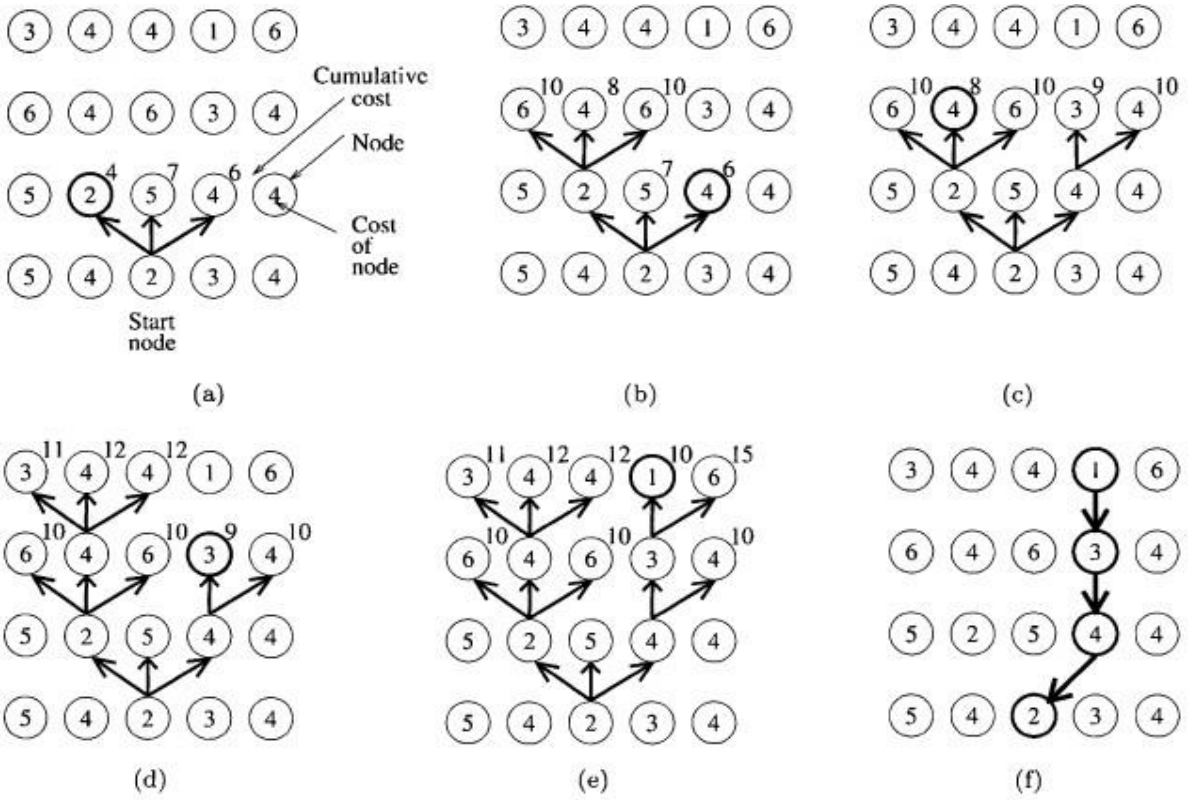


Figure 6.21: Example of a graph searching sequence using the A-algorithm (see text for description of progress of algorithm). (a) Step 1, expansion of the start node. (b) Step 2. (c) Steps 3 and 4. (d) Step 5. (e) Step 6. (f) The optimal path is defined by back-tracking.

If no additional requirements are set on the graph construction and search, this process can easily result in an infinite loop (see Figure 6.22). To prevent this behavior, no node expansion is allowed that puts a node on the OPEN list if this node has already been visited and put on the OPEN list in the past. A simple solution to the loop problem is not to allow searching in a backward direction. This approach can be used if a priori information about the boundary location and its local direction is available. In this case, than others. When included into the border, a border element value can be weighted by the distance ^disV from the approximate boundary, the distance having either additive or multiplicative influence on the cost dist(xi,approximate_boundary) . F.20)

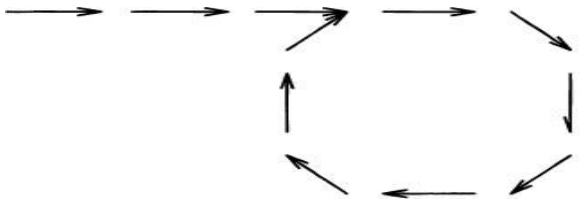


Figure 6.22: Example of following a closed loop in image data.

Thus, edge strengths close to the expected value will be preferred in comparison to edges of lower or higher edge strength. A variety of such transforms may be developed: a set of generally useful cost transforms. Overall, a good cost function will very often consist of several components combined together.

Graph-based border detection methods very often suffer from extremely large numbers of expanded nodes stored in the OPEN list, these nodes with pointers back to their predecessors representing the searched part of the graph. The cost associated with each node in the OPEN list is a result of all the cost increases on the path from the starting node to that node. This implies that even a good path can generate a higher cost in the current node than costs of the nodes on worse paths which did not get so far from the starting node. This results in expansion of these 'bad' nodes representing shorter paths with lower total costs, even with the general view that their probabilities are low. An excellent way to solve this problem is to incorporate a heuristic estimate hfa) into the cost evaluation, but unfortunately, a good estimate of the path cost from the current node to the goal is not usually available. Some modifications which make the method more practically useful, even if some of them no longer guarantee the minimum-cost path, are available:

• **Pruning the solution tree:** The set of nodes in the OPEN list can be reduced during the search. Deleting those paths that have high average cost per unit length, or deleting paths that are too short whenever the total number of nodes in the OPEN list exceeds a defined limit, usually gives good results.

• **Least maximum cost:** The strength of a chain may be given by the strength of the weakest element—this idea is included in cost function computations. The cost of the current path is then set as the cost of the most expensive arc in the path from the starting node to the current node, whatever the sum of costs along the path. The path cost does not therefore necessarily grow with each step, and this is what favors expansion of good paths for a longer time.

• **Branch and bound:** This modification is based on maximum allowed cost of a path, no path

being allowed to exceed this cost. This maximum path cost is either known beforehand or it is computed and updated during the graph search. All the paths that exceed the allowed maximum path cost are deleted from the OPEN list.

• **Lower bound:** Another way to increase the search speed is to reduce the number of poor edge candidate expansions. Poor edge candidates are always expanded if the cost of the best current path exceeds that of any worse but shorter path in the graph. If the cost of the best successor is set to zero, the total cost of the path does not grow after the node expansion and the good path will be expanded again.

• **Multi-resolution processing:** The number of expanded nodes can be decreased if a sequence of two graph search processes is applied. The first search is done in lower resolution, therefore a smaller number of graph nodes is involved in the search and a smaller number is expanded, compared to full resolution. The low-resolution search detects an approximate boundary. The second search is done in full resolution using the low-resolution results as a model, and the full-resolution costs are weighted by a factor representing the distance from the approximate boundary acquired in low resolution (equation F.20)). The weighting function should increase with the distance in a non-linear way. This approach assumes that the approximate boundary location can be detected from the low-resolution image .

• **Incorporation of higher-level knowledge:** Including higher-level knowledge into the graph search may significantly decrease the number of expanded nodes. The search may be directly guided by a priori knowledge of approximate boundary position. Another possibility is to incorporate a boundary shape model into the cost function Computation. Graph searching techniques offer a convenient way to ensure global optimality of the detected contour. This technique has often been applied to the detection of approximately straight contours. The detection of closed structure contours would involve geometrically transforming the image using a polar-to-rectangular co-ordinate transformation in order to 'straighten' the contour, but this may prevent the algorithm from detecting the non-convex parts of the contour. To overcome this problem, the image may be divided into two segments (iteratively, if necessary) and separate, simultaneous searches can be conducted in each segment. The searches are independent and proceed in opposite directions from a start point until they meet at the dividing line between the two image segments.

The approaches discussed above search for optimal borders between two specific image points, but searching for all the borders in the image without knowledge of the start and end points is more complex. In an approach based on magnitudes and directions of edges in the image, edges are merged into edge chains (i.e., partial borders). Edge chains are constructed by applying a bi-directional heuristic search in which half of each 8-neighborhood expanded node is considered as lying in front of the edge, the second half as lying behind the edge (see Figure 6.26).

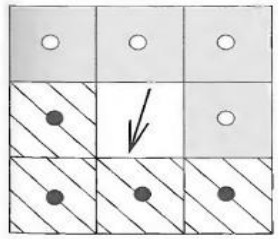


Figure 6.26: Bidirectional heuristic search: Edge predecessors (marked ○) and successors (marked •).

Algorithm12: Heuristic search for image borders

1. Search for the strongest edge in the image, not considering edges previously marked or edges that already are part of located borders. Mark the located edge. If the edge magnitude is less than the preset threshold or if no image edge was found, go to step 5.
2. Expand all the image edges positioned in front of the specified starting edge until no new successors can be found.
3. Expand all the image edges positioned behind the specified starting edge until no new predecessors can be found. In steps 2 and 3 do not include edges that are already part of any existing edge chain.
4. If the resulting edge chain consists of at least three edges, it is stored in the chain list, otherwise it is deleted. Proceed to step 1.
5. Modify edge chains according to the rules given in Figure 6.27.
6. Repeat step 5 until the resulting borders do not change substantially from step to step. Border detection as dynamic programming

Dynamic programming is an optimization method based on the principle of optimality . It searches for optima of functions in which not all variables are simultaneously interrelated.

Consider the following simple boundary-tracing problem (Figure 6.28). The aim is to find the best path (minimum cost) between one of the possible start points A, B. C and one of the possible ending points G. H, I. The boundary must be contiguous in 8-connectivity. The graph representing the problem, together with assigned partialcosts, is shown in Figure 6.28a,b. As can be seen, there are three ways to get to the node E. Connecting A E gives the cost g(A, E) = 2; connecting BE, cost g(B, E) = 6; connecting C-E, cost g(C,E) = 3.

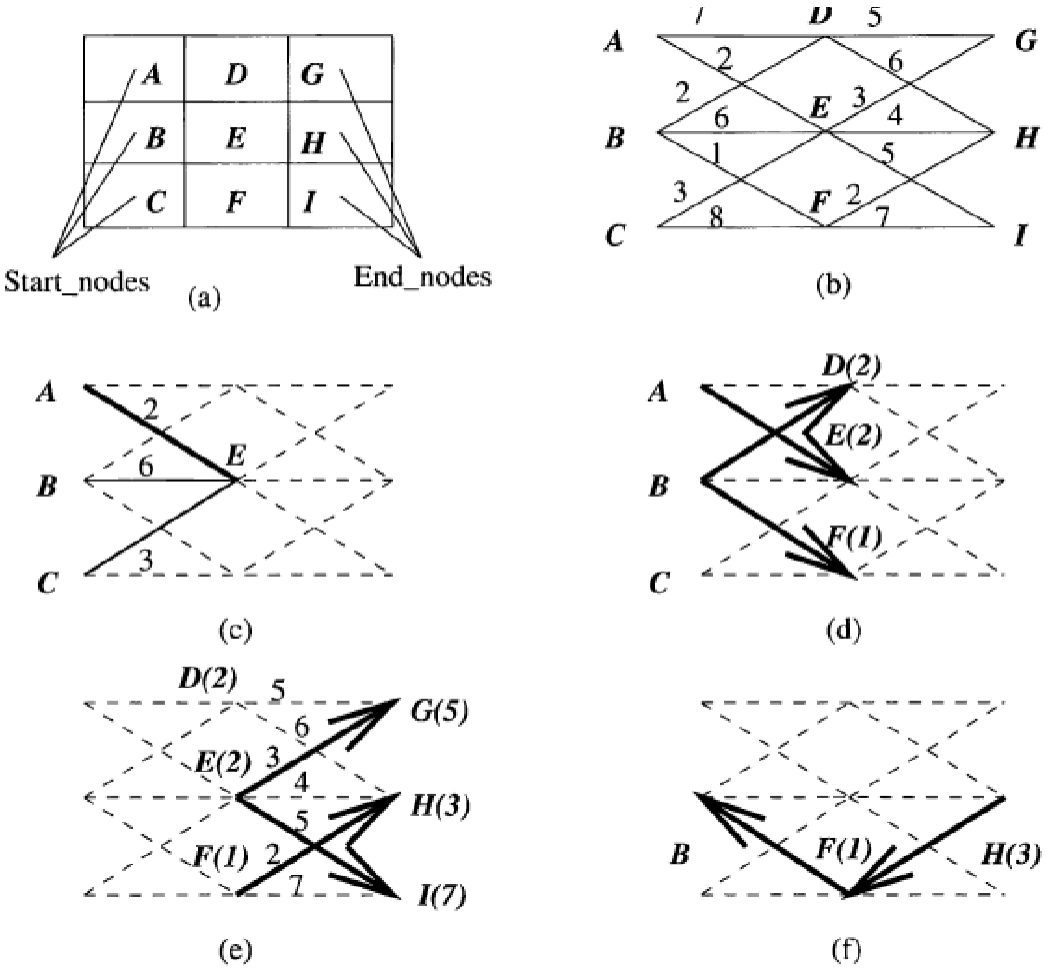


Figure 6.28: Edge following as dynamic programming. (a) Edge image. (b) Corresponding graph, partial costs assigned. (c) Possible paths from any start point to E, $A-E$ is optimal. (d) Optimal partial paths to nodes D, E, F. (e) Optimal partial paths to nodes G, H, I. (f) Back-tracking from H defines the optimal boundary.

optimization as shown in Figure 6.30a

$$C(x_k^{m+1}) = \min_i \left(C(x_i^m) + g^m(i,k) \right), \tag{6.22}$$

where $C(x_k^{m+1})$ is the new cost assigned to the node x_k^{m+1}, and $g^m(i,k)$ is the partial path cost between nodes x_i^m and x_k^{m+1}. For the complete optimization problem,

$$\min \left(C(x^1, x^2, \ldots, x^M) \right) = \min_{k=1,\ldots,n} \left(C(x_k^M) \right), \tag{6.23}$$

where x_k^M are the end point nodes, M is the number of graph layers between start points and end points (see Figure 6.30b), and $C(x^1, x^2, \ldots, x^M)$ denotes the cost of a path between the first and the last (M^{th}) graph layer. Requiring an 8-connected border and assuming n nodes x_i^m in each graph layer m, $3n$ cost combinations must be computed for each layer, $3n(M-1)+n$ being the total number of cost combination computations. Compared to the brute-force enumerative search, where $n(3^{M-1})$ combinations must be

The main idea of the principle of optimality is: Whatever the path to the node E was, there exists an optimal path between E and the end point. In other words, if the optimal path start point endpoint goes through E, then both its parts start point-E and E-end point, are also optimal

In our case, the optimal path between the start point and E is the partial path A E (see Figure 6.28c). Only the following information need be stored for future use; to get to E, the optimal path is A-E, cost C(E) = 2. Using the same approach, to get to D

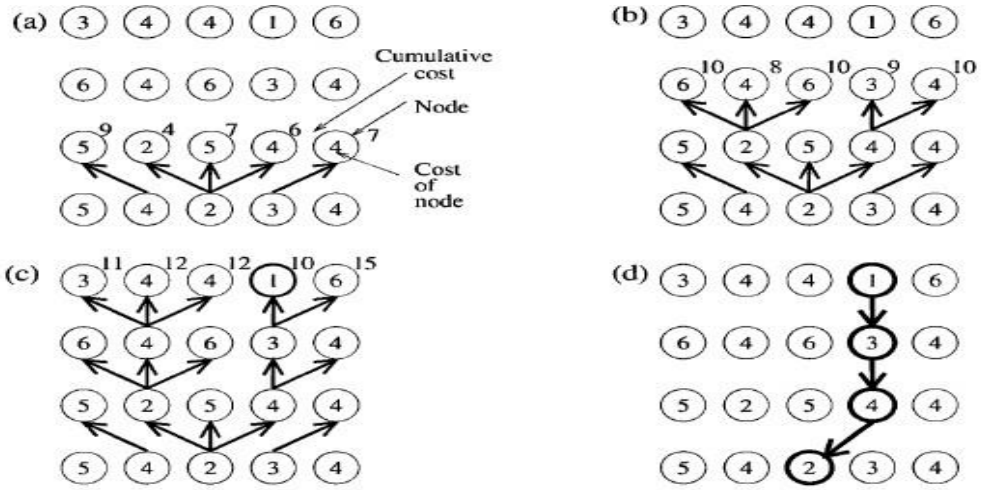


Figure 6.29: Example of a graph searching sequence using dynamic programming: (a) step 1, expansion of the first graph layer; (b) step 2; (c) step 3—the minimum-cost node in the last layer marked; (d) the optimal path is defined by back-tracking.

The optimal path is B-D, cost C(D) = 2; the best path to F is B-F, cost C(F) = 1 (see Figure 6.28d). The path may get to node G from either D or E. The cost of the path through the node D is a sum of the cumulative cost C(D) of the node D and the partial path cost g(D, G). This cost C(Gp) = 7 represents the path B-D-G because the best path to D is from B. The cost to get to G from E is C(Ge) = 5, representing the path A-E-G. It is obvious that the path going through node E is better, the optimal path to G is the path A-E-G with cost C(G) = 5 (see Figure 6.28e). Similarly, cost C(H) = 3 (B-F-H) and cost C(I) = 7 (A-E-I). Now, the end point with the minimum path cost represents the optimum path; node H is therefore the optimal boundary end point, and the optimal boundary is B-F H (see Figure 6.28f). Figure 6.29 gives an example in which node costs are used (not arc costs as in Figure 6.28). Note that the graph, the cost function, and the resulting path are identical to those used in Figure 6.21. If the graph has more layers (in Figure 6.28 just three layers were present), the

process is repeated until one of the end points is reached. Each repetition consists of a simpler Hough transforms.

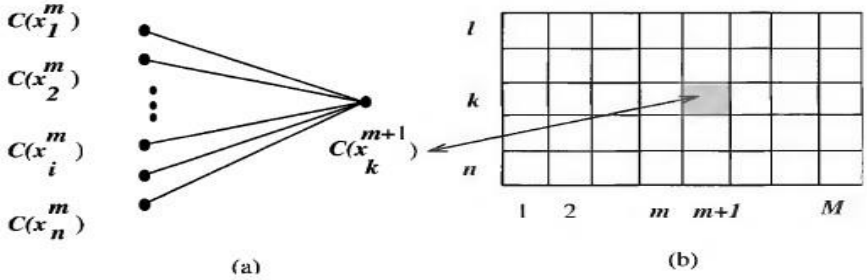


Figure 6.30: Dynamic programming: (a) one step of the cost calculation; (b) graph layers, node notation.

If an image consists of objects with known shape and size, segmentation can be viewed as a problem of finding this object within an image. Typical tasks are to locate circular pads in printed circuit boards, or to find objects of specific shapes in aerial or satellite data, etc. One of many possible ways to solve these problems is to move a mask with an appropriate shape and size along the image and look for correlation between the image and the mask.Unfortunately, the specified mask often differs too much from the object's representation in the processed data, because of shape distortions, rotation, zoom, etc.

Hough Transform

One very effective method that can solve this problem is the Hough transform, which can even be used successfully in segmentation of overlapping or semi-occluded objects. The original Hough transform was designed to detect straight lines and curves [Hough, 1962], and this original method can be used if analytic equations of object borderlines are known—no prior knowledge of region position is necessary. A big advantage of this approach is robustness of segmentation results; that is, segmentation is not too sensitive to imperfect data or noise. Nevertheless, it is often impossible to get analytic expressions describing borders. Later, a generalized Hough transform will be described that can find objects even if an analytic expression of the border is not known. The basic idea of the method can be seen from the simple problem of detecting a straight line in an image. A straight line is defined by two points A — (?1,2/1) and D — {x2,y2) (shown in Figure 6.33a). All straight lines going through the point A are given by the expression y\ — kxy + q for some values of k and q. This means that the same equation can be interpreted as an equation in the parameter space fc, q\ all the straight lines going through the point A are then represented by the

equation q = —x\k -f y\ (see Figure 6.33b). Straight lines going through the point B can likewise be represented as q = —x2k + 3/2- The only common point of both straight lines in the A;, q parameter space is the point which in the original image space represents the only existing straight line connecting points A and B.

This means that any straight line in the image is represented by a single point in the k,q parameter space and any part of this straight line is transformed into the same point. The main idea of line detection is to determine all the possible line pixels in the image, to transform all lines that can go through these pixels into corresponding points in the parameter space, and to detect the points (a, b) in the parameter space that frequently resulted from the Hough transform of lines y = ax + b in the image. These main steps will be described in more detail. Detection of all possible line pixels in the image may be achieved by applying an edge detector to the image; then, all pixels with edge magnitude exceeding some threshold can be considered possible line pixels (referred to as edge pixels below). In the most general case, nothing is known about lines in the image, and therefore lines of any direction may go through any of the edge pixels. In reality, the number of these lines is infinite; however, for practical purposes, only a limited number of line directions may be considered.

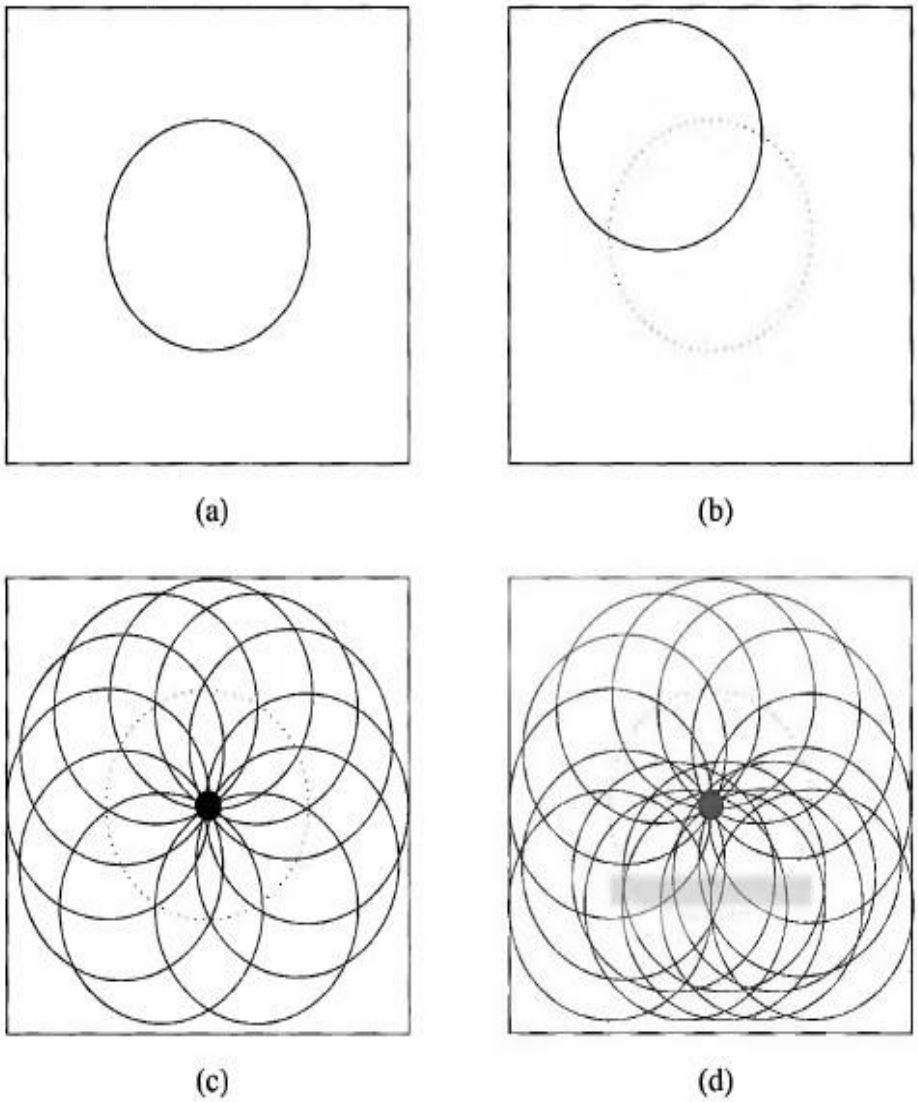


Figure 6.32: Hough transform—example of circle detection. (a) Original image of a dark circle (known radius r) on a bright background. (b) For each dark pixel, a potential circle-center locus is defined by a circle with radius r and center at that pixel. (c) The frequency with which image pixels occur in the circle-center loci is determined—the highest-frequency pixel represents the center of the circle (marked by •). (d) The Hough transform correctly detects the circle (marked by •) in the presence of incomplete circle information and overlapping structures. (See Figure 6.37 for a real-life example.)

The possible directions of lines define a discretization of the parameter k. Similarly, the parameter q is sampled into a limited number of values. The parameter space is not continuous any

more, but rather is represented by a rectangular structure of cells. This array of cells is called the accumulator array A, whose elements are accumulator cells A(k,q).

For each edge pixel, parameters k, q are determined which represent lines of allowed directions going through this pixel. For each such line, the values of line parameters k. q are used to increase the value of the accumulator cell A(k,q). Clearly, if a line represented by an equation y = ax+b is present in the image, the value of the accumulator cell A(a, b) will be increased many times—as many times as the line y = ax + b is detected as a line possibly going through any of the edge pixels. For any pixel P, lines going through it may have any direction k (from the set of allowed directions), but the second parameter q is constrained by the image co-ordinates of the pixel P and the direction k. Therefore, lines existing in the image will cause large values of the appropriate accumulator cells in the image, while other lines possibly going through edge pixels, which do not correspond to lines existing in the image, have different fc, q parameters for each edge pixel, and therefore the corresponding accumulator cells are increased only rarely. In other words, lines existing in the image may be detected as high-valued accumulator cells in the accumulator array, and the parameters of the detected line are specified by the accumulator array co-ordinates. As a result, line detection in the image is transformed to detection of local maxima in the accumulator space.

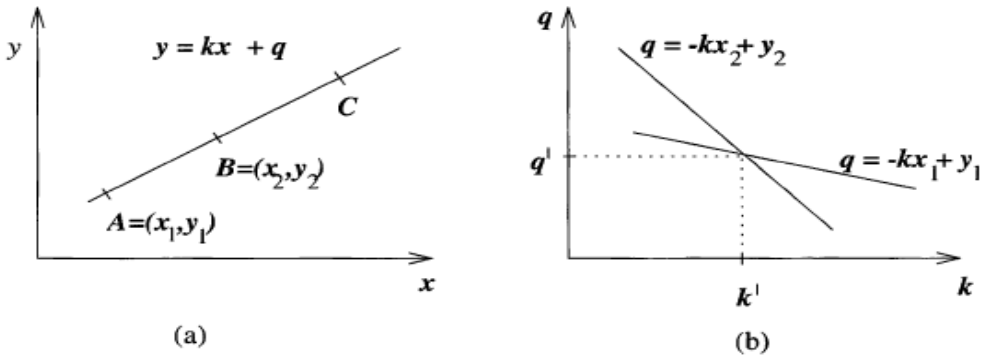


Figure 6.33: Hough transform principles. (a) Image space. (b) k, q parameter space.

It has been noted that an important property of the Hough transform is its insensitivity to missing parts of lines, to image noise, and to other non-line structures co-existing in the image. Insensitivity to data imprecision and noise can be seen in Figure 6.34. This is caused by the robustness of

transformation from the image space into the accumulator space—a missing part of the line will cause only a lower local maximum because a smaller number of edge pixels contributes to the corresponding accumulator cell. A noisy or only approximately straight line will not be transformed into a point in the parameter space, but rather will result in a cluster of points, and the cluster center of gravity can be considered the straight line representation.

Note that the parametric equation of the line y = kx + q is appropriate only for explanation of the Hough transform principles—it causes difficulties in vertical line detection (fc —> oo) and in non-linear discretization of the parameter k. If a line is represented as s = x cos 0 + y sin 0 , F.25) the Hough transform does not suffer from these limitations. Again, the straight line is transformed to a single point (see Figure 6.35). A practical example showing the segmentation of an MR image of the brain into the left and right hemispheres is given in

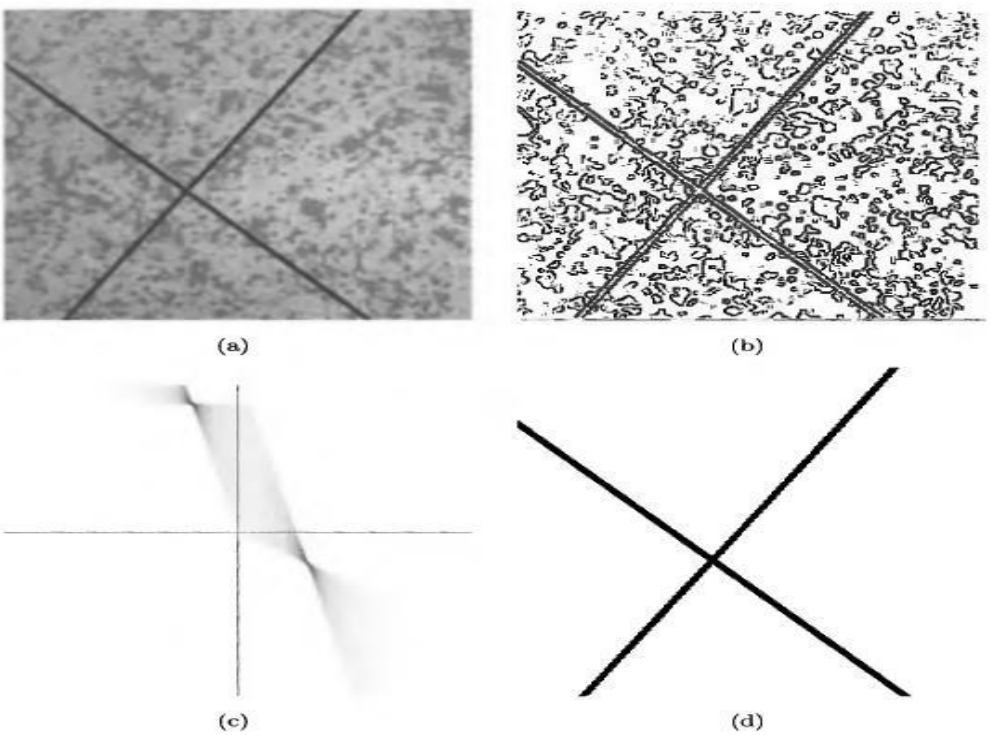


Figure 6.34: Hough transform—line detection. (a) Original image. (b) Edge image (note many edges, which do not belong to the line). (c) Parameter space. (d) Detected lines.

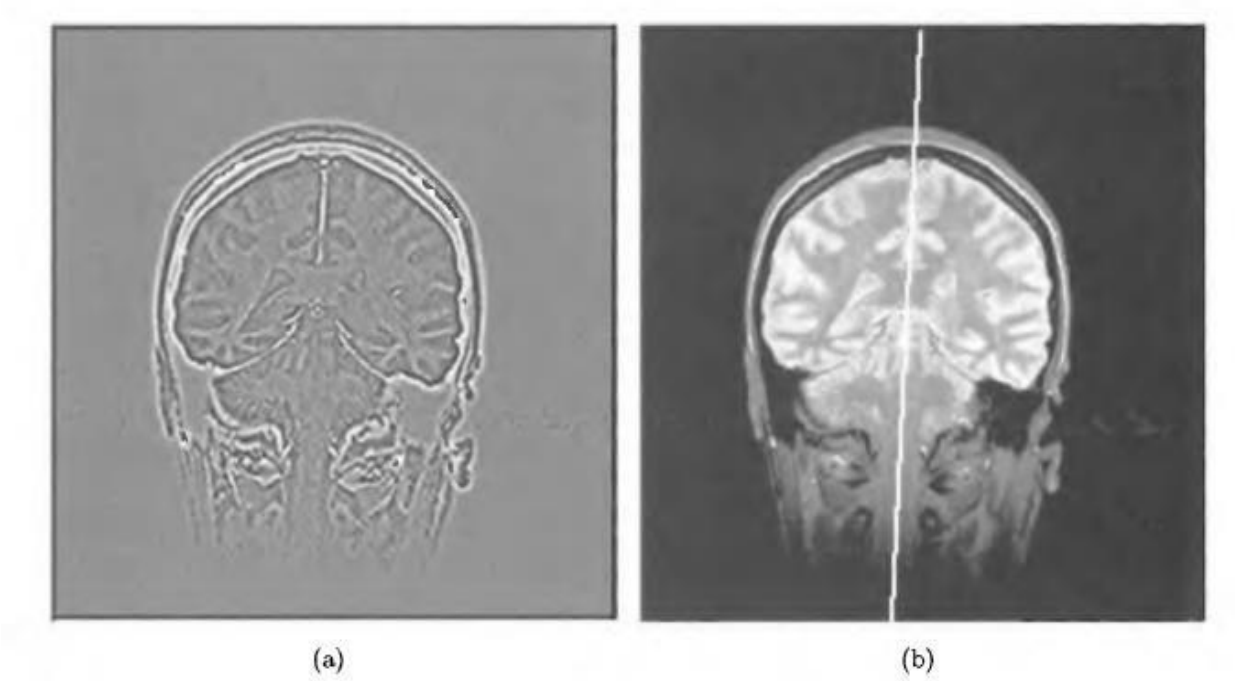


Figure 6.36: Hough transform line detection used for MRI brain segmentation to the left and right hemispheres. (a) Edge image. (b) Segmentation line in original image data.

Curve detection using the Hough transform

1. Quantize parameter space within the limits of parameters **a**. The dimensionality n of the parameter space is given by the number of parameters of the vector **a**.
2. Form an n-dimensional accumulator array $A(\mathbf{a})$ with structure matching the quantization of parameter space; set all elements to zero.
3. For each image point (x_1, x_2) in the appropriately thresholded gradient image, increase all accumulator cells $A(\mathbf{a})$ if $f(\mathbf{x}, \mathbf{a}) = 0$

$$A(\mathbf{a}) = A(\mathbf{a}) + \Delta A$$

for all **a** inside the limits used in step 1.

4. Local maxima in the accumulator array $A(\mathbf{a})$ correspond to realizations of curves $f(\mathbf{x}, \mathbf{a})$ that are present in the original image.

If we are looking for circles, the analytic expression $f(\mathbf{x}, \mathbf{a})$ of the desired curve is

$$(x_1 - a)^2 + (x_2 - b)^2 = r^2 , \qquad (6.26)$$

Border detection using border location information

If any information about boundary location or shape is known, it is of benefit to use it. The information may, for instance, be based on some higher-level knowledge, or can result from segmentation applied to a lower-resolution image.

One possibility is to determine a boundary in an image as the location of significant edges positioned close to an assumed border if the edge directions of these significant edges match the assumed boundary direction. The new border pixels are sought in directions perpendicular to the assumed border (see Figure 6.39). If a large number of border elements satisfying the given conditions is found, an approximate curve is computed based on these pixels, and a new, more accurate, border results. Another possibility is based on prior knowledge of end points—this approach assumes low image noise and relatively straight boundaries. The process iteratively partitions the border and searches for the strongest edge located on perpendiculars to the line

connecting end points of each partition; perpendiculars are located at the center of the connecting straight line. The strongest significant edge located on the perpendicular that is close to the straight line connecting the end points of the current partition is accepted as a new border element. The iteration process is then repeated.

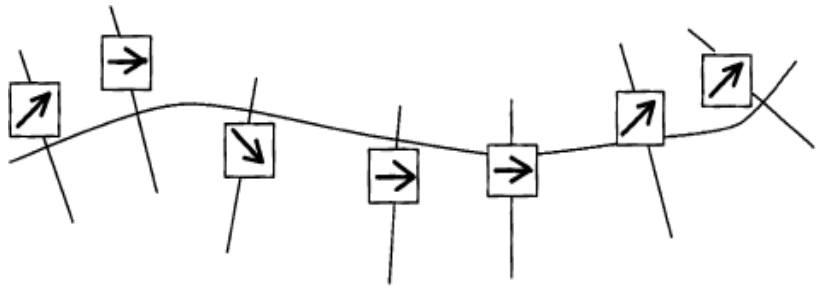


Figure 6.39: A priori information about boundary location.

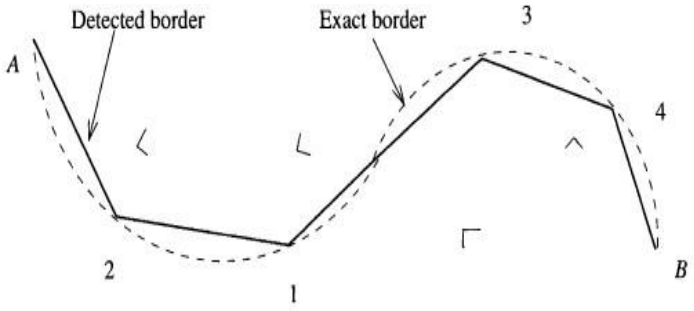


Figure 6.40: Divide-and-conquer iterative border detection; numbers show the sequence of division steps.

8 Region construction from borders

All methods considered hitherto have focused on the detection of borders that partially or completely segmented the processed image. If a complete segmentation is achieved, the borders segment an image into regions; but if only a partial segmentation results, regions are not defined uniquely and region determination from borders may be a very complex task requiring cooperation with higher-level knowledge. However, methods exist that are able to construct regions from partial borders which do not form closed boundaries. These methods do not always find acceptable regions, but they are useful in many practical situations.

One of them is the superslice method, which is applicable if regions have dominant gray-

level properties. The approach assumes that some border-part locations are known in the image; the image data is then thresholded using different thresholds. Regions resulting from the thresholding for which the detected boundaries best coincide with assumed boundary segments are then accepted as correct.

Algorithm 16: Region forming from partial borders

1. For each border pixel x, search for an opposite edge pixel within a distance not exceeding a given maximum M. If an opposite edge pixel is not found, process the next border pixel in the image. If an opposite edge pixel is found, mark each pixel on the connecting straight line as a potential region member.

2. Compute the number of markers for each pixel in the image (the number of markers tells how often a pixel was on a connecting line between opposite edge pixels). Let 6(x) be the number of markers for the pixel x.

3. The weighted number of markers B(x) is then determined as follows:

$$\begin{aligned} B(\mathbf{x}) &= 0.0 \quad \text{for } b(\mathbf{x}) = 0\,, \\ &= 0.1 \quad \text{for } b(\mathbf{x}) = 1\,, \\ &= 0.2 \quad \text{for } b(\mathbf{x}) = 2\,, \\ &= 0.5 \quad \text{for } b(\mathbf{x}) = 3\,, \\ &= 1.0 \quad \text{for } b(\mathbf{x}) > 3\,. \end{aligned} \tag{6.28}$$

The confidence that a pixel x is a member of a region is given as the sum $\sum_i B(\mathbf{x}_i)$ in a 3x3 neighborhood of the pixel x. If the confidence that a pixel x is a region member is one or larger, then pixel x is marked as a region pixel, otherwise it is marked as a background pixel.

Note that this method allows the construction of bright regions on a dark background as well as dark regions on a bright background by taking either of the two options in the search for opposite edge pixels—step 1. Search orientation depends on whether relatively dark or bright regions are constructed. If $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$ are directions of edges, the condition that must be satisfied for x and y to be opposite is

$$\frac{\pi}{2} < \left| (\phi(\mathbf{x}) - \phi(\mathbf{y})) \bmod (2\pi) \right| < \frac{3\pi}{2}\,.$$

Note that it is possible to take advantage of prior knowledge of maximum region

sizes—this information defines the value of M in step 1 of the algorithm, the maximum search length for the opposite edge pixel.

3 Region-based segmentation

The aim of the segmentation methods described in the previous section was to find borders between regions; the following methods construct regions directly. It is easy to construct regions from their borders, and it is easy to detect borders of existing regions. However, segmentations resulting from edge-based methods and region-growing methods are not usually exactly the same, and a combination of results may often be a good idea. Region growing techniques are generally better in noisy images, where borders are extremely difficult to detect. Homogeneity is an important property of regions and is used as the main segmentation criterion in region growing, whose basic idea is to divide an image into zones of maximum homogeneity. The criteria for homogeneity can be based on gray-level, color, texture, shape, model (using semantic information), etc. Properties chosen to describe regions influence the form, complexity, and amount of prior information in the specific region-growing segmentation method. Methods that specifically address region-growing segmentation of color images. Further assumptions needed in this section are that regions must satisfy the following conditions:

$$H(R_i) = \text{TRUE}, \quad i = 1, 2, \ldots, S, \tag{6.30}$$

$$H(R_i \cup R_j) = \text{FALSE}, \quad i \neq j, \quad R_i \text{ adjacent to } R_j, \tag{6.31}$$

where S is the total number of regions in an image and H(Ri) is a binary homogeneity evaluation of the region Ri. Resulting regions of the segmented image must be both homogeneous and maximal, where by 'maximal' we mean that the homogeneity criterion would not be true after merging a region with any adjacent region.

(Here 6.41 Figure)

We will discuss simpler versions of region growing first, that is, the merging, splitting, and split-and-merge approaches. Of especial interest are the homogeneity criteria, whose choice is the most important factor affecting the methods mentioned; general and specific heuristics may also be incorporated. The simplest homogeneity criterion uses an average

gray-level of the region, its color properties, simple texture properties, or an m-dimensional vector of average gray values for multi-spectral images. Considering three-dimensional connectivity constraints, homogeneous regions (volumes) of a three-dimensional image can be determined using three-dimensional region growing

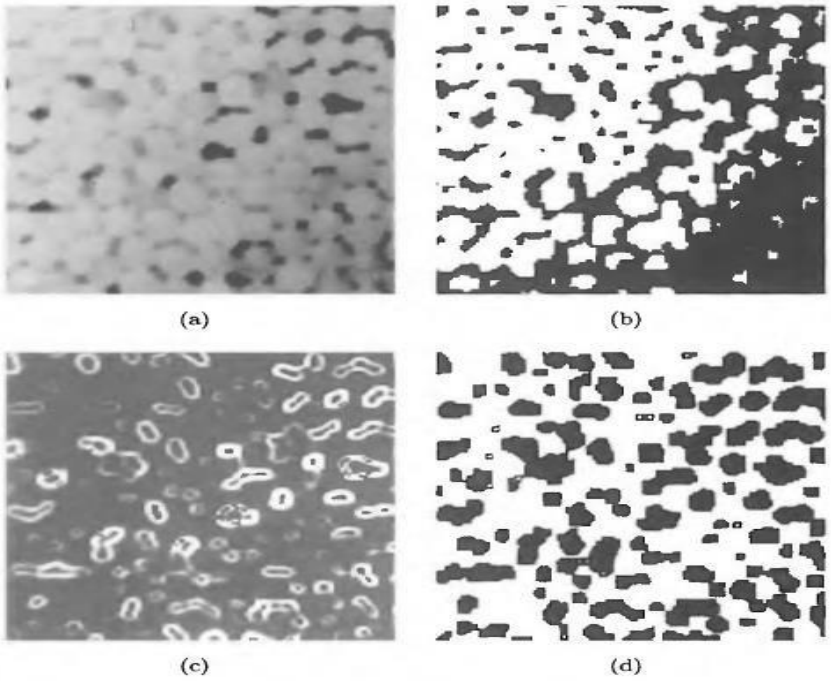


Figure 6.41: Region forming from partial borders. (a) Original image. (b) Thresholding. (c) Edge image. (d) Regions formed from partial borders.

Region merging

The most natural method of region growing is to begin the growth in the raw image data, each pixel representing a single region. These regions almost certainly do not satisfy the condition of equation 6.31, and so regions will be merged as long as equation 6.30. remains satisfied.

Algorithm 17: Region merging (outline)

1. Define some starting method to segment the image into many small regions satisfying condition (6.30).
2. Define a criterion for merging two adjacent regions.
3. Merge all adjacent regions satisfying the merging criterion. If no two regions can

be merged maintaining condition (6.30), stop.

This algorithm represents a general approach to region merging segmentation. Specific methods differ in the definition of the starting segmentation and in the criterion for merging. In the descriptions that follow, regions are those parts of the image that can be sequentially merged into larger regions satisfying equations (6.30) and (6.31). The result

of region merging usually depends on the order in which regions are merged, meaning that segmentation results will probably differ if segmentation begins, for instance, in the upper left or lower right corner. This is because the merging order can cause two similar adjacent regions R1 and R2 not to be merged, since an earlier merge used Ri and its new characteristics no longer allow it to be merged with region R2. If the merging process used a different order, this merge may have been realized.

The simplest methods begin merging by starting the segmentation using regions of 2x2, 4x4, or 8x8 pixels. Region descriptions are then based on their statistical gray-level properties—a regional gray-level histogram is a good example. A region description is compared with the description of an adjacent region; if they match, they are merged into a larger region and a new region description is computed. Otherwise, regions are marked as non-matching. Merging of adjacent regions continues between all neighbors, including newly formed ones. If a region cannot be merged with any of its neighbors, it is marked 'final'; the merging process stops when all image regions are so marked.

State space search is one of the essential principles of problem solving in AI, According to this approach, pixels of the raw image are considered the starting state, each pixel being a separate region. A change of state can result from the merging of two regions or the splitting of a region into sub-regions. The problem can be described as looking for permissible changes of state while producing the best image segmentation. This state space approach brings two advantages; first, well-known methods of state space search can be applied which also include heuristic knowledge; second, higher-level data structures can be used which allow the possibility of working directly with regions and their borders, and no longer require the marking of each image element according to its region marking. Starting regions are formed by pixels of the same gray-level—these starting regions are small in real images. The first state changes are based on crack edge computations (Section

2.3.1), where local boundaries between regions are evaluated by the strength of crack edges along their common border. The data structure used in this approach (the so-called supergrid) carries all the necessary information (see Figure 6.42);
this allows for easy region merging in 4-adjacency when crack edge values are stored in the so' elements. Region merging uses the following two heuristics.

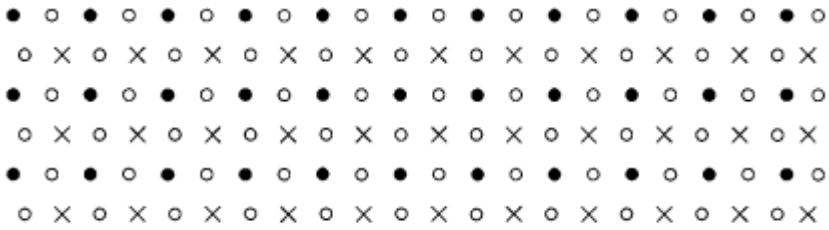


Figure 6.42: Supergrid data structure: ×,image data; ○, crack edges; •, unused.

• Two adjacent regions are merged if a significant part of their common boundary consists of weak edges (significance can be based on the region with the shorter perimeter; the ratio of the number of weak common edges to the total length of the region perimeter may be used).

• Two adjacent regions are also merged if a significant part of their common boundary consists of weak edges, but in this case not considering the total length of the region borders.

Of the two given heuristics, the first is more general and the second cannot be used alone because it does not consider the influence of different region sizes.

Edge significance can be evaluated according to the formula

$$\begin{aligned} v_{ij} &= 0 \quad \text{if } s_{ij} < T_1, \\ &= 1 \quad \text{otherwise,} \end{aligned} \tag{6.32}$$

where $v_{ij} = 1$ indicates a significant edge, $v_{ij} = 0$ a weak edge, T_1 is a preset threshold, and s_{ij} is the crack edge value $s_{ij} = \left| f(\mathbf{x}_i) - f(\mathbf{x}_j) \right|$.

Algorithm 6.18: Region merging via boundary melting

1. Define a starting image segmentation into regions of constant gray-level. Construct a supergrid edge data structure in which to store the crack edge information.

2. Remove all weak crack edges from the edge data structure (using equation 6.32) and threshold Ti).

3. Recursively remove common boundaries of adjacent regions R_i, R_j, if

$$\frac{W}{\min(l_i, l_j)} \geq T_2 ,$$

where W is the number of weak edges on the common boundary, l_i, l_j are the perimeter lengths of regions R_i, R_j, and T_2 is another preset threshold.

4. Recursively remove common boundaries of adjacent regions R_i, R_j if

$$\frac{W}{l} \geq T_3 \qquad (6.33)$$

or, using a weaker criterion [Ballard and Brown, 1982]

$$W \geq T_3 , \qquad (6.34)$$

where l is the length of the common boundary and T_3 is a third threshold.

Note that even if we have described a region growing method, the merging criterion is based on border properties and so the merging does not necessarily keep condition 6.30) true. The super grid data structure allows precise work with edges and borders, but a big disadvantage of this data structure is that it is not suitable for the representation of regions—it is necessary to refer to each region as a part of the image, especially if semantic information about regions and neighboring regions is included. This problem can be solved by the construction and updating of a data structure describing region adjacencies and their boundaries, and for this purpose a good data structure to use can be a planar-region adjacency graph and a dual-region boundary graph, Figure 6.43 gives a comparison of region merging methods. An original image and its pseudo-color representation (to see the small gray-level differences) are given in Figures 6.43a,b.

The original image cannot be segmented by thresholding because of the significant and continuous gray-level gradient in all regions. Results of a recursive region merging method, which uses a simple merging criterion that allows pixels to be merged in the row-first fashion as long as they do not differ by more than a pre-specified parameter from the seed pixel is shown in Figure 6.43c; note the resulting horizontally elongated regions corresponding to vertical changes of image gray-levels. If region merging via boundary melting is applied, the segmentation results improve dramatically;

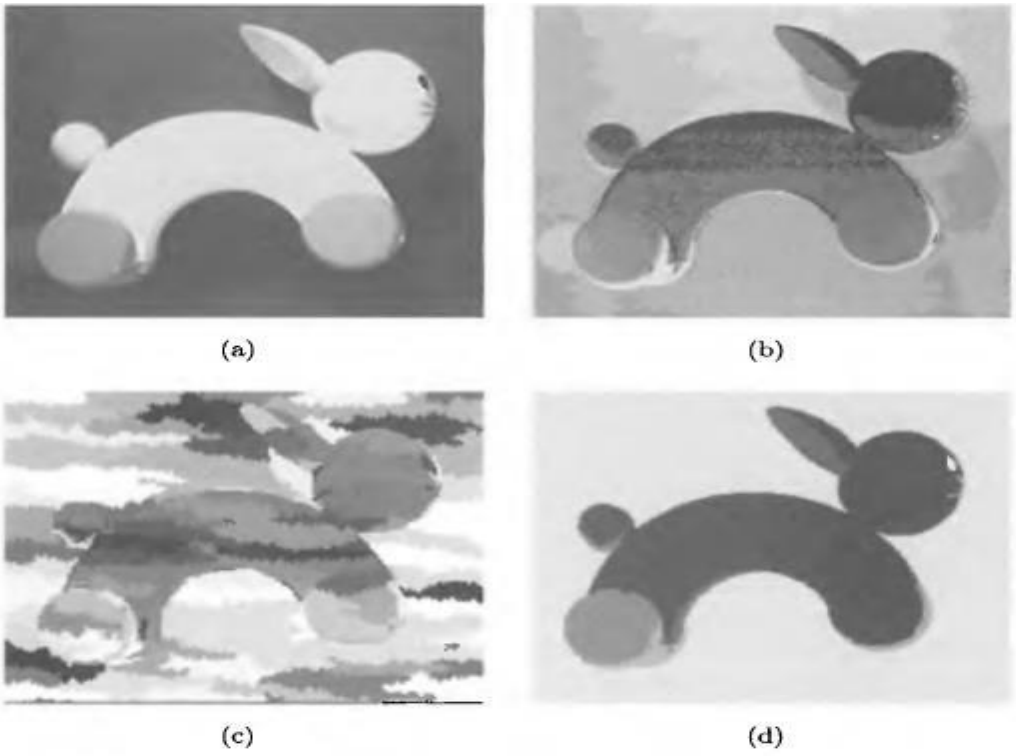


Figure 6.43: Region merging segmentation. (a) Original image. (b) Pseudo-color representation of the original image (in grayscale). (c) Recursive region merging. (d) Region merging via boundary melting. *Courtesy of R. Marik, Czech Technical University.*

Region splitting

Region splitting is the opposite of region merging, and begins with the whole image represented as a single region which does not usually satisfy condition 6.30. Therefore, the existing image regions are sequentially split to satisfy 6.1, 6.30 and 6.31. Even if this approach seems to be dual to region merging, region splitting does not result in the same segmentation even if the same homogeneity criteria are used. Some regions may be homogeneous during the splitting process and therefore are not split any more; considering the homogeneous regions created by region merging

procedures, some may not be constructed because of the impossibility of merging smaller sub-regions earlier in the process. A fine black-and-white chessboard is an example: Let a homogeneity criterion be based on variance of average gray-levels in the quadrants of the evaluated region in the next lower pyramid level —if the segmentation process is based on region splitting, the image will not be split into sub-regions because its quadrants would have the same value of the measure as the starting region consisting of the whole image. The region merging approach, on the other hand, begins with merging single pixel regions into larger regions, and this process will stop when regions match the chessboard squares. Thus, if splitting is applied, the whole image will be considered one region; whereas if merging is applied, a chessboard will be segmented into squares as shown in Figure 6.44. In this particular case, considering gray-level variance within the entire region as a measure of region homogeneity, and not considering the variance of quadrants only, would also solve the problem. However, region merging and region splitting are not dual.

Region splitting methods generally use similar criteria of homogeneity as region merging methods, and differ only in the direction of their application. The multi-spectral segmentation discussed in considering thresholding can be seen as an example of a region splitting method. As mentioned there, other criteria can be used to split regions (e.g., cluster analysis, pixel classification, etc.).

Splitting and merging

A combination of splitting and merging may result in a method with the advantages of both approaches. Split-and-merge approaches work using pyramid image representations; regions are square shaped and correspond to elements of the appropriate pyramid level.

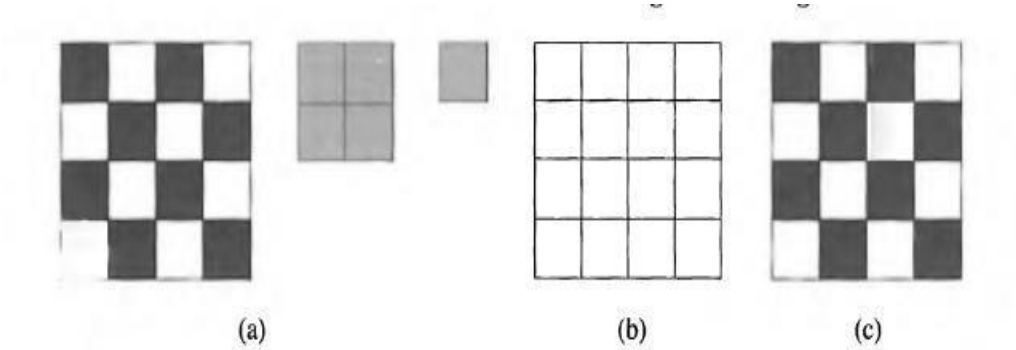


Figure 6.44: Different segmentations may result from region splitting and region merging approaches. (a) Chessboard image, corresponding pyramid. (b) Region splitting segmentation (upper pyramid level is homogeneous, no splitting possible). (c) Region merging segmentation (lowest pyramid level consists of regions that cannot be merged).

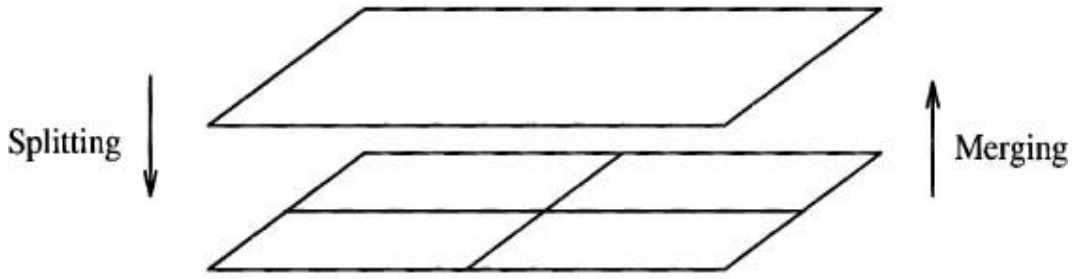


Figure 6.45: Split-and-merge in a hierarchical data structure.

If any region in any pyramid level is not homogeneous (excluding the lowest level), it is split into four sub-regions—these are elements of higher resolution at the level below. If
four regions exist at any pyramid level with approximately the same value of homogeneity measure, they are merged into a single region in an upper pyramid level (see Figure 6.45).

The segmentation process can be understood as the construction of a segmentation quad tree where each leaf node represents a homogeneous region—that is, an element of some pyramid level. Splitting and merging corresponds to removing or storing information building parts of the segmentation quadtree—the number of leaf nodes of the tree corresponds to the number of segmented regions after the segmentation process is over. These approaches are sometimes called split-arid-link methods if they use segmentation trees for about adjacent regions. Split-and-merge methods usually store the adjacency information in region adjacency graphs (or similar data structures).

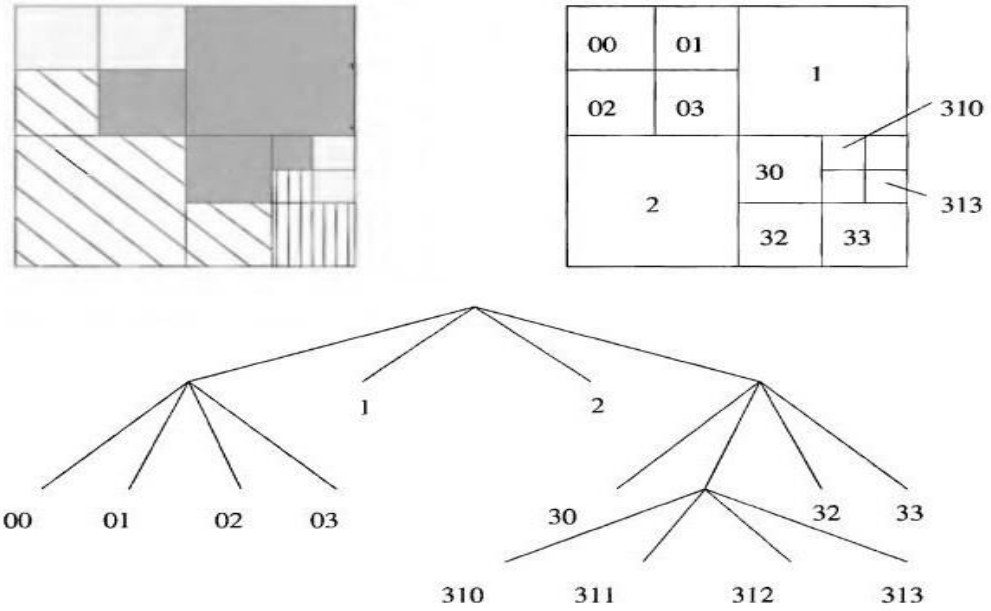


Figure 6.46: Segmentation quadtree.

Using segmentation trees, in which regions do not have to be contiguous, An unpleasant drawback of segmentation quadtrees is the square- region shape assumption (see Figure 6.46), and it is therefore advantageous to add more processing steps that permit the merging of regions which are not part of the same branch of the segmentation tree. Starting image regions can either be chosen arbitrarily or can be based on prior knowledge. Because both split-and-merge processing options are available, the starting segmentation does not have to satisfy either condition 6.30 or 6.31.

Algorithm 19: Split and merge

1. Define an initial segmentation into regions, a homogeneity criterion, and a pyramid data structure.
2. If any region R in the pyramid data structure is not homogeneous [$H(R)$ = FALSE], split it into four child-regions; if any four regions with the same parent can be merged into a single homogeneous region, merge them. If no region can be split or merged, go to step 3.
3. If any two adjacent regions R_i, R_j (even if they are in different pyramid levels or do not have the same parent) can be merged into a homogeneous region, merge them.
4. Merge small regions with the most similar adjacent region if it is necessary to remove small-size regions.

A pyramid data structure with overlapping regions is an interesting modification of this method. In this data structure, each region has four potential parent elements in the upper pyramid level and 16 possible child elements in the lower pyramid level. Segmentation tree generation begins in the lowest pyramid level. Properties of each region are compared with properties of each of its potential parents and the segmentation branch is linked to the most similar of them. After construction of the tree is complete, all the homogeneity values of all the elements in the pyramid data structure are recomputed to be based on child-region properties only.

This recomputed pyramid data structure is used to generate a new segmentation tree, beginning again at the lowest level. The pyramid updating process and new segmentation

tree generation is repeated until no significant segmentation changes can be detected between steps. Assume that the segmented image has a maximum of 2n (non-contiguous)

regions. Any of these regions must link to at least one element in the highest allowed pyramid level—let this pyramid level consist of 2n elements. Each element of the highest

pyramid level corresponds to one branch of the segmentation tree, and all the leaf nodes of this branch construct one region of the segmented image. The highest level of the segmentation tree must correspond to the expected number of image regions, and the pyramid height defines the maximum number of segmentation branches. If the number of regions in an image is less than 2n,

some regions can be represented by more than one element in the highest pyramid level. If this is the case, some specific processing steps can either allow merging of some elements in the highest pyramid level or can restrict some of these elements to be segmentation branch roots. If the number of image regions is larger than 2n, the most similar regions will be merged into a single tree branch, and the method will not be able to give acceptable results.

Algorithm 20: Split and link to the segmentation tree

1. Define a pyramid data structure with overlapping regions. Evaluate the starting region description.

2. Build a segmentation tree starting with leaves. Link each node of the tree to that one of the four possible parents to which it has the most similar region properties.

Build the whole segmentation tree. If there is no link to an element in the higher pyramid level, assign the value zero to this element.

3. Update the pyramid data structure; each element must be assigned the average of the values of all its existing children.

4. Repeat steps 2 and 3 until no significant segmentation changes appear between iterations (a small number of iterations is usually sufficient).

Algorithm 21: Single-pass split-and-merge

1. Search an entire image line by line except the last column and last line. Perform the following steps for each pixel.

2. Find a splitting pattern for a 2x2 pixel block.

3. If a mismatch between assigned labels and splitting patterns in overlapping blocks is found, try to change the assigned labels of these blocks to remove the mismatch (discussed below).

4. Assign labels to unassigned pixels to match a splitting pattern of the block.

5. Remove small regions if necessary.

4. Watershed segmentation

The concepts of watersheds and catchment basins are well known in topography. Watershed lines divide individual catchment basins. The North American Continental Divide is a textbook example of a watershed line with catchment basins formed by the Atlantic and Pacific Oceans. image data may be interpreted as a topographic surface where the gradient image gray-levels

represent altitudes. Thus, region edges correspond to high watersheds and low-gradient region interiors correspond to catchment basins. According to equation F.30), the goal of region growing segmentation is to create homogeneous regions; in watershed segmentation, catchment basins of the topographic surface are homogeneous in the sense that all pixels belonging to the same catchment basin are connected with the basin's region of minimum altitude (gray-level) by a simple path of pixels that have monotonically decreasing altitude (gray-level) along the path. Such catchment basins then represent the regions of the segmented image (Figure 6.48). While the concept of watersheds and catchment basins is quite straightforward, development of algorithms for watershed segmentation is a complex task, with many of the early methods resulting in either slow or inaccurate execution. The first algorithms for watershed segmentation were developed for topographic digital elevation models. Most of the existing algorithms start with extraction of potential watershed line pixels using a local 3x3 operation, which are then connected into geornorphological networks in subsequent steps. Due to the local character of the first step, these approaches are often inaccurate. There are two basic approaches to watershed image segmentation. The first one starts with finding a downstream path from each pixel of the image to a local minimum of image surface altitude. A catchment basin is then defined as the set of pixels for which their respective downstream paths all end up in the same altitude minimum. While the downstream paths are easy to determine for continuous altitude surfaces by calculating the local gradients, no rules exist to define the downstream paths uniquely for digital surfaces.

While the given approaches were not efficient because of their extreme computational demands and inaccuracy, the second watershed segmentation approach represented by a seminal paper [Vincent and Soille, 1991] makes the idea practical. This approach is essentially dual to the first one; instead of identifying the downstream paths, the catchment basins fill from the bottom. As was explained earlier, each minimum represents one catchment basin, and the strategy is to start at the altitude minima. Imagine that there is a hole in each local minimum, and that the topographic surface is immersed in water. As a result, the water starts filling all catchment basins, minima of which are under the water level. If two catchment basins would merge as a result of further immersion, a dam is built all the way to the highest surface altitude and the dam represents the watershed line. The algorithm is based on sorting the pixels in increasing order of their gray values,

followed by a flooding step consisting of a fast breadth-first scanning of all pixels in the order of their gray-levels. During the sorting step, a brightness histogram is computed.

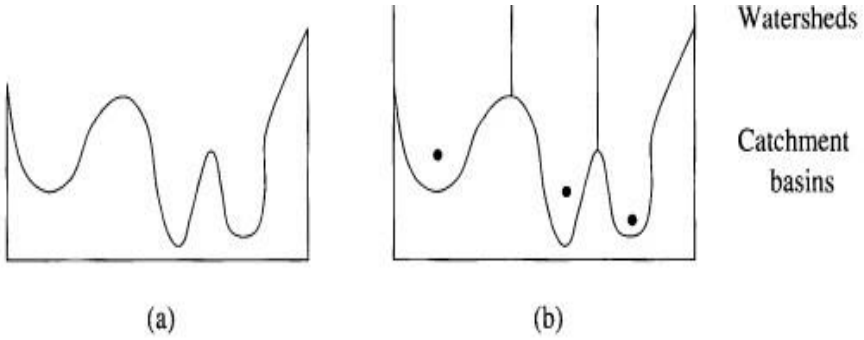


Figure 6.48: One-dimensional example of watershed segmentation. (a) Gray-level profile of image data. (b) Watershed segmentation—local minima of gray-level (altitude) yield catchment basins, local maxima define the watershed lines.

Simultaneously, a list of pointers to pixels of gray-level h is created and associated with each histogram gray-level to enable direct access to all pixels of any gray-level. Information about the image pixel sorting is used extensively in the flooding step. Suppose the flooding has been completed up to a level (gray-level, altitude) k. Then every pixel having gray-level less than or equal to k has already been assigned a unique catchment basin label. Next, pixels having gray-level k + 1 must be processed; all such pixels can be found in the list that was prepared in the sorting step—consequently, all these pixels can be accessed directly. A pixel having gray-level k +1 may belong to a catchment basin labeled I if at least one of its neighbors already carries this label. Pixels that represent potential catchment basin members are put in a first-in first-out queue and await further processing.

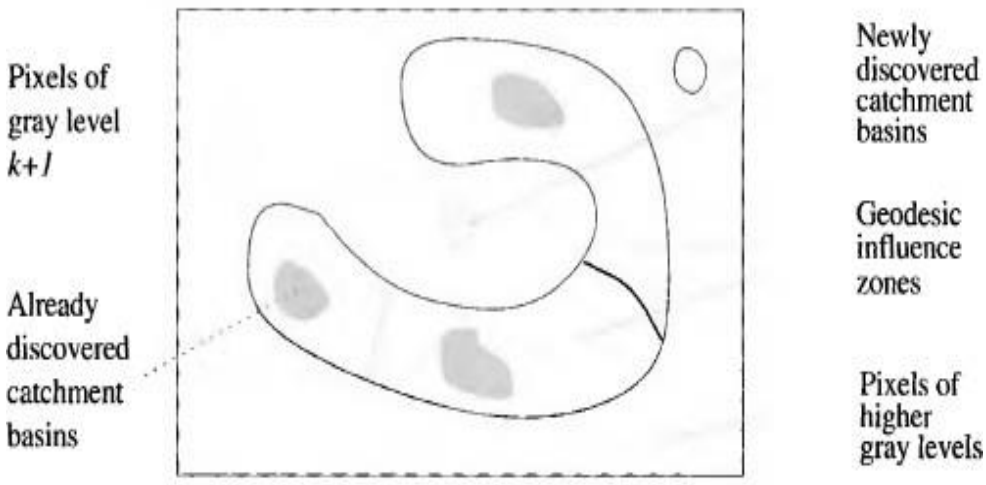


Figure 6.49: Geodesic influence zones of catchment basins.

Geodesic influence zones are computed for all hitherto determined catchment basins. A geodesic influence zone of a catchment basin bL is the locus of non-labeled image pixels of gray-level k + 1 that are contiguous with the catchment basin U (contiguous within the region of pixels of gray-level k + 1) for which their distance to U is smaller than their distance to any other catchment basin lj (Figure 6.49). All pixels with gray-level A; + 1 that belong to the influence zone of a catchment basin labeled I are also labeled with the label I, thus causing the catchment basin to grow. The pixels from the queue are processed sequentially, and all pixels from the queue that cannot be assigned an existing label represent newly discovered catchment basins and are marked with new and unique labels.

Matching

Matching is another basic approach to segmentation that can be used to locate known objects in an image, to search for specific patterns, etc. Figure 6.52 shows an example of a desired pattern and its location found in the image. Matching is widely applicable; it can be used to determine stereoscopic scene properties if more than one image of the same scene taken from different locations is available. Matching in dynamic images (e.g., moving cars, clouds, etc.) is another application area. Generally speaking, one image canbe used to extract objects or patterns, and directed search is used to look for the same (or similar) patterns in the remaining images. The best match is based on some criterion of optimality which depends on object properties and object

relations.

Figure 6.52: Segmentation by matching; matched pattern and location of the best match.

Matched patterns can be very small, or they can represent whole objects of interest. While matching is often based on directly comparing gray-level properties of image sub- regions, it can be equally well performed using image-derived features or higher-level image descriptors. In such cases, the matching may become invariant to image transforms. Criteria of optimality can compute anything from simple correlations up to complex approaches of graph matching.

I. Matching criteria

Match-based segmentation would be extremely easy if an exact copy of the pattern of interest could be expected in the processed image; however, some part of the pattern is usually corrupted in real images by noise, geometric distortion, occlusion, etc. Therefore,

it is not possible to look for an absolute match, and a search for locations of maximum match is more appropriate.

Algorithm 6.23: Match-based segmentation

1. Evaluate a match criterion for each location and rotation of the pattern in the image.
2. Local maxima of this criterion exceeding a preset threshold represent pattern locations in the image.

Control strategies of matching

Match-based segmentation localizes all image positions at which close copies of the searched pattern are located. These copies must match the pattern in size and orientation, and the

geometric distortion must be small. To adapt match-based methods to detect patterns that are rotated, and enlarged or reduced, it would be necessary to consider patterns of all possible sizes and rotations. Another option is to use just one pattern and match an image with all possible geometric transforms of this pattern, and this may work
well if some information about the probable geometric distortion is available. Note that there is no difference in principle between these approaches.

However, matching can be used even if an infinite number of transformations are allowed. Let us suppose a pattern consists of parts, these parts being connected by rubber
links. Even if a complete match of the whole pattern within an image may be impossible,
good matches can often be found between pattern parts and image parts. Good matching
locations may not be found in the correct relative positions, and to achieve a better match, the rubber connections between pattern parts must be either pushed or pulled. The final goal can be described as the search for good partial matches of pattern parts in locations that cause minimum force in rubber link connections between these parts.

A good strategy is to look for the best partial matches first, followed by a heuristic graph construction of the best combination of these partial matches in which graph nodes represent pattern parts.

Match-based segmentation is time consuming even in the simplest cases with no geometric transformations, but the process can be made faster if a good operation sequence is found. The sequence of match tests must be data driven. Fast testing of image locations with a high probability of match may be the first step; then it is not necessary to test all possible pattern locations. Another speed improvement can be realized if a mismatch can be detected before all the corresponding pixels have been tested.

If a pattern is highly correlated with image data in some specific image location, then typically the correlation of the pattern with image data in some neighborhood of this specific location is good. In other words, the correlation changes slowly around the best matching location. If this is the case, matching can be tested at lower resolution first, looking for an exact match in the neighborhood of good low-resolution matches only. The mismatch must be detected as soon as possible since mismatches are found much more often than matches. Considering formulae 6.37-

6.39, testing in a specified position must stop when the value in the denominator (measure of mismatch) exceeds some preset threshold. This implies that it is better to begin the correlation test in pixels.

Digital Image Processing

Unit-3 Possible Questions

2 Marks Questions:

1. Define image segmentation ?
2. Mention the three groups of segmentation methods?
3. Write short notes on thresholding?
4. Define P-tile thresholding
5. Define optimal thresholding
6. Write notes on multispectral thresholding
7. Write notes on edge relaxation
8. Mention the uses of edge image thresholding
9. Write notes on border tracing
10. Define hough transform
11. Mention the advantage of using A-algorithm
12. Define region growing in image segmentation
13. Mention the approaches adopted in region growing
14. Write notes on watershed segmentation
15. Mention the uses of matching technique
16. Define chamfer matching and its uses

8 Marks Questions:

1. Discuss in detail the concept of thresholding in image segmentation with suitable example
2. Discuss in detail about region based segmentation with suitable example
3. Explain in detail about matching concept with suitable algorithm
4. Illustrate in detail the concept of edge based segmentation with example
5. Explain about edge relaxation
6. Explain the following:
   i) Inner boundary tracing ii)Outer boundary tracing iii) Extended boundary tracing

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC ACT 1956)
M.sc Computer Science
DEPARTMENT OF CS, CA & IT
Third Semester(Digital Image processing)

S.no	Questions	OPT1	OPT2	OPT3	OPT4	Answer
1	Using more than one thresholds for image segmentation is called	basic thresholding	adaptive thresholding	band thresholding	semithresholding	adaptive thresholding
2	_____ is a brightness constant to segment objects	threshold	view	image	boundary	threshold
3	Segmentation using variable thresholds is called _____	basic thresholding	adaptive thresholding	band thresholding	semithresholding	adaptive thresholding
4	Which thresholding makes human-assisted analysis easier?	basic thresholding	adaptive thresholding	band thresholding	semithresholding	semithresholding
5	Threshold detection method using bimodel histogram is _____	mean method	mode method	median method	data method	mode method
6	Segmenting part of image is _____	partial segmentation	complete segmentation	full segmentation	threshold	partial segmentation
7	Segmenting complete image is _____	partial segmentation	complete segmentation	full segmentation	both b& c	both b& c
8	Segmenting partial of image is _____	partial segmentation	complete segmentation	full segmentation	threshold	partial segmentation
9	Segmentation using single threshold is called _____	basic thresholding	adaptive thresholding	band thresholding	semithresholding	basic thresholding
10	Segmentation has ____ categories.	1	2	3	4	3
11	_____ is the simple, inexpensive and fast segmentation process that are characterized by constant reflectivity or light absorption of their surface.	White level thresholding	Gray level thresholding	Fast segmentation	Slow segmentation	Gray level thresholding

12	_____ results in a set of disjoint regions uniquely corresponding with objects in the input image	image restoration	Complete segmentation	image objects	Partial segmentation	Complete segmentation
13	_____ results in regions that do not correspond directly with image objects	fast segmentation	image objects	Partial segmentation	Slow segmentation	Partial segmentation
14	In _____, the threshold value varies over the image as a function of local image characteristics	image restoration	both a & d	image thresholding	Adaptive thresholding	Adaptive thresholding
15	_____ aims to segment an image into regions of pixels with gray levels from a set D and into background otherwise and also serve as border detection g(i,j) = 1 for f(i,j) in D and g(i,j) = 0 otherwise	Band-thresholding	edge detection	Semithresholding	adaptive thresholding	Band-thresholding
16	_____ aims to mask out the image background leaving gray level information present in the objects g(i,j) = f(i,j) for f(i,j) >= T and g(i,j) = 0 otherwise	Semithresholding	stochastic	border detection	edge detection	Semithresholding
17	_____ choose a threshold T (based on the image histogram) such that 1/p of the image area has gray values less than T and the rest has gray values larger than T and is used in text segmentation.	full resolution	edge detection	P-tile-thresholding	adaptive thresholding	P-tile-thresholding
18	Bimodal histogram threshold detection algorithms usually finds the highest local maxima first and detect the threshold as a minimum between them; this technique is called _____.	border method	Mode method	edge method	tile method	Mode method
19	_____ is based on approximation of the histogram of an image using a weighted sum of two or more probability densities with normal distribution	Optimal thresholding	border detection	multi thresholding	adaptive thresholding	Optimal thresholding

20	_____ is appropriate for color or multiband images.	single thresholding	multi thresholding	Multispectral thresholding	adaptive thresholding	Multispectral thresholding
21	The aim of _____ is to detect the presence of a region in a low resolution image, and to give the region more precision in image of higher to full resolution.	border thresholding	Semithresholding	Hierarchical thresholding	adaptive thresholding	Hierarchical thresholding
22	_____ relies on edges found in an image by edge detecting operators which marks image locations of discontinuities.	Edge image thresholding	border thresholding	image thresholding	Edge based segmentation	Edge based segmentation
23	_____ is based on construction of an edge image that is processed by an appropriate threshold	Edge image thresholding	border thresholding	image thresholding	adaptive thresholding	Edge image thresholding
24	In _____ , edge properties are considered in the context of neighboring edges	edge detection	edge relaxation	border detection	adaptive thresholding	edge relaxation
25	Three types of region borders may be formed : inner, outer and _____	expanded	extended	enclosed	adaptive thresholding	extended
26	If the criterion of optimality is defined, globally optimal borders can be determined using _____	edge searching	point searching	curve searching	graph searching	graph searching
27	Graph searching is otherwise called as _____	graph	segmentation	dynamic programming	adaptive thresholding	dynamic programming
28	_____ is the key to successful border detection	planner	Cost definition	plotter	C-algorithm	Cost definition
29	Graph searching uses _____ that guarantees optimality.	A- algorithm	B-algorithm	C-algorithm	D-algorithm	A- algorithm
30	_____ may substantially increase search speed with some additional constraints	B-algorithm	complete borders	Heuristic graph search	C-algorithm	Heuristic graph search

31	Using the _____ to search a graph, it is not necessary to construct the entire graph since the costs associated with the expanded nodes are calculated only if needed	A-algorithm	C-algorithm	D-algorithm	B-algorithm	A-algorithm
32	_____ segmentation is applicable if objects of known shape are to be detected within a image.	Hough transform	border detection	rough transform	C-algorithm	Hough transform
33	While forming the regions from complete borders is trivial, region determination from _____ may be very complex task	image borders	partial borders	graph	both a & b	partial borders
34	A graph based method called the _____ can be introduced that allows searching to proceed in any direction.	graph	Graph searching	gradient field transform	partial borders	gradient field transform
35	In Hough transform, the parameter space is represented by a rectangular structure of cells called the _____	accumulator cells	accumulator group	accumulator slice	accumulator array	accumulator array
36	The elements of accumulator array are called as _____	segmentation	accumulator cells	accumulator array	array	accumulator cells
37	The method construct regions from partial borders and is applicable if regions have dominant gray level properties.	Graph searching	super slice	crack edges	weak edge	super slice
38	The edge type 0-0 is called that has negative influence on the edge confidence	crack edges	weak edge	isolated edge	strong edge	isolated edge
39	_____is an iterative method, with edge confidences converging	optimality	connectivity	adjacency	both a & b	optimality
40	A _____ knowledge is obtained by analyzing the degraded image.	spectral	artificial	posteriori	segmented	posteriori
41	_____ deals with direct construction of regions and are generally better in noisy images where edges are extremely difficult to detect.	surface – based segmentation	adjacency	Region-based segmentation	Graph searching	Region-based segmentation

42	_____ of regions is used as the main segmentation criterion in region growing.	Homogeneity	Graph searching	segmentation	noisy images	Homogeneity
43	The data structure called _____ carries all the necessary information for region merging in 4-adjacency using crack edges.	supergrid	super region	superguide	both a & b	supergrid
44	_____ begins with the whole image represented as a single region which does not usually satisfy the condition of homogeneity.	edge spliting	boundary splitting	area spliting	Region splitting	Region splitting
45	The segmentation process can be understood as the construction of a segmentation _____ where each leaf node represents a homogeneous region.	segmenttree	adjacency	quadtree	area spliting	quadtree
46	An unpleasant drawback of segmentation _____ is the square region shape assumption and merging of regions are not part of the same branch of the segmentation tree	segmenttree	quadtrees	quadtree	adjacency	quadtrees
47	Lower memory requirements can be found in _____ segmentation. A local splitting pattern is detected in each 2x2 pixel image block and regions are merged in overlapping blocks of the same size	single-pass split-and-merge	single-pass split	single-pass merge	area spliting	single-pass split-and-merge
48	_____ divide individual catchment basins and are well known in topography.	Watershed lines	catchment basins	both a & b	quadtree	Watershed lines
49	Image data may be interpreted as a where the gradient image gray-levels represent altitudes.	dam	catchment basins	topographic surface	quadtree	topographic surface
50	_____ correspond to high watersheds and low-gradient region interiors correspond to catchment basins.	water region	both a & c	watersheds	Region edges	Region edges
51	_____ represent the regions of the segmented image.	Catchment basins	watersheds	water basins	none	Catchment basins

52	Raw watershed segmentation produces a severely over segmented image with hundreds or thousands of catchment basins. To overcome this problem, _____ and other approaches have been suggested to generate good segmentation.	watersheds	catchment basins	region markers	Watershed lines	region markers
53	Region growing often results in undergrowing or overgrowing as a result of non-optimal parameter setting. Hence _____ are used to decrease the number of small regions in the segmented image that cannot be merged with any adjacent region according to the originally applied homogeneity criteri	computer graphics	optimality	post-processors	water region	post-processors
54	_____ is another basic approach to segmentation that can be used to locate known objects in an image, to search for specific patterns, etc. and best match is based on some criterion of optimality which depends on object properties and object relations.	homogeneity	Raw watershed	both a & d	Matching	Matching
55	_____ can be defined as correlation between a pattern and the searched image data	image criteria	adjacent region	Matching criteria	water region	Matching criteria
56	A printed text sheet may be an example if we know that characters of the text cover _____of the sheet area	1/p	2/p	3/p	4/p	1/p
57	bimodal histogram - if objects have approximately the same gray level that differs from the gray level of the _____	background	image	boundary	edge	background
58	multimodal histogram - more thresholds may be determined at minima between any _____maxim	one	two	three	four	two
59	_____ it is often impossible to interpret the significance of local histogram maxima	multimodal histogram	bimodal histogram	both a & b	histogram	bimodal histogram

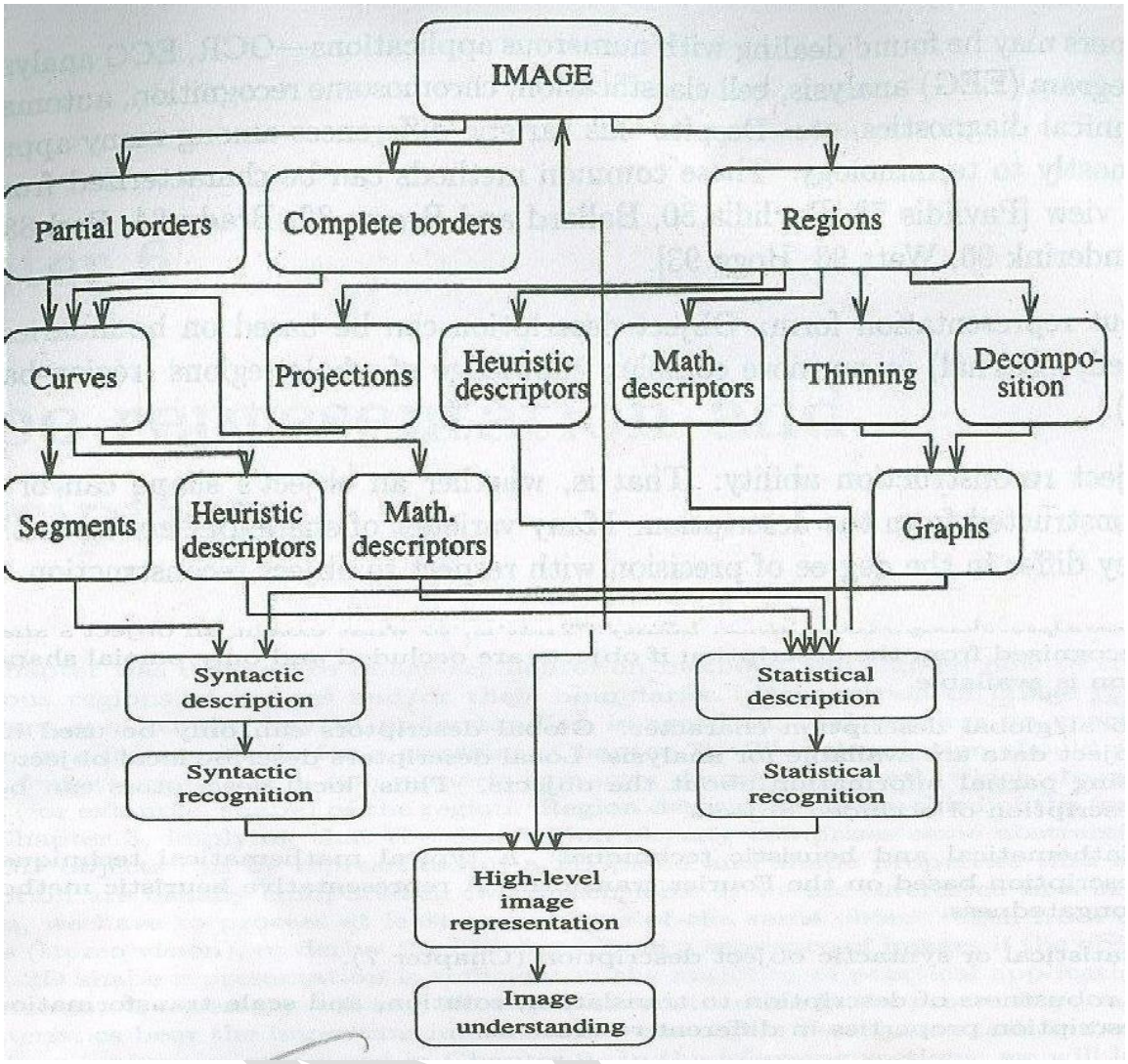
60	The border detection process is transformed into a search for the optimal path in the _____ graph.	weighted	non weighted	costs	root	weighted

UNIT IV

SYLLABUS:

Shape Representation and Description: Region Identification - Contour Based Representation And Description – Region Based Shape Representation And Description

Recognition of image regions is an important step on the way to understanding image data, and requires an exact region description in a form suitable for a classifier. This description should generate a numeric feature vector, or a non-numeric syntactic description word, which characterizes properties (for example, shape) of the region. Region description is the third of the four levels, implying that the description already comprises some abstraction—for example, 3D objects can be represented in a 2D plane and shape properties that are used for description are usually computed in two dimensions. If we are interested in a 3D object description, we have to process at least two images of the same object taken from different viewpoints (stereo vision), or derive the 3D shape from a sequence of images if the object is in motion. A 2D shape representation is sufficient in the majority of practical applications, but if 3D information is necessary—if, say, 3D object reconstruction is the processing goal, or the 3D characteristics bear the important information—the object description task is much more difficult.



Defining the shape of an object can prove to be very difficult. Shape is usually represented verbally or in figures, and people use terms such as elongated, rounded, with sharp edges, etc. The computer era has introduced the necessity to describe even very complicated shapes precisely, and while many practical shape description methods exist, there is no generally accepted methodology of shape description. Further, it is not known what is important in shape.

Location and description of substantial variations in the first derivative of object boundaries often yield suitable information. Examples include alphanumeric optical character recognition (OCR), technical drawings, electro-cardiogram (ECG) curve characterization, etc.

Shape is an object property which has been carefully investigated in recent years and many papers may be found dealing with numerous applications—OCR, ECG analysis, electro-encephalogram (EEG) analysis, cell classification, chromosome recognition, automatic inspection, technical diagnostics, etc. Despite this variety, differences among many approaches are limited mostly to terminology. These common methods can be characterized from different points of view:

• Input representation form: Object description can be based on boundaries (contour- based, external) or on more complex knowledge of whole regions (region-based, internal).

• Object reconstruction ability: That is, whether an object's shape can or cannot be reconstructed from the description. Many varieties of shape-preserving methods exist. They differ in the degree of precision with respect to object reconstruction.

• Incomplete shape recognition ability: That is, to what extent an object's shape can be recognized from the description if objects are occluded and only partial shape information is available.

• Local/global description character: Global descriptors can only be used if complete object data are available for analysis. Local descriptors describe local object properties using partial information about the objects. Thus, local descriptors can be used for description of occluded objects.

• Mathematical and heuristic techniques: A typical mathematical technique is shape description based on the Fourier transform. A representative heuristic method may be **elongatedness**. :

$$elongatedness = \frac{area}{(2d)^2}$$

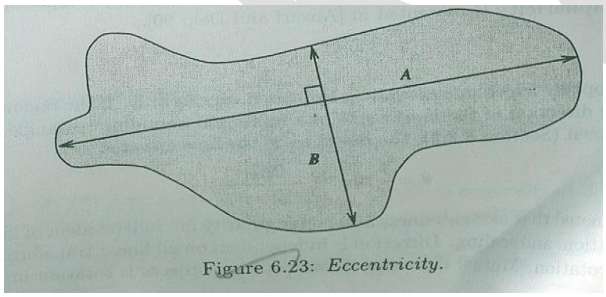
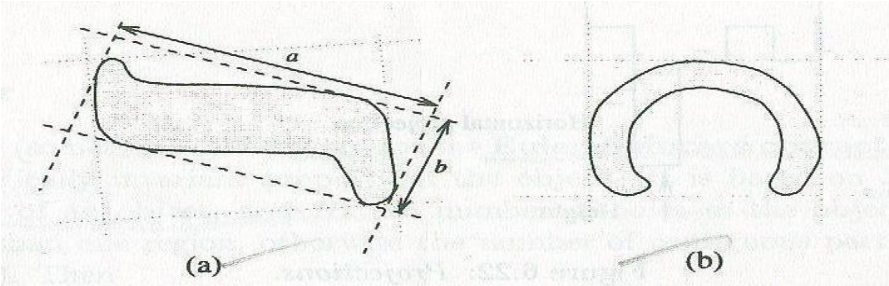


Figure 6.23: *Eccentricity.*

• Statistical or syntactic object description.

• A robustness of description to translation, rotation, and scale transformations:

Shape description properties in different resolutions.



$$\alpha(x,y) = x\cos\theta + y\sin\theta \qquad \beta(x,y) = -x\sin\theta + y\cos\theta$$

Problems of scale (resolution) are common in digital images. Sensitivity to scale is even more serious if a shape description is derived, because shape may change substantially with image resolution. Contour detection may be affected by noise in high resolution, and small details may disappear in low resolution (see Figure 8.2). Therefore, shape has been studied in multiple resolutions which again causes difficulties with matching corresponding shape representations from different resolutions. Moreover, the conventional shape descriptions change discontinuously. This approach is not a new technique itself, but is an extension of existing techniques, and more robust shape methods may result from developing and retaining their parameters over a range of scales. Consider an object with planar faces and imagine how many very different 2D shapes may result from a given face if the position and 3D orientation of this simple object changes with respect to an observer. In some special cases, such as circles which transform to ellipses, or planar polygons, projectively invariant features (called invariants) can be found. Unfortunately, no existing shape descriptor is perfect; in fact, they are all far from being perfect. Therefore, a very careful choice of descriptors resulting from detailed analysis of the shape recognition problem must precede any implementation, and whether or not a 2D representation is capable of describing a 3D shape must also be considered. For some 3D shapes, their 2D projection may bear enough information for recognition—aircraft contours are a good example; successful recognition of airplanes from projections are known even if they change their position and orientation in space. In many other cases, objects must be seen from a specific direction to get enough descriptive information—human faces are such a case.

Object occlusion is another hard problem in shape recognition. However, the situation is easier

here (if pure occlusion is considered, not combined with orientation variations yielding changes in 2D projections as discussed above), since visible parts of objects may

be used for description. Here, the shape descriptor choice must be based on its ability to describe local object properties—if the descriptor gives only a global object description (e.g., object size, average boundary curvature, perimeter), such a description is useless if only a part of an object is visible. If a local descriptor is applied (e.g., description of local boundary changes), this information may be used to compare the visible part of the object to all objects which may appear in the image. Clearly, if object occlusion occurs, the local or global character of the shape descriptor must be considered first. This classification of shape description methods corresponds to previously described boundary-based and region-based segmentation methods. However, both contour-based and region-based shape descriptors may be local or global and differ in sensitivity to translation, rotation, scaling, etc.

1 Region identification

Region identification is necessary for region description. One of the many methods for region identification is to label each region (or each boundary) with a unique (integer) number; such identification is called labeling or coloring (also connected component labeling), and the largest integer label usually gives the number of regions in the image. Another method is to use a smaller number of labels, and ensure that no two neighboring regions have the same label; then information about some region pixel must be added to the description to provide full region reference. This information is usually stored in a separate data structure. Alternatively, mathematical morphology approaches (Chapter 13) may be used for region identification.

Assume that the segmented image R consists of m disjoint regions Ri (as in equation 6.1)). The image R often consists of objects and a background m where Rp is the set complement, Rb is considered background, and other regions are considered objects. Input to a labeling algorithm is usually either a binary or multi-level image, where background may be represented by zero pixels, and objects by non-zero values. A multi-level image is often used to represent the labeling result, background being represented by zero values, and regions represented by their non-zero labels.

Algorithm 1presents a sequential approach to labeling a segmented image.

Algorithm 1: 4-neighborhood and 8-neighborhood region identification

1. First pass: Search the entire image R row by row and assign a non-zero value v to each non-zero pixel R(i,j). The value v is chosen according to the labels of the pixel's neighbors, where the property neighboring is defined by Figure 8.3. ('neighbors' outside the image R are not considered),

• If all the neighbors are background pixels (with pixel value zero), R(i,j) is assigned a new (and as yet) unused label.

• If there is just one neighboring pixel with a non-zero label, assign this label to the pixel R(i,j). • If there is more than one non-zero pixel among the neighbors, assign the label of any one to the labeled pixel. If the labels of any of the neighbors differ(label collision), store the label pair as being equivalent. Equivalence pairs are stored in a separate data structure—an equivalence table.

2. Second pass: All of the region pixels were labeled during the first pass, but some regions have pixels with different labels (due to label collisions). The whole image is scanned again, and pixels are re-labeled using the equivalence table information (for example, with the lowest value in an equivalence class).

Label collision is a very common occurrence—examples of image shapes experiencing this are U-shaped objects, mirrored E C) objects, etc. (see Figure 8.3c). The equivalence table is a list of all label pairs present in an image; all equivalent labels are replaced by a unique label in the second step. Since the number of label collisions is usually not known beforehand, it is necessary to allocate sufficient memory to store the equivalence table in an array. A dynamically allocated data structure is recommended. Further, if pointers are used for label specification, scanning the image for the second time is not necessary (the second pass of the algorithm) and only rewriting labels to which these pointers are pointing is much faster.

The algorithm is basically the same in 4-connectivity and 8-connectivity, the only difference being in the neighborhood mask shape (Figure 8.3b). It is useful to assign the region labels incrementally to permit the regions to be counted easily in the second pass.

Region identification can be performed on images that are not represented as straightforward matrices; the following algorithm may be applied to images that are run length encoded.

Algorithm 2: Region identification in run length encoded data

1. First pass: Use a new label for each continuous run in the first image row that is not part of the background.

2. For the second and subsequent rows, compare positions of runs.

• If a run in a row does not neighbor (in the 4- or 8-sense) any run in the previous row, assign a new label.

• If a run neighbors precisely one run in the previous row, assign its label to the new run.

2. Contour-based shape representation and description

Region borders must be expressed in some mathematical form. The rectangular representation of xn pixel co-ordinates as a function of the path length n is most common.

• Polar co-ordinates, in which border elements are represented as pairs of angle <f> and distance r;

• Tangential co-ordinates, which codes the tangential directions 0(xn) of curve points as a function of path length n.

Chain codes

Chain codes describe an object by a sequence of unit-size line segments with a given orientation. The first element of such a sequence must bear information about its position to permit the region to be reconstructed. The process results in a sequence of numbers; to exploit the position invariance of chain codes the first element, which contains the position information, is omitted. This definition of the chain code is known as Freeman's code. Note that a chain code object description may easily be obtained as a by-product of border detection; for a description of border detection algorithms. If the chain code is used for matching, it must be independent of the choice of the first border pixel in the sequence. One possibility for normalizing the chain code is to find the pixel in the border sequence which results in the minimum integer number if the description chain is interpreted as a base 4 number—that pixel is then used as the starting pixel . A 'mod 4 or mod 8 difference code, called a chain code derivative, is another numbered sequence that represents relative directions of Simple geometric border representation The following descriptors are based

mostly on geometric properties of described regions. Because of the discrete character of digital images, all of them are sensitive to image resolution.

Boundary length

Boundary length is an elementary region property, that is simply derived from the chain code representation. Vertical and horizontal steps have unit length, and the length of diagonal steps in 8-connectivity is \/2. It can be shown that the boundary is longer in 4-connectivity, where a diagonal step consists of two rectangular steps with a total length of 2. A closed-boundary length (perimeter) can also be easily evaluated from run length or quadtree representations. Boundary length increases as the image raster resolution increases; on the other hand, region area is not affected by higher resolution and converges to some limit.To provide continuous-space perimeter properties (area computation from the boundary length, shape features, etc.), it is better to define the region border as being the outer or extended border. If inner borders are used, some properties are not satisfied e.g., the perimeter of a 1-pixel region is 4 if the outer boundary is used, and 1 if the inner is used.

Curvature

In the continuous case, curvature is defined as the rate of change of slope. In discrete space, the curvature description must be slightly modified to overcome difficulties resulting from violation of curve smoothness. The curvature scalar descriptor (also called boundary straightness) finds the ratio between the total number of boundary pixels (length) and the number of boundary pixels where the boundary direction changes significantly. The smaller the number of direction changes, the straighter the boundary. The evaluation algorithm is based on the detection of angles between line segments positioned b boundary pixels from the evaluated boundary pixel in both directions. The angle need not be represented numerically; rather, relative position of line segments can be used as a property. The parameter b determines sensitivity to local changes of the boundary direction.

Curvature computed from the chain code can be found in [Rosenfeld, 1974], and the tangential border representation is also suitable for curvature computation. Values of the curvature

at all boundary pixels can be represented by a histogram; relative numbers then provide information on how common specific boundary direction changes are. **Histograms** of boundary angles, such as the /? angle, can be built in a similar way—such histograms can be used for region description. Another approach to calculating curvature from digital curves is based on convolution with the truncated Gaussian kernel

Bending energy

The bending energy (BE) of a border (curve) may be understood as the energy necessary to bend a rod to the desired shape, and can be computed as a sum of squares of the border curvature c(k) over the border length L. Bending energy can easily be computed from Fourier descriptors using Parseval's theorem. To represent the border, Freeman's chain code or its smoothed version may be used; see Figure 8.8. Bending energy does not permit shape reconstruction.

Signature

The signature of a region may be obtained as a sequence of normal contour distances. The normal contour distance is calculated for each boundary element as a function of the path length. For each border point A, the shortest distance to an opposite border point B is sought in a direction perpendicular to the border tangent at point A: see Figure 8.9. Note that being opposite is not a symmetric relation (compare Algorithm 6.16). Signatures are noise sensitive, and using smoothed signatures or signatures of smoothed contours reduces noise sensitivity. Signatures may be applied to the recognition of overlapping objects or whenever only partial contours are available [Vernon, 1987]. Position, rotation, and scale-invariant modifications based on gradient-perimeter and angle-perimeter plots .

Chord distribution

A line joining any two points of the region boundary is a chord, and the distribution of lengths and angles of all chords on a contour may be used for shape description. Let b(x,y) = 1 represent the contour points, and b(x,y) — 0 represent all other points. The chord distribution can be computed

Fourier transforms of boundaries

Suppose C is a closed curve (boundary) in the complex plane (Figure 8.11a). Traveling anti-

clockwise along this curve keeping constant speed, a complex function z(t) is obtained, where t is a time variable. The speed should be chosen such that one circumnavigation of the boundary takes time 27r; then a periodic function with period 2tt is obtained after multiple passes around the curve. This permits a Fourier representation of z(t)

z(t) = ^Tneint. (8.6)

The coefficients Tn of the series are called the Fourier descriptors of the curve C. It is more useful to consider the curve distance s in comparison to time t = 2ns/L , where L is the curve length. The Fourier descriptors Tn are given by The descriptors are influenced by the curve shape and by the initial point of the curve. Working with digital image data, boundary co-ordinates are discrete and the function z(s) is not continuous. Assume that z(k) is a discrete version of z(s), where 4-connectivity is used to get a constant sampling interval; the descriptors Tn can be computed from the discrete Fourier transform (DFT) of z(k)

A closed boundary can be represented as a function of angle tangents versus the distance between the boundary points from which the angles were determined (Figure 8.11b). Let ifk be the angle measured at the kth boundary point, and let 4 be the distance between the boundary starting point and the kth boundary point.

The discrete Fourier transform is used in all practical applications. The high-quality boundary shape representation obtained using only a few lower-order coefficients is a favorable property common to Fourier descriptors. We can compare the results of using the Sn and Tn descriptors: The Sn descriptors have more high-frequency components present in the boundary function due to more significant changes of tangent angles, and as a result, they do not decrease as fast as the Tn descriptors. In addition, the Sn descriptors are not suitable for boundary reconstruction since they often result in a non-closed boundary. A method for obtaining a closed boundary using Sn descriptors is given in [Strackee and Nagelkerke, 1983]. The Tn descriptor values decrease quickly for higher frequencies, and their reconstruction always results in a closed boundary. Moreover, the Sn descriptors cannot be applied for squares, equilateral triangles, etc. unless the solution methods introduced in [Wallace and Wintz, 1980] are applied. Fourier descriptors can also be used for calculation of region area, location of centroid, and computation of second-order moments .Fourier

descriptors are a general technique, but problems with describing local information exist. A modified technique using a combined frequency-position space that deals better with local curve properties exists, another modification that is invariant under rotation, translation, scale, mirror reflection, and shifts in starting points.

Conventional Fourier descriptors cannot be used for recognition of occluded objects. Nevertheless, classification of partial shapes using Fourier descriptors is introduced in [Lin and Chellappa, 1987]. Boundary detection and description using elliptic Fourier decomposition of the.

Boundary description using segment sequences

Representation of a boundary using segments with specified properties is another option for boundary (and curve) description. If the segment type is known for all segments, the boundary can be described as a chain of segment types, a code word consisting of representatives of a type alphabet. An example is given in Figure 8.14 which will be discussed later in more detail. This sort of description is suitable for syntactic recognition.

A polygonal representation approximates a region by a polygon, the region being represented using its vertices. Polygonal representations are obtained as a result of a simple boundary segmentation. The boundary can be approximated with varying precision; if a more precise description is necessary, a larger number of line segments may be employed. Any two boundary points xi, X2 define a line segment, and a sequence of points xi, X2, X3 represents a chain of line segments—from the point xi to the point X2, and from X2 to x3. If xi— X3, a closed boundary results. There are many types of straight-segment boundary representations ,the problem lies in determining the location of boundary vertices, one solution to which is to apply a split-and-merge algorithm. The merging step consists of going through a set of boundary points and adding them to a straight segment as long as a segment straightness criterion is satisfied. If the straightness characteristic of the segment is lost, the last connected point is marked as a vertex and construction of a new straight segment begins.

Boundary vertices can be detected as boundary points with a significant change of boundary direction using the curvature (boundary straightness) criterion. This approach works well for boundaries with rectilinear boundary segments. Another method for determining the boundary

vertices is a tolerance interval approach based on setting a maximum allowed difference e. Assume that point xi is the end point of a previous segment and so by definition the first point of a new segment.

Define points X2, X3 positioned a distance e from the point xi to be rectilinear—xl5 x2, x3 are positioned on a straight line—see Figure 8.12. The next step is to locate a segment which can fit between parallels directed from points X2 and X3. Resulting segments are sub-optimal, although optimality can be achieved with a substantial increase in computational effort.

The methods introduced above represent single-pass algorithms of boundary segmentation using a segment-growing approach. Often they do not result in the best possible boundary segmentation because the vertex which is located often indicates thatthe real vertex should have been located a few steps back. The splitting approach of segmenting boundaries into smaller segments can sometimes help, and the best results can be anticipated using a combination of both methods. If the splitting approach is used, segments are usually divided into two new, smaller segments until the new segments meet the final requirements. A simple procedure for splitting begins from end points xi and X2 of a curve; these end points are connected by a line segment. The next step searches all the curve points for the curve point X3 with the largest distance from the line segment. If the point located is within a preset distance between itself and the line segment, the segment X1-X2 is an end segment and all curve vertices are found, the curve being represented polygonally by vertices xi and X2. Otherwise the point X3 is set as a new vertex and the process is applied recursively to both resulting segments X1--X3 and X3-X2. circular arcs and straight lines is presented in [Rosin and West, 1989]. Segments are considered as primitives for syntactic shape recognition procedures—a typical example is the syntactic description and recognition of chromosomes [Fu, 1974], where boundary segments are classified as convex segments of large curvature, concave segments of large curvature, straight segments, etc. Other syntactic object recognition methods based on a contour partitioning into primitives from a specified set .

Partitioning of the contour using location of points with high positive curvatures (corners) is, together with applications to occluded contours. A discrete curvature function based on a chain code representation of a boundary is used with a morphological approach to obtain segments of

constant curvature. Contour partitioning using segments of constant intensity and polygonal representation used in a hypothesize and verify approach to recognition of occluded objects may be found This technique is based on application of a unique Gaussian smoothing kernel to a one-dimensional signal (e.g., a curvature function) over a range of sizes and the result is differentiated twice. To determine the peaks of curvature, the zero-crossing of the second derivative is detected; the positions of zero-crossings give the positions of curve segmentation points. Different locations of segmentation points are obtained at varying resolution (different Gaussian kernel size). An important property of the Gaussian kernel is that the location of segmentation points changes continuously with resolution which can be seen in the scale-space image of the curve, Figure 8.15a. Fine details of the curve disappear in pairs with increasing size of the Gaussian smoothing kernel, and two segmentation points always merge to form a closed contour, showing that any segmentation point existing .

Other contour-based shape description approaches

Many other methods and approaches can be used to describe two-dimensional curves and contours. Mathematical morphology can be used for shape description, typically in connection with region skeleton construction. A different approach is introduced in, where a geometrical correlation function represents two-dimensional continuous or discrete curves. This function is translation, rotation, and scale invariant and may be used to compute basic geometrical properties.

Neural networks can be used to recognize shapes in raw boundary representations directly. Contour sequences of noiseless reference shapes are used for training, and noisy data are used in later training stages to increase robustness; effective representations of closed planar shapes result. Another neural network shape representation system uses a modified Walsh-Hadamard transform to achieve position-invariant shape representation. and detailed survey of projective geometry for machine vision. Even if shape invariance is a novel approach in machine vision, invariant theory is not new, and many of its principles were introduced in the nineteenth century.

Collinearity is the simplest example of a projectively invariant image feature. Any straight line is projected as a straight line under any projective transform. Similarly, the basic idea of the projection-invariant shape description is to find such shape features that

are unaffected by the transform between the object and the image plane.

A standard technique of projection-invariant description is to hypothesize the pose (position and orientation) of an object and transform this object into a specific co-ordinate system; then shape characteristics measured in this co-ordinate system yield an invariant description. However, the pose must be hypothesized for each object and each image, which makes this approach difficult and unreliable.

Application of invariant theory, where invariant descriptors can be computed directly from image data without the need for a particular co-ordinate system, represents another approach. In addition, invariant theory can determine the total number of functionally independent invariants for a given situation, therefore showing completeness of the description invariant set. Invariant theory is based on a collection of transforms that can be composed and inverted. In vision, the plane-projective group of transforms is considered which contains all the perspectives as a subset. The group approach provides a mathematical tool for generating invariants; if the transform does not satisfy the properties of a group, this machinery. Therefore, the change of co-ordinates due to the plane-projective transform is generalized as a group action. Lie group theory is especially useful in designing new invariants. Let corresponding entities in two different co-ordinate systems be distinguished by capital and lowercase letters. An invariant of a linear transformation is defined as follows:

1. Cross ratio: The cross ratio represents a classic invariant of a projective line. As mentioned earlier, a straight line is always projected as a straight line. Any four collinear points A,B,C,D may be described by the cross-ratio invariant

(A-C)(B-D)

(A-D)(B-C) ' (8.24)

where (A — C) represents the distance between points A and C.

Note that the cross ratio depends on the order in which the four collinear points are labeled.

Systems of lines or points: A system of four co-planar concurrent lines (meeting at the same point) is dual to a system of four collinear points and the cross ratio is its invariant. A system of five general co-planar lines forms two invariants

If the three lines forming the matrix M^ are concurrent, the matrix becomes singular and the invariant is undefined.

A system of five co-planar points is dual to a system of five lines and the same two invariants are formed. These two functional invariants can also be formed as two cross ratios of two co-planar concurrent line quadruples; see Figure 8.20. Note that even though combinations other than those given in Figure 8.20 may be formed, only the two presented functionally independent invariants exist.

Plane conies: A plane conic may be represented by an equation

$ax2 + bxy + cy2 + dx + ey + / = 0$ or $x = (x, y, 1)T$. Then the conic may also be defined by a matrix C

Region-based shape representation and description

We can use boundary information to describe a region, and shape can be described from the region itself. A large group of shape description techniques is represented by heuristic approaches which yield acceptable results in description of simple shapes. Region area, rectangularity, elongatedness, direction, compactness, etc., are examples of these methods.

Unfortunately, they cannot be used for region reconstruction and do not work for more complex shapes. Other procedures based on region decomposition into smaller and simpler sub-regions must be applied to describe more complicated regions, then sub-regions can be described separately using heuristic approaches. Objects are represented by a planar graph with nodes representing sub-regions resulting from region decomposition, and region . Are translation and rotation invariant; position and rotation can be included in the graph definition.

• Are insensitive to small changes in shape.

• Are highly invariant with respect to region magnitude.

• Generate a representation which is understandable.

• Can easily be used to obtain the information-bearing features of the graph.

• Are suitable for syntactic recognition.

On the other hand, the shape representation can be difficult to obtain and the classifier-learning stage is not easy either .Nevertheless, if we are to get closer to the reality of computer

vision, and to understand complex images, there is no alternative.

1 Simple scalar region descriptors

A number of simple heuristic shape descriptors exist which relate to statistical feature description. These methods are basic and are used for description of sub-regions in complex regions, and may then be used to define graph node classification. Area The simplest and most natural property of a region is its area, given by the number of pixels of which the region consists. The real area of each pixel may be taken into consideration to get the real size of a region, noting that in many cases, especially in satellite imagery, pixels in different positions correspond to different areas in the real world. If an image is represented as a rectangular raster, simple counting of region pixels will provide its area. If the image is represented by a quadtree, however, it may be more difficult to find the region area. Assuming that regions have been identified by labeling, the following algorithm may be used.

Algorithm4: Calculating area in quadtrees

1. Set all region area variables to zero, and determine the global quadtree depth H; for example, the global quadtree depth is H = 8 for a 256 x 256 image.
2. Search the tree in a systematic way. If a leaf node at a depth h has a non-zero label, proceed to step 3.
3. Compute: area[region_label] = area[region_label] + 4\H~h> ,
4. The region areas are stored in variables area[region_label].

The region can be represented by n polygon vertices (ik,jk), and (io,Jo) = {in^jn)-

The area is given by —the sign of the sum represents the polygon orientation. If a smoothed boundary is used to overcome noise sensitivity problems.

Various smoothing methods and accurate area-recovering techniques.

If the region is represented by the (anti-clockwise) Freeman chain code, the following algorithm provides the area.

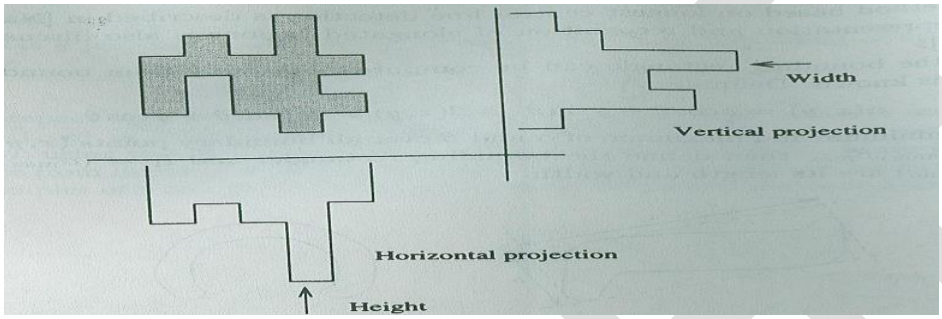
Euler's number Euler's number ϑ (sometimes called genus or the Euler-Poincare characteristic) describes a simple, topologically invariant property of the object. It is based on 5, the number of contiguous parts of an object, and ϑ , the number of holes in the. Then

$$\vartheta = S - N$$

Projections

Horizontal and vertical region projections Ph{i) and pv(j) are defined as

$$p_h(i) = \sum_j f(i,j) \qquad p_v(j) = \sum_i f(i,j)$$



Region description by projections is usually connected to binary image processing. Projections can serve as a basis for definition of related region descriptors; for example, the width (height) of a region with no holes is defined as the maximum value of the horizontal (vertical) projection of a binary image of the region. These definitions are illustrated in Figure 8.22.

Eccentricity

he simplest eccentricity characteristic is the ratio of the length of the maximum chord A to the maximum chord B which is perpendicular to A (the ratio of major and minor axes of an object), Figure 6.23. Another approximate eccentricity measure
is based on a ratio of main region axes of inertia.

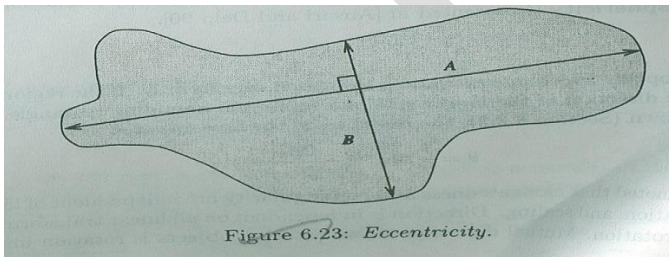


Figure 6.23: *Eccentricity.*

Elongatedness

Elongatedness is a ratio between the length and width of the region bounding rectangle. This is the rectangle of minimum area that bounds the shape, which is located by turning in discrete steps until a minimum is located (see Figure 8.24a). This criterion cannot succeed in curved regions (see Figure 8.24b), for which the evaluation of elongatedness must be based on maximum region thickness. Elongatedness can be evaluated as a ratio of the region area and the square of its thickness. The maximum region thickness (holes must be filled if present) can be determined as the number of erosion steps that may be applied before the region totally disappears. If the number of erosion steps is d,

$$elongatedness = \frac{area}{(2d)^2}$$

Rectangularity

$$rectangularity = \max_k(F_k)$$

compactness:

Moments

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dx\,dy$$

$$m_{pq} = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} i^p j^q f(i,j)$$

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-x_c)^p (y-y_c)^q f(x,y)\,dx\,dy$$

Let Fk be the ratio of region area and the area of a bounding rectangle, the rectangle having the direction fc. The rectangle direction is turned in discrete steps as before, and rectangularity measured as a maximum of this ratio Fk:

The direction need only be turned through one quadrant. Rectangularity assumes values

from the interval @,1], with 1 representing a perfectly rectangular region. Sometimes, it may be more natural to draw a bounding triangle; a method for similarity evaluation between two triangles called sphericity is presented in [Ansari and Delp, 1990].

Direction

Direction is a property which makes sense in elongated regions only. If the region is elongated, direction is the direction of the longer side of a minimum bounding rectangle.

If the shape moments are known (Section 8.3.2), the direction 0 can be computed as

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$$

It should be noted that clongatedness and rectangularity are independent of linear transformations—translation, rotation, and scaling. Direction is independent on all linear transformations which do not include rotation. Mutual direction of two rotating objects is rotation invariant.

Compactness

Compactness is a popular shape description characteristic independent of linear transformations given by area.

$$compactness = \frac{(region\_border\_length)^2}{area}$$

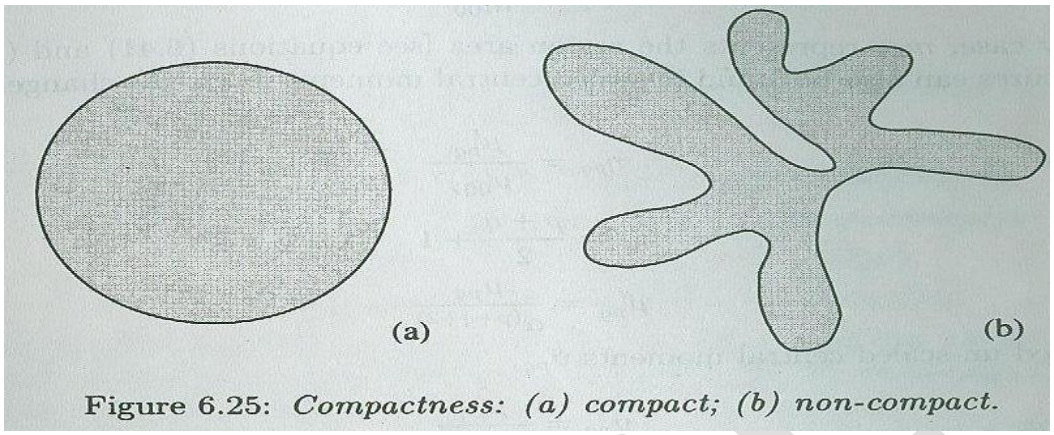


Figure 6.25: *Compactness: (a) compact; (b) non-compact.*

The most compact region in a Euclidean space is a circle. Compactness assumes values in the interval [1, 00) in digital images if the boundary is defined as an inner boundary using the outer boundary, compactness assumes values in the interval [16, 00). Independence from linear transformations is gained only if an outer boundary representation is used. Examples of a compact and a non-compact region

Moments

Region moment representations interpret a normalized gray-level image function as a probability density of a 2D random variable. Properties of this random variable can be described using statistical characteristics. Assuming that non-zero pixel values represent regions, moments can be used for binary or gray-level region description. A moment of order (p + q) is dependent on scaling, translation, rotation, and even on gray-level transformations and is given by

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dx\,dy$$

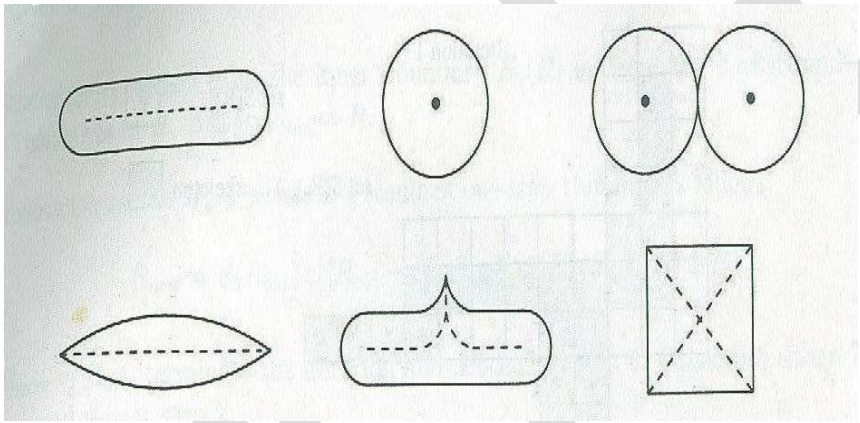
$$m_{pq} = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} i^p j^q f(i,j)$$

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-x_c)^p (y-y_c)^q f(x,y)\,dx\,dy$$

Convex hull

A region convex if and only if for any two points Xi, X2 E R, the whole line segment X1X2 defined by its end points X],x2 is inside the region R. The convex hull of a region is the smallest convex region H which satisfies the condition R, C HThe convex hull has some special properties in digital data which do not exist in the continuous case. For instance, concave parts can appear and disappear in digital data due to rotation, and therefore the convex hull is not rotation invariant in digital space. The convex hull can be used to describe region shape properties and can be used to build a tree structure of region concavity. A discrete convex hull can be defined by the following algorithm which may also be used for convex hull construction. This algorithm has complexity 0(n2) and is presented here as an intuitive way of detecting the convex hull. Algorithm 8.7 describes a more efficient approach.

REGION SKELETON:



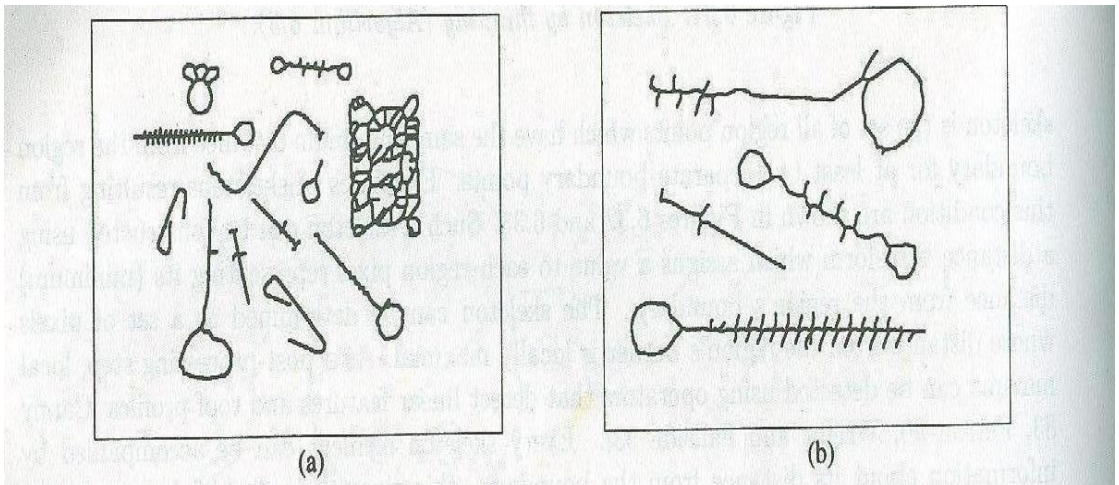
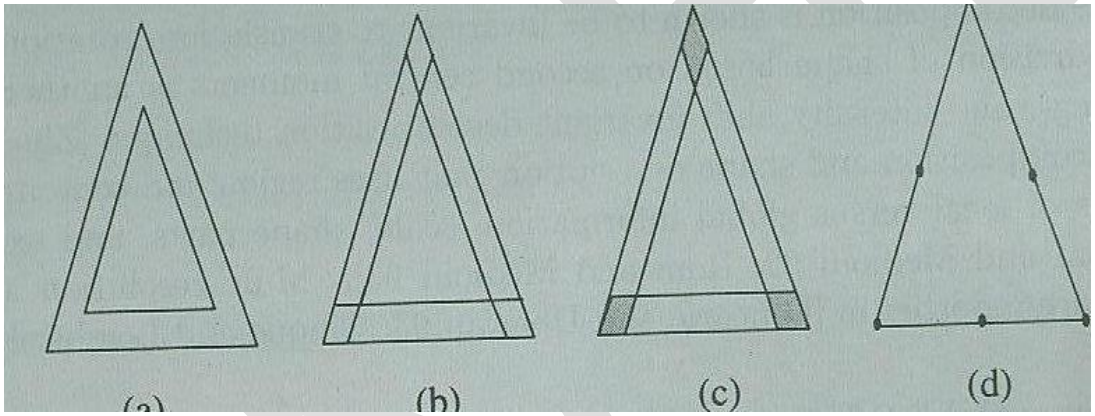


Figure 6.33: *Region skeletons, see Figures 5.1a and 6.2a for original images; thickened for visibility.*

REGION DECOMPOSITION



(a) Region (b) Primary region (c) Primary region and skeleton(d)decomposition graph

ALGORITHM 6: REGION CONVEX HULL CONSTRUCTION

1. Find all pixels of a region R with the minimum row co-ordinate; among them, find the pixel Pi with the minimum column co-ordinate. Assign P/~ = Pi, v = @, —1); the vector v represents the direction of the previous line segment of the convex hull.

2. Search the region boundary in an anti-clockwise direction (Algorithm 7) and compute the angle orientation (pn for every boundary point Pn which lies after the point Pi (in the direction of boundary search—see Figure 26). The angle orientation tpn is the angle of vector PfcPn. The point

Pq satisfying the condition <pq = minn (pn is an element (vertex) of the region convex hull.

3. Assign v = PfcPg, Pk = Pq.

4. Repeat steps 2 and 3 until P^ — Pi. Graph representation based on region skeleton

This method corresponds significantly curving points of a region boundary to graph nodes.

The main disadvantage of boundary-based description methods is that geometrically close points can be far away from one another when the boundary is described—graphical representation methods overcome this disadvantage. Shape properties are then derived from the graph properties.

The region graph is based on the region skeleton, and the first step is the skeleton construction. There are four basic approaches to skeleton construction:

• Thinning—iterative removal of region boundary pixels.

• Wave propagation from the boundary.

• Detection of local maxima in the distance-transformed image of the region.

• Analytical methods.

Most thinning procedures repeatedly remove boundary elements until a pixel set with maximum thickness of 1 or 2 is found. The following algorithm constructs a skeleton of maximum thickness 2.

Algorithm 8.8: Skeleton by thinning

1. Let R be the set of region pixels, Hi(R) its inner boundary, and H0(R) its outer boundary. Let S(R) be a set of pixels from the region R which have all their neighbors in 8-connectivity either from the inner boundary Hi(R) or from the background—from the residuum of R.

2. Construct a region Rnew which is a result of one-step thinning as follows Rnew = S(Roid) U [.Rold ~ Hi(Ro\d)] U [H0[S(R0\d)) H i?old] •

3. If i?new = Roid, terminate the iteration and proceed to step 4. Otherwise assign Ro\d — #new and repeat step 2.

4. Rnew is a set of skeleton pixels, the skeleton of the region R.

Thinning procedures often use a medial axis transform (also symmetric axis transform) to construct a region skeleton .This method corresponds significantly curving points of a region boundary to graph nodes.

The main disadvantage of boundary-based description methods is that geometrically close points can be far away from one another when the boundary is described—graphical representation methods overcome this disadvantage. Shape properties are then derived from the graph properties.

The region graph is based on the region skeleton, and the first step is the skeleton construction. There are four basic approaches to skeleton construction:

• Thinning—iterative removal of region boundary pixels.

• Wave propagation from the boundary.

• Detection of local maxima in the distance-transformed image of the region.

• Analytical methods.

Most thinning procedures repeatedly remove boundary elements until a pixel set with maximum thickness of 1 or 2 is found. The following algorithm constructs a skeleton of

Algorithm 8.9: Region graph construction from skeleton

1. Assign a point description to all skeleton points—end point, node point, normal point.

2. Let graph node points be all end points and node points. Connect any two graph nodes by a graph edge if they are connected by a sequence of normal points in the region skeleton.

It can be seen that boundary points of high curvature have the main influence on the graph. They are represented by graph nodes, and therefore influence the graph structure. close, they can become ambiguous if A, B, C are regions. For instance , the relation to be left of can be defined in many different ways:

• All pixels of A must be positioned to the left of all pixels of B.

• At least one pixel of A must be positioned to the left of some pixel of B.

• The center of gravity of A must be to the left of the center of gravity of B.

All of these definitions seem to be satisfactory in many cases, but they can sometimes be unacceptable because they do not meet the usual meaning of being left of. Human observers are generally satisfied with the definition:

• The center of gravity of A must be positioned to the left of the leftmost point of B and (logical AND) the rightmost pixel of A must be left of the rightmost pixel of B.

Many other inter-regional relations are defined in [Winston, 1975], where relational descriptions are

studied in detail.

An example of applying geometrical relations between simply shaped primitives to shape representation and recognition may be found, where recognition is based on a hypothesize and verify control strategy. Shapes are represented by region neighborhood graphs that describe geometrical relations among primitive shapes.

The model-based approach increases the shape recognition accuracy and makes partially occluded object recognition possible. Recognition of any new object is based on a definition of a new shape model.

KARPAGAM ACADEMY OF HIGHER EDUCATION
COIMBATORE – 641 021

Digital Image Processing

Unit-4 Possible Questions

2 Marks Questions:

1. Write notes on shape representation and description
2. Define shape classes
3. Write notes on region identification
4. Write notes on quad tree region identification
5. Mention some of the contour based shape representation and description
6. Write short notes on chain codes
7. Define boundary length
8. Define curvature and bending energy
9. Write notes on chord distribution
10. Write notes on b-spline representation
11. Define shape invariants
12. Define convex hull
13. Write notes on region skeleton method
14. Mention some advantages of region decomposition
15. Define shape classes with example
16. Discuss about region neighborhood graphs

8 Marks Questions:

1. Illustrate in detail some simple geometric border representation with suitable diagram
2. Explain in detail simple scalar region descriptors with algorithm
3. Explain about region identification using run length encoded data and quad tree identification
4. Explain about region based shape representation and description
5. Explain about contour based shape representation and description

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC ACT 1956)
M.sc Computer Science
DEPARTMENT OF CS, CA & IT
Third Semester(Digital Image processing)

S.no	Questions	OPT1	OPT2	OPT3	OPT4	Answer
1	Region identification is to label each region (or each boundary) with a unique (integer) number; such identification is called	coloring,	colore boundary	cooling	color pallet	coloring,
2	Coloring is otherwise called as _____	colore boundary	connected component	color pallet	coloring label	connected component
3	_____ is a very common occurrence that are found in image shapes such as U-shaped objects, mirrored E objects,	shape collision	image label collision	speed collision	Label collision	Label collision
4	_____ describe an object by a sequence of unit-size line segments with a given orientation.	mirror code	unit code	computer	Chain codes	Chain codes
5	A *mod* 4 or *mod* 8 difference is called a chain code .	derivative	polygon region	curve region	square region	derivative
6	_____ are mostly based on geometric properties of described regions and all of them are sensitive to image	Curvature	Boundary length	Descript	Bending energy	Descriptors
7	_____ are mostly based on geometric properties of described regions *I Boundary length II . Curvature III*	I & II	I & III	I, II & III	II & III	I, II & III
8	A _____ approximates a region by a polygon, the region being represented using its vertices.	curve	square	polygon chain	polygonal representation	polygonal representation
9	Another method for determining the boundary vertices is a based on setting a maximum allowed	tolerance interval	space interval	homogen	both a & b	tolerance interval
10	Sensitivity of shape descriptors to scale is also important if a curve is to be divided into segments and a to	scale-space	polygon chain	sequentia	spacial approach	scale-space approach
11	A multiscale curve description can be represented by an .	internal tree	parent tree	interval tree	quard tree	interval tree
12	Representation of curves using piecewise _____ to obtain smooth curves is widely used in computer graphics.	Contour sequences	polygon - chain	both a & d	polynomial interpolation	polynomial interpolation

13	_____ are piecewise polynomial curves whose shape is closely related to their control polygon - a chain of	P-splines	C-splines	B-splines	E-splines	B-splines
14	Each part of a _____ curve is a third-order polynomial, meaning that it and its first and second derivatives	cubic C-splines	cubic B-spline	H-splines	cubic H-splines	cubic B-spline
15	The _____ has excellent shape description abilities that can be used to describe two-dimensional curves and	Contour sequences	homogeneity	Hough transfor	histogram	Hough transform
16	Region-based shape description use _____.	fuzzy inverse	statistical moments	computer	fuzzy logic	statistical moments
17	_____ approach to shape is gaining attention in image shape description.	Fractal	neural	viewpoin	view	Fractal
18	_____ can be used for shape description, typically in connection with region skeleton construction.	Mathemati	fuzzy logic	homogen	view	Mathematical
19	_____ can be used to recognize shapes in raw boundary representations directly. Contour sequences of	viewpoin	Neural networks	fuzzy logic	hybrid logic	Neural networks
20	_____ represent a very active current research area in machine vision that represent properties of such geometric	Shape invariants	object recognition	Contour sequence	edge	Shape invariants
21	_____ depends on viewpoint, meaning that object recognition may often be impossible as a result of changed	image plane	Contour sequences	none	Shape descriptors	Shape descriptors
22	_____ is the simplest example of a projectively invariant image feature. Any straight line is projected as a	full resolution	image plane	Collinear	edge	Collinearity
23	The basic idea of the projection-invariant _____ is to find such shape features that are unaffected by the transform	viewpoint	shape description	object recogniti	Contour sequences	shape description
24	Application of _____, where invariant descriptors can be computed directly from image data without the need for	minor axes	region bounding	invariant theory	mirror theory	invariant theory
25	Area, rectangularity, elongatedness & direction are examples of	B-splines	region bounding	heuristic region	viewpoint	heuristic region
26	_____ describes a simple topologically invariant property of the object.	Euler's number	Roger number	Millar number	view	Euler's number
27	Euler's number is sometimes called as _____	planar graph	Genus or the Euler-	minor axes	Eccentricity	Genus or the Euler-
28	_____ is the ratio of major and minor axes of an object	machine vision	tree structure	Eccentiry	Eccentricity	Eccentricity

29	_____ is a ratio between the length and width of the region bounding rectangle	Eccentricit	Elongatednes	Euler-Poincare	Euler's number	Elongatednes
30	Properties of this random variable can be described using statistical characteristics - .	regions	subregions	moments	viewpoint	moments
31	The _____ can be used to describe region shape properties and can be used to build a tree structure of region	Convex hull	Millar hull	Millar hall	Millar hill	Convex hull
32	A _____ is generated recursively during the construction of a convex hull.	region concavity	Eccentricity	tree structure	binary tree	region concavity tree
33	Objects can be represented by a planar graph with nodes representing subregions resulting from region decomposition,	region bounding	planar graph	viewpoin	region shape	region shape
34	The first approach to acquire a graph of sub regions is region thinning, which leads to .	nodes	thresholding	region skeleton	subregions	region skeleton
35	The second option to acquire a graph of sub regions starts with the into subregions, which are then	distributed graph	region decompositio	regions	distributed regions	region decompositio
36	The region graph is based on the region skeleton, and the first step is the	skeleton constructio	boundary noise	subregio	region	skeleton construction.
37	_____ is one of the four basic approaches to skeleton construction:	darking	queeing	coloring	thinning	thinning
38	_____ is generally a time-consuming process, although sometimes it is not necessary to look for a skeleton,	thicking	sharping	Thinning	merging	Thinning
39	Thinning procedures often use a _____ transform to construct a region skeleton.	minor axes	poly axis	bio axis	medial axis	medial axis
40	A _____ approach to skeleton construction may also result in decreased sensitivity to boundary noise.	cell resolution	multi-resolution	parallel boundary	contour	multi-resolution
41	_____ are defined at the lower level, primitives being the simplest elements which form the region.	lower level	Shape primitives	high level	none	Shape primitives
42	_____ co-ordinates are represented as pairs of angle Φ and distance r	modal	mode	Polar	pole	Polar
43	_____ co-ordinates codes the tangential directions $\theta(x_n)$ of curve points as a function of path length *n*	planer	zone	skeleton	Tangential	Tangential
44	_____ can be applied to closed curves, co-ordinates of which can be treated as periodic signals.	tree structure	Fourier shape descriptors	Eulear shape	contour	Fourier shape descriptors

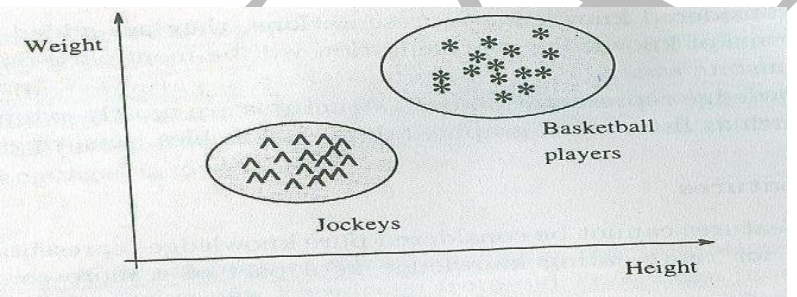
45	Chain codes describe an object by a sequence of unit-size line segments with a given orientation called	parallel boundary	Freeman's code	fuzzy code	freedom code	Freeman's code
46	_____ represent the general shapes of an objects belonging to the same class	region thinning	skeleton	Shape classes	skeleton classes	Shape classes
47	A closed boundary length can be easily evaluated from run length or	graph	modal tree	quad tree	polygon tree	quad tree
48	_____ is defined as the rate of change of slope that must be slightly modified to overcome difficulties of curve	Curvature	Euler-Poincare	quad tree	polygon tree	Curvature
49	The curvature scalar descriptor also called _____ find the ratio between the total number of boundary pixels and	region thinning	boundary straightness	boundary	contour	boundary straightness
50	_____ is computed as a sum of squares of the border curvature over border length	Bending energy	boundary energy	Binding energy	bend energy	Bending energy
51	_____ represents shape description as two dimensional continuous or discrete curves.	neighborin	boundary thinning	region thinning	Geometrical correlation	Geometrical correlation
52	_____ is a popular shape description characteristics independent of linear transformations.	completene	Compactness	effective	collectiveness	Compactness
53	_____ interpret a normalized gray level image function as probability density of a 2D random variable.	region thinning	Statistical moments	mathema	logical moments	Statistical moments
54	The two general approaches to acquire a graph of sub regions are region thinning and	region mapping	region merging	region decompo	both c & b	region decompositio
55	_____ represents every region as a graph node and nodes of neighboring regions are connected by edges.	region decomposit	region merging	region mapping	Region neighborhood	Region neighborhood
56	_____ is a special case of the region neighborhood graph	chord graph	Region adjacency	region merging	both c & b	Region adjacency
57	An unconventional approach to skeletonization that is derived from boundary data considering the scale, the intensity axes of	chord	cores	choss	chock	cores
58	Shape can be represented as a sequence of _____ with specified properties.	region	boundary	segments	edge	segments
59	_____ is a type of projection invariant shape descriptors	Plane conics	acquit angle projection	pie projectio	right angle projection	Plane conics
60	A set of primitive curvature discontinuities is defined with first and second derivatives of a Gaussian in multiple resolutions.	curvature primal	curvature main sketch	curvature	both c & b	curvature primal sketch

UNIT V

SYLLABUS:

Image Recognition: Introduction – Statistical Pattern Recognition - Neural Net- Syntactic Pattern Recognition - Graph Matching – Clustering

Not even the simplest machine vision tasks can be solved without the help of recognition. Pattern recognition is used for region and object classification, and basic methods of pattern recognition must be understood in order to study more complex machine vision processes. Classification of objects or regions has been mentioned several times; recognition is then the last step of the bottom-up image processing approach. It is also often used in other control strategies for image understanding. Almost always, when information about an object or region class is available, some pattern recognition method is used. Consider a simple recognition problem. Two different parties take place at the same hotel at the same time—the first is a celebration of a successful basketball season,



and the second a yearly meeting of jockeys. The doorman is giving directions to guests, asking which party they are to attend. After a while the doorman discovers that no questions are necessary and he directs the guests to the right places, noticing that instead of questions, he can just use the obvious physical features of basketball players and jockeys. Maybe he uses two features to make a decision, the weight and the height of

the guests. All small and light men are directed to the jockey party, all tall and heavier guests are sent to the basketball party. Representing this example in terms of recognition theory, the early guests answered the doorman's question as to which party they are going to visit. This information, together with characteristic features of these guests, resulted in the ability of the doorman to classify them based only on their features. Plotting the guests' height and weight in a two-dimensional space,it is clear that jockeys and basketball players form two easily separable classes and that this recognition task is extremely simple. Although real object recognition problems are often more difficult, and the classes do not differ so substantially, the main principles remain the same. No recognition is possible without knowledge. Decisions about classes or groups into which recognized objects are classified are based on such knowledge—knowledge about objects and their classes gives the necessary information for object classification. Both specific knowledge about the objects being processed and hierarchically higher and more general knowledge about object classes is required. First, common knowledge representation techniques will be introduced, because the concept of knowledge representation in suitable form for a computer may not be straightforward.

1 Knowledge representation

Knowledge as well as knowledge representation problems are studied in artificial intelligence (AI), and computer vision takes advantage of these results. Use of AI methods is very common in higher processing levels, and a study of AI is necessary for a full appreciation of computer vision and image understanding. Here we present a short outline of common techniques as they are used in AI, and an overview of some basic knowledge representations.

• The syntax of a representation specifies the symbols that may be used and the ways that they may be arranged.

• The semantics of a representation specifies how meaning is embodied in the symbols and the symbol arrangement allowed by the syntax.

• A representation is a set of syntactic and semantic conventions that make it possible to describe things.

The main knowledge representation techniques used in AI are formal grammars and languages, predicate logic, production rules, semantic nets, and frames. Even if features and descriptions are not usually considered knowledge representations, they are added for practical reasons; these low-level forms of knowledge representation will be mentioned many times throughout the coming sections. Note that knowledge representation data structures are mostly extensions of conventional data structures such as lists, trees, graphs, tables, hierarchies, sets, rings, nets, and matrices.

Descriptions, features

Descriptions and features cannot be considered pure knowledge representations. Nevertheless, they can be used for representing knowledge as a part of a more complex

representation structure. Descriptions usually represent some scalar properties of objects, and are called features. Typically, a single description is insufficient for object representation, therefore the descriptions are combined into feature vectors. Numerical feature vectors are inputs for statistical pattern recognition techniques.

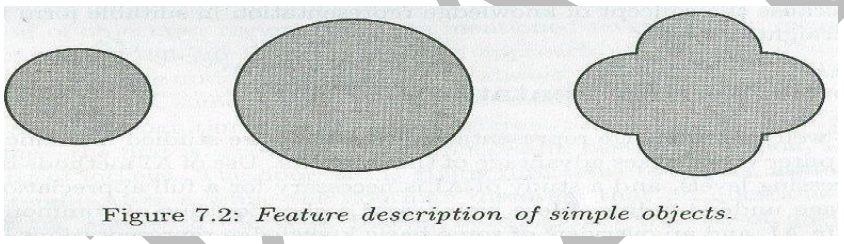


Figure 7.2: *Feature description of simple objects.*

Grammars, languages

If an object's structure needs to be described, feature description is not appropriate. A structural description is formed from existing primitives (elementary structural properties of the objects) and the relations between them. Primitives are represented by information about their types. The simplest form of structure representations are chains, trees, and general graphs. Structural description of chromosomes using border segments as primitives is a classic example of structural

object description, where borders are represented by a chain of symbols, the symbols representing specific types of border primitives. Hierarchical structures can be represented by trees—the concavity tree of serves as an example. Grammars and languages (similar to natural languages) provide rules defining how the chains, trees, or graphs can be constructed from a set of symbols (primitives).

Predicate logic

Predicate logic plays a very important role in knowledge representation- -it introduces a mathematical formalism to derive new knowledge from old knowledge by applying mathematical deduction. Predicate logic works with combinations of logic variables, quantifiers C,V), and logic operators (and, or, not, implies, equivalent). The logic variables are binary (true, false). The idea of proof and rules of inference such as modus ponens and resolution are the main building blocks of predicate logic. Predicate logic forms the essence of the programming language PROLOG, which is widely used if objects are described by logic variables. Requirements of 'pure truth' represent the main weakness of predicate logic in knowledge representation, since it does not allow work with uncertain or incomplete information. Predicate logic incorporates logic conditions and constraints into knowledge.

Production rules

Production rules represent a wide variety of knowledge representations that are based on condition action pairs. The essential model of behavior of a system based on production rules (a production system) can be described as follows: if condition X holds then action Y is appropriate.

Information about what action is appropriate at what time represents knowledge. The procedural character of knowledge represented by production rules is another important property—not all the information about objects must be listed as an object property. Consider a simple knowledge base where the following knowledge is presented **using a production rule:**

if ball then circular.

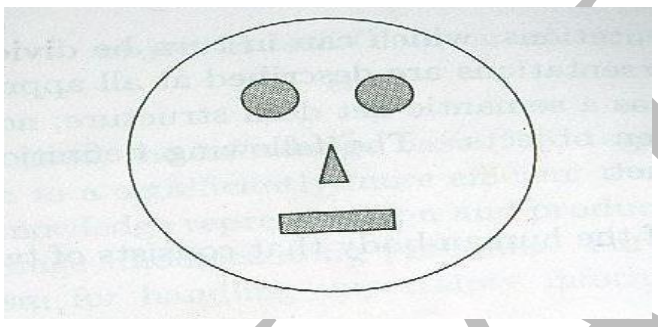
Let the knowledge base also include the statements

object A is_a ball, object B is a ball, object C is_a shoe, etc.

To answer the question how many objects are circular?, if enumerative knowledge representation is used, the knowledge must be listed as

object A is_a (ball, circular),

object B is_a (ball, circular), etc. If procedural knowledge is used, the knowledge base together with the knowledge gives the same information in a significantly more efficient manner. Both production rule knowledge representation and production systems appear frequently in computer vision and image understanding problems. Furthermore, production systems, together with a mechanism for handling uncertainty information, form a basis of expert systems.



Fuzzy logic

Fuzzy logic has been developed to overcome the obvious limitations of numerical or crisp representation of information. Consider the use of knowledge represented by production rule for recognition of balls; using the production rule, the knowledge about balls may be represented as if circular then ball.

If the object in a two-dimensional image is considered circular then it may represent a ball. Our experience with balls, however, says that they are usually close to, but not perfectly, circular. Thus, it is necessary to define some circularity threshold so that all reasonably circular objects from our set of objects are labeled as balls. Here is the fundamental problem of crisp descriptions: how circular must an object be to be

considered circular? If humans represent such knowledge, the rule for ball circularity may look like if circularity is HIGH then object is a ball with HIGH confidence.

Clearly, high circularity is a preferred property of balls. Such knowledge representation is very close to common sense representation of knowledge, with no need for exact specification of the circularity/non-circularity threshold. Fuzzy rules are of the form if X is A then Y is B, where X and Y represent some properties and A and B are linguistic variables. Fuzzy logic can be used to solve object recognition and other decision-making tasks, among others.

Semantic nets

Semantic nets are a special variation of relational data structures. The semantics distinguish them from general nets—semantic nets consist of objects, their description, and a description of relations between objects (often just relations between neighbors). Logical forms of knowledge can be included in semantic nets, and predicate logic can be used to represent and/or evaluate the local information and local knowledge. Semantic nets can also represent common sense knowledge that is often imprecise and needs to be treated in a probabilistic way. Semantic nets have a hierarchical structure; complex representations consist of less complex representations, which can in turn be divided into simpler ones, etc. Relations between partial representations are described at all appropriate hierarchical levels.

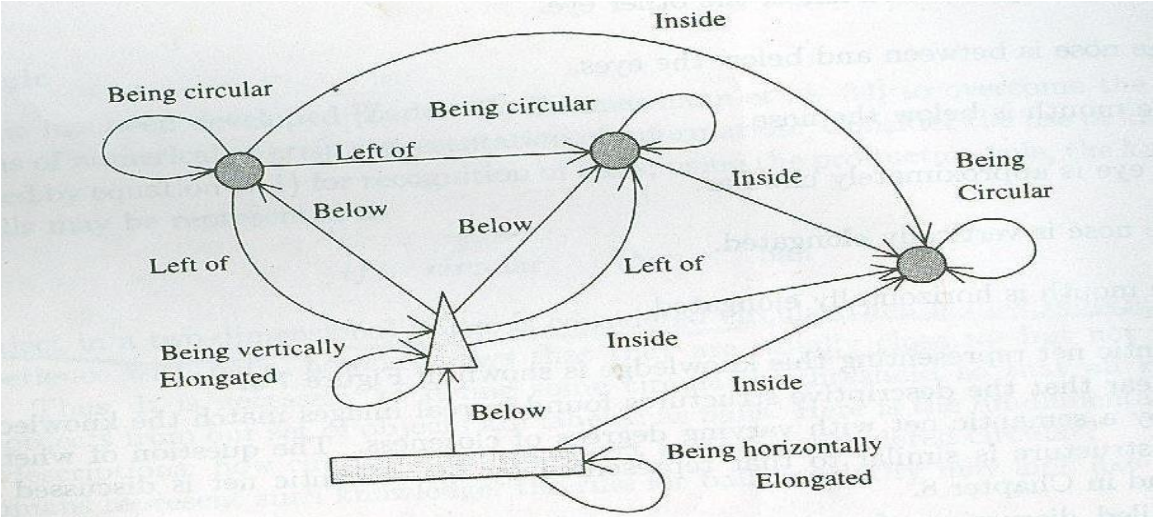


Figure 7.3: *Semantic nets: human face model and its net.*

Evaluated graphs are used as a semantic net data structure; nodes represent objects and arcs represent relations between objects. The following definition of a human face is an example of a simple semantic net: • A face is a circular part of the human body that consists of two eyes, one nose, and one mouth.

• One eye is positioned left of the other eye.

• The nose is between and below the eyes.

• The mouth is below the nose.

• An eye is approximately circular.

• The nose is vertically elongated.

• The mouth is horizontally elongated. Frames, scripts

Frames provide a very general method for knowledge representation which may contain all the knowledge representation principles discussed so far. They are sometimes called scripts because of their similarity to film scripts. Frames are suitable for representing common sense knowledge under specific circumstances. Consider a frame called plane_start this frame may consist of the following sequence of actions:

1. Start the engines.

2.Taxi to the runway.

3.Increase RPMs of engines to maximum.

4. Travel along runway increasing speed.
5. Fly.

Assuming this frame represents knowledge of how planes usually start, the situation of a plane standing on a runway with engines running causes the prediction that the plane will start in a short time. The frame can be used as a substitute for missing information which may be extremely important in vision-related problems.

Assuming that one part of the runway is not visible from the observation point, using the plane_start frame, a computer vision system can overcome the lack of continuous information between the plane moving at the beginning of the runway and flying when it next appears. If it is a passenger plane, the frame may have additional items such as time of departure, time of arrival, departure city, arrival city, airline, flight number, etc., because in a majority of cases it makes sense to be interested in this information if we identify a passenger plane.

From a formal point of view, a frame is represented by a general semantic net accompanied by a list of relevant variables, concepts, and concatenation of situations. No standard form of frame exists. Frames represent a tool for organizing knowledge in prototypical objects, and for description of mutual influences of objects using stereotypes of behavior in specific situations. Examples of frames can be found. Frames are considered high-level knowledge representations.

2 Statistical pattern recognition

An object is a physical unit, usually represented in image analysis and computer vision by a region in a segmented image. The set of objects can be divided into disjoint subsets, that, from the classification point of view, have some common features and are called classes. The definition of how the objects are divided into classes is ambiguous and depends on the classification goal. Object recognition is based on assigning classes to objects, and the device that does these assignments is called the classifier. The number of classes is usually known beforehand, and typically can be derived from the problem specification. Nevertheless, there are approaches in which the number of classes may not be known .The classifier (similarly to a human) does

not decide about the class from the object itself—rather, sensed object properties serve this purpose. For example, to distinguish steel from sandstone, we do not have to determine their molecular structures, although this would describe these materials well. Properties such as texture, specific weight, hardness, etc., are used instead. This sensed object is called the pattern, and the classifier does not actually recognize objects, but recognizes their patterns. Object recognition and pattern recognition are considered synonymous.

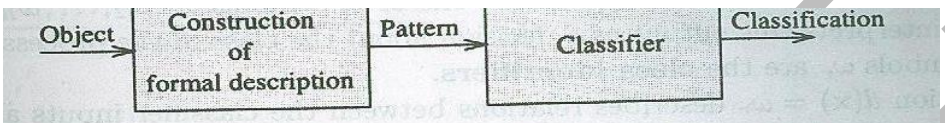


Figure 7.4: *Main pattern recognition steps.*

The main pattern recognition steps are shown in Figure 7.4. The block 'Construction of formal description' is based on the experience and intuition of the designer. A set of elementary properties is chosen which describe some characteristics of the object; these properties are measured in an appropriate way and form the description pattern of the object. These properties can be either quantitative or qualitative in character and their form can vary (numerical vectors, chains, etc.). The theory of recognition deals with the problem of designing the classifier for the specific (chosen) set of elementary object descriptions.

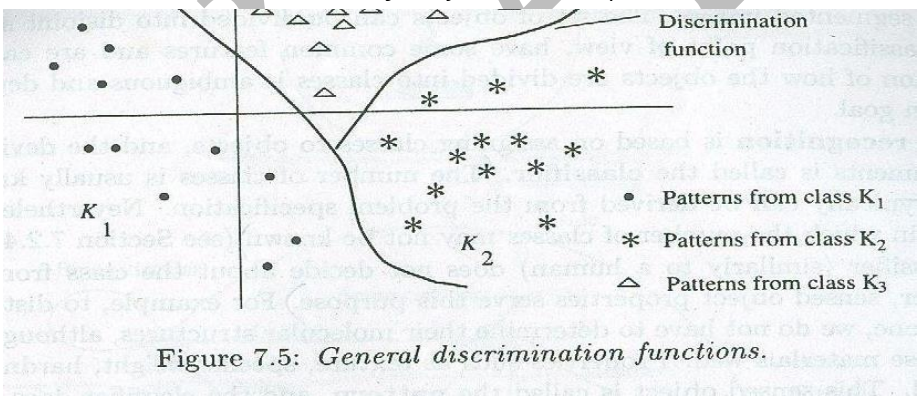


Figure 7.5: *General discrimination functions.*

The pattern (also referred to as pattern vector, or feature vector) x = (xi,.T2,...

,xn) that describes an object is a vector of elementary descriptions, and the set of all possible patterns forms the pattern space X (also called feature space). If the elementary descriptions were appropriately chosen, similarity of objects in each class results in the proximity of their patterns in pattern space. The classes form clusters in the feature space, which can be separated by a discrimination curve (or hyper-surface in a multi-dimensional feature space).

If a discrimination hyper-surface exists which separates the feature space such that only objects from one class are in each separated region, the problem is called a recognition task with separable classes. If the discrimination hyper-surfaces are hyper-planes, it is called a linearly separable task. If the task has separable classes, each pattern will represent only objects from one class. Intuitively, we may expect that separable classes can be recognized without errors.

The majority of object recognition problems do not have separable classes, in which case the locations of the discrimination hyper-surfaces in the feature space can never separate the classes correctly and some objects will always be misclassified.

2.1 Classification principles

A statistical classifier is a device with n inputs and 1 output. Each input is used to enter the information about one of n features $x_1, x_2, \ldots, x_n$ that are measured from an object to be classified. An R-class classifier will generate one of R symbols $\omega_1, \omega_2, \ldots, \omega_R$ as an output, and the user interprets this output as a decision about the class of the processed object. The generated symbols cjr are the class identifiers.

The function $d(x) = \omega_r$ describes relations between the classifier inputs and the output; this function is called the decision rule. The decision rule divides the feature space into R disjoint subsets Kr, r = 1,... ,i?, each of which includes all the feature representation vectors x' of objects for which $d(x') = \omega_r$. The borders between subsets Kr, r — 1,..., R form the discrimination hyper-surfaces mentioned earlier. The determination of discrimination hyper-surfaces (or definition of the decision rule) is the goal of classifier design.

$$g_r(x) \geq g_s(x)$$

If all the discrimination functions of the classifier are linear, it is called a linear classifier. Another possibility is to construct classifiers based on the minimum distance principle.

$$g_r(x) - g_s(x) = 0$$

The resulting classifier is just a special case of classifiers with discrimination functions, but they have computational advantages and may easily be implemented on digital computers.

If each class is represented by just one exemplar, a linear classifier results. If more than one exemplar represents some class, the classifier results in piece wise linear discrimination hyper-planes Non-linear classifiers usually transform the original feature space Xn into a new feature space Xm applying some appropriate non-linear function <?, where the superscripts n, m refer to the space dimensionality

Classifier setting

A classifier based on discrimination functions is a deterministic machine—one pattern x will always be classified into the same class. Note that the pattern x may represent objects from different classes, meaning that the classifier decision may be correct for some objects and incorrect for others. Therefore, setting of the optimal classifier should be probabilistic. Incorrect classifier decisions cause some losses to the user, and according to the definition of loss, different criteria for optimal classifier settings will be obtained.

Let the classifier be considered a universal machine that can be set to represent any decision rule from the rule set D. The set D may be ordered by a parameter vector q that refers to particular discrimination rules. The value of the mean loss J(q) depends on the decision rule that is applied: uj — d(x, q). In comparison with the definition of decision rule used in the previous section, the parameter vector q has been added to represent the specific decision rule used by the classifier. The decision rule $\omega = d(x, q^*)$ (9.12)

that gives the minimum mean loss J(q) is called the optimum decision rule, and q* is called the vector of optimal parameters

$$J(\mathbf{q}^*) = \min_{\mathbf{q}} J(\mathbf{q}) \qquad d(\mathbf{x},\mathbf{q}) \in D \quad . \quad (9.13)$$

The minimum error criterion (Bayes criterion, maximum likelihood) uses loss functions of the form where A(.) is the number that describes quantitatively the loss incurred if a pattern x which should be classified into the class us is incorrectly

classified into the class $\omega_r = d(\mathbf{x},\mathbf{q})$. (9.14)

$$J(\mathbf{q}) = \int_X \sum_{s=1} \lambda[d(\mathbf{x},\mathbf{q})|\omega_s] p(\mathbf{x}|\omega_s) P(\omega_s)\, d\mathbf{x}$$

A classifier that has been set according to the minimum-loss optimality criterion is easy to construct using discrimination functions

If the statistical properties of patterns are known, the necessary sizes of the training sets can be estimated, but the problem is that in reality they are not usually known. The training set is actually supposed to substitute this missing statistical information. Only after processing of the training set can the designer know whether it was sufficient, and whether an increase in the training set size is necessary.

The training set size will typically be increased several times until the correct classification setting is achieved. The problem, which originally could not be solved, uses more and more information as the training set size increases, until the problem specifications can be met.

The general idea of sequential increase in training set size can be understood as presenting small portions of a large training set to the classifier, whose performance is checked after each portion. The smallest portion size is one element of the training set. Sequential processing of information (which cannot be avoided in principle) has some substantial consequences in the classifier setting process. All the properties of the classifier setting methods given have analogies in the learning process of living organisms. The basic properties of learning can be listed as follows.

• Learning is the process of automated system optimization based on the sequential presentation of examples.

• The goal of learning is to minimize the optimality criterion. The criterion may be represented by the mean loss caused by incorrect decisions.

• The finite size of the training set requires the inductive character of learning. The goal of learning must be achieved by generalizing the information from examples, before all feasible examples have been presented. The examples may be chosen at random.

• The unavoidable requirements of sequential information presentation and the finite size of system memory necessitate the sequential character of learning. Therefore, learning is not a one-step process, but rather a step-by-step process of improvement. The learning process searches out the optimal classifier setting from examples. The classifier system is constructed as a universal machine that becomes optimal after processing the training set examples (supervised learning), meaning that it is not necessary to repeat the difficult optimal system design if a new application appears. Learning methods do not depend on the application; the same learning algorithm can be applied if a medical diagnostics classifier is set just as when an object recognition classifier for a robot is set.

The quality of classifier decisions is closely related to the quality and amount of information that is available. From this point of view, the patterns should represent as complex a description as possible. On the other hand, a large number of description features would result. Therefore, the object description is always a trade-off between the permissible classification error, the complexity of the classifier construction, and the time required for classification. This results in a question of how to choose the best features from a set of available features, and how to detect the features with the highest contribution to the recognition success.

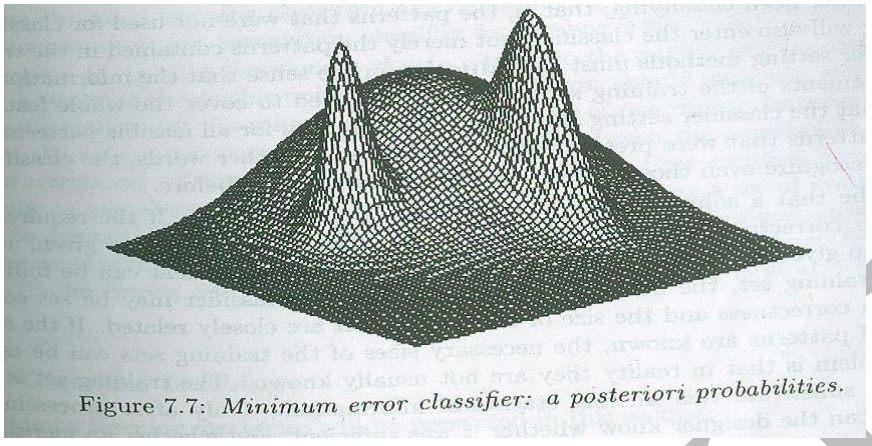


Figure 7.7: *Minimum error classifier: a posteriori probabilities.*

2.3 Classifier learning

Two common learning strategies will be presented in this section:

• Probability density estimation estimates the probability densities p(x.¥ujr) and probabilities P(ccv), r — 1,..., R. The discrimination functions are computed according to the minimum error criterion

• Direct loss minimization finds the decision rule uj = d(x, q*) by direct minimization of losses J(q) without estimation of probability densities and probabilities. The criterion of the best approximation is applied. Probability density estimation methods differ in computational difficulty according to the amount of prior information available about them. If some prior information is available, it usually describes the shape of probability density functions p(x.¥ur). The parameters describing the distribution are not usually known, and learning must find the estimate of these parameters. Therefore, this class of learning methods is sometimes called parametric learning. Learning and classification based on estimates of probability densities assuming the normal distribution

1. Learning: Compute the estimates of the mean value vector fir and the dispersion matrix *r, equations (9.22) and/or (9.25), (9.29).
2. Compute the estimates of the a priori probability densities p(x|oy), equation (9.20).
3. Compute the estimates of the a priori probabilities of classes, equation (9.32).
4. Classification: Classify all patterns into the class r if = max (p(x.¥ui) P(ui))

If no prior information is available (i.e., even the distribution type is not known), the

computation is more complex. In such cases, if it is not necessary to use the minimum error criterion, it is advantageous to use a direct loss minimization method. No probability densities or probabilities are estimated in the second group of methods based on direct minimization of losses. The minimization process can be compared to gradient optimization methods, but pure gradient methods cannot be used because of unknown probability densities, so the gradient cannot be evaluated.

Syntactic pattern recognition

Quantitative description of objects using numeric parameters (the feature vector) is used in statistical pattern recognition, while qualitative description of an object is a characteristic of syntactic pattern recognition. The object structure is contained in
the syntactic description. Syntactic object description should be used whenever feature
description is not able to represent the complexity of the described object and/or when
the object can be represented as a hierarchical structure consisting of simpler parts.
The elementary properties of the syntactically described objects are called primitives
. Graphical or relational descriptions of objects where primitives represent sub-regions of specific shape is another .After each primitive has been assigned a symbol, relations between primitives in the object are described, and a relational structure results. As in statistical recognition, the design of description primitives and their relation is not algorithmic. The design is based on the analysis of the problem, designer experience, and abilities. However, there are some principles that are worth following:

1. The number of primitive types should be small.
2. The primitives chosen must be able to form an appropriate object representation.
Primitives should be easily segment able from the image.
4. Primitives should be easily recognizable using some statistical pattern recognition method.
5. Primitives should correspond with significant natural elements of the object (image)

structure being described.

For example, if technical drawings are described, primitives are line and curve segments, binary relations describe relations such as to be adjacent, to be left of, to be above, etc. This description structure can be compared with the structure of a natural language. The text consists of sentences, sentences consist of words, words are constructed by concatenation of letters. Letters are considered primitives in this example; the set of all letters is called the alphabet. The set of all words in the alphabet that can be used to describe objects from one class (the set of all feasible descriptions) is named the description language and represents descriptions of all objects in the specific class. In addition, a grammar represents a set of rules that must be followed when words of the specific language are constructed from letters (of the alphabet).

Assume that the object is appropriately described by some primitives and their relations. Moreover, assume that the grammar is known for each class that generates descriptions of all objects of the specified class. Syntactic recognition decides whether the description word is or is not syntactically correct according to the particular class grammars, meaning that each class consists only of objects whose syntactic description can be generated by the particular grammar. Syntactic recognition is a process that looks for the grammar that can generate the syntactic word that describes an object. We mentioned relational structure in the correspondence with the syntactic description of objects. Each relational structure with multiple relations can be transformed to a relational structure with at most binary relations; the image object is then represented by a graph which is planar if relations with adjacent regions only are considered. A graphical description is very natural, especially in the description of segmented images. Each planar graph can be represented either by a graph grammar or by a sequence of symbols (chain, word, etc.) over an alphabet. Sequential representation is not always advantageous in image object recognition because the valuable correspondence between the syntactic description and the object may be lost. Nevertheless, work with chain grammars is more straightforward and understandable,

and all the main features of more complex grammars are included in chain grammars. Therefore, we will discuss principally sequential syntactic descriptions and chain grammars.

The syntactic recognition process is described by the following algorithm:

Algorithm: Syntactic recognition

1. Learning: Based on the problem analysis, define the primitives and their possible relations.
2. Construct a description grammar for each class of objects using either hand analysis of syntactic descriptions or automated grammar inference (see Section 9.4.3).
3. Recognition: For each object, extract its primitives first; recognize the primitives' classes and describe the relations between them. Construct a description word representing an object.
4. Based on the results of the syntactic analysis of the description word, classify an object into that class for which its grammar (constructed in step 2) can generate the description word.

It can be seen that the main difference between statistical and syntactic recognition is in the learning process. Grammar construction can rarely be algorithmic using today's approaches, requiring significant human interaction. It is usually found that the more complex the primitives are, the simpler is the grammar, and the simpler and faster is the syntactic analysis. More complex description primitives on the other hand make step 3 of the algorithm more difficult and more time consuming; also, primitive extraction and evaluation of relations may not be simple.

Grammars and languages

Assuming that the primitives have been successfully extracted, all the inter-primitive relations can then be described syntactically as ?z-ary relations; these relations form structures (chains, trees, graphs) called words that represent the object or the pattern. Each pattern is therefore described by a word. Primitive classes can be understood as letters from the alphabet of symbols called terminal symbols. Let the alphabet of terminal symbols be V$.

The set of patterns from a particular class corresponds to a set of words. This set of words is called the formal language and is described by a grammar. The grammar is a mathematical model of a generator of syntactically correct words (words from theparticular language); it is a quadruple The set P is a non-empty finite subset of the set F x P; elements of P are called the substitution rules. The set of all words that can be generated by the grammar G is called the languageL(G). Grammars that generate the same language are called equivalent.

Grammars can be divided into four main groups ordered from the general

1. Type 0—General Grammars

There are no limitations for the substitution rules.

2. Type 1—Context-Sensitive Grammars

3. Type 2—Context-Free Grammars ,The substitution rule S —» e may be included.

All the grammars discussed so far have been non-deterministic. The same left-hand side might appear in several substitution rules with different right-hand sides, and no rule exists that specifies which rule should be chosen. A non-deterministic grammar generates a language in which no words are 'preferred'. If it is advantageous to generate some words (those more probable) more often than others, substitution rules can be accompanied by numbers (for instance, by probabilities) that specify how often the substitution rule should be applied. If the substitution rules are accompanied by probabilities, the grammar is called stochastic. If the accompanying numbers do not satisfy the properties of probability (unit sum of probabilities for all rules with the same left-hand side), the grammar is called fuzzy. Note that the evaluation of the frequency with which each substitution rule should be used can substantially increase the efficiency of syntactic analysis in the recognition stage.

Syntactic analysis, syntactic classifier

If appropriate grammars exist that can be used for representation of all patterns in their classes, the last step is to design a syntactic classifier which assigns the pattern (the word) to an appropriate class. It is obvious that the simplest way is to construct a separate grammar for each class; an unknown pattern x enters a parallel

structure of blocks that can decide if x ? L(Gj), where j = 1,2,... R and R is the number of classes; L(Gj) is the language generated by the jth grammar. If the jth block's decision is positive, the pattern is accepted as a pattern from the jth class and the classifier assigns the pattern to the class j. Note that generally more than one grammar can accept a pattern as belonging to its class.

The decision of whether or not the word can be generated by a particular grammar is made during syntactic analysis. Moreover, syntactic analysis can construct the pattern derivation tree which can represent the structural information about the pattern. If a language is finite (and of a reasonable size), the syntactic classifier can search for a match between the word being analyzed and all the words of the language. Another simple syntactic classifier can be based on comparisons of the chain word descriptions with typical representatives of classes comparing primitive type presence only. This method is very fast and easily implemented, though it does not produce reliable results since the syntactic information is not used at all. However, impossible classes can be rejected in this step, which can speed up the syntactic analysis process.

Syntactic analysis is based on efforts to construct the tested pattern by the application of some appropriate sequence of substitution rules to the start symbol. If the substitution process is successful, the analysis process stops and the tested pattern can be generated by the grammar. The pattern can be classified into the class represented by the grammar.

If the substitution process is unsuccessful, the pattern is not accepted as representing an object of this class. If the class description grammar is regular (type 3), syntactic analysis is very simple.

The pure top-down approach is not very efficient, since too many incorrect paths are generated. The number of misleading paths can be decreased by application of consistency tests. Tree pruning is often used if an exhaustive search cannot be completed because the search effort would exceed any reasonable bounds. Note that pruning can mean that the final solution is not optimal or may not be found at all. This

depends on the quality of the a priori information that is applied during the pruning process.

There are two main principles for recovery from following a wrong path. The first one is represented by the back-tracking mechanism already mentioned, meaning that the generation of words returns to the nearest point in the tree where another substitution rule can be applied which has not yet been applied. This approach requires the ability to re-construct the former appearances of generated sub-words and/or remove some branches of the derivation tree completely.

The second approach does not include back-tracking. All possible combinations of the substitution rules are applied in parallel and several generation trees are constructed simultaneously. If any tree succeeds in generating the analyzed word, the generation process ends. If any tree generation ends with a non-successful word, this tree is abandoned. The latter approach uses more brute force, but the algorithm is simplified by avoiding back-tracking.

It is difficult to compare the efficiency of these two and the choice depends on the application; Bottom-up analysis is more efficient for some grammars, and top-down is more efficient for others. As a general observation, the majority of syntactic analyzers which produce all generated words is based on the top-down principle. This approach is appropriate for most grammars but is usually less efficient.

Another approach to syntactic analysis uses example relational structures of classes. The syntactic analysis consists of matching the relational structure that represents the analyzed object with the example relational structure. The main goal is to find an isomorphism of both relational structures. These methods can be applied to n-ary relational structures as well. Relational structure matching is a perspective approach to syntactic recognition, a perspective way of image understanding.

Syntactic classifier learning, grammar inference

To model a language of any class of patterns as closely as possible, the grammar rules should be extracted from a training set of example words. This process of grammar construction from examples is known as grammar inference.

The source of words generates finite example words consisting of the terminal symbols. Assume that these examples include structural features that should be represented by a grammar G which will serve as a model of this source. All the words that can be generated by the source are included in the language ?(??), and the words that cannot be generated by the source represent a residuum of this set LC(G). This information enters the inference algorithm whose goal is to find and describe the grammar G. Words that are included in the language L(G) can be acquired simply from the source of examples. However, the elements of LC(G) must be presented by a teacher that has additional information about the grammar properties.

This property of grammar inference is extremely unsuitable for syntactic analysis because of the computational complexity. Therefore, the main role in syntactic analyzer learning is still left to a human analyst, and the grammar construction is based on heuristics, intuition, experience, and prior information about the problem. If the recognition is based on sample relational structures, the main problem is in its automated construction. The conventional method for the sample relational structure construction is described in here the relational descriptions of objects from the training set are used. The training set consists of examples and counter-examples.

5 Recognition as graph matching

The following section is devoted to recognition methods based on graph comparisons. Graphs with evaluated nodes and evaluated arcs will be considered as they appear in the image description using relational structures. The aim is to decide whether the reality represented by an image matches prior knowledge about the image incorporated into the graphical models. An example of a typical graph matching .If this task is presented as an object recognition problem, the object graph must match the object model graph exactly. If the problem is to find an object (represented by a model graph) in the graphical representation of the image, the model must match a sub-graph in the image graph exactly..

Graph isomorphism and sub-graph isomorphism evaluation is a classical problem in graph theory and is important from both practical and theoretical points of

view. Isomorphism of graphs and sub-graphs Regardless of whether graph or sub-graph isomorphism is required, the problems can be divided into **three main classes .**
1. Graph isomorphism. Given two graphs G¥ = (V¥,Ei) and G2 = (V2.E2), find a one-to-one and onto mapping (an isomorphism) / between V¥ and V2 such that for each edge of E¥ connecting any pair of nodes v,vf ? Vi, there is an edge of E2 connecting f(v) and /(?/); further, if f(v) and f(vf) are connected by an edge in G2, v and v1 are connected in G¥.
2.Sub-graph isomorphism. Find an isomorphism between a graph G¥ and sub-graphs of another graph G2. This problem is more difficult than the previous one.
3.Double sub-graph isomorphism. Find all isomorphisms between sub-graphs of a graph G¥ and sub-graphs of another graph G2. This problem is of the same order of difficulty as number 2.

The sub-graph isomorphism and the double sub-graph isomorphism problems are ATP-complete, meaning that, using known algorithms, the solution can only be found in time proportional to an exponential function of the length of the input. It is still not known whether the graph isomorphism problem is iVP-complete. Despite extensive effort, there is neither an algorithm that can test for graph isomorphism in polynomial time, nor is there a proof that such an algorithm cannot exist. However, non-deterministic algorithms for graph isomorphism that use heuristics and look for sub-optimal solutions give a solution in polynomial time in both graph and sub-graph isomorphism testing. Isomorphism testing is computationally expensive for both non-evaluated and evaluated graphs. Evaluated graphs are more common in recognition and image understanding, where nodes are evaluated by properties of regions they represent, and graph arcs are evaluated by relations between nodes they connect.

The evaluations can simplify the isomorphism testing. More precisely, the evaluation may make disproof of isomorphism easier. Isomorphic evaluated graphs have the same number of nodes with the same evaluation, and the same number of arcs with the same evaluation. An isomorphism test of two evaluated graphs G¥ = (Vi, E¥) and G2 = (V2, E2) can be based on partitioning the node sets V¥ and V2 in a

consistent manner looking for inconsistencies in the resulting set partitions. The goal of the partitioning is to achieve a one-to-one correspondence between nodes from sets V¥ and V2 for all nodes of the graphs Gi and G2. The algorithm consists of repeated node set partitioning steps, and the necessary conditions of isomorphism are tested after each step (the same number of nodes of equivalent properties in corresponding sets of both graphs). The node set partitioning may, for example, be based on **the following properties:**

- Node attributes (evaluations).
- The number of adjacent nodes (connectivity).
- The number of edges of a node (node degree).
- Types of edges of a node.
- The number of edges leading from a node back to itself (node order).
- The attributes of adjacent nodes.

Algorithm 9: Graph isomorphism

1. Take two graphs Gx = (VuEi), G2 = (V2, E2).
2. Use a node property criterion to generate subsets Vu, V2i of the node sets V¥ and V2. Test whether the cardinality conditions hold for corresponding subsets. If not, the isomorphism is disproved.
3. Partition the subsets Vh, V2i into subsets Wij, W2j satisfying the conditions given in equation (9.74) (no two subsets W¥j or W2j contain the same node). Test whether the cardinality conditions hold for all the corresponding subsets W¥j, W2j. If not, the isomorphism is disproved.
4. Repeat steps 2 and 3 using another node property criterion in all subsets Wij,W2j generated so far. Stop if one of the three above-mentioned situations occurs.
5. Based on the situation that stopped the repetition process, the isomorphism either was proved, disproved, or some additional procedures (such as back-tracking) must be applied to complete the proof or disproof.

Digital Image Processing

Unit-5 Possible Questions

2 Marks Questions:

1. Define syntax and semantics
2. Define representation
3. Write notes on descriptions and features
4. Write notes on grammars and languages
5. Mention the use of predicate logic
6. Mention the representation for production rules with example
7. Write notes on fuzzy logic
8. Write notes on semantic nets
9. Define frames and scripts
10. Write notes on classification principle
11. Define learning process and its goal
12. Mention the two common learning strategies adopted in classifier learning
13. Define neural nets
14. Write notes on cluster analysis
15. Mention the two main groups of cluster analysis methods
16. Define primitives
17. Mention the four main groups of grammars
18. Mention the three main classes of graph isomorphism

8 Marks Questions:

1. Discuss in detail about statistical pattern recognition with suitable diagram
2. Explain in detail about mac queen k-means cluster analysis method with example
3. Explain in detail about the various knowledge representation techniques used in ai
4. Discuss in detail about feed-forward neural networks with algorithm
5. Explain the following
   I). unsupervised learning ii) hop-field neural nets
6. Discuss in detail about syntactic pattern recognition
7. Illustrate with diagram the concept of recognition as graph matching

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC ACT 1956)
M.sc Computer Science
DEPARTMENT OF CS, CA & IT
Third Semester(Digital Image processing)

S.no	Questions	OPT1	OPT2	OPT3	OPT4	Answer
1	_____ is used for region and object classification	pattern mapping	pattern generation	Pattern recognition	pattern finding	Pattern recognition
2	Decisions about classes or groups into which recognized objects are classified are based on , which gives	database	knowledge	data	informa	knowledge
3	A _____ is a set of syntactic and semantic conventions that makes it possible to describe things.	mapping	discovery	representatio	plotting	representation
4	Descriptions usually represent some scalar properties of objects and are called	features	color	character	texture	features
5	A single description is insufficient for object representation, therefore the descriptions are combined into	feature vectors	character vectors	texture vectors	color vectors	feature vectors
6	The whole class of objects cannot be described by a single chain, a single tree etc. but a class of structurally described	definitions	grammars and languages	grammars	both a & c	grammars and languages
7	_____ uses mathematical formalism to derive new knowledge from old knowledge by applying mathematical	pre-gate logic	post- logic	Predicate logic	pre-logic	Predicate logic
8	The idea of proof and rules of inference such as are the main building blocks of predicate	modus pones	region	both a & b	modus pones	modus pones and resolution
9	_____ represent a wide variety of knowledge representations that are based on condition action pairs.	production	Production rules	production logic	product	Production rules
10	_____ was developed to overcome the obvious limitations of numeric and crisp representation of information	fuzzy algorithm	Fuzzy logic	Dclar's logic	Dclar's algorith	Fuzzy logic
11	_____ are special variations of relational data structures that consists of objects, their descriptions and a	Semantic roots	Semantic web	Semantic nets	both c & b	Semantic nets
12	_____ provide a very general method for knowledge representation	Only frames	only codes	only scripts	Frames or	Frames or scripts

13	_____ is based on assigning classes to objects	Object orientation	Object recognition	Object linking	Object binding	Object recognition
14	The device that assigns classes to objects is called the	classification	classifier	class builder	class	classifier
15	The classifier that decides the class for an object from its properties and the sensed object is called the	texture	color group	gray level	pattern	pattern
16	The set of all possible patterns forms the pattern space also called	texture space	texture matching	feature detection	feature space	texture space
17	The generated symbols $\mathbf{w_r}$ that is used as a decision about the class of the processed object are called as	class identifiers	class builder	classifier	both c & b	class builder
18	The function $\mathbf{d(x)= w_r}$ that describes relations between the classifier input and the output is called the	decision chart	decision rule	both a & b	pattern	both a & b
19	If all the discrimination functions of the classifier are linear, it is called	classifier	non linear classifier	linear classifier	pre-classifi	linear classifier
20	The different criteria for optimal classifier settings can be obtained using minimum error criterion or	least approximatio	best optimal criterion	best approximati	minimu	best approximation criterion
21	The ability to set classification parameters from a set of examples is called	classifier learning	pre- classifier	class builder	both c & b	classifier learning
22	The set of patterns and their classes is called the	tutorial	training set	tutorial set	tutorial guide	training set
23	_____ is the process of automated system optimization based on the sequential presentation of examples.	Learning	reasoning	reading	both c & b	Learning
24	The finite size of the training set requires the _____ character of learning.	automata	beta	infinite	inducti	inductive
25	The unavoidable requirements of sequential information presentation and the finite size of system memory necessitate the	sequential character of	parallel character of	good character of	both a & b	sequential character of
26	The two common learning strategies are probability density estimation and	sequential character of	high estimation	evaluation learning	direct loss	direct loss minimization
27	_____ can be applied in classification if training set cannot be prepared or if example with known class evaluation	C analysis	B analysis	cluster analysis	quad-analysi	cluster analysis
28	_____ is a non-parametric cluster analysis that is based on non hierarchical approach.	MinQueen k-means	MacQueen k-means	MacQueen L-means	MinQu	MacQueen k-means

29	Most neural approaches are based on combination of elementary processors called	neurons	protons	nephrons	neutron	neurons
30	A threshold and _____ is associated with every neuron.	projection function	rotation function	invariant function	transfer	transfer function
31	If the output vector is binary and contains only a single one, the position of the one classifies the input pattern into one of m	binary tree	m patterns	classificatio	pie clasific	classification
32	_____ causes the neural network to regenerate the input function at the outputs.	neon association	atom association	net association	Auto associat	Auto association
33	The restrictions of classification such as linearly separable can be overcome by	back propagation	forward propagation	K propagation	Net propaga	back propagation algorithm
34	In _____ , data are admitted at the inputs and travel in one direction towards the outputs, at which the answer may be	form feed network	feed- forward network	feed-backward	d pull back	feed- forward network
35	During back propagation learning process, if the output does not match the required output vector, then adjust the weights called	reasoning rate	grasping rate	learning rate	knowle	learning rate
36	The convergence process of the back propagation algorithm can be speeded up using .	amplifier	synthesizer	momentum	momen t	momentum
37	_____ is an example of unsupervised learning	Kohonen maps	Kenneth map	Kernaph map	all the above	Kohonen maps
38	In _____, the class of self teaching networks is not exposed to training set with known information about classes,	self learning	unsupervised learning	inference learning	group learnin	unsupervised learning
39	_____ are used mostly in optimization problems, that find the maximum similarity between a pattern and one of	Hope field	Hoop	Hopfield	both b & c	Hopfield
40	_____ description of an object is a characteristic of syntactic pattern recognition	Quantitative	Qualitative	register	suspen	Qualitative
41	The elementary properties of the syntactically described objects are called	primitives	regenerative	black elements	pixels	primitives
42	After each primitive has been assigned a symbol, relations between primitives in the objects are described, and a	relational	rational structure	relational structure	discrete	relational structure
43	The set of all words in the alphabet that can be used to describe objects from one class is named the	D language	OOPS	description language	both b & c	description language
44	A _____ represents a set of rules that must be followed when words of the specific language are constructed	rule	grammar	procedures	syntax	grammar

45	Each relational structure with multiple relations can be transformed to a relational structure with at most binary	planar graph	rotational graph	coplanar graph	distanc	planar graph
46	The inter primitive relations can be described syntactically to form structures called that represent the	character	sentence	letter	words	words
47	Primitive classes can be understood from the alphabet of symbols called	classes symbols	auto symbols	terminal symbols	both b & c	terminal symbols
48	The set of patterns from a particular class corresponds to a set of words called the	procedural language	D language	symbol language	formal languag	formal language
49	Type – 1 grammars are otherwise called _____	Context-Sensitive	Context-Switching	both a & b	none	Context-Sensitive grammars
50	Type – 2 grammars are otherwise called _____	Context-Switching	Context-Sensitive	Context-Free	all the above	Context-Free grammars
51	Type – 3 grammars are otherwise called _____	Context-Free	Context-Sensitive	Regular grammars	Context-	Regular grammars
52	Grammars can be classified into non-deterministic, stochastic and	automated	deterministic	neural	fuzzy	fuzzy
53	_____ is often used if an exhaustive search cannot be completed because the search effort exceed any reasonable	Tree queering	Tree turning	Tree pruning	Tree queuing	Tree pruning
54	To model a language of any class of patterns, the grammar rules should be extracted from a training set of examples words and	grammar building	grammar inference	grammar interface	gramm ar	grammar inference
55	The grammar inference methods can be divided into two groups based on and induction	evolution	elaborating	enumeration	evaluati on	enumeration
56	In graphical representation of the image, the model must match a sub-graph in the image graph exactly called graph	isomorphism	cluster	both a & b	clique	isomorphism
57	The double sub-graph isomorphism problem can be translated into a sub-graph isomorphism problem using a	clique	cliff	quad	quack	clique
58	The search for two sub-graphs is transformed to a clique search using the	polynomial graph	search graph	assignment graph	tree graph	assignment graph
59	The similarity of two strings can be based on the , which is defined as the smallest number of	s distance	Chord distance	Levenshtein distance	bounda	Levenshtein distance
60	The _____ area of each pixel may be taken into consideration to get the real size of a region.	skeleton	real	shape class	curvatu	real