Digital Image Processing

Ming Jiang

May 19, 2010

Part I
Introduction and Course Overview

Outline

Contents

1 Digital image processing: What, Why and How

Vision

- Image is better than any other information form for our human being to perceive. Vision allows humans to perceive and understand the world surrounding us.
- Human are primarily visual creatures. Not all animals depend on their eyes, as we do, for 99% or 90% of the information received about the world [Russ, 1995, Zhao and Zhong, 1982].

Computer Vision

- Computer vision aims to duplicate the effect of human vision by electronically perceiving and understanding an image.
- Books other than this one would dwell at length on this sentence and the meaning of the word *duplicate*
  - whether computer vision is *simulating* or *mimicking* human systems is a philosophical territory,
  - and one very fertile territory, too.

Figure 1: A frame from a video of a typical farmyard scene: the cow is one of a number walking naturally from right to left.

3D vs 2D

- Giving computers the ability to see is not an easy task — we live in a three-dimensional (3D) world.
- When computers try to analyze objects in 3D space, the visual sensors available (e.g., TV cameras) usually give two-dimensional (2D) images.
- This projection from 3D to a lower number of dimensions incurs an enormous loss of information.
- Sometimes, equipment will deliver images that are 3D but this may be of questionable value:
  - analyzing such datasets is clearly more complicated than 2D;
  - sometimes the 'three-dimensionality' is less than intuitive to us;
  - terahertz scans are an example of this.

Video Analysis

- Dynamic scenes such as those to which we are accustomed, with moving objects or a moving camera, are increasingly common and represent another way of making computer vision more complicated.

Video Analysis: easy for human

- There are many reasons why we might wish to study scenes such as this, which are attractively simple to us
  - the beast is moving slowly;
  - it is clearly black and white;
  - its movement is rhythmic.
- However, automated analysis is very fraught.

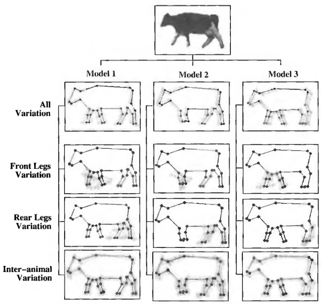
Video Analysis: difficult for computer

- The animal's boundary is often very difficult to distinguish clearly from the background;
- the motion of the legs is self occluding;
- (subtly) the concept of *cow-shaped* is not something easily encoded.

Figure 2: Three frames from a cow sequence: notice the model can cope with partial occlusion as the animal enters the scene, and the different poses exhibited.

Video Analysis: procedures

- The application from which this picture was taken made use of many of the algorithms presented in this book:
  - starting at a low level moving features were identified and grouped;
  - a *training phase* taught the system what a cow might look like in various poses (see the figure on the right), from which a model of a *moving* cow could be estimated.



Various models for a cow silhouette: a straight-line boundary approximation has been learned from training data and is able to adapt to different animals and different forms of occlusion.

Video Analysis: operations

- These models could then be fitted to new (*unseen*) video sequences.
- At this stage anomalous behavior such as lameness could be detected by the model failing to fit properly, or well.
- Thus we see a sequence of operations
  - image capture,
  - early processing,
  - segmentation,
  - model fitting,
  - motion prediction,
  - qualitative and/or quantitative conclusion,
- that is characteristic of image understanding and computer vision problems.

Video Analysis: models and cow detection

Video Analysis: disscusions

- Each of these phases (which may not occur sequentially!) may be addressed by a number of algorithms which we shall cover in due course.
- The application was serious; there is a growing need in modern agriculture for automatic monitoring of animal health, for example to spot lameness.
- *A limping cow is trivial for a human to identify, but it is very challenging to do this automatically.*

Video Analysis: disscusions

- This example is relatively simple to explain, but serves to illustrate that many computer vision techniques use the results and methods of
  - mathematics,
  - pattern recognition,
  - artificial intelligence (AI),
  - psycho-physiology,
  - computer science,
  - electronics,
  - and other scientific disciplines.

2 What Are the Difficulties

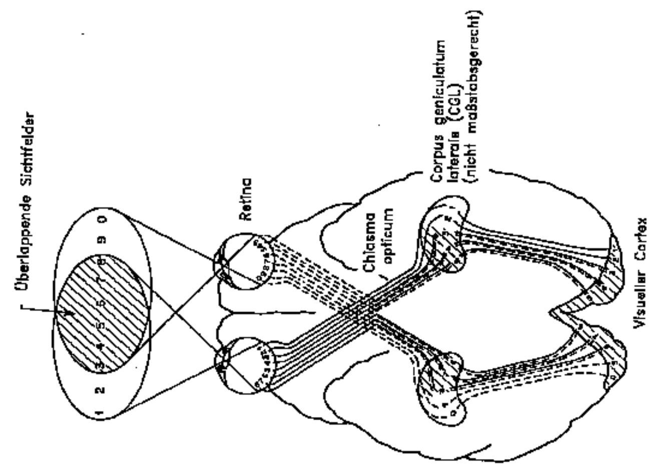
Difficuties???

- Consider a single gray-scale (monochromatic) image, write down a few reasons why you feel automatic inspection and analysis of it may be difficult.

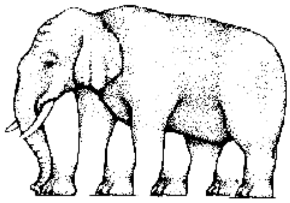
2.1 Poor understanding of human vision system

Human Vision

- How the human perceive process and store the visual information?



How many legs does this elephant have?



- From the Home of Vision Illusion:
  - http://www.123opticalillusions.com/pages/opticalillusions15.php

What is it?

- Explanation and further information can be found at
  - http://www.psychologie.tu-dresden.de/i1/kaw/diverses%20Material/www.illusionworks.com/html/perceptual_ambiguity.html

Old woman and Young woman: discussions

- Do you see an old woman or a young woman in this illustration?
- They are both present, but you will not be able to see both of them simultaneously.
- Once you perceive both figures, see if you can get them to fluctuate back and forth between the two interpretations.

Old woman and Young woman: discussions

- This type of reversible figure concerns the meaningful content of what is interpreted by your brain from the same static image.
- Your perception of each figure tends to remain stable until you attend to different regions or contours.
- Certain regions and contours tend to favor one perception, others the alternative.
- Your visual system tends to group like or related regions together.
- It does not present you with some odd mixture of the two alternatives.
- Attending to different regions or contours does tend to initiate a change of perception.

Human Vision

- We do not have a clear understanding how the human perceive, process and store the visual information.
- We do not even know how the human measures internally the image visual quality and discrimination.

Perception ≡ Description

- If this image is looked at with a steady eye, it will still change, though less often.
- Researchers have stabilized the image directly onto the retina to eliminate any effects that may arise from eye movements.
- Even under these conditions, a perceptual reversal may occur.

- This indicates that higher cortical processing occurs that strives to make meaning out of a stable image presented to the retina.
- This illustrates once more that vision is an active process that attempts to make sense of incoming information.
- As the late David Marr said, "*Perception is the construction of a description.*"

History of this illustration

- For many years the creator of this famous figure was thought to be British cartoonist W. E. Hill, who published it in 1915. Hill almost certainly adapted the figure from an original concept that was popular throughout the world on trading and puzzle cards.
- This anonymous dated German postcard (shown at the top of the page) from 1888 depicts the image in its earliest known form.

The 1890 example on the left shows quite clearly its association as "My Wife and Mother-in-Law." Both of these examples predate the Punch cartoon that was previously thought to serve as the figure's inspiration.

History of this illustration

- The figure was later altered and adapted by others, including the two psychologists, R. W. Leeper and E. G. Boring who described the figure and made it famous within psychological circles in 1930. It has often been referred to as the "Boring figure."
- Versions of the figure proved to be popular and the image was frequently reprinted; however, perceptual biases started to occur in the image, unbeknownst to the plagiarizing artists and psychologists who were reprinting the images.
- Variations have appeared in the literature that unintentionally are biased to favor one interpretation or another, which defeats its original purpose as a truly ambiguous figure.

History of this illustration

- In the three versions shown above, can you tell which one is biased toward the young girl, the old woman?

History of this illustration

- In 1961, J, Botwinick redesigned this figure once again, and entitled it, "Husband and Father-in-Law."

2.2 Internal representation is not directly understandable

Images as 2D functions

- Images are usually represented as a two dimensional function.
- Digitized images are usually represented by two dimensional array.
- However, those representations are not suitable for machine understanding, while the computer is able to process those representations.
- General knowledge, domain-specific knowledge, and information extracted from the image will be essential in attempting to **understanding** those arrays of numbers.

Experiment with images as 2D functions

- Read and display a image file as a two dimensional function.
- The example matlab script file is here matlab display example.

Images as 2D functions: discussions

- Both presentations contain exactly the same information.
- But for a human observer it is very difficult to find a correspondence between both.
- The point is that a lot of a priori knowledge is used by humans to interpret the images;
- the machine only begins with an array of numbers and so will be attempting to us more likely the first display than the second display.

2.3 Why is computer vision difficult?

Why is computer vision difficult?

- This philosophical question provides some insight into the rather complex landscape of computer vision.
- It can be answered in many ways: we offer six.
- Here, we mention the reasons only briefly — most of them will be discussed in more detail later in the book.

I. Loss of information

- Loss of information in projections from 3D to 2D is a phenomenon which occurs in typical image capture devices such as a camera or an eye.
- Their geometric properties have been approximated by a pinhole model for centuries (a box with a small hole in it, called in Latin a `camera obscura` [dark room]).

Pinhole camera

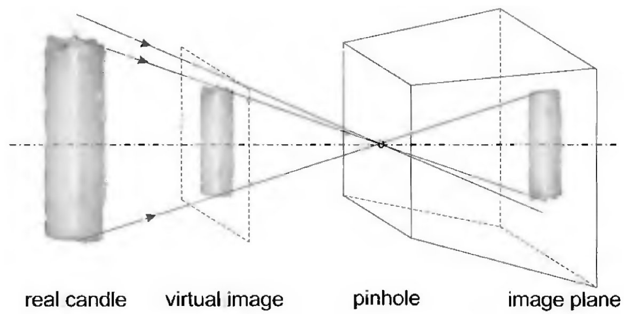


Figure 3: The pinhole model of imaging geometry does not distinguish size of objects.

- This physical model corresponds to a mathematical model of perspective projection.
- The projective transformation maps points along rays but does not preserve angles and collinearity.

II. Interpretation of images

- Interpretation of image(s) constitutes the principal tool of computer vision to approach problems which humans solve unwittingly.
- When a human tries to understand an image then previous knowledge and experience is brought to the current observation.
- Human ability to reason allows representation of long-gathered knowledge, and its use to solve new problems.
- Artificial intelligence has invested several decades in attempts to endow computers with the capability to understand observations;
- while progress has been tremendous, the practical ability of a machine to understand observations remains very limited.
- Attempting to solve related multidisciplinary scientific problems under the name cognitive systems is seen as a key to developing intelligent machines.

Interpretation of images

- From the mathematical logic and/or linguistics point of view, interpretation of images can be seen as a mapping interpretation:
$$\text{image data} \rightarrow \text{ model} \tag{1}$$
- The (logical) model means some specific world in which the observed objects make sense.
- Examples
  - nuclei of cells in a biological sample,
  - rivers in a satellite image,
  - or parts in an industrial process being checked for quality.
- There may be several interpretations of the same image(s).

Semantics of images

- Introducing interpretation to computer vision allows us to use concepts from mathematical logic, linguistics as syntax (rules describing correctly formed expression), and semantics (study of meaning).
- Considering observations (images) as an instance of formal expressions, semantics studies relations between expressions and their meanings.
- The interpretation of image(s) in computer vision can be understood as an instance of semantics.
- Practically, if the image understanding algorithms know into which particular domain (model in logical terminology) the observed world is constrained, then automatic analysis can be used for complicated problems.

III. Noise

- Noise is inherently present in each measurement in the real world.
- Its existence calls for mathematical tools which are able to cope with uncertainty; an example is probability theory.
- Of course, more complex tools make the image analysis much more complicated compared to standard (deterministic) methods.

IV. Too much data

- Images and video sequences are huge.
- An A4 sheet of paper scanned monochromatically at 300 dots per inch (dpi) at 8 bits per pixel corresponds to 8.5 MB.
- Non-interlaced RGB 24 bit color video 512×768 pixels, 25 frames per second, makes a data stream of 225 Mb per second.
- If the processing we devise is not very simple, then it is hard to achieve real-time performance; i.e., to process 25 or 30 images per second.

V. Complexity in mage formation

- Brightness measured in the image is given by complicated image formation physics.
- The `radiance` (`brightness`, `image intensity`) depends on the `irradiance` (light source type, intensity and position), the observer's position, the surface local geometry, and the surface reflectance properties.
- The inverse tasks are ill-posed — for example, to reconstruct local surface orientation from intensity variations.

VI. Local window vs. for global view

- Commonly, image analysis algorithms analyze a particular storage bin in an operational memory (e.g., a pixel in the image) and its local neighborhood;
- the computer sees the image through a keyhole.
- Seeing the world through a keyhole makes it very difficult to understand more global context.
- It is often very difficult to interpret an image if it is seen only locally or if only a few local keyholes are available.

Local parts of an image

Figure 4: Illustration of the world seen through several keyholes providing only a very local context.

The global view of the image

3 Image representation and image analysis tasks

Two Approaches

- There are philosophically two approaches: `bionics` and engineering (that is project attempt coordinated), approaches.
- The bionics approach has not been so successful, since we do have a through understanding about the biological vision system.

Figure 5: How context is taken into account is an important facet of image analysis.

Image Understanding

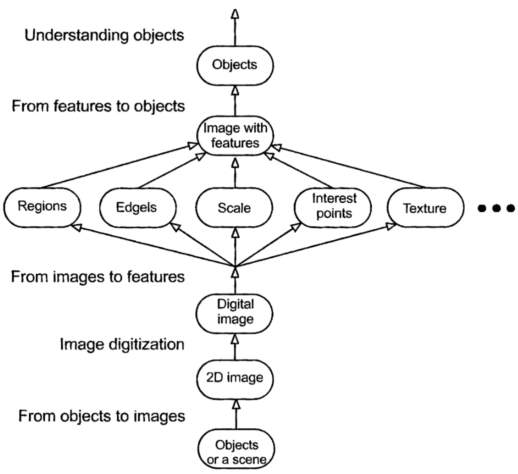
- `Image understanding` by a machine can be seen as an attempt to find a relation between input image(s) and previously established models of the observed world.
- Transition from the input image(s) to the model reduces the information contained in the image to relevant information for the application 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

- `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.

Image Representation & Hierarchy of Computer Vision

- This suggests a bottom up way of information processing, from signals with almost no abstraction, to the highly abstract description needed for image understanding.
- The flow of information does not need to be unidirectional.
- Feedback loops are often introduced to allow the modification of algorithms according to intermediate results.



Two Levels

- 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;
  - high-level image understanding.

Low-level processing

- Low-level processing methods usually use very little knowledge about the content of images.
- Low-level methods often include image compression, pre-processing methods for noise filtering, edge extraction, and image sharpening.
- Ee shall discuss in this course.
- Low-level image processing uses data which resemble the input image.
- Very often, such a data set will be part of a video stream with an associated frame rate.
- E.g., an input image captured by a TV camera is 2D in nature, being described by an image function $f(x, y, t)$ whose value, at simplest, is usually brightness depending on parameters x, y and t.

High-level processing I

- High-level processing is based on knowledge, goals, and plans of how to achieve those goals.
- 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 processing II

- 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.
- When differences emerge, partial matches (or sub-goals) are sought that overcome the mismatches.
- The omputer 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.
- A feedback loop is introduced in which high-level partial results create tasks for low-level image processing.
- The iterative image understanding process should eventually **converge** to the global goal.

Low- vs. high-level representations

- Both representations contain exactly the same information.
- But **for a human observer** it is not difficult to find a correspondence between them, and without the second, it is unlikely that one would recognize the face of a child.
- The point is that a lot of a priori knowledge is used by humans to interpret the images.
- A machine only begins with an array of numbers and so will be attempting to make identifications and draw conclusions from data that to us are more uncomprehensible.
- Increasingly, data capture equipment is providing very large data, sets that do not lend themselves to straightforward interpretation by humans.
- We have already mentioned terahertz imaging as an example.
- General knowledge, domain-specific knowledge, and information extracted from the image will be essential in attempting to "understand" these arrays of numbers.

Low-level processing

The following sequence of processing steps is commonly recognized:

- Image Acquisition: An image is captured by a sensor (such as a TV camera) and digitized. Image may come in many **formats** and ways.
- Preprocessing: Image reconstruction or restoration, denoising and enhancement. E.g., computer tomography.
- Image coding and compression: this is important for transferring images.
- Image segmentation: computer tries to separate objects from the image background.
- Object description and classification in a totally segmented image is also understood as part of low-level image processing.

Image Segmentation

- Image segmentation is 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.
- Example: optical character recognition, OCR.
- Een 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.

Low-level Image Processing

- Low-level computer vision techniques overlap almost completely with digital image processing, which has been practiced for decades.
- 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 vs High

- 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.
- 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. —
- E.g., object size, shape, and mutual relations between objects in the image.
- High-level data are usually expressed in symbolic form.

Image Processing

- Most current low-level image processing methods were proposed in the 1970s or earlier.
- Recent research is trying to find more efficient and more general algorithms, implementations.
- The requirement for better and faster algorithms is fuelled by technology delivering larger images (better spatial resolution), and color.

- A complicated and so far unsolved problem is how to order low-level steps to solve a specific task, and the aim of automating this problem has not yet been achieved.
- It is usually still a human operator who finds a sequence of relevant operations.
- Domain- specific knowledge and uncertainty cause much to depend on this operator's intuition and previous experience.

High-level Vision

- 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.
- Unsurprisingly this task is very complicated and computationally intensive.
- David Marr's book [Marr, 1982] influenced computer vision considerably throughout the 1980s.
- It described a new methodology and computational theory inspired by biological vision systems.
- Developments in the 1990s moved away from dependence on this paradigm, but interest in properly understanding and then modeling human visual systems.
- It remains the case that the only known solution to the "vision problem" is our own brain!

3D Vision Problems

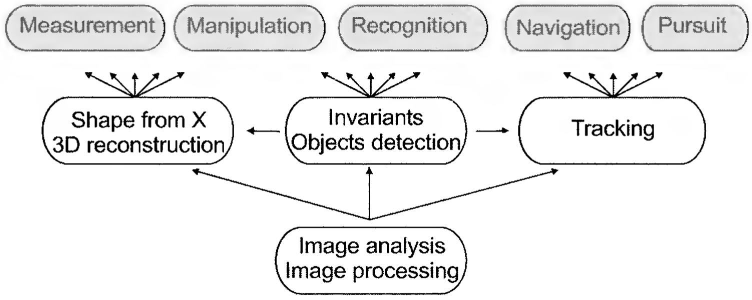


Figure 6: Several 3D vision tasks and algorithmic components expressed on different abstraction levels. We adopt the user's view, i.e., what tasks performed routinely by humans would be good to accomplish by machines.

3D Vision Problems

- What is the relation of these 3D vision tasks to low-level (image processing) and high- level (image analysis) algorithmic methods?
- There is no widely accepted view in the academic community.
- Links between (algorithmic) components and representation levels are tailored to the specific application solved, e.g., navigation of an autonomous vehicle.
- These applications have to employ **specific knowledge** about the problem solved to be competitive with tasks which humans solve.
- More general theories are expected to emerge.
- Many researchers in different fields work on related problems.
- There is a belief that research in 'cognitive systems' could be the key.

4 Course Overview

Course Overview

- Digital image processing, image analysis, image understanding are related branches of computer vision.
- This course is about digital image processing.
- The following topics are to be covered in this course.

Course Syllabus

- Introduction and Course Overview
- Image representations and properties
  - Images as a stochastic processes or linear systems, etc.
  - Metric and topological properties of digital images
  - Histograms
  - Noise in images
- Data Structures for Image Analysis

- Image Pre-processing
  - Various pre-processing operators
- Image Segmentation
  - Thresholding, edge-based, region growing, segmentation method.
- Scale Space Theory
  - Image processing and partial differential equations.

The textbook and web resource

- Milan Sonka, V. Hlavac, R. Boyle: Image Processing, Analysis and Machine Vision, 3rd edition. Thomson Learning, 2008.
- Image Processing, Analysis, and Machine Vision: A MATLAB Companion, http://visionbook.felk.cvut.cz/

References

- Kenneth R. Castleman: Digital Image Processing. Prentice-Hall International, Inc. 1996. Or Tsinghua University Press, 1998.
- Rongchun Zhao: Introduction to Digital Image Processing (in Chinese). Northwestern Polytechnical University Press, 2000.

Summary

- Human vision is natural and seems easy; computer mimicry of this is difficult.
- We might hope to examine pictures, or sequences of pictures, for quantitative and qualitative analysis.
- Many standard and advanced AI techniques are relevant.
- "High" and "low" levels of computer vision can be identified.
- Processing moves from digital manipulation, through pre-processing, segmentation, and recognition to understanding — but these processes may be simultaneous and co-operative.
- An understanding of the notions of heuristics, a priori knowledge, syntax, and semantics is necessary.
- The vision literature is large and growing; books may be specialized, elementary, or advanced.
- A knowledge of the research literature is necessary to stay up to date with the topic.
- Developments in electronic publishing and the Internet are making access to vision simpler.

Part II
The Digitized Image and its Properties

Outline

Contents

5 Basic concepts

Signals

- Fundamental concepts and mathematical tools are introduced in this chapter which will be used throughout the course.
- Mathematical models are often used to describe images and other signals.
- A signal is a function depending on some variable(s) with physical meaning. Signals can be
  - one-dimensional (e.g., audio signal dependent on time);
  - two-dimensional (e.g., images dependent on two co-ordinates in a plane);
  - three-dimensional (e.g., describing an object in space or video signal);
  - or higher-dimensional.

Images as 2D functions: discussions

- A scalar function may be sufficient to describe a monochromatic image.
- Vector functions are to represent, for example, color images consisting of three component colors.

- Functions we shall work with may be categorized as continuous, discrete and digital.
- A continuous function has continuous domain and range;
- If the domain set is discrete, then we get a `discrete function`;
- if the range set is also discrete, then we have a `digital function`.

5.1 Image functions

Images as 2D functions: discussions

- An image can be modeled by a continuous function of two or three variables;
- Its arguments are co-ordinates x and y in a plane;
- If images change in time a third variable t might be added.
- The image function values correspond to the brightness at image points.

Intepretation of image values

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

Homework

Discuss the various factors that influence the brightness of a pixel in an image.

2D images

- The image on the human eye retina or on a TV camera sensor is intrinsically 2D.
- We shall call such a 2D image bearing information about brightness points an `intensity image`.
- The real world which surrounds us is intrinsically 3D.
- The 2D intensity image is the result of a perspective projection of the 3D scene.

Ill-posed problem I

- When 3D objects are mapped into the camera plane by perspective projection a lot of information disappears as such a transformation 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.
- The aim is to recover a full 3D representation such as may be used in computer graphics.

Ill-posed problem II

- The second problem is how to understand image brightness.
- The only information available in an intensity image is the brightness of pixels.
- They are dependent on a number of independent factors such as
  - object surface reflectance properties (given by the surface material, micro-structure and marking),
  - illumination properties,
  - object surface orientation with respect to a viewer and light source.
- This is a non-trivial and again ill-posed problem.

2D or 3D

- Some applications work with 2D images directly; for example,
  - an image of the flat specimen viewed by a microscope with transparent illumination;
  - a character drawn on a sheet of paper;
  - the image of a fingerprint, etc.
- Many basic and useful methods used in digital image analysis do not depend on whether the object was originally 2D or 3D (e.g, FFT).

5.2 Image formation

Image formation: I

- The image formation process is described in [Horn, 1986, Wang and Wu, 1991].
- Related disciplines are photometry which is concerned with brightness measurement, and colorimetry which studies light reflectance or emission depending on wavelength.

Image formation: II

- A light source energy distribution $C(x, y, t, \lambda)$ depends in general on image co-ordinates (x, y), time t, and wavelength λ.
- For the human eye and most technical image sensors (e.g., TV cameras), the "brightness" f depends on the light source energy distribution C and the spectral sensitivity of the sensor, $S(\lambda)$ (dependent on the wavelength)

$$f(x, y, t) = \int_0^\infty C(x, y, t, \lambda) S(\lambda) \, d\lambda \tag{2}$$

- An intensity image $f(x, y, t)$ provides the brightness distribution.

Multi-spetral images

- In a color or multi-spectral image, the image is represented by a real vector function f

$$f(x, y, t) = (f_1(x, y, t), f_2(x, y, t), \cdots, f_n(x, y, t)) \tag{3}$$

  where, for example, there may be red, green and blue, three components.

Monochromatic static image

- Image processing often deals with static images, in which time t is constant.
- A monochromatic static image is represented by a continuous image function $f(x, y)$ whose arguments are two co-ordinates in the plane.
- Most methods introduced in this course is primarily for intensity static image.
- It is often the case that the extension of the techniques to the multi-spectral case is obvious.

Monochromatic static image

- Computerized image processing uses digital image functions which are usually represented by matrices.
- Co-ordinates are integer numbers.
- The domain of the image function is a region R in the plane

$$R = \{(x, y) : 1 \leq x \leq x_m, 1 \leq y \leq y_n\} \tag{4}$$

  where x_m and y_n represent maximal image co-ordinates.

Limited domain

- The image function has a limited domain — infinite summation or integration limits can be used, as it is assumed that the image function is zero outside the domain.
- The customary orientation of co-ordinates in an image is in the normal Cartesian fashion (horizontal x axis, vertical y axis).
- The (row, column) orientation used in matrices is also quite often used in digital image processing.

Limited range

- The range of image function values is also limited; by convention, in intensity images the lowest value corresponds to black and the highest to white.
- Brightness values bounded by these limits are `gray levels`.
- The gray level range is $0, 1, \cdots, 255$, represented by 8 bits, the data type used is **unsigned char**. In some applications, 14 bits or more is used, e.g, for medical images.
- The usual computer display supports 8 bit gray level.

Homework

- How to display a 16 bit gray level image? Generate an image of 16 bit and try to display it with your computer.
- If a discrete image is of continuous range, the image matrix is of type **float** or **double**. How to display it? Generate an image of **float** or **double** type and try to display it with your computer.

Image quality

- The quality of a digital image grows in proportion to the spatial, spectral, radiometric, and time resolution.
- The `spatial resolution` is given by the proximity of image samples in the image plane.
- The `spectral resolution` is given by the bandwidth of the light frequencies captured by the sensor.
- The `radiometric resolution` (or `contrast resolution`, or `density resolution`) corresponds to the number of distinguishable gray levels.
- The `time resolution` is given by the interval between time samples at which images are captured.

5.3 Image as a stochastic process

Image as a stochastic process

- Images $f(x, y)$ can be treated as deterministic functions or as realizations of stochastic processes.
- Images are statistical in nature due to random changes and noise.
- It is sometimes of advantages to treat image functions as realizations of a stochastic process.

Image as a stochastic process: a typical model

- Mathematical tools used in image description have roots in linear system theory, integral transformations, discrete mathematics and the theory of stochastic processes, [Horn, 1986, Wang and Wu, 1991].
- A typical image formation model is described by a `linear spatial invariant system`,

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(a, b) h(x - a, y - b) \, da db + n(x, y) \quad (5)$$

$$= h * g(x, y) + n(x, y) \quad (6)$$

where h is called the `point spread function` (PSF)and n is an additive noise.

6 Image digitization

Image sampling and digitization

- 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 $f(x, y)$.

Two questions for sampling

- Two questions should be answered in connection with image function sampling:
  - First, the sampling period should be determined – the distance between two neighboring sampling points in the image;
  - Second, the geometric arrangement of sampling points (sampling grid) should be set.

6.1 Sampling

Sampling intervals

- A continuous image function $f(x, y)$ can be sampled using a discrete grid of sampling points in the plane.
- The image is sampled at points $x = j\Delta x$, $y = k\Delta y$
- Two neighboring sampling points are separated by distances Δx along the x axis and Δy along the y axis.
- Distances Δx and Δy are called the sampling interval.
- The matrix of samples constitutes the discrete image.

Sampled Image

- The ideal sampling $s(x, y)$ in the regular grid can be represented using a collection of Dirac distributions

$$s(x, y) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(x - j\Delta x, y - k\Delta y) \tag{7}$$

- The sampled image is the product of the continuous image $f(x, y)$ and the sampling function $s(x, y)$

$$f_s(x, y) = s(x, y)f(x, y) \tag{8}$$

Sampled Image

- The collection of Dirac distributions in equation (7) can be regarded as periodic with period Δx, Δy.
- It can be expanded into a Fourier series (assuming for a moment that the sampling grid covers the whole plane (infinite limits))

$$s = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_{mn} \mathbf{e}^{2\pi i \left(\frac{mx}{\Delta x} + \frac{ny}{\Delta y} \right)} \tag{9}$$

Fourier expansion

- The coefficients of the Fourier expansion can be calculated as

$$a_{mn} = \frac{1}{\Delta x \Delta y} \int_{-\frac{\Delta x}{2}}^{\frac{\Delta x}{2}} \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(x - j\Delta x, y - k\Delta y) \mathbf{e}^{2\pi i \left(\frac{mx}{\Delta x} + \frac{ny}{\Delta y} \right)} \, dxdy \tag{10}$$

- Noting that only the term for $j = 0$ and $k = 0$ in the sum is nonzero in the range of integration (for other j and k, the center of the Delta function is outside the integral interval), the coefficients are

$$a_{mn} = \frac{1}{\Delta x \Delta y} \tag{11}$$

Fourier expansion in the frequency domain

- Then, (8) can be rewritten as

$$f_s(x,y) = f(x,y)\frac{1}{\Delta x \Delta y}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \mathbf{e}^{2\pi i\left(\frac{mx}{\Delta x}+\frac{ny}{\Delta y}\right)} \tag{12}$$

- In the frequency domain then

$$F_s(u,v) = \frac{1}{\Delta x \Delta y}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} F(u-\frac{m}{\Delta x}, v-\frac{n}{\Delta y}) \tag{13}$$

  where F and F_s are the Fourier transform of f and f_s respectively.

Fourier transform of the sampled image

- Recall the Fourier transform is

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\mathbf{e}^{-2\pi i(ux+vy)}\, dxdy \tag{14}$$

- Thus the Fourier transform of the sampled image is the sum of periodically repeated Fourier transforms $F(u,v)$ of the origin image.
- 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.

Aliasing

- There is no aliasing if the image function $f(x,y)$ has a band limited spectrum, its Fourier transform $F(u,v) = 0$ outside a certain interval of frequencies $|u| > U$ and $|v| > V$.

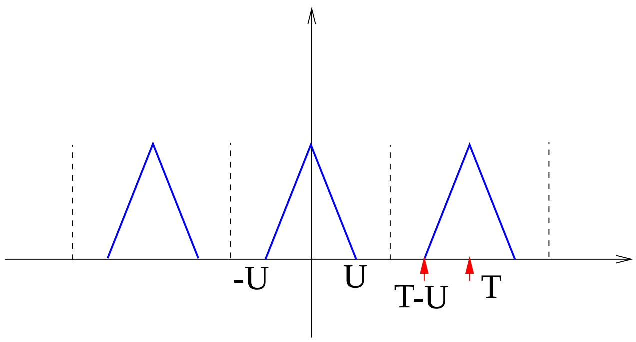


Figure 7: Where $T = \frac{1}{\Delta x}$.

Shannon sampling theorem

- From general sampling theory [Oppenheim et al., 1997], aliasing can be prevented if the sampling interval is chosen according to

$$\Delta x \le \frac{1}{2U}, \qquad \Delta x \le \frac{1}{2V}. \tag{15}$$

- This is the Shannon sampling theorem that has a simple physical interpretation in image analysis
  - the sampling interval should be chosen such that it is less than or equal to half of the smallest interesting detail in the image.

Sampling function in practice

- The sampling function is not the Dirac distribution in real digitizers – narrow impulses with limited amplitude are used instead.
- Assume a rectangular sampling grid which consists of $M \times N$ such equal and non-overlapping impulses $h_s(x,y)$ with sampling period Δx and Δy.
- Ideally, $h_s(x,y) = \delta(x,y)$.
- The function $h_s(x,y)$ simulates realistically the real image sensors.
- Outside the sensitive area of the sensor, the sampling element $h_s(x,y) = 0$.

Sampled image in practice

- The sampled image is then given by the following convolution

$$f_s(x,y) = f(x,y) \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h_s(x - j\Delta x, y - k\Delta y) \tag{16}$$

- The sampled image f_s is distorted by the convolution of the original image f and the limited impulse h_s.
- The distortion of the frequency spectrum of the function F_s can be expressed as follows

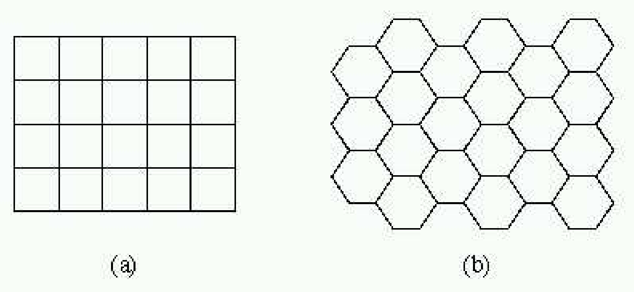
$$F_s(u,v) = \frac{1}{\Delta x \Delta y} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} F(u - \frac{m}{\Delta x}, v - \frac{n}{\Delta y}) H_s(\frac{m}{\Delta x}, \frac{n}{\Delta y}). \tag{17}$$

Homework

- Prove Eq. (17) from Eq. (16).

Homework

- There are other sampling schemes.
- These sampling points are ordered in the plane and their geometric relation is called the grid.
- Grids used in practice are mainly square or hexagonal



(a) Square grid, (b) hexagonal grid.

Pixels

- One infinitely small sampling point in the grid corresponds to one picture element (`pixel`) in the digital image.
- The set of pixels together covers the entire image.
- Pixels captured by a real digitization device have finite sizes.
- A pixel is a unit which is not further divisible.
- Sometimes pixels are also called points.

Real digitizers

- In real image digitizers, a sampling interval about ten times smaller than that indicated by the Shannon sampling theorem (15) is used.
- This is because algorithms which reconstruct the continuous image on a display from the digitized image function use only a step function.
- E.g., a line in the image is created from pixels represented by individual squares.

- The situation is more complicated.
- `Average sampling` happens in practice.

Sampling Examples

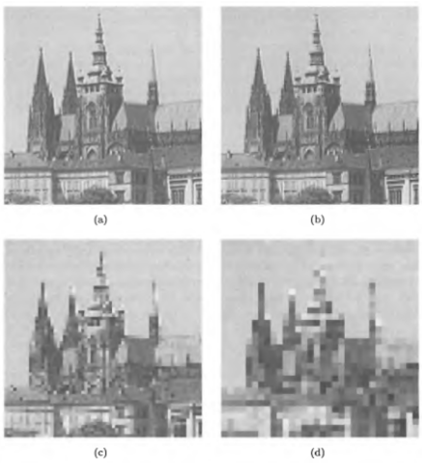


Figure 8: Digitizing, (a) 256×256. (b) 128×128. (c) 64×64. (d) 32×32. Images have been enlarged to the same size to illustrate the loss of detail.

6.2 Quantization

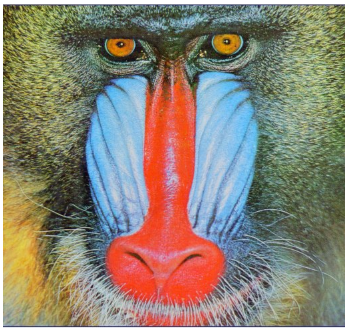
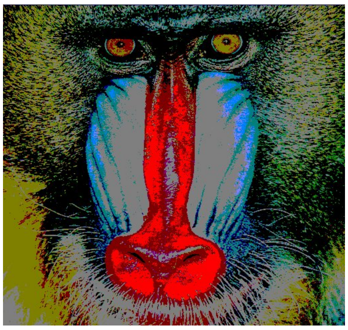
Quantization

- The magnitude of a sampled image 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 for human perception of fine shading details in the image.

False Contours in Images

- The occurrence of false contours is the main problem in image which have been quantized with insufficient brightness levels.
- This effect arises when the number of brightness levels is lower than that which humans can easily distinguish.


False Contours in Color Images

Discussions on Quantization

- This number is dependent on many factors: e.g., the average local brightness.
- Displays which avoids 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 brightnesses in the image is enlarged.
- These gray-scale transformation techniques are considered in later sections.
- Most digital image processing devices use quantization into k equal intervals.
- If b bits are used ... the number of brightness levels is $k = 2^b$.
- Eight bits per pixel are commonly used, specialized measuring devices use 12 and more bits per pixel.

Quantization Experiment with matlab

Do you observe false contours when the quantization levels is decreasing?

7 Digital image properties

Digital Properties

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

7.1 Metric and topological properties of digital images

Continuous Property May not Hold

- A digital image consists of picture elements of finite size.
- Usually pixels are arranged in a rectangular grid.
- A digital image is represented by a two-dimensional matrix whose elements are integer 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.

7.1.1 Metric properties of digital images

Distance

- Distance is an important example.
- The distance between two pixels in a digital image is a significant quantitative measure.
- The distance between points with co-ordinates (i, j) and (h, k) may be defined in several different ways.

Euclidean Distance

- The Euclidean distance D_E is defined by

$$D_E[(i, j), h, k] = \sqrt{(i - h)^2 + (j - k)^2} \tag{18}$$

- The advantage of the Euclidean distance is the fact that it is intuitively obvious.
- The disadvantages are costly calculation due to the square root, and its not-integer value.

TAXI Distance

- The distance between two points can also expressed as the minimum number of elementary steps in the digital grid which are needed to move from the starting point to the end point.
- If only horizontal and vertical moves are allowed, the distance D_4 or city block distance is obtained:

$$D_4[(i, j), h, k] = |i - h| + |j - k| \tag{19}$$

- This is the analogy with the distance between two locations in a city with a rectangular grid of streets and closed blocks of buildings.

Chess-board Distance

- If moves in diagonal directions are allowed in addition, the distance D_8 or the chess-board distance is obtained:

$$D_8[(i, j), h, k] = \max\{|i - h|, |j - k|\} \tag{20}$$

7.1.2 Topological properties of digital images

Pixel Neighborhoods

- Pixel adjacency is another important concept in digital images.
- Any two pixels are called 4-neighbors if they have distance $D_4 = 1$ from each other.
- 8-neighbors are two pixels with $D_8 = 1$.
- 4-neighbors and 8-neighbors are illustrated in Figure 111.

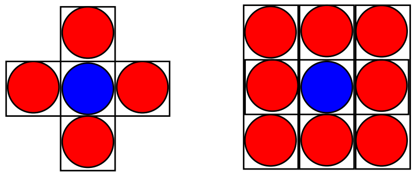


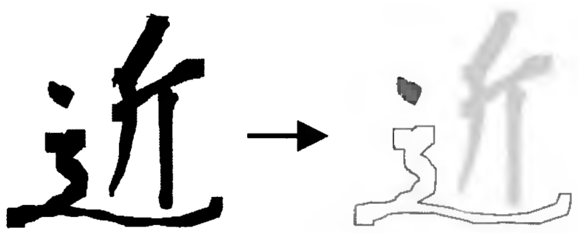
Figure 9: Pixel neighborhoods.

Regions and Paths

- It will become necessary to consider important sets consisting of several adjacent pixels — regions.
- `Region` is a contiguous (touching, neighboring, near to) set.
- A `path` from pixel P to pixel Q as a sequence of points $A_1, A_2, \cdots, A_n$, where $A_1 = P$ and $A_n = Q$, and A_{i+1} is a neighbor of A_i, $i = 1, \cdots, n-1$.
- A `region` is a set of pixels in which there is a path between any pair of its pixels, all of whose pixels also belong to the set.

Contiguity

- If there is a path between two pixels in the set of pixels in the image, these pixels are called `contiguous`.
- The relation to be contiguous is reflexive, symmetric and transitive and therefore defines a decomposition of the set (in our case image) into equivalence classes (regions).
- The following image illustrates a binary image decomposed by the relation `contiguous` into three regions.



Some Terms

- Assume that R_i are disjoint regions in the image and that these regions do not touch the image boundary (to avoid special cases).
- Let R be the union of all regions R_i. Let R^C be the complement of R with respect to the image.
- The subset of R^C, which is contiguous with the image boundary, is called `background`, and the rest of the complement R^C is called `holes`.
- If there are no holes in a region we call it a `simply contiguous` region.
- A region with holes is called `multiply contiguous`.

Image Segmentation

- Note that the concept of region uses only the property to be contiguous.
- Secondary properties can be attached to regions which originate in image data interpretation.
- It is common to call some regions in the image `objects`.
- A process which determines which regions in an image correspond to objects in the world is part of `image segmentation`.

Contiguous Example

- The brightness of a pixel is a property used to find objects in some images.
- If a pixel is darker than some other predefined values (threshold), then it belongs to some object.
- All such points which are also contiguous constitute one object.
- A hole consists of points which do not belong to the object and surrounded by the object, and all other points constitute the background.

- An example is the black printed text on the white paper, in which individual letters are objects.
- White areas surrounded by the letter are holes, e.g., the area inside a letter 'O'.
- Other parts of the paper are background.

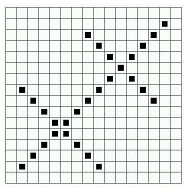
7.1.3 Contiguity paradoxes

Contiguity paradoxes

- These neighborhood and contiguity definitions on the square grid create some paradoxes.

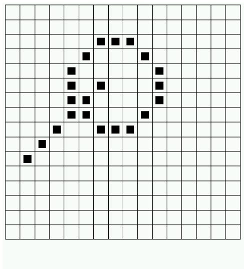
Line Contiguous Paradox

The following figure shows three digital lines with 45^o and -45^o slope.



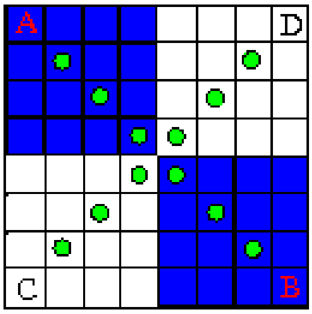
- If 4-connectivity is used, the lines are not contiguous at each of their points.
- An even worse conflict with intuitive understanding of line properties is:
  - two perpendicular lines do intersect in one case (upper right intersection) and do not intersect in another case (lower left), as they do not have any common point.

Jordon Paradox



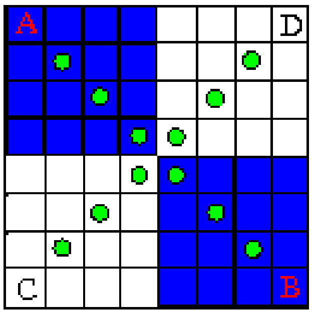
- Each closed curve divides the plane into two non-contiguous regions.
- If image are digitized in a square grid using 8-connectivity, there is a line from the inner part of a closed curve into the outer part without intersecting the curve.
- This implies that the inner and outer parts of the curve constitute one contiguous region.

Connectivity paradox



- If we assume 4-connectivity, the figure contains four separate contiguous regions A, B, C and D.
  - $A \cup B$ are disconnected, as well as $C \cup D$.
  - A topological contradiction.
  - Intuitively, $C \cup D$ should be connected if $A \cup B$ are disconnected.

Connectivity paradox



- If we assume 8-connectivity, there are two regions, $A \cup B$ and $C \cup D$.
- Both sets contain paths AB and CD entirely within themselves, but which also intersect!

Ad Hoc Solutions to Paradoxes

- One possible solution to contiguity paradox is to treat objects using 4-neighborhoods and background using 8-neighborhoods (or vice versa).
- More exact treatment of digital contiguity paradox and their solution for binary images and images with more brightness levels can be found in [Pavlidis, 1977].
- These problems are typical on square grids — a hexagonal grid (96) solves many of them.
- However, a grid of this type has also a number of disadvantages, [Pavlidis, 1977], p. 60.
- For reasons of simplicity and ease of processing, most digitizing devices use a square grid despite the stated drawbacks.
- We do not pursue further into this topic in this course, but use the simple approach, although there are some paradoxes.

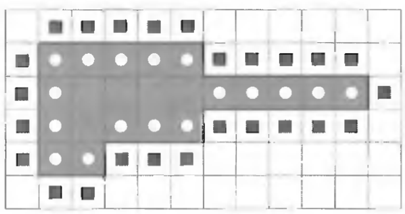
Solutions to Paradoxes

- An alternative approach to the connectivity problems is to use discrete topology based on CW complex theory in topology.
- It is called cell complex in [Kovalevski, 1989].
- This approach develops a complete strand of image encoding and segmentation.
- The idea, first proposed by Riemann in the nineteenth century, considers families of sets of different dimensions:
  - points, which are 0-dimensional, may then be assigned to sets containing higher dimensional structures (such as pixel array), which permits the removal of the paradoxes we have seen.
  - line segments, which are 1-dimensional, gives precise definition of edge and border.

7.1.4 Other topological and geometrical properties

Border

- `Border` 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.
- This definition of border is sometimes referred to as `inner border`, to distinguish it from the `outer border`,
  - it is the border of the background (i.e., the complement of) of the region.



Edge

- `Edge` is a local property of a pixel and its immediate neighborhood — it is a vector given by a magnitude and direction.
- The edge direction is perpendicular to the gradient direction which points in the direction of image function growth.

Border and Edge

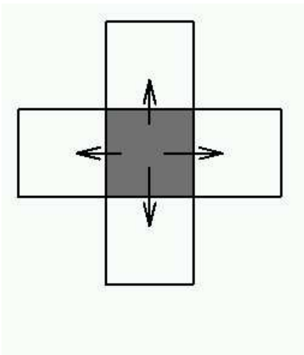
- The border is a global concept related to a region, while edge expresses local properties of an image function.
- The border and edge are related as well.
- One possibility for finding boundaries is chaining the significant edges (points with high gradient of the image function).

Crack Edge I

- The edge property is attached to one pixel and its neighborhood.
- It is of advantage to assess properties between pairs of neighboring pixels.
- The concept of the `crack edge` comes from this idea.

Crack Edge II

- Four crack edges are attached to each pixel, which are defined by its relation to its 4-neighbors.
- The direction of the crack edge is that of increasing brightness, and is a multiple of 90 degrees.
- Its magnitude is the absolute difference between the brightness of the relevant pair of pixels.



Convex Hull

- Convex hull is used to describe geometrical properties of objects.
- The convex hull is the smallest convex 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.

Lakes and Bays

- An object can be represented by a collection of its topological components.
- The sets inside the convex hull which does not belong to an object is called the deficit of convexity.
- This can be split into two subsets.
  1. `lakes` are fully surrounded by the objects.
  2. `bays` are contiguous with the border of the convex hull of the object.
- The convex hull, lakes and bays are sometimes used for object description.

7.2 Histogram

Brightness Histogram

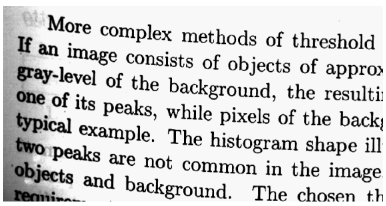
- Brightness histogram provides the frequency of the brightness value in the image.
- The brightness histogram $h_f(z)$ is a function showing,
  - for each brightness value z,
  - the number of pixels in the image f that have that brightness value z.

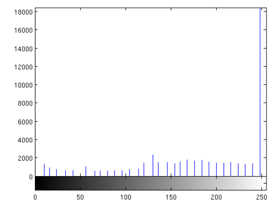
Computing Brightness Histogram

- The histogram of an image with L gray levels is represented by a one-dimensional array with L elements.
- For a digital image of brightness value ranging in $[0, L-1]$, the following algorithm produces the brightness histogram:
  1. Assign zero values to all element of the array h_f;
  2. For all pixels (x, y) of the image f, increment $h_f[f(x, y)]$ by 1.

Brightness Histogram Example

- The histogram is often displayed as a bar graph.

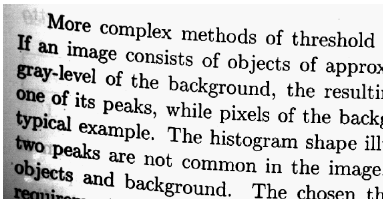

More complex methods of threshold
If an image consists of objects of approx
gray-level of the background, the resulti
one of its peaks, while pixels of the back
typical example. The histogram shape ill
two peaks are not common in the image
objects and background. The chosen th

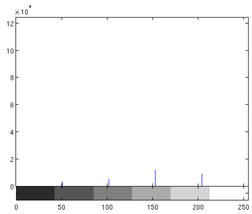


Computation Options

- Computing brightness histogram is similar to generating the histogram of a random variable from a given group of samples.
- In the above algorithm, the starting value is 0, bin-width 1, and bin number L.
- This algorithm can be modified to generate brightness histogram of arbitrary bin-width and bin number.
- For multi-spectral band images, histogram of each individual band can be generated in a similar way.

Brightness Histogram Example: non-convential bin


More complex methods of threshold
If an image consists of objects of appro
gray-level of the background, the resulti
one of its peaks, while pixels of the back
typical example. The histogram shape ill
two peaks are not common in the image
objects and background. The chosen th



Histogram: discussions I

- The histogram provides a natural bridge between images and a probabilistic description.
- Wc might want to find a first-order probability function $p_I(z; x, y)$ 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.*

Histogram: discussions II

- 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;
  - e.g., a change of the object position on a constant background does not affect the histogram.

Histogram: discussions II

- 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 algorithm would need some boundary adjustment, and carries no guarantee of removing all local minima.
- Other techniques for smoothing exist, notably Gaussian blurring.

Histogram Experiment with matlab

- Creating an image histogram using imhist.

7.3 Entropy

Entropy

- Image information content can be estimated 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`.

Entropy As a Measure of uncertainty

- 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.

Entropy: definition

- The entropy is defined formally assuming a discrete random variable X with possible outcomes (called also states) $x_1, \cdots, x_n$.
- Let $p(x_k)$ be the probability of the outcome x_k, $k = 1, \cdots, n$.
- Then the entropy is defined as

$$H(x) = \sum_{k=1}^{n} p(x_k) \log \frac{1}{p(x_k)} = -\sum_{k=1}^{n} p(x_k) \log p(x_k). \tag{21}$$

- $\log \frac{1}{p(x_k)}$ is called the surprisal of the outcome x_k.
- The entropy is the expected value of its outcome's surprisal.

Entropy: discussions

- The base of the logarithm in this formula determines the unit in which entropy is measured.
- If this base is two then the entropy is given in bits.
- The entropy is often estimated using a gray-level histogram in image analysis.
- Because entropy measures the uncertainty about the realization of a random variable, it is used to assess redundancy in an image for image compression.

7.4 Visual perception of the image

Visual Perception

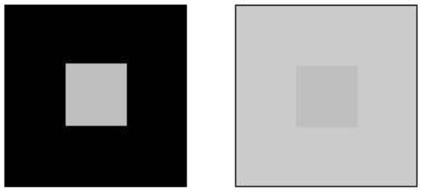
- Anyone who creates or uses algorithms or devices for digital image processing should take into account the principle of human visual perception.
- There are psycho-physical parameters such as contrast, border, shape, texture, color, etc.
- Humans will find objects in images only if they may be distinguished effortlessly from the background.
- Human perception of image provokes many illusions, the understanding of which provides valuable clues about visual mechanisms.
- The topic is covered exhaustively from the point of view of computer vision in [Frisby, 1979].

7.4.1 Contrast

Contrast

- Contrast is the local change in brightness and is defined as the ratio between average brightness of an object and the background brightness.
- The human eye is logarithmically sensitive to brightness.
- Gamma correction is used to calibrate the differences among different computer monitors.
- Apparent brightness depends very much on the brightness of the local background; this effect is called conditional contrast.

Conditional Contrast



- The figure illustrates this effect with two small squares of the same brightness on a dark and a light background.
- Human perceives the brightness of the small squares as different.

7.4.2 Acuity

Acuity

- Acuity is the ability to detect details in image.
- The human eye is less sensitive to slow and fast brightness changes but is more sensitive to intermediate changes.
- 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.

Resolution

- Resolution in optics is defined as the inverse value of a maximum viewing angle between the viewer and two proximate points which human cannot distinguish, and so fuse together.
- Human vision has the best resolution for objects which are at a distance of about 25 cm from an eye under illumination of about 500 lux.
- This illumination is provided by a 60 W from a distance 40 cm.
- Under this conditions the distance between two distinguishable points is approximately 0.16 mm.
- Another report says that the minimal distinguishable distance is 0.47 mm [Kutter, 1999].

Quiz
Given the above two minimal distinguishable distance, what is the resolution in DPI needed for a printer to produce perfect output?
DPI means "Dots Per Inch" (1 in = 2.54 cm).

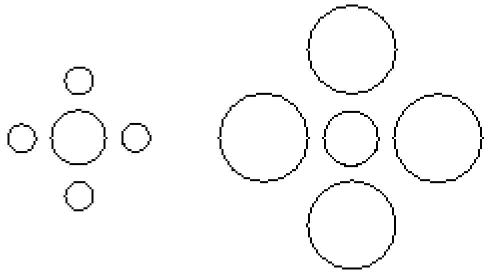
7.4.3 Visual Illusions

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.

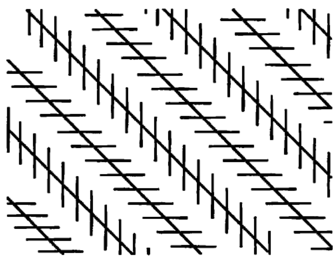
The Ebbinghaus Circles

- Object borders carry a lot of information.
- Boundaries of objects and simple patterns such as blobs or lines enable adaption effects similar to conditional contrast.
- The Ebbinghaus illusion is a well known example — two circles of the same diameter in the center of images appear to have different sizes.



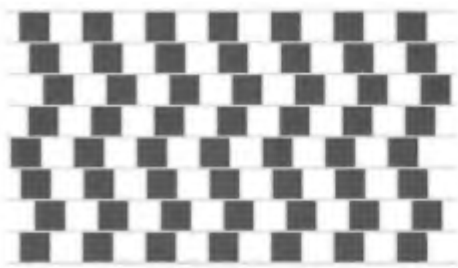
Parallel Lines

- Perception of one dominant shape can be fooled by nearby shapes.
- This figure shows parallel diagonal line segments which are not perceived as parallel.



Zigzag Lines

- This figure contains rows of black and white squares which are all parallel.
- However, the vertical zigzag squares disrupt our horizontal perception.



7.5 Image quality

Image Quality

- An image might be degraded during capture, transmission, or processing.

- 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 Quality

- Subjective methods are often used in television technology.
- The ultimate criterion is the perception of a selected group of professional and lay viewers.
- They appraise an image according to a list of criteria and give appropriate marks.

Objective Quality

- Objective quantitative methods measuring image quality are more interesting for our purposes.
- Ideally such a method also provides a good subjective test, and is easy to apply;
- we might then use it as a criterion in parameter optimization.

Image Quality: MSE, etc.

- The quality of the image $f(x, y)$ is usually estimated by comparison with a known reference image $g(x, y)$.
- A synthesized image $g(x, y)$ is often used for this purpose.
- One class of methods uses simple measures such as the mean quadratic difference (or mean squared error, MSE)
$$\mathrm{MSE}(g, f) = \frac{1}{N} \sum_{x,y} (g(x, y) - f(x, y))^2 \tag{22}$$
where N is the number of pixels.
- The problem here is that it is not possible to distinguish a few big differences from a lot of small differences.
- Instead of the mean quadratic difference, the mean absolute difference or simply maximal absolute difference may be used.
- Correlation between images f and g is another alternative.

Image Quality: SNR

- Signal to noise ratio SNR is also used as a image degradation measure.
- Let $f(x, y)$ be the original image and $f'(x, y)$ be the degraded image, the degree of degradation is measured by
$$\mathrm{SNR}(f', f) = 10 \log_{10} \frac{\sum_{x,y} f(x, y)^2}{\sum_{x,y} (f(x, y) - f'(x, y))^2} \quad \text{(db)} \tag{23}$$

Image Quality: PSNR

- Peak signal to noise ratio PSNR is another measure in this class.
- PSNR is defined as

$$\mathrm{PSNR}(f', f) = 10 \log_{10} \frac{\max_{x,y} f(x,y)^2}{MSE(f', f)} \tag{24}$$

$$= 10 \log_{10} \frac{N \max_{x,y} f(x,y)^2}{\sum_{x,y} (f(x,y) - f'(x,y))^2} \quad \text{(db)} \tag{25}$$

  where N is the number of pixels.

- Experimentally, a PSNR larger than 32db means invisible visual degradation.

Image Quality Measures

- Measures of image similarity are becoming more important since they may be used in image retrieval from multimedia databases.
- There are many other measures of image similarity based on distance functions [D. R. Wilson and T. R. Martinez, 19

7.6 Noise in images

7.6.1 Image Noise

Image Noise

- Images are often degraded by random noise.
- Noise can occur during image capture, transmission or processing, and may be dependent on or independent of image content.
- Noise is usually described by its probabilistic characteristics.

White Noise

- White noise — constant power spectrum (its intensity does not decrease with increasing frequency);
- it is frequently applied as a crude approximation of image noise in most cases.
- Its auto-correlation is the delta function. So it is un-correlated at two different instances.
- The advantage is that it simplifies the calculations.

Gaussian Noise

- A special case of noise is Gaussian noise.
- Gaussian noise is a very good approximation of noise that occurs in many practical cases.
- Probability density of the random variable is given by the Gaussian function.
- 1D Gaussian noise — μ is the mean and σ is the standard deviation of the random variable.

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{26}$$

7.6.2 Noise Type

Noise Type: additive

- Noise may be

  additive the noise ν and image signal g are independent

$$f(x, y) = g(x, y) + \nu(x, y). \tag{27}$$

- During image transmission, noise is usually independent of the image signal occurs.
- The degradation can be modeled as additive noise.

Noise Type: multiplicative and impulse

- Noise may be

  multiplicative the noise is a function of signal magnitude

$$\begin{aligned} f(x, y) &= g(x, y) + \nu(x, y)g(x, y) && (28) \\ &= g(x, y)(1 + \nu(x, y)) && (29) \\ &= g(x, y)n(x, y). && (30) \end{aligned}$$

  impulse an image is corrupted with individual noisy pixels whose brightness differs significantly from that of the neighborhood.

Noise: discussions

- The term "salt and pepper noise" is used to describe saturated impulsive noise — an image corrupted with white and/or black pixel is an example.
- The problem of suppressing noise in images is addressed in subsequent lectures of this course.

7.6.3 Simulation of noise

Simulation of Gaussian Noise

- We first consider the simulation of Gaussian noise.
- **Theorem 7.1.** *(Box-Muller Method) Let U_1 and U_2 be i.i.d (independent identical distributed) uniformly distributed random variables on $(0, 1)$. Then the random variables*

$$N_1 = \sqrt{-2 \log U_1} \cos(2\pi U_2) \tag{31}$$

$$N_2 = \sqrt{-2 \log U_1} \sin(2\pi U_2) \tag{32}$$

  are independent standard Gaussian.

Proof of Box-Muller Method I

- To prove it we need the following
- **Theorem 7.2.** *(Transformation Theorem for Densities) Let Z_1, Z_2 and U_1, U_2 be random variables. Assume*

  – *(U_1, U_2) takes values in the open set G' of $\mathbf{R}^2$ and has density f on G';*

– (Z_1, Z_2) *takes values in the open set* G *of* $\mathbf{R}^2$*;*
– $\varphi : G \longmapsto G'$ *is a continuously differentiable bijection with continuously differentiable inverse* $\varphi^{-1} : G' = \varphi(G) \longmapsto G$.

Given

$$\begin{pmatrix} U_1 \\ U_2 \end{pmatrix} = \varphi \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}. \tag{33}$$

the random vector (Z_1, Z_2) *on* G *has the density*

$$g(z) = f \circ \varphi(z) |J_\varphi(z)| \tag{34}$$

where $J_\varphi(z) = \frac{\partial(\varphi_1, \varphi_2)}{\partial(z_1, z_2)}$ *is the Jacobian of* φ.

Proof of Box-Muller Method II

- Frst we determine the map φ from the last theorem.
- We have

$$N_1^2 = -2 \log U_1 \cos^2(2\pi U_2) \tag{35}$$
$$N_2^2 = -2 \log U_1 \sin^2(2\pi U_2). \tag{36}$$

- Hence

$$N_1^2 + N_2^2 = -2 \log U_1 \tag{37}$$

and

$$U_1 = \mathbf{e}^{-\frac{N_1^2 + N_2^2}{2}}. \tag{38}$$

- Moreover, by (31),

$$\frac{N_2}{N_1} = \tan(2\pi U_2), \tag{39}$$

i.e.,

$$U_2 = \frac{1}{2\pi} \arctan\left(\frac{N_2}{N_1}\right). \tag{40}$$

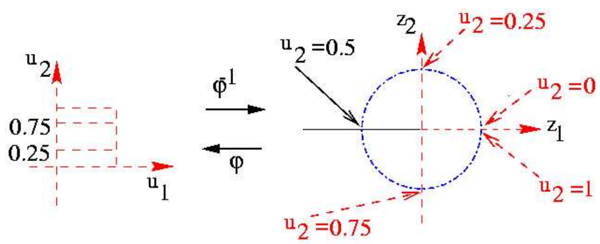
Proof of Box-Muller Method III

- Hence

$$\varphi(z_1, z_2) = \begin{pmatrix} \mathbf{e}^{-\frac{z_1^2 + z_2^2}{2}} \\ \frac{1}{2\pi} \arctan(\frac{z_2}{z_1}) \end{pmatrix} \tag{41}$$

- The transform domains are

$$G = \mathbf{R}^2 \setminus [\{z_1 = 0\} \cup \{z_2 = 0 \text{ and } z_1 > 0\}], \tag{42}$$
$$G' = (0,1) \times (0,1) \setminus \{u_2 = \frac{1}{4} \text{ or } u_2 = \frac{3}{4}\}. \tag{43}$$



Proof of Box-Muller Method IV

- The partial derivatives of φ are

$$\frac{\partial \varphi_1}{\partial z_1}(z) = -z_1 \mathbf{e}^{-\frac{z_1^2+z_2^2}{2}}, \qquad \frac{\partial \varphi_1}{\partial z_2}(z) = -z_2 \mathbf{e}^{-\frac{z_1^2+z_2^2}{2}} \tag{44}$$

$$\frac{\partial \varphi_1}{\partial z_1}(z) = \frac{1}{2\pi} \frac{-z_2}{z_1^2+z_2^2}, \qquad \frac{\partial \varphi_1}{\partial z_2}(z) = \frac{1}{2\pi} \frac{z_1}{z_1^2+z_2^2}. \tag{45}$$

- It follows that

$$J_\varphi(z) = \frac{1}{2\pi} \mathbf{e}^{-\frac{z_1^2+z_2^2}{2}} = \frac{1}{\sqrt{2\pi}} \mathbf{e}^{-\frac{z_1^2}{2}} \cdot \frac{1}{\sqrt{2\pi}} \mathbf{e}^{-\frac{z_2^2}{2}}. \tag{46}$$

Proof of Box-Muller Method V

- Since (U_1, U_2) has density $\chi_{(0,1)\times(0,1)}$, which is identically 1 on $\chi_{(0,1)\times(0,1)}$, (N_1, N_2) has density

$$\frac{1}{\sqrt{2\pi}} \mathbf{e}^{-\frac{z_1^2}{2}} \cdot \frac{1}{\sqrt{2\pi}} \mathbf{e}^{-\frac{z_2^2}{2}} \tag{47}$$

  on G. Therefore they are independent Gaussian variables. □

7.6.4 The rejection method

The Rejection Method I

- This method uses an auxiliary density for generation of random quantities from distributions not amenable to analytic treatment [Press et al., 1992].
- Consider the generation of samples from a density π.
- Consider also an auxiliary density q for which we know how to generate samples.
- The idea is to use q to make samples from π.
- The method is general enough to generate samples from π without knowing the complete expression for π.

The Rejection Method II

- It is extremely common in statistics to encounter such situations where the kernel of π is known but the constant ensuring it integrals to 1 cannot be obtained analytically.
- The only mathematical restriction over q is that there must exist a constant A such that

$$\pi(x) \le Aq(x) \tag{48}$$

  for every possible value of x.

The Rejection Method III

- The method consists of independently drawing X from q and $U \sim U[0,1]$ and accepting X as a sample generated from π if

$$AUq(X) \le \pi(X). \tag{49}$$

  Otherwise X is not accepted as a sample from π and the process must be reinitialized until a value X is accepted.
- Hence the name of the method, which is also known as the acceptance/rejection method.

Proof of the Rejection Method I

- To prove the rejection procedure effectively generates samples from π, we need to show that the density function of X is a constant multiple of π, when conditioned by the acceptance condition (49).
- The joint density of (X, U) is $f(x, u) = q(x)\chi_{[0,1]}(u)$, by the independence between X and U and the uniformity of U.
- The conditional probability is computed by the following formula:

$$\mathbf{Pr}(A|B) = \frac{\mathbf{Pr}(A, B)}{\mathbf{Pr}(B)}. \tag{50}$$

Proof of the Rejection Method II

$$\begin{aligned}
&\mathbf{Pr}(X \le t | AUq(X) \le \pi(X)) = \mathbf{Pr}(X \le t | U \le \frac{\pi(X)}{Aq(X)}) \\
&= \frac{\mathbf{Pr}(X \le t, U \le \frac{\pi(X)}{Aq(X)})}{\mathbf{Pr}(U \le \frac{\pi(X)}{Aq(X)})} \\
&= \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} q(x)\chi_{[0,1]}(u)\chi_{\left\{x \le t, u \le \frac{\pi(x)}{Aq(x)}\right\}}(x, u)\, dxdu}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} q(x)\chi_{[0,1]}(u)\chi_{\left\{u \le \frac{\pi(x)}{Aq(x)}\right\}}(x, u)\, dxdu} \\
&= \frac{\int_{-\infty}^{t} dx \int_{0}^{\frac{\pi(x)}{Aq(x)}} q(x)\chi_{[0,1]}(u)\, du}{\int_{-\infty}^{\infty} dx \int_{0}^{\frac{\pi(x)}{Aq(x)}} q(x)\chi_{[0,1]}(u)\, du} \\
&= \frac{\int_{-\infty}^{t} \frac{\pi(x)}{Aq(x)} q(x)\, dx}{\int_{-\infty}^{\infty} \frac{\pi(x)}{Aq(x)} q(x)\, dx} = \frac{\int_{-\infty}^{t} \pi(x)\, dx}{\int_{-\infty}^{\infty} \pi(x)\, dx}.
\end{aligned}$$

Extension of the Rejection Method

- The samples to be generated can have any form: scalar, vector or matrix.
- In each case, the rejection step is based on a comparison of densities with aid of a scalar uniform random variable.

Discussions on the Rejection Method

- q should be a density that is easy to draw samples from.
- The overall acceptance probability is

$$\mathbf{Pr}(U \le \frac{\pi(X)}{Aq(X)}) = \frac{1}{A}\int_{-\infty}^{\infty} \pi(x)\, dx \tag{51}$$

- Hence, A must be chosen as close as possible to $\int_{-\infty}^{\infty} \pi(x)\, dx$.

Rejection Method for Truncated Distributions

- A special case of rejection sampling is given for truncated distributions.
- Let q be any density and π be its truncation to the region C, i.e., $\pi = q\chi_C \leq q$.
- Taking $A = 1$, the acceptance condition is

$$Uq(X) \leq \pi(X) = q\chi_C \tag{52}$$

  which is satisfied, almost surely, if and only if $X \in C$.

- Hence, to generate sample from q restricted to C, one simply has to draw sample from q and accept it if and only if it is in C.

7.6.5 The Polar method

Polar Method

- This a variant of the Box-Muller method to generate standard normal deviates. It is due to G. Marsaglia.
- It is substantially faster than the Box-Muller method since it avoids the calculation of the trigonometric functions (but still slower than other methods, [Knuth, 1981]) and it has essential perfect accuracy.

Box-Muller Method Revisited

- The Box-Muller method may be rephrased as follows:
  - given (W, Θ) uniformly distributed on $[0, 1] \times [0, 2\pi]$;
  - the variables

$$N_1 = \sqrt{-2 \log W} \cos(\Theta) \tag{53}$$

$$N_2 = \sqrt{-2 \log W} \sin(\Theta) \tag{54}$$

    are independent standard Gaussian.

- The rejection method allows us to sample directly from $\sqrt{W} \cos \Theta$ and $\sqrt{W} \sin \Theta$, thus avoiding to calculate the sines and cosines.

Sampling $\sqrt{W} \cos \Theta$ and $\sqrt{W} \sin \Theta$

- Given (Z_1, Z_2) uniformly distributed on the unit disk;
- $Z_1 = R \cos \Theta$ and $Z_2 = R \sin \Theta$ in polar coordinates R, Θ;
- Then $W = R^2$ and Θ have joint density

$$\frac{1}{\pi}\chi_{\{|r|\leq 1\}}(\sqrt{W} \cos \Theta, \sqrt{W} \sin \Theta)|\frac{\partial Z_1, Z_2}{\partial W, \Theta}| \tag{55}$$

$$=\frac{1}{\pi}\chi_{(0,1]\times[0,2\pi)} \left| \begin{pmatrix} \frac{\cos \Theta}{2\sqrt{W}} & \frac{\sin \Theta}{2\sqrt{W}} \\ -\sqrt{W} \sin \Theta & \sqrt{W} \cos \Theta \end{pmatrix} \right| \tag{56}$$

$$=\frac{1}{2\pi}\chi_{[0,1]\times[0,2\pi)} \tag{57}$$

  on $(0, 1] \times [0, 2\pi)$;

- Hence (W, Θ) constructed above is uniform and independent on $(0, 1] \times [0, 2\pi)$;

Box-Muller Method in Polar Representation

- Clearly $W = Z_1^2 + Z_2^2$ and $\cos\Theta = \frac{Z_1}{\sqrt{W}}$, and $\sin\Theta = \frac{Z_2}{\sqrt{W}}$;
- Then the Box-Muller transform can be written as

$$N_1 = \sqrt{\frac{-2\log W}{W}} Z_1 \tag{58}$$

$$N_2 = \sqrt{\frac{-2\log W}{W}} Z_2 \tag{59}$$

$$W = Z_1^2 + Z_2^2 \tag{60}$$

Polar Method

To sample from the unit disk, we adopt the truncated rejection method:

- sample (V_1, V_2) uniformly from $[-1, 1] \times [-1, 1]$;
- until $0 < V_1^2 + V_2^2 \le 1$, set $(Z_1, Z_2) = (V_1, V_2)$.

7.6.6 Poissonian noise

Poissonian noise

- Each image data $I_{i,j}$ is the random sampling from a random variable, which is Poissonian distributed with mean $d_{i,j}$.
- Usually those random variables are independent.
- Then we have

$$\mathbf{Pr}(I|d) = \prod_{i,j} \mathbf{Pr}(I_{i,j}|d_{i,j}) \qquad = \prod_{i,j} \frac{{d_{i,j}}^{I_{i,j}}}{I_{i,j}!} \exp\left(-d_{i,j}\right) \tag{61}$$

7.6.7 Mixed noise: Poissonian + Gaussian

Mixed noise: Poissonian + Gaussian

- In this model, the image data $I_{i,j}$ consist of two parts,

$$I_{i,j} = P_{i,j} + G_{i,j} \tag{62}$$

  where

  - $O_{i,j}$ are the data from the object, which are independently Poissonian distributed random variables with mean $d_{i,j}$;
  - $g_{i,j}$ are real-valued random variables accounting for the noise that is present in the device and environment.

Homework: Image Quality and Image Noise

- Use images corrupted with various noises to demonstrated the performance of image quality measures such as MSE (22), SNR (23) and PSNR (24), etc.
- Which measure do you think *statitically reasonable*?

- You cal also consider to use those measures from [D. R. Wilson and T. R. Martinez, 1997].
- Please refer to [Wang et al., 2004] and references therein.

8 Color Images

Color

- 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 in which it is perceived.
- There is considerable literature on the variants in the naming of colors across languages — a very subtle affair.

Color Constancy

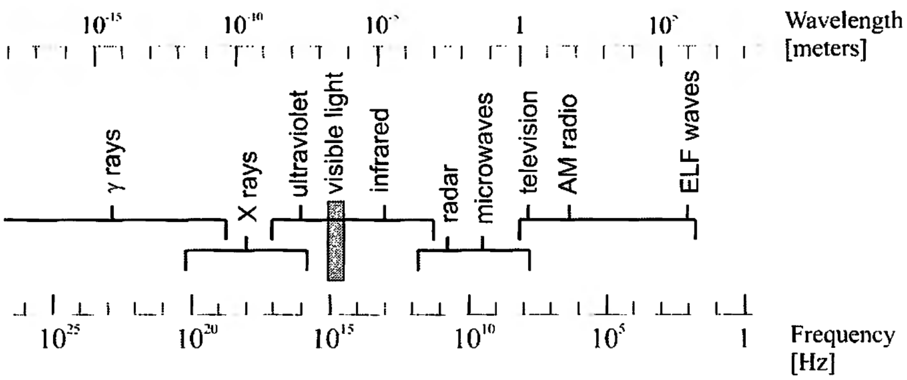
- The human visual system is not very precise in perceiving- color in absolute terms;
  - if we wish to express our notion of color precisely,
  - we would describe it relative to some widely used color which is used as a standard.
- There are whole industries which present images to humans, and hence 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**.

Newton

- Newton reported in the 17th century that white light from the sun is a spectral mixture,
- and used the optical prism to perform decomposition.
- This was a radical idea to propose at time.
- Over 100 years later influential scientists and philosophers such as Goethe refused to believe it.

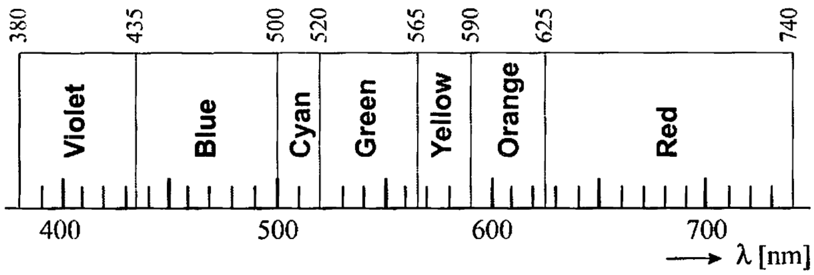
8.1 Physics of Color

Electromagnetic Spectrum



Visual Spectrum

- The electromagnetic spectrum visible to a human, is with the wavelength λ from approximately 380 nm to 740 nm.
- The intensity of irradiation for different wavelengths λ is called the `power spectrum` (or `power spectrum distribution` $S(\lambda)$.
- Visible colors with the wavelengths shown are called spectral colors



Primary Colors

- Colors can be represented as combinations of the primary colors, e.g., red, green, and blue;
- For the purposes of standardization they have been defined as 700 nm, 546.1 nm, and 435.8 nm, respectively [Pratt, 1978].
- But this standardization does not imply that all colors can be synthesized as combinations of these three.

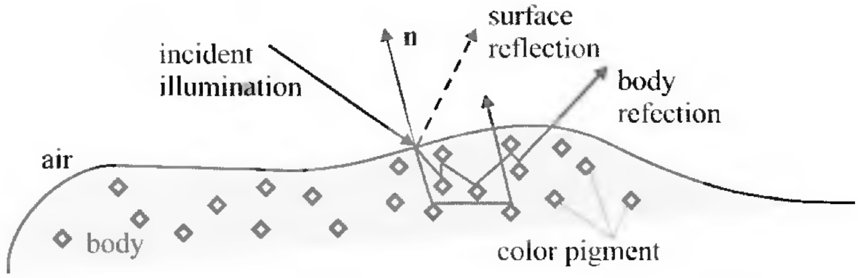
Surface Reflection

- 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'.

Body Reflection

- 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.
- Colors are caused by the properties of pigment particles which absorb certain wavelengths from the incoming illuminant wavelength spectrum.

Surface and Body Reflections



Spectrophotometer

- 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 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.

Multispectral images

- Light 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.
- `Multispectral 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.
- Seven or a dozen wavelength bands are common.
- For instance, the LANDSAT 4 satellite transmits digitized images in five spectral bands from near-ultraviolet to infrared.

8.2 Color Perceived by Humans

Receptors

- 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

- Cones are categorized into three types based on the sensed wavelength range
  - S (short) with maximum sensitivity at ≈ 430 nm,
  - M (medium) at ≈ 560 nm,
  - L (long) at ≈ 610 nm.
- Cones S, M, L are occasionally called cones B, G and R, respectively.
- This 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.

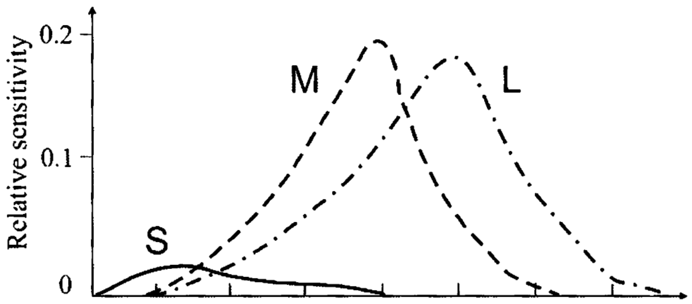
Spectral Response

- The reaction of a photoreceptor or output from a sensor in a camera can be modeled mathematically.
- Let i be the specific type of sensor, $i = 1, 2, 3$, (the retinal cone type S, M, L in the human case).
- Let $R_i(\lambda)$ be the spectral sensitivity of the sensor;
- $I(\lambda)$ be the spectral density of the illumination,
- and $S(\lambda)$ describe how the surface patch reflects each wavelength of the illuminating light.
- The spectral response q_i of the i-th sensor, can be modeled by integration over a certain range of wavelengths

$$q_i = \int_{\lambda_1}^{\lambda_2} I(\lambda) R_i(\lambda) S(\lambda) \tag{63}$$

Cones Relative Sensitivity

- Consider the cone types S, M, L.
- Only in the ideal case, when the illumination is perfectly white, i.e., $I(\lambda) = 1$,
- (q_S, q_M, q_L) is an estimate of the color of the surface.
- The figure illustrates qualitatively the relative sensitivities of S, M, L cones.
- Measurements carefully were taken with the white light source at the cornea so that absorption of wavelength in cornea, lens and inner pigments of the eye is taken into account [Wandell, 1995].



Relative sensitivities of S, M, L cones of the human eye to wavelength.

Color Metamer

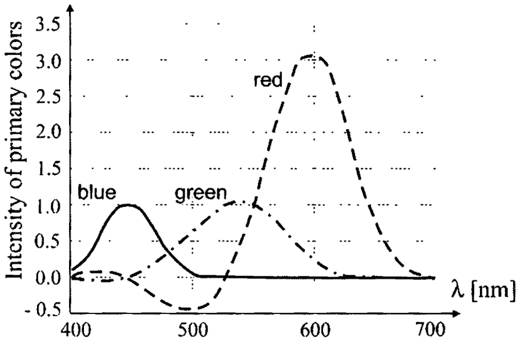
- A phenomenon called color metamer is relevant.
- A metamer, in general, means two things that are physically different but perceived as the same.
- Red and green adding to produce yellow is a color mctamer, because yellow could have also been produced by a spectral color.
- The human visual system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

Color Matching

- Consider a color matching experiment in which someone is shown a pattern consisting of two adjacent color patches.
- The first patch displays a test light — a spectral color of certain wavelength.
- The second patch is created as an additive combination of three selected primary lights.
- The observer is asked to control the red, green and blue intensities until both patches look identical.
- This color matching experiment is possible because of the color metamer.

Color Matching with Negative Values

- Negative lobes can be seen on the curves for red and green in this figure [Wandell, 1995].
- If the perceptual match has to be obtained then the observer has to add the intensity to the patch corresponding to the spectral color.
- This increase of this intensity is depicted as a decrease in the color matching function. Hence the negative values.



Color matching functions obtained in the color matching experiment. Intensities of the selected primary colors which perceptually match spectral color of given wavelength λ.

Human Vision

- 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.

XYZ Color Space

- 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 of numbers, typically as three or four values of color components.
- Being motivated by the press and the development of color film, CIE[1] issued a technical standard called XYZ color space in 1931.

CIE Standard

- The standard is given by
  - the three imaginary lights $X = 700.0\,\text{nm}$, $Y = 546.1\,\text{nm}$, $Z = 435.8\,\text{nm}$,
  - the color matching functions $X(\lambda)$, $Y(\lambda)$ and $Z(\lambda)$ corresponding to the perceptual ability of an average human viewing a screen through an aperture providing a 2 deg field of view.
- The standard is artificial because there is no set of physically realizable primary lights that would yield the color matching functions in the color matching experiment.
- Nevertheless, if we wanted to characterize the imaginary lights then, very roughly speaking, $X \approx$ red, $Y \approx$ green and $Z \approx$ blue.

[1] International Commission on Illumination, still acting in Lausanne, Switzerland.

XYZ Standards

- The XYZ color standard fulfills three requirements:

  A 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;

  B The value of $Y(\lambda)$ should coincide with the brightness (luminance);

  C 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).

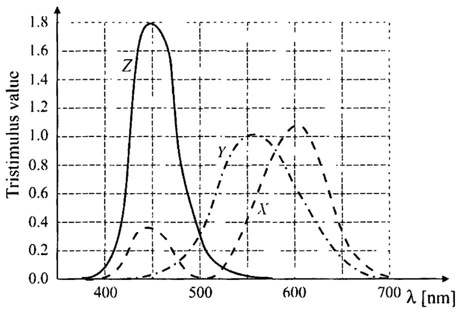
Color Match in XYZ

- A color is a mixture (more precisely a convex combination)

$$c_X X + c_Y Y + c_Z Z \tag{64}$$

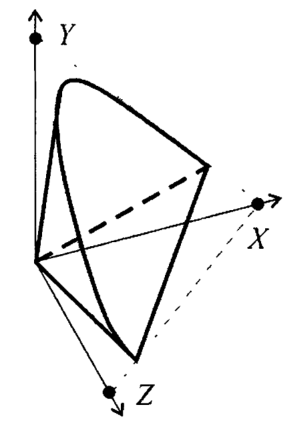
  where $0 \leq c_X, c_Y, c_Z le 1$, are weights (intensities) in the mixture.

- The resulting color matching functions are shown in the figure [Wandell, 1995].



Color Gamut in XYZ

- The subspace of colors perceivable by humans is called the `color gamut` and is demonstrated in the figure.



Projection View of Color Gamut in XYZ

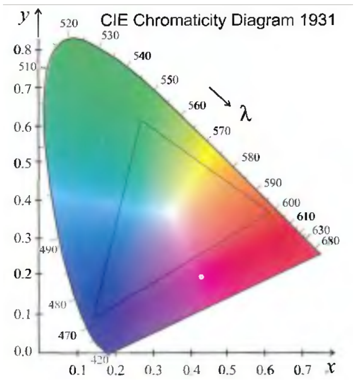
- The projection plane is given by the plane passing through extremal points on all three axes, `CIE chromaticity diagram`.

- The new 2D coordinates (x, y) are obtained as

$$x = \frac{X}{X+Y+Z}, \tag{65}$$

$$y = \frac{Y}{X+Y+Z}, \tag{66}$$

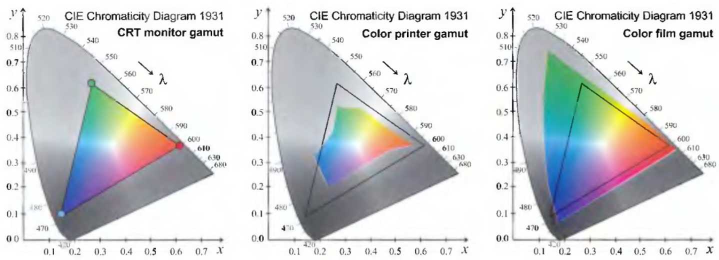
$$z = 1 - x - y. \tag{67}$$



All monochromatic spectra visible to humans map into the curved part of the horseshoe. The triangle depicts a subset of colors spanned by red, green, and blue.

Color Gamuts in XYZ

- 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.



Other Standards

- The CIE standard is an example of an absolute standard, i.e., defining unambiguous representation of color which does not depend on other external factors.
- There are more recent and more precise absolute standards: CIELAB 1976 (ISO 13665) and Hunter-Lab (www.hunterlab.com).
- Later, we will also discuss relative color standards such as RGB color space.
- There are several RGB color spaces used — two computer devices may display the same RGB image differently.

8.3 Color Spaces

Color Spaces

- Several different primary colors and corresponding color spaces are used in practice, and these 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.

RGB Color Space

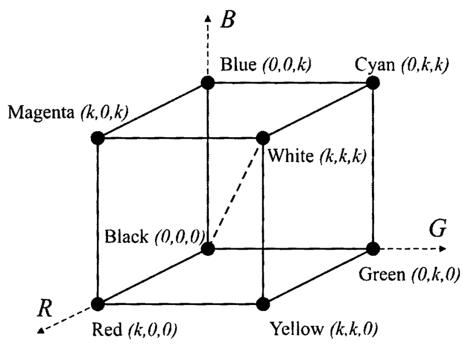
- The RGB color space has its origin in color television where Cathode Ray Tubes (CRT) were used.
- RGB color space is an example of a relative color standard (as opposed to the absolute one, e.g., CIE 1931).
- The primary colors (R-red, G-green and B-blue) mimicked phosphor in CRT luminophore.
- The 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.
- A transformation to a different color space is expressed by a transformation by a 3×3 matrix.

RGB Representation

- Assume that values for each primary are quantized to $m = 2^n$ values;
- let the highest intensity value be $k = m - 1$;
- then $(0, 0, 0)$ is black, (k, k, k) is (television) white, $(k, 0, 0)$ is 'pure' red, and so on.
- The value $k = 255 = 2^8 - 1$ is common, i.e., 8 bits per color channel.
- There are $256^3 = 2^{24} = 16,777,216$ possible colors in such a discretized space.

RGB Color Space Diagram

- RGB color space with primary colors red, green, blue and secondary colors yellow, cyan, magenta.
- Gray-scale images with all intensities lie along the dashed line connecting black and white colors in RGB color space.
- The RGB model may be thought of as a 3D co-ordinatization of color space; note the secondary colors which are combinations of two pure primaries.



RGB vs XYZ

- There are specific instances of the RGB color model as sRGB, Adobe RGB and Adobe Wide Gamut RGB.
- They differ slightly in transformation matrices and the gamut.
- One of transformations between RGB and XYZ color spaces is

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.24 & -1.54 & -0.50 \\ -0.98 & 1.88 & 0.04 \\ 0.06 & -0.20 & 1.06 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} ,$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.41 & 0.36 & 0.18 \\ 0.21 & 0.72 & 0.07 \\ 0.02 & 0.12 & 0.95 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} .$$

YIQ Color Space

- The US and Japanese color television formerly used YIQ color space.
- The Y component describes intensity
- and I, Q represent colors, corresponding approximately to the amounts of blue and red in the color.
- YIQ is another example of additive color mixing.

YUV Color Space

- This color space corresponds closely to the YUV color model in the PAL television norm.
- YIQ color space is rotated 33 deg with respect to the YUV color space.
- The YIQ color model is useful since the Y component provides all that is necessary for a monochrome display;
- It exploits advantageous properties of the human visual system, in particular our sensitivity to luminance.

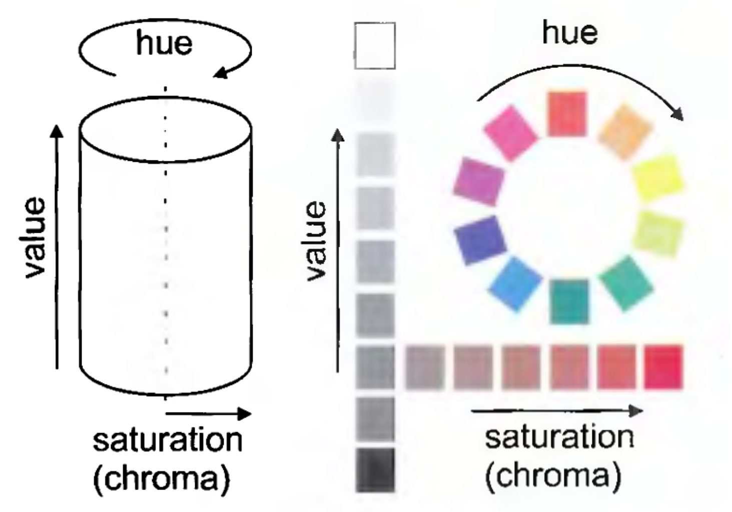
CMY Color Space

- 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.
- 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 Color Space

- 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.
- 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 (andor black) to obtain the desired lower saturation.

HSV Color Model



HSV, HSL, HSI

- HSV decouples intensity information from color, while hue and saturation correspond to human perception.
- This representation is very useful for developing image processing algorithms.
- This will become clearer as we proceed to describe image enhancement algorithms.
- 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.**

Summary

Models	Color spaces	Applications
Colorimetric	XYZ	Colorimetric calculations
Device oriented, nonuniform spaces	RGB, UIQ	Storage, processing, coding, color TV
Device oriented, Uniform spaces	LAB, LUV	Color difference, analysis
User oriented	HSL, HSI	Color perception, computer graphics

- Be careful for formulas in books.

Multi-spectral image demonstration

- Multi-spectral image captured by Land-sat TM-5.
- The example uses the images from its 3rd, 4th and 5th spectral bands.
- The matlab script for this example is multi_spectral_image.m.

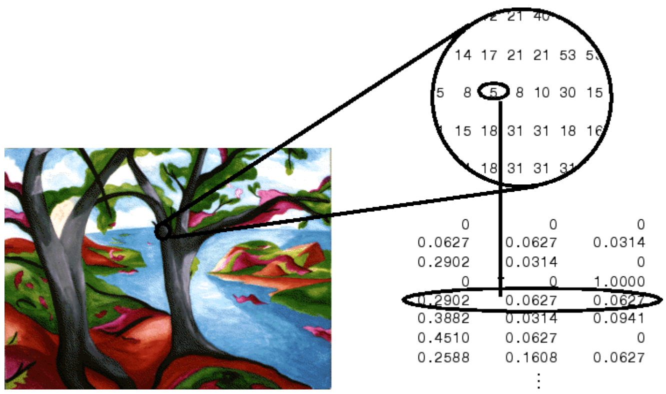
8.4 Palette Images

Palette Images

- Palette images (*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 of colors (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, 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.
- This approach would reduce data size to 1/3 (plus size of the look up table).

Image Formats

- In the RGB color model, values for red, green and blue are provided.
- Image formats for raster images such as TIFF, PNG and GIF can store palette images.



(image from www.mathworks.com)

Not Enough Colors?

- 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
  - a subset of colors has to be chosen,
- and a loss of information occurs.

Color Selection

- 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 $8 \times 8 \times 8 = 256$ cubes.
  - for a green frog in green grass there will not be enough shades of green in the lookup table to display the image well.
- It is better to
  - check which colors appear in the image by creating histograms for all three color components
  - quantize them 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 the lookup table is used to represent the color.

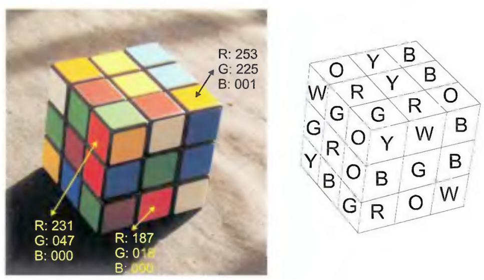
Pseudocolor

- 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 is an issue.

8.5 Color Constancy

Color Constancy: a challenging problem

- Consider the situation in which the same surface is seen under different illumination.
- The same surface colors are shown fully illuminated and in shadow.
- The human vision system is able to abstract to a certain degree from the illumination changes and perceive several instances of a particular color as the same.
- This phenomenon is called `color constancy`.



Part III
Data structures for image analysis

Outline

Contents

Data and Algorithm

- Data and an algorithm are the two basic parts of any program.
- Computer program = data + algorithm.
- Data organization can considerably affect the simplicity of the selection and the implementation of an algorithm.
- The choice of data structures is fundamental when writing a program.

Inappropriate Representation for Images

- The difficulties in image processing, image analysis and computer vision come from the bad representation or organization of the data involved.
- In fact, the visual information representation and organization inside human brain is not well understood at present.
- Although we are to discuss some representations used so far, none of them are appropriate for a general purpose processing target.

9 Levels of Representation

Levels of Representation

- The aim of computer visual perception is to find a relation between an input image and the models of the real world.
- During the transition from the raw image to the model, `semantic knowledge` about the interpretation of image data is used more and more.
- Several levels of visual information representation are defined on the way between the input image the model.

Levels of Representation and Algorithms

- Computer vision then comprises a design of the
  - Intermediate representations (data structures).
  - Algorithms used for the creation of representation and introduction of relations between them.

Four Levels of Representations

- The representation can be stratified in four levels.
  - `Iconic images`
  - `Segmented images`
  - `Geometric representations`
  - `Relational models`

Discussions on Four Levels of Representations

- However, there are no strict borders between them and a more detailed classification of the representational levels may be used in some applications.
- For some specific uses, some representations can be omitted.
- These four representational levels are ordered from signals at low level of abstraction to the description that a human can understand.
- The information flow between the levels may be bi-directional.

Iconic Images

- `Iconic images` — consists of images containing original data; integer matrices with data about pixel brightness.
- E.g., outputs of pre-processing operations (e.g., filtration or edge sharpening) used for highlighting some aspects of the image important for further treatment.

Segmented Images

- `Segmented images` — parts of the image are joined into groups that probably belong to the same objects.
- E.g., the output of the segmentation of a scene with polyhedrons is either line segments coinciding with borders or two-dimensional regions corresponding with faces of bodies.
- It is useful to know something about the application domain while doing image segmentation; it is then easier to deal with noise and other problems associated with erroneous image data.

Geometric Representations

- `Geometric representations` — hold knowledge about 2D and 3D shapes.
- The quantification of a shape is very difficult but very important.
- It is the inverse problem of computer graphics.

Relational Models

- `Relational models -` give the ability to treat data more efficiently and at a higher level of abstraction.
- A priori knowledge about the case being solved is usually used in processing of this kind.
- Example - counting planes standing at an airport using satellite images
  - position of the airport (e.g., from a map).
  - relations to other objects in the image (e.g., to roads, lakes, urban areas).
  - geometric models of planes for which we are searching.
  - etc.
- AI techniques are often explored.
- Information gained from the image may be represented by semantic nets or frames.

10 Traditional Image Data Structures

Traditional Image Data Structures

- Traditional image data structures, such as
  - matrices
  - chains
  - graphs
  - lists of object properties
  - relational databases
  - etc.

  are important not only for the direct representation of image information, but also a basis of more complex hierarchical methods of image representation.

10.1 Matrices

Matrices

- This is the most common data structure for low level image representation.
- Elements of the matrix are integer, real or complex numbers, corresponding to brightness, or to another property of the corresponding pixel of the sampling grid.
- Image data of this kind are usually the direct output of the image capturing device, e.g., a scanner.
- Pixels of both rectangular and hexagonal sampling grids (Fig. 96) can be represented by a matrix.
- The correspondence between data and matrix elements is obvious for a rectangular grid; with a hexagonal grid every row in the image is shifted half a pixel to the right or left.

Images Represented by Matrices

- Some special images that are represented by matrices are:
  - Binary images (an image with two brightness levels only) is represented by a matrix containing zeros and ones.
  - Several matrices can contain information about one multi-spectral image. Each of these matrices contains one image corresponding to one spectral band.
  - Matrices of different resolution are used to obtain hierarchical image data structures.
  - A representation of a segmented image by a matrix usually saves memory than an explicit list of all spatial relations between all objects, although sometimes we need to record other relations among objects.

Images = Matrices = Arrays

- Image information in the matrix is accessible through the co-ordinates of a pixel that correspond with row and column indexes.
- Most programming language use a standard array data structure to represent a matrix.

Images = Matrices = Arrays in C

- In C, an image of 256 gray-levels with dimension $M \times N$ can be stored in a two dimensional array

```
unsigned char image [M][N]
```

- The above stored array may be dis-continuous in memory.
- Then another way to represent the image is by a one-dimensional array, which is continuous in the memory

```
unsigned char image [M * N]
```

- We need to know the image dimension M and N usually.

Images = Matrices = Structure in C (I)

- Another method is to represent an image by a structure type,

```
typedef struct _image
{
   int rows;
   int columns;
   char *values;
   ¡other useful information¿;
} Image;
```

or use a pointer to the above structure

```
typedef Image * ImageX;
```

or declared as

```
typedef struct _image * ImageX;
```

Images = Matrices = Structure in C (II)

- When initialized,

```
value = (char *) malloc (sizeof(char) * rows * columns);
```

- To access one element of the matrix at pixel (m, n), use the following macro

```
#define Pixel(IX, m, n)
                                        (IX-¿values[m*(IX-¿columns)+n])
```

where $0 \leq m \leq M-1$ and $0 \leq n \leq N-1$.

Other Image Data Types

- You may use other type such as

```
float *values;
```

or

```
double *values;
```

to implement much precise computation.

10.2 Chains

Chains

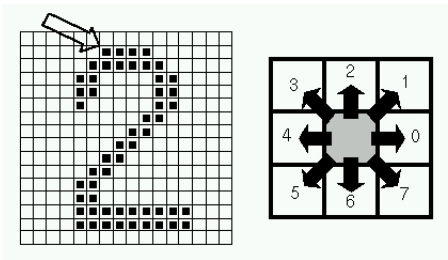
- Chains are used for the representation of object borders in computer vision.
- One element of the chain is a basic symbol, which corresponds to some kind of primitives in the image.
- Chains are appropriate for data that can be arranged as a sequence of symbols.
- The neighboring symbols in a chain usually correspond to the neighboring of primitives in the image.

Chain Codes

- `Chain codes` or `Freeman codes` [Freeman, 1961] are often used for the description of object borders, or other one-pixel-wide primitives in images.
- Chain codes describe an object by a sequence of unit-size length line segments with a given orientation.
- The border is defined by the co-ordinates of its reference pixel and the sequence of symbols for lines of unit length in several predefined orientations.
- The first element of such a sequence must bear information about its position to permit the region to be reconstructed.
- The result is a sequence of numbers, indicating the orientation.
- A chain code is of relative nature; data are expressed with respect to some reference point.

Chain Codes: Example

- An example of a chain code is shown in the following figure, where 8-neighborhoods are used — it is possible to define chain code using 4-neighborhoods as well. The reference pixel is marked by an



arrow. The chain code is

00077665555556600000006444444442221111112234445652211

Chain Codes: Discussions

- To exploit the position invariance of chain codes, e.g., when used for matching, the first element, which contains the position information, should be omitted. One need a method to normalize the chain codes.

- A chain code is very sensitive to noise and arbitrary changes in scale and rotation may cause problems if used fro recognition.
- The description of an image by chains is appropriate for syntactic pattern recognition that is based on formal language theory approaches.
- Chains can be represented using static data structures (e.g., 1D arrays); their size is the longest length of the chain expected.
- Dynamic data structures are more advantageous to save memory.

10.3 Topological Data Structures

Topological Data Structures by Graphs

- Topological data structures describe the image as a set of elements and their relations.
- These relations are often represented using graphs.
- A graph $G = (V, E)$ is an algebraic structure which consists of a set of nodes

$$V = \{v_1, v_2, \cdots, v_n\}, \tag{68}$$

  and a set of arcs

$$E = \{e_1, e_2, \cdots, e_m\}. \tag{69}$$

- Each arc e_k is naturally connected with an unordered pair of nodes $\{v_i, v_j\}$ which are not necessarily distinct.
- The degree of the node is equal to the number of incident arcs of the node.

Evaluated Graphs

- An evaluated graph is a graph in which values are assigned to arcs, to nodes or to both — these values may, e.g., represent weights, or costs.

Region Adjacency Graphs

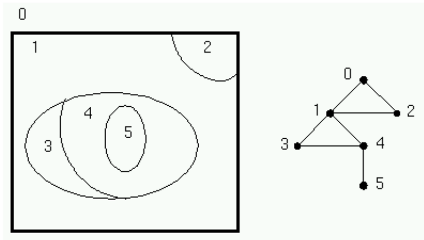
- The `region adjacency graph` is typical of this class of data structures.
- Nodes correspond to regions and neighboring regions are connected by an arc.
- The segmented image consists of regions of pixels with similar properties (brightness, texture, color, ...) that correspond to some entities in the scene.
- The `neighborhood relation` is fulfilled when the regions have some common border.

Region Adjacency Graphs: Example

An example of an image with regions labeled by numbers and corresponding region adjacency graph is shown in the following figure The label 0 is denotes pixels out of the image. This value is used to indicate regions that touch borders of the image in the region adjacency graph.

Region Adjacency Graphs: Discussions

- The region adjacency graph has several attractive features.
- If a region enclose other regions, then the part of the graph corresponding with the areas inside can be separated by a cut in the graph.
- Nodes of degree 1 represent simple holes.
- The region adjacency graph can be used for matching with a stored pattern for recognition purpose.



Region Adjacency Graphs: Creation

- The region adjacency graph is usually created from the `region map`, which is a matrix of the same dimension as the original image matrix whose elements are identification labels of the regions.
- To created the region adjacency graph, borders of all regions in the image are traced, and labels of all neighboring regions are stored.
- The region adjacency graph can easily be created from an image represented by a `quadtree` in § 11.2 as well.

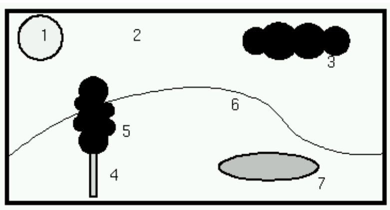
10.4 Relational Structures

Relational Structures

- Relational databases can also be used for representation of information from an image.
- All the information is then concentrated in relations between semantically important parts of the image — objects — that are the result of segmentation.
- Relations are recorded in the form of tables.
- Individual objects are associated with their names and other features, e.g., the top-left pixel of the corresponding region in the image.
- Relations between objects are expressed in the relational table as well.

Relational Structures: Example

An example of such a representation is shown in the following figure and table.



(a) Image

No.	Object name	Color	Min. row	Min. col.	Inside
1	sun	white	5	40	2
2	sky	blue	0	0	-
3	cloud	gray	20	180	2
4	tree trunk	brown	95	75	6
5	tree crown	green	53	63	-
6	hill	light green	97	0	-
7	pond	blue	100	160	6

(b) Database

Relational Structures: Discussions

- Description by means of relational structures is appropriate for higher level image understanding.
- Search using keys, similar to database searches, can be used to speed up the whole process.

11 Hierarchical Data Structures

Huge Dataset

- Computer vision is by its nature very computationally expensive, if for no other reason than the large amount of data to be processed.
- Systems which we might call sophisticated must process considerable quantities of image data — hundreds of kilobytes to tens of megabytes.
- The visual information perceived by the two human eyes is about 3000 MB/s (add reference or `estimate it`!!!).
- One of the solutions is using parallel computers (in other words brute force).
- Unfortunately many computer vision problems are difficult to divide among processors, or decompose in any way.

Hierarchical Data Structures

- `Hierarchical data structures` make it possible to use algorithms which decide a strategy for processing on the basis of relatively small quantities of data.
- They work at the finest resolution only with those parts of the image for which it is necessary, using knowledge instead of brute force to ease and speed up the processing.
- We are going to introduce two typical hierarchical structures, pyramids and quadtrees.

11.1 Pyramids

Pyramids

- Pyramids are among the simplest hierarchical data structures.
- We distinguish between `M-pyramids` (matrix pyramids) and `T-pyramids` (tree pyramids).

M-Pyramids

- A M-pyramid is a sequence $\{M_L, M_{L-1}, \cdots, M_0\}$ of images.
- M_L has the same dimensions and elements as the original image.
- M_{i-1} is derived from the M_i by reducing the resolution, **usually**, by one half.

M-Pyramids: Creation

- When creating pyramids, it is customary to work with square matrices with dimensions equal to powers of 2.
- For images with arbitrary dimensions, a resampling procedure is needed in creating the pyramids.
- M_0 corresponds to one pixel only.
- The number of image pixels used by an M-pyramid for storing all matrices is

$$N^2 \left(1 + \frac{1}{4} + \frac{1}{16} + \cdots\right) = 1.33N^2$$

M-Pyramids: Discussions

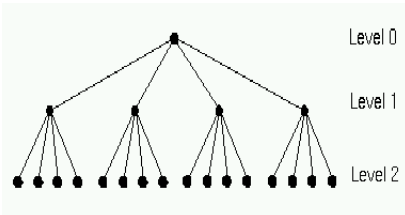
- M-pyramid created by shrinking the image dimensions. The matlab script for this example is m_pyramid.m.

M-Pyramids: Discussions

- M-pyramids are used when it is necessary to work with an image at different resolutions simultaneously.
- An image having one degree smaller resolution in a pyramid contains four times less data, so that it is processed approximately four times as quickly.

T-Pyramids

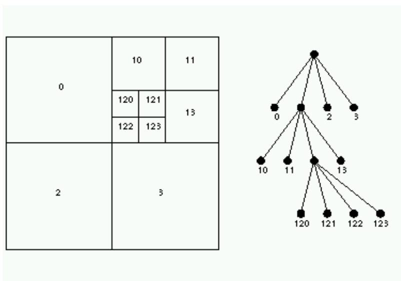
- Often it is advantageous to use several images of the same resolution simultaneously rather than to create just one image at a resolution in the M-pyramid.
- E.g., we use images at a resolution, containing additional information at this resolution, texture, orientation and segmentation properties, etc.
- Such images can be represented using tree pyramids — T-pyramids.
- The following figure is a example T-pyramid tree. Every node of the T-pyramid has 4 child nodes.



11.2 Quadtrees

Quadtrees

- Quadtrees are modifications of T-pyramids.
- Every node of the tree except the leaves has four children (NW: north-western, NE: north-eastern, SW: south-western, SE: south-eastern).
- The image is divided into four quadrants at each hierarchical level, though it is not necessary to keep nodes at all levels.
- If a parent node has four children of the same (e.g., brightness) value, (which is often characterized by a similarity measure), it is not necessary to record them.



Quadtrees: Structures

- Quadtrees are usually represented by recording the whole tree as a list of its individual nodes.
- Every node being a record with several items characterizing it.
- An example is given as following

```
node = {
        node_type,
        pointer_to_NW_son,
        pointer_to_NE_son,
        pointer_to_SW_son,
        pointer_to_SE_son,
        pointer_to_Father,
        other_data
      }
```

- In the item "node_type", there is information about whether the node is a leaf or inside the tree.
- "other_data" can be the level of the node in the tree, position in the picture, brightness for this node, etc.
- This kind of representation is redundant and expansive in memory. Its advantage is easy access to any node.

Quadtrees: Example

- Launch matlab, run the demo "qtdemo".
- matlab uses sparse matrix to store the quadtree decomposition, without the brightness value information for each node or block.

Quadtrees: Discussions

- Disadvantages associated with image pyramid representation:
  - Dependence on the position, orientation and relative size of objects.
  - Two similar images with just very small differences can have very different pyramid or quadtree representations.
  - Even two images depicting the same, slightly shifted scene, can have entirely different representations.
  - These disadvantages can be overcome using a normalized shape of quadtree in which do not create the quadtree for the whole image, but for its individual objects.
  - Please refer the text book for further discussions.

Part IV
Image Pre-processing

Pre-processing

- Pre-processing is a common name for operations with images at the lowest level of abstraction — both input and output are intensity images.

- These iconic images are of the same kind as the original data captured, with an intensity image usually represented by a matrix of image function values (brightness).
- The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.

Pre-processing: Classification

- Four categories of image pre-processing methods according to the size of the pixel neighborhood that is used for the calculation of a new pixel brightness:
  - pixel brightness transformations.
  - geometric transformations.
  - pre-processing methods that use a local neighborhood of the processed pixel.
  - image restoration that requires knowledge about the entire image.
- Other classifications of image pre-processing methods exist.

Pre-processing: basic idea

- Image pre-processing methods use the considerable redundancy in images.
- Neighboring pixels corresponding to one object in real images have essentially the same or similar brightness value — `spatial coherence` or `spatial correlation`.
- Thus, distorted pixel can often be restored as an average value of neighboring pixels.

Pre-processing: *a priori* information

- If pre-processing aims to correct some degradation in the image, the nature of *a priori* `information` is important and is used to different extent:
  - no knowledge about the nature of the degradation is used; only very general properties of the degradation are assumed.
  - using knowledge about the properties of the image acquisition device, and conditions under which the image was obtained. The nature of noise (usually its spectral characteristics) is sometimes known.
  - using knowledge about objects that are searched for in the image, which may simplify the pre-processing quite considerably.
- If knowledge about objects is not available in advance it can be estimated during the processing.

Pre-processing: Strategy

- The following strategy is possible.
  - First the image is coarsely processed to reduce data quantity and to find image objects.
  - The image information derived is used to create a hypothesis about image object properties and this hypothesis is then verified in the image at finer resolution.
  - Such an iterative process cab be repeated until the presence of knowledge is verified or rejected.
  - This feedback may span more than pre-processing, since segmentation also yields semantic knowledge about objects — thus feedback can be initiated after the object segmentation.

12 Pixel Brightness Transformations

Pixel Brightness Transformations

- Brightness transformations modify pixel brightness — the transformation depends on the properties of a pixel itself.
- There are two brightness transformations:
- `Brightness corrections`
  - consider the original brightness
  - and pixel position in the image.
- `Gray scale transformation`
  - change brightness without regard to position in the image.

12.1 Position dependent brightness correction

Position dependent brightness correction

- Ideally, the sensitivity of image acquisition and digitization devices should not depend on position in the image, but this assumption is not valid in practice.
- Sources of degradation:
  - non-homogeneous property of optical system;
    The lens attenuates light more if it passes farther from the optical axis.
  - non-homogeneous sensitivity of light sensors;
    The photo sensitive part of the sensor (vacuum-tube camera, CCD camera elements) is not of identical sensitivity.
  - non-homogeneous object illumination.
- Systematic degradation can be suppressed by brightness correction.

Multipilcation Degradation

- Let a multiplicative error coefficient $e(i,j)$ describe the change from the ideal identity transfer function
  - $g(i,j)$ is the original undegraded image (or desired image);
  - $f(i,j)$ is the image containing degradation.

$$f(i,j) = e(i,j)g(i,j) \tag{70}$$

Correction for Multipilcation Degradation

- If a reference image $g_c(i,j)$ is known (e.g., constant brightness c)
  - the degraded result is $f_c(i,j)$
  - systematic brightness errors can be suppressed:

$$g(i,j) = \frac{f(i,j)}{e(i,j)} = \frac{g_c(i,j)f(i,j)}{f_c(i,j)} = \frac{c}{f_c(i,j)}f(i,j) \tag{71}$$

Discussions I

- This method can be used only if the image degradation process is stable.
- If we wish to suppress this kind of degradation in the image capture process, we should calibrate the device from time to time (find error coefficients $e(i, j)$)
- This method implicitly assumes linearity of the transformation, which is not true in reality as the brightness scale is limited into some interval.
  - overflow is possible in (71). Then the limits of the brightness scale are used instead in (71).
  - The best reference image should have brightness that is far enough from both limits.

Discussions II

- If the gray scale has 256 brightnesses the ideal image has constant brightness value 128.
  - Most TV/DV cameras have automatic control of the gain which allows them to operate under changing illumination conditions.
  - If systematic errors are suppressed using error coefficients, this automatic gain control should be switched off first.

Brightness Correction Example

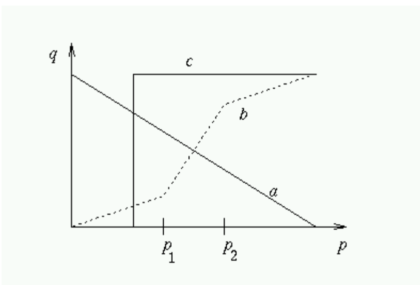
- The matlab script for this example is brightness_correction.m.

12.2 Grey scale transformation

Grey scale transformation

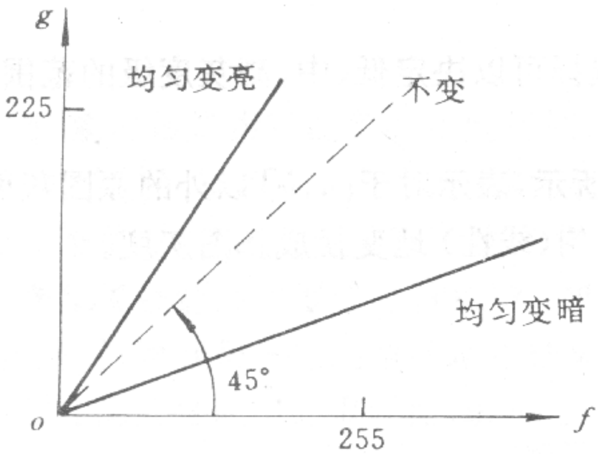
- Grey scale transformations do not depend on the position of the pixel in the image.
- Brightness transform is a monotonic function:

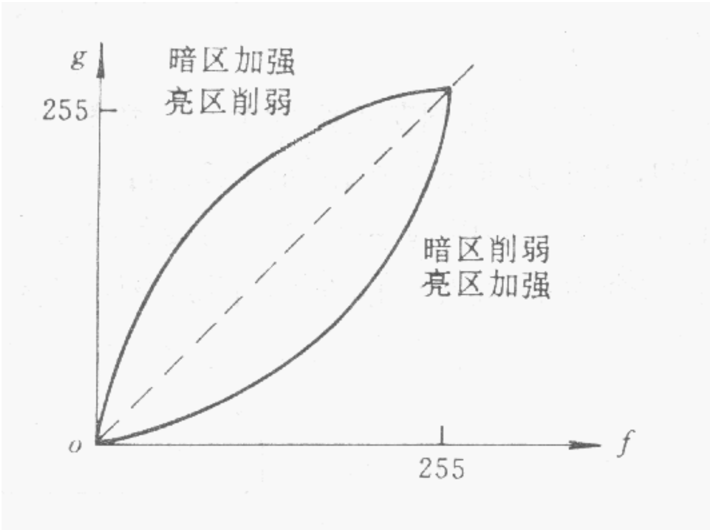
$$q = T(p) \tag{72}$$

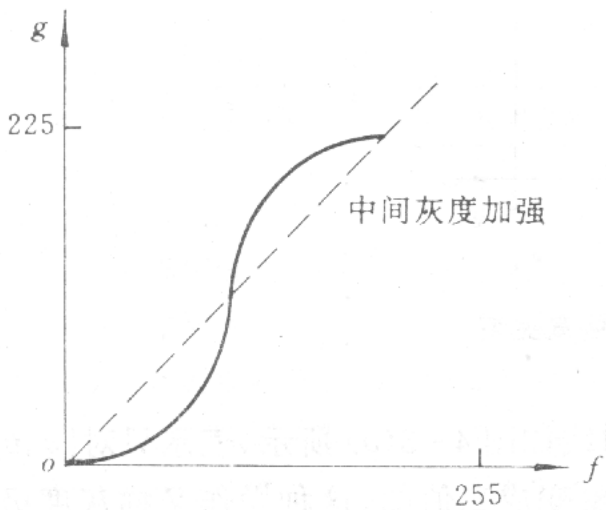


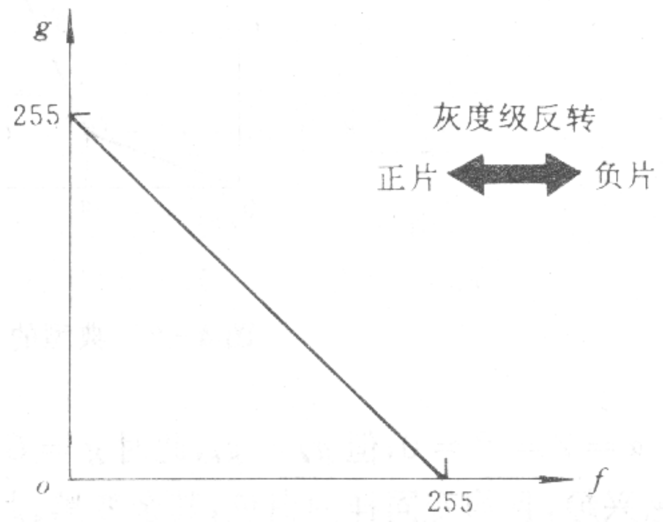
  - a - Negative transformation
  - b - contrast enhancement (between p_1 and p_2)
  - c - Brightness thresholding

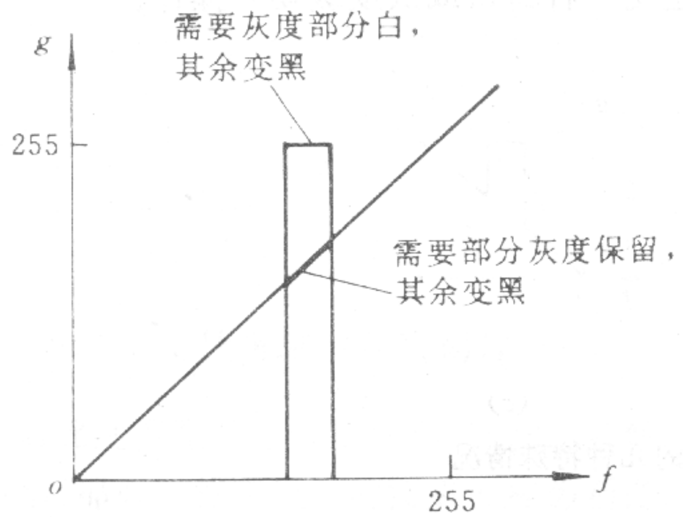
Some Other Examples

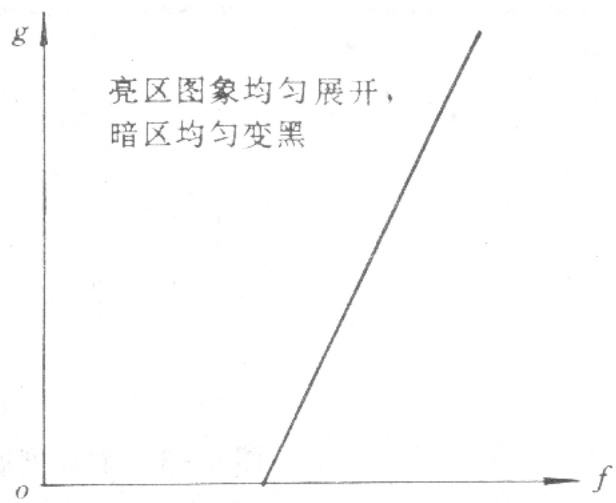

g
225
均匀变亮
不变
均匀变暗
45°
o
255
f


g
暗区加强
亮区削弱
255
暗区削弱
亮区加强
o
255
f


g
225
中间灰度加强
o
255
f


g
255
灰度级反转
正片
负片
o
255
f


需要灰度部分白，
其余变黑
g
255
需要部分灰度保留，
其余变黑
o
255
f


g
亮区图象均匀展开，
暗区均匀变黑
o
f

Look-up Tables

- Grey scale transformations can be performed using look-up tables.
- Grey scale transformations are mostly used if the result is viewed by a human.

12.2.1 Windows and level

Windows and level

- It is an interactive contrast enhancement tool.
- It is an expansion of the contrast of pixels within a given window range.
- Two parameters define the range: the middle point Level, and the width of the range Window.
- Another name for this operation is `Intensity of Interest` (IOI).

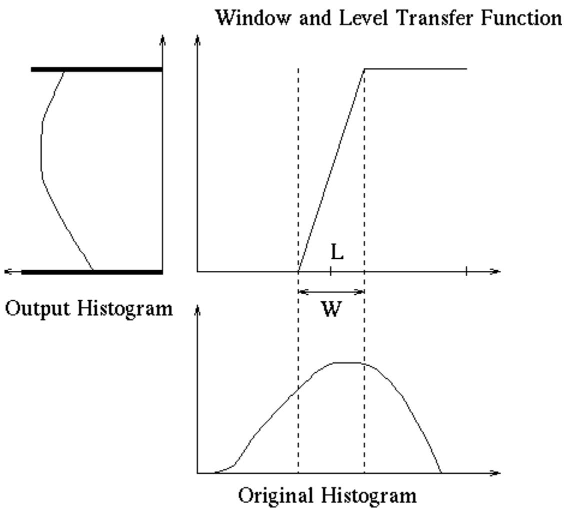
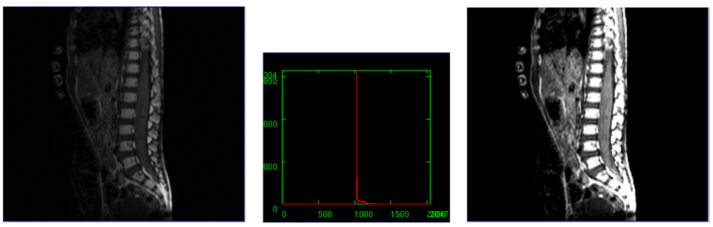


Figure 10: A graphic visualization of this process.

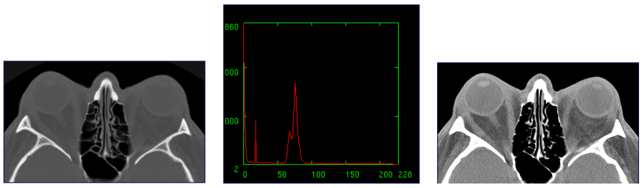
Example I



(a) Original (b) Hisogram (c) After Windows and Level Contrast Change

Figure 11: First find the minimum and maximum values, decide the initial value, bin-width and number of bins for computing the histogram. From the histogram, chose the lower and upper cutoff value.

Example II



(a) Original (b) Hisogram (c) After Windows and Level Contrast Change

12.2.2 Histogram stretching

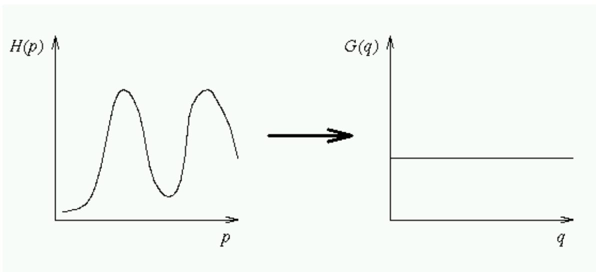
Histogram stretching

- Histogram stretching can be seen as a Window and Level contrast enhancement technique where the window ranges from the minimum to the maximum pixel values of the image.
- This normalization or histogram stretching operation is automatically performed in many display operators.
- To fully illustrate the difference between the two displayed images a gray level scale was superimposed in both images to guarantee that the display operator will use the same gray-level scale.

12.2.3 Histogram Equalization

Histogram Equalization I

- Histogram equalization is **to produce an image with equally distributed brightness levels over the whole brightness scale**.



Histogram Equalization II

- Let $H(p)$ be the input histogram and the input gray-scale range be $[p_0, p_k]$.
- We are to find a monotonic pixel brightness transform $q = T(p)$ such that the output histogram $G(q)$ is uniform over the whole output brightness scale $[q_0, q_k]$.
- The histogram is a discrete probability density function.
- The monotonic property of the transform T implies

$$\sum_{i=0}^{j} G(q_i) = \sum_{i=0}^{j} H(p_i) \tag{73}$$

where $q_i = T(p_i)$.

- The sum in the above equation can be interpreted as discrete distribution function.

Histogram Equalization III

- Assume that the image has M rows and N columns. The total number of pixels is MN.
- The equalized histogram corresponding to the uniform probability density function f whose function value satisfies

$$f(q)(q_k - q_0) = MN. \tag{74}$$

- The continuous version of (73) is

$$\int_{q_0}^{q} f(r)\, dr = \int_{p_0}^{p} H(s)\, ds, \quad q = T(p). \tag{75}$$

- Therefore, we obtain,

$$MN \int_{q_0}^{q} \frac{1}{q_k - q_0}\, dr = \frac{MN(q - q_0)}{q_k - q_0} = \int_{p_0}^{p} H(s)\, ds. \tag{76}$$

Histogram Equalization IV

- The desired pixel brightness transformation T can then be derived as

$$q = T(p) = \frac{q_k - q_0}{MN} \int_{p_0}^{p} H(s)\, ds + q_0. \tag{77}$$

- The integral in the above equation is called the cumulative histogram, which is approximated by a sum for digital images,

$$F[p] = \int_{p_0}^{p} H(s)\, ds = \sum_{i=0}^{j} H(p_i), \quad p_j = \text{round}(p).$$

- Therefore, resulting histograms are not generally equalized ideally.
- The discrete approximation of the continuous pixel brightness transformation form the above equation is

$$q_j = T(p_j) = \frac{q_k - q_0}{MN} \sum_{i=0}^{j} H(p_i) + q_0. \tag{78}$$

Histogram Equalization Algorithm

1. For an $N \times M$ image of G gray-levels (often 256), create two arrays H and T of length G initialized with 0 values.
2. Form the image histogram: scan every pixel and increment the relevant member of H — if pixel X has intensity p, perform

$$H[p] = H[p] + 1 \tag{79}$$

3. Form the cumulative image histogram H_c. We may use the same array H to stored the result.

$$H[0] = H[0]$$
$$H[p] = H[p-1] + H[p]$$

   for $p = 1, \cdots, G - 1$.

4. Set

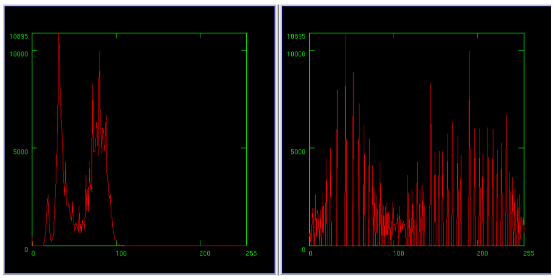
$$T[p] = \text{round}\left[\frac{G-1}{MN} H[p]\right]. \tag{80}$$

   Note the new gray-scale is assumed to be the same as the input image, i.e., $q_k = G - 1$ and $q_0 = 0$.

5. Rescan the image and write an output image with gray-levels q, setting

$$q = T[p]. \tag{81}$$

(d) Images before and after histoeqlization



(e) Hisograms

Example I

Example II

- The matlab script from `visionbook` is visionbook/05Preproc/hist_equal_demo.m.

Discussions

The following comments are from [Klette and Zamperoni, 1996, p. 148]:

- The gray value range equalization may be used for improving the image quality if the original image covers only a part of the full gray scale.
- An insufficient exploitation of the full gray scale is mostly due to image acquisition circumstances, as e.g., low scene illumination or automatic gain control of the camera.
- In case of good gray value dynamics of the input image, an equalization can lead even to quality losses in form of unsharp edges.

Example III

- Another histogram equalization example. The matlab script is ../program/histo_eq_demo.m.
- Some edges become unsharp.

12.2.4 Histogram Matching

Histogram Matching I

- `Histogram matching` is to produce an image with desired distributed brightness levels over the whole brightness scale.
- Assume the desired probability density function is $G(q)$.
- Let the desired pixel brightness transform be T.
- Similarly we have

$$F[p] = \int_{q_0}^{q=T[p]} G[s]\,ds \tag{82}$$

  where $F[p]$ is the cumulative histogram of the input image. Assumed that the cumulative histogram is normalized in $[0, 1]$.
- From the above equation, it is possible to find the transformation T.

Histogram Matching II

- E.g., if G is the exponential distribution,

$$G[q] = \alpha \mathbf{e}^{-\alpha(q-q_0)} \tag{83}$$

  for $q \geq q_0$.

- We have

$$F[p] = 1 - \mathbf{e}^{-\alpha(T[p]-q_0)} \tag{84}$$

- Then we find the transformation

$$T[p] = q_0 - \frac{1}{\alpha} \log(1 - F[p]) \tag{85}$$

- In the discrete case, the transformation can be implemented by building look-up tables.

Homework

Implementation of those transforms are left as homework. The homework includes:

1. Grey scale transformations with given functions such as exponent, logarithm, user defined (by some control points for a piece wise linear function), etc.;
2. Histogram stretching to a given range;
3. Histogram equalization;
4. Histogram matching: find other probability density functions in [Rongchun Zhao, 2000, p. 81]. Implemente the histogram matching transforms for those probability density functions.
5. ★ Histogram matching to the histogram from a given image.

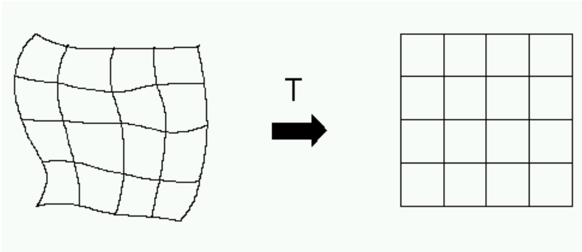
Note:

- Histograms should be plotted when histograms are involved.
- Your final score will also depend on the interface of your program.
- Be sure to give us a friendly interface.
- Be sure to choose illustrative images.

13 Geometric Transformations

Geometric Transformations

- Geometric transformations are common in computer graphics, and are often used in image analysis.
- Geometric transforms permit the elimination of 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:
  - matching remotely sensed images of the same area taken after one year, when the recent image is not taken from the same orientation and/or position.
  - correcting for document skew in OCR, which occurs when an image with an obvious orientation (e.g., a printed page) is scanned, or otherwise captured, at a different orientation.



Geometric Transformations

- A geometric transform is a vector function T that maps the pixel (x, y) to a new position (x', y'),

$$x' = T_x(x, y) \quad y' = T_y(x, y) \tag{86}$$

Geometric Transformations

- The transformation equations are
  - either known in advance,
  - or can be determined from known original and transformed images,
  - or can be estimated from known obvious orientations (e.g., OCR applications).
  - Several pixels in both images with known correspondence are used to derive the unknown transformation.

Geometric Transformations

- A geometric transform consists of two basic steps:
  1. determining the pixel co-ordinate transformation
     - mapping of 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.
  2. determining the brightness of the points in the digital grid.

Geometric Transformations

- The brightness values are usually computed as an interpolation of the brightnesses of several points in the neighborhood.
- This idea enables the classification of geometric transformation among other pre-processing techniques, the criterion being that only the neighborhood of a processed pixel is needed for the calculation.
- Geometric transformations are on the boundary between point and local operations.

Geometric Transformations: Areas of Images

- A geometric transformation applied to the whole image may change the co-ordinate system, and a Jacobean J provides information about how the co-ordinate system changes

$$J(x, y) = \frac{\partial(x', y')}{\partial(x, y)} \tag{87}$$

- The area of the image is invariant if and only if $|J| = 1$.

13.1 Pixel Co-ordinate Transformations

13.1.1 Polynomial Approximation

Polynomial Approximation

- The transformation is usually approximated by a polynomial equation (of degree m)

$$x' = \sum_{r=0}^{m} \sum_{k=0}^{m-r} a_{rk} x^r y^k \tag{88}$$

$$y' = \sum_{r=0}^{m} \sum_{k=0}^{m-r} b_{rk} x^r y^k. \tag{89}$$

- This transform is linear with respect to the coefficients a_{rk} and b_{rk}.
- If pairs of corresponding points (x, y), (x', y') in both images are known, it is possible to determine a_{rk} and b_{rk} by solving a set of linear equations.
- More points than coefficients are usually used to get robustness. The mean square method (least squares fitting) is often used.

Polynomial Approximation: Orders

- If the geometric transformation 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.
- The higher the degree of the approximating polynomial, the more sensitive to the distribution of the pairs of corresponding points the geometric transform.

13.1.2 Bilinear Transformations

Bilinear Transformations

- In practice, the geometric transform is often approximated by `bilinear transformations`:

$$x' = a_0 + a_1 x + a_2 y + a_3 xy, \tag{90}$$

$$y' = b_0 + b_1 x + b_2 y + b_3 xy. \tag{91}$$

- 4 pairs of corresponding points are sufficient to find transformation coefficients.

Affine Transformations

- Even simpler is the `affine transformation` for which three pairs of corresponding points are sufficient to find the coefficients:

$$x' = a_0 + a_1 x + a_2 y, \tag{92}$$

$$y' = b_0 + b_1 x + b_2 y. \tag{93}$$

- The affine transformation includes typical geometric transformations such as rotation, translation, scaling and skewing (shear).

13.1.3 Important Transformations

Rotations and Scaling

- Rotation by the angle ϕ

$$x' = x \cos \phi + y \sin \phi \quad (94)$$
$$y' = -x \sin \phi + y \cos \phi \quad (95)$$
$$J = 1 \quad (96)$$

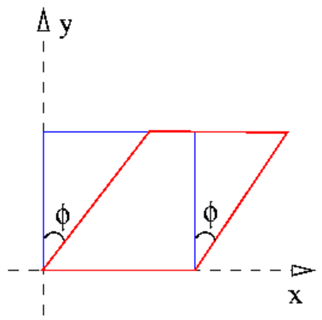
- Change of scale a in the x-axis and b in the y-axis

$$x' = ax \quad (97)$$
$$y' = by \quad (98)$$
$$J = ab \quad (99)$$

Skew

- Skew by the angel ϕ

$$x' = x + y \tan \phi \quad (100)$$
$$y' = y \quad (101)$$
$$J = 1 \quad (102)$$



Discussions

- It is possible to approximate complex geometric transformations (distortion) by partitioning an image into smaller rectangular sub-images.
- For each sub-image, a simple geometric transformation, such as the affine, is estimated using pairs of corresponding pixels.
- The geometric transformation (distortion) is then performed separately in each sub-image.

13.2 Brightness Interpolation

Brightness Interpolation

- Assume that the planar transformation has been accomplished, and new point co-ordinates (x', y') were obtained.

- The position of the pixel transformed does not in general fit the discrete grid of the output image.
- Values on the integer grid are needed.
- Each pixel value in the output image can be obtained by brightness interpolation of some neighboring non-integer samples, transformed from the input image.

Brightness Dual Interpolation

- 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.

Brightness Dual Interpolation

- Assume that we wish to compute the brightness value of the pixel (x', y') in the output image where x' and y' lie on the discrete grid.
- The co-ordinates of the point (x, y) in the original image can be obtained by inverting the transformation

$$(x, y) = T^{-1}(x', y'). \tag{103}$$

- In general the real co-ordinates (x, y) after inverse transformation do not fit the input image discrete grid, and so brightness is not known.

Brightness Interpolation or Image Resampling

- To get the brightness value of the point (x, y) the input image is re-sampled or interpolated:

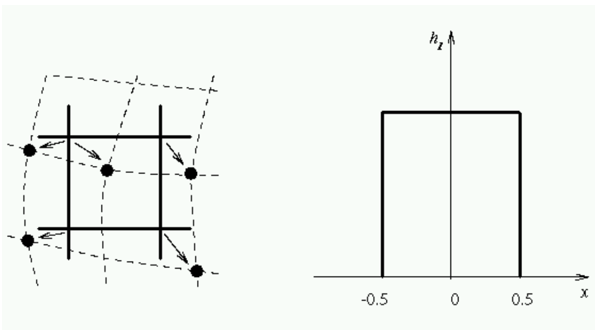
$$f_n(x, y) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} g_s(l\Delta x, k\Delta y) h_n(x - l\Delta x, y - k\Delta y) \tag{104}$$

  where $f_n(x, y)$ is the result of interpolation and h_n is the interpolation kernel. n distinguishes different interpolation methods.
- Usually, a small neighborhood is used, outside which h_n is zero.

13.2.1 Nearest Neighbor Interpolation

Nearest Neighbor Interpolation

- Assign to the point (x, y) the brightness value of the nearest point g in the discrete raster.



- The right side of the above figure shows how the new brightness is assigned.
- Dashed lines show how the inverse planar transformation maps the grids of the output image into the input image — solid lines show the grids of the input image.

Nearest Neighbor Interpolation

- Nearest neighbor interpolation is given by

$$f_1(x, y) = g_s(\text{round}(x), \text{round}(y)). \tag{105}$$

- The interpolation kernel h_1 as in (104) is

$$h_1(x, y) = h_1^1(x)h_1^1(y), \tag{106}$$

  where,

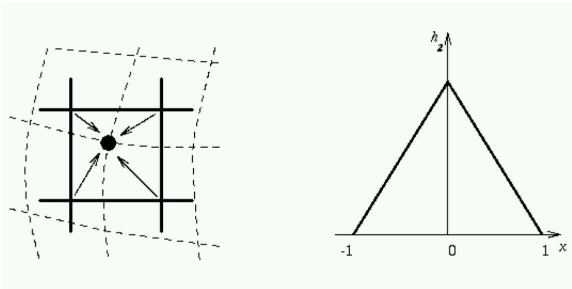
$$h_1^1(t) = \begin{cases} 1, & \text{if } t \in [-0.5, 0.5], \\ 0, & \text{otherwise.} \end{cases} \tag{107}$$

- 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.

13.2.2 Bilinear Interpolation

Bilinear Interpolation

- Bilinear interpolation explores four points neighboring the point (x, y), and assumes that the brightness function is bilinear in this neighborhood.



Bilinear Interpolation
Bilinear interpolation is given by

$$\begin{aligned} f_2(x, y) =& (1-a)(1-b)g_s(l, k) + a(1-b)g_s(l+1, k) \\ & (1-a)bg_s(l, k+1) + abg_s(l+1, k+1) \\ =& g_s(l, k) \\ & + (g_s(l+1, k) - g_s(l, k))a \\ & + (g_s(l, k+1) - g_s(l, k))b \\ & + (g_s(l, k) + g_s(l+1, k+1) - g_s(l+1, k) - g_s(l, k+1))ab, \end{aligned}$$

where

$$l = \text{floor}(x), \quad a = x - l, \tag{108}$$
$$k = \text{floor}(y), \quad b = y - k. \tag{109}$$

Proof of Bilinear Interpolation Formula

- By construction

$$x = a \cdot (l+1) + (1-a) \cdot l,$$
$$y = b \cdot (k+1) + (1-b) \cdot k.$$

- Since f_2 is bilinear,

$$f_2(x,k) = (1-a)g_s(l,k) + ag_s(l+1,k)$$
$$f_2(x,k+1) = (1-a)g_s(l,k+1) + ag_s(l+1,k+1).$$

- Then

$$\begin{aligned} f_2(x,y) =& bf_2(x,k+1) + (1-b)f_2(x,k) \\ =& b(1-a)g_s(l,k+1) + bag_s(l+1,k+1) \\ &+ (1-b)(1-a)g_s(l,k) + (1-b)ag_s(l+1,k). \end{aligned}$$

Bilinear Interpolation

- The interpolation kernel h_2 is

$$h_2(x,y) = h_2^1(x)h_2^1(y), \tag{110}$$

where

$$h_2^1(t) = h_1^1 * h_1^1(t) = \begin{cases} 1-t, & \text{if } t \in [0,1], \\ t+1, & \text{if } t \in [-1,0], \\ 0, & \text{otherwise.} \end{cases} \tag{111}$$

- Linear interpolation can cause a small decrease in resolution and blurring due to its averaging nature.
- The problem of step like straight boundaries with the nearest neighborhood interpolation is reduced.

13.2.3 Bi-cubic interpolation

Bi-cubic Interpolation

- Bi-cubic interpolation improves the model of the brightness function by approximating it locally by a bicubic polynomial surface;
- 16 neighboring points are used for interpolation.
- interpolation kernel ('Mexican hat') is defined via

$$h_3^1(t) = \begin{cases} 1 - 2|t|^2 + |t|^3, & \text{if } |t| < 1 \\ 4 - 8|t| + 5|t|^2 - |t|^3, & \text{if } 1 \le |t| < 2 \\ 0, & \text{otherwise} \end{cases} \tag{112}$$

by

$$h_3(x,y) = h_3^1(x)h_3^1(y). \tag{113}$$

Example

- The matlab script from `visionbook` is visionbook/05Preproc/imgeomt_demo.m.

Discussions

- Bicubic interpolation does not suffer from the step-like boundary problem of nearest neighborhood interpolation, and copes with linear interpolation blurring as well.
- Bicubic interpolation is often used in raster displays that enable zooming to an an arbitrary scale.
- Bicubic interpolation preserves fine details in the image very well.

14 Local pre-processing

Local Pre-processing

- 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 filtration (or filtering) if signal processing terminology is used.
- Local pre-processing methods can be divided into the two groups according to the goal of the processing:
  - `smoothing operators`
  - `gradient operators`

Smoothing Operators

- They aim to suppress noise or other small fluctuations in the image.
- Smoothing is equivalent to the suppression of high frequencies in the frequency domain.
- Unfortunately, smoothing also blurs all sharp edges that bear important information about the image.
- If objects are rather large, an image can be enhanced by smoothing of small degradations.
- Smoothing operators will benefit if some general knowledge about image degradation is available; this might, e.g., be statistical parameters of the noise.

Gradient Operators

- They are based on local derivatives of the image function.
- Derivatives are bigger at locations of the image where the image function undergoes rapid changes.
- The aim of gradient operators is to indicate such locations in the image.
- Gradient operators have a similar effect as suppressing low frequencies in the frequency domain.
- Noise is often high frequency in nature; unfortunately, if a gradient operator is applied to an image, the noise level increases simultaneously.

Local Pre-processing: Variants

- 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 nonlinear transformations can be distinguished.

Local Pre-processing: Convolution Mask I

- Linear operations 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.
- The contribution of the pixels in the neighborhood is weighted by coefficients h

$$f(i,j) = \sum_{(m,n)\in\mathcal{O}}\sum h(i-m, j-n)g(m,n) \tag{114}$$

- The above equation is equivalent to discrete convolution with the kernel h, that is called a `convolution mask`.

Local Pre-processing: Convolution Mask II

- Rectangular neighborhoods $\mathcal{O}$ are often used with an odd number of pixels in rows and columns, enabling the specification of the central pixel of the neighborhood.
- The choice of the local transformation, size, and shape of the neighborhood $\mathcal{O}$ depends strongly on the size of objects in the processed image.
- Convolution-based operators (filters) can be used for smoothing, gradient operators, and line detectors.
- There are methods that enable the speed-up of calculations to ease implementation in hardware — examples are recursive filters or separable filters.

Local Pre-processing: a priori Knowledge

- 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 $\mathcal{O}$ of the processed pixel is small.

14.1 Image smoothing

Image smoothing

- Image smoothing is to the suppression image noise — it uses redundancy in the image data.
- New pixel value is the averaging of brightness values in some neighborhood $\mathcal{O}$.
- Smoothing could blur sharp edges.
- There are smoothing methods which are edge preserving.
- The average is computed only from those pixels in the neighborhood which have similar properties to the pixel under processing.
- Local image smoothing can effectively eliminate impulsive noise or degradations appearing as thin stripes.
- It does not work well if degradations are large blobs or thick stripes.
- For complicated degradations, image restoration techniques in § **??** can be applied.

14.1.1 Averaging

Averaging with Images

- Assume that the noise ν at each pixel is an independent random variable with zero mean and standard deviation σ.
- E.g., the same image can be captured for the same static scene several times.
- The average of the same n images $g_1, \cdots, g_n$ with noise $\nu_1, \cdots, \nu_n$, is

$$\frac{g_1 + \cdots + g_n}{n} + \frac{\nu_1 + \cdots + \nu_n}{n}. \tag{115}$$

- The second term describes the effect of the noise,
  - it is again a random value with zero mean and standard deviation $\frac{\sigma}{\sqrt{n}}$.
- The resultant standard deviation is decreased by a factor $\sqrt{n}$.
- Thus if n images of the same scene are available, the smoothing can be accomplished without blurring the image by

$$f(i,j) = \frac{1}{n} \sum_{k=1}^{n} g_k(i,j) \tag{116}$$

Averaging with Pixels

- In many cases only one image with 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 assumption.

Averaging with Pixels

- Averaging is a special case of discrete convolution. For a 3×3 neighborhood, the convolution mask h is

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{117}$$

- The significance of the central pixel may be increased,

$$h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{118}$$

- Larger convolution masks for averaging can be created.

14.1.2 Averaging with Data Validity

Averaging with Data Validity

- Methods that average with limited data validity try to avoid blurring by averaging only those pixels which satisfy some criterion.
- They prevent involving pixels that are part of a separate feature.
- A very simple criterion is to use only pixels in the original image with brightness in a predefined interval $[\min, \max]$.
- For a pixel (m, n), the convolution mask is calculated in the neighborhood $\mathcal{O}$ from the nonlinear formula
$$h(i, j) = \begin{cases} 1, & \text{for } g(m+i, n+j) \in [\min, \max] \\ 0, & \text{otherwise.} \end{cases} \tag{119}$$
- The interval $[\min, \max]$ represents valid data.

Averaging with Data Validity: Variants

- The second method performs the averaging only if the computed brightness change of a pixel is in some predefined interval.
- This method permits repair to large-area errors resulting from slowly changing brightness of the background without affecting the rest of the image.
- The third method uses edge strength (i.e., magnitude of a gradient) as a criterion.
- The magnitude of some gradient operator is first computed for the entire image, and only pixels in the input image with a gradient magnitude smaller than a predefined threshold are used in averaging.

14.1.3 Averaging According to Inverse Gradient

Averaging According to Inverse Gradient

- The idea is that the brightness change within a region is usually smaller than between neighboring regions.
- Let (i, j) be the pixel under processing. The inverse gradient at the pixel (m, n) with respect to (i, j) is
$$\delta(i, j, m, n) = \begin{cases} \frac{1}{|g(m,n) - g(i,j)|}, & \text{if } g(m, n) \neq g(i, j); \\ 2 & \text{if } g(m, n) = g(i, j). \end{cases}$$
- The inverse gradient is in the interval $(0, 2]$, and is smaller at edges than in the interior of a homogeneous region.
- Weight coefficients in the convolution mask h are normalized by the inverse gradient,
$$h(i, j, m, n) = \frac{\delta(i, j, m, n)}{\sum_{(m', n') \in \mathcal{O}} \delta(i, j, m', n')}, \quad \text{if } (m, n) \in \mathcal{O}.$$

Averaging According to Inverse Gradient

- The above method assumes sharp edges.
- Isolated noise points within homogeneous regions have small values of the inverse gradient; points from the neighborhood take part in averaging and the noise is removed.
- When the convolution mask is close to an edge, pixels from the region have larger coefficients than pixels near the edge, and they is not blurred.

14.1.4 Averaging Using a Rotating Mask

Averaging Using a Rotating Mask

- This method avoids edge blurring by searching for the homogeneous part of the current pixel neighborhood.
- The resulting image is in fact sharpened.
- Brightness average is calculated only within the homogeneous region.
- A brightness dispersion σ^2 is used as the region homogeneity measure.
- Let n be the number of pixels in a region R and $g(i, j)$ be the input image. Dispersion σ^2 is calculated as

$$\sigma^2 = \frac{1}{n} \sum_{(i,j)\in R} \left[g(i,j) - \frac{1}{n} \sum_{(i',j')\in R} g(i',j') \right]^2 \tag{120}$$

Dispersion Calculation

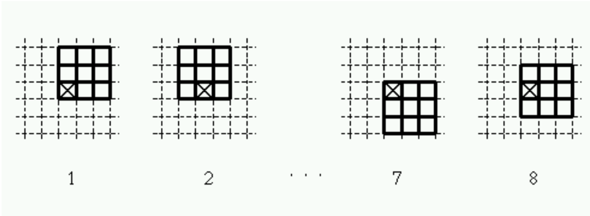
$$\begin{aligned}
&\sigma^2 \\
=&\frac{1}{n} \sum_{(i,j)\in R} \left\{ g(i,j)^2 - 2g(i,j) \frac{\sum_{(i',j')\in R} g(i',j')}{n} + \left[\frac{\sum_{(i',j')\in R} g(i',j')}{n} \right]^2 \right\} \\
=&\frac{1}{n} \left\{ \sum_{(i,j)\in R} g(i,j)^2 - 2 \frac{\left[\sum_{(i',j')\in R} g(i',j')\right]^2}{n}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

Rotating Mask

- Having computed region homogeneity, we consider its shape and size.
- The eight possible 3×3 masks that cover a 5×5 neighborhood of a current pixel (marked by small cross in the following figure) are shown in the following figure The ninth mask is the 3×3 neighborhood of the current pixel itself.

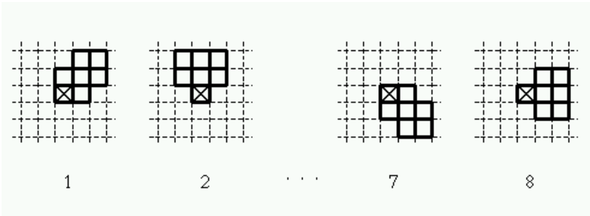
Algorithm: smoothing using a rotating mask

1. For each image pixel (i, j), calculate dispersion in the mask for all possible mask rotations about pixel (i, j).
2. Choose the mask with minimum dispersion.
3. Assign to the pixel (i, j) in the output image the average brightness in the chosen mask.



Other Masks

- Another set of eight masks covering a 5×5 neighborhood of the current pixel Again the ninth mask



  is the 3×3 neighborhood of the current pixel itself.

Other Masks

- Another possibility is to rotate a small 2×1 mask to cover the 3×3 neighborhood of the current pixel.
- This algorithm can be used iteratively. (What about other algorithms?)
- The iterative process converges quite quickly to stable state (that is, the image does not change any more).

Example

- The matlab script from `visionbook` is is visionbook/05Preproc/rotmask_demo.m.

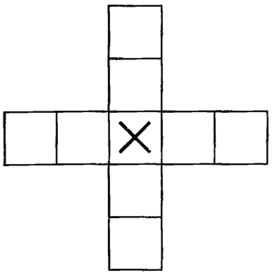
14.1.5 Median Filtering

Median Filtering

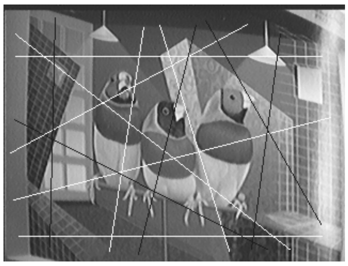
- In a set of ordered values, the median is the central value.
- Median filtering assigns the output pixel brightness value to be the median value of the brightness values in a neighborhood of the pixel.
- Median filtering reduces blurring of edges.
- The median of the brightness in the neighborhood is not affected by individual noise spikes and so median smoothing eliminates impulsive noise quite well.
- As median filtering does not blur edges much, it can be applied iteratively.

Median Filtering

- 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.
- Variants of median filtering is to choose the maximum and minimum values in the neighborhood.
- This leads to the dilation and erosion operators in mathematical morphology.

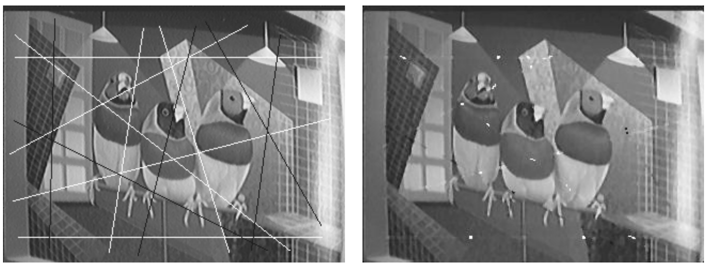


Example



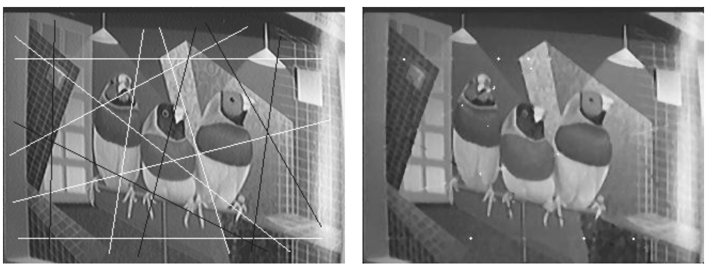
(f) Original

Figure 12: Iterative application of median filter in 3×3 neighborhoods.



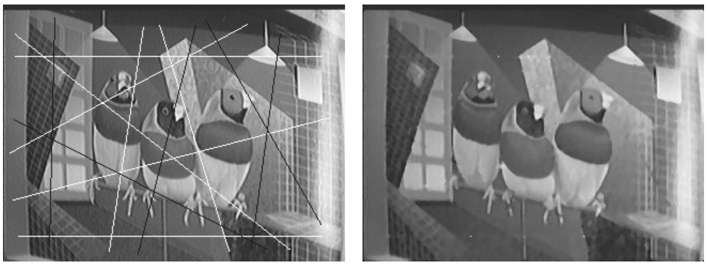
(a) Original (b) 1st median filtering

Figure 13: Iterative application of median filter in 3×3 neighborhoods.



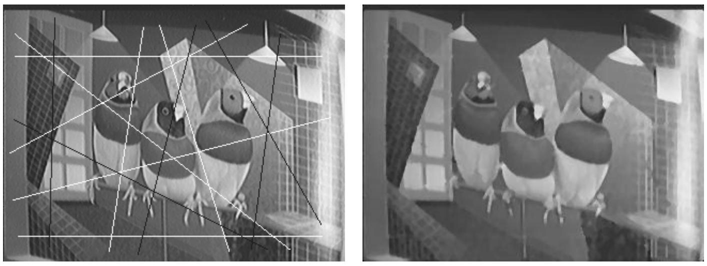
(a) Original (b) 2nd median filtering

Figure 14: Iterative application of median filter in 3×3 neighborhoods.



(a) Original (b) 3rd median filtering

Figure 15: Iterative application of median filter in 3×3 neighborhoods.



(a) Original (b) 4th median filtering

Figure 16: Iterative application of median filter in 3×3 neighborhoods.

Example II

- The matlab script from `visionbook` is is visionbook/05Preproc/medfilt_demo.

14.1.6 Non-linear Mean Filtering

Non-linear Mean Filtering

- The non-linear mean filter is another generalization of averaging techniques.
- It is defined by

$$f(i,j) = u^{-1}\left(\frac{\sum_{(m,n)\in\mathcal{O}} a(m,n)u[g(m,n)]}{\sum_{(m,n)\in\mathcal{O}} a(m,n)}\right) \tag{121}$$

  - $f(i,j)$ result of the filtering,
  - $g(m,n)$ pixel in the input image,
  - $\mathcal{O}$ local neighborhood of the current pixel (i,j).
  - u^{-1} the inverse of the function u,
  - $a(m,n)$ weight coefficients.

Homomorphic Filters

- If the coefficients $a(i,j)$ are constants, the filter is called homomorphic.
- Some homomorphic filters used in image processing are
  - Arithmetic mean, $u(g) = g$.
  - Harmonic mean, $u(g) = \frac{1}{g}$.
  - Geometric mean, $u(g) = \log g$.

14.1.7 Variational Properties of Smoothing Operators

Theorem

Theorem 14.1. *Given $x_1 \le x_2 \le \cdots \le x_N$, then*

a $\arg\min_a \sum_{i=1}^N |x_i - a|^2$ *is the arithmetic mean of* $x_1, x_2, \cdots, x_N$*;*

b $\arg\min_a \sum_{i=1}^N |x_i - a|$ *is the median of* $x_1, x_2, \cdots, x_N$*;*

c $\arg\min_a \max_{1\le i\le N} |x_i - a|$ *is* $\frac{x_1+x_N}{2}$.

Proof of (a)

- Let

$$g(a) = \sum_{i=1}^N |x_i - a|^2. \tag{122}$$

The result follows immediately by calculus.

Proof of (b) I

- Let

$$g(a) = \sum_{i=1}^N |x_i - a|. \tag{123}$$

- Let a_0 and a_1 be arbitrary numbers such that

$$a_0 < x_1 \le x_N < a_1. \tag{124}$$

- Then

$$g(a_0) = \sum_{i=1}^N x_i - Na_0, \tag{125}$$

$$g(a_1) = Na_1 - \sum_{i=1}^N x_i. \tag{126}$$

- If $a \in [x_1, x_N]$, assume that

$$x_k \le a < x_{k+1}. \tag{127}$$

- Then

$$g(a) = \sum_{i=1}^k (a - x_i) + \sum_{i=k+1}^N (x_i - a) \tag{128}$$

$$= \sum_{i=k+1}^N x_i - \sum_{i=1}^k x_i + (2k - N)a. \tag{129}$$

Proof of (b) II

- If $2k \le N$,

$$\begin{aligned} g(a_0) - g(a) &= 2\sum_{i=1}^k x_i - Na_0 + (N - 2k)a \\ &= 2\sum_{i=1}^k (x_i - a_0) + (N - 2k)(a - a_0) \ge 0. \end{aligned}$$

- If $2k \geq N$,

$$
\begin{aligned}
g(a_1) - g(a) &= Na_1 - 2\sum_{i=k+1}^{N} x_i + (N-2k)a \\
&= 2\sum_{i=k+1}^{N}(a_1 - x_i) + (N - 2(N-k))a_1 + (N-2k)a \\
&= 2\sum_{i=k+1}^{N}(a_1 - x_i) + (2k-N)(a_1 - a) \geq 0.
\end{aligned}
$$

Proof of (b) II

- Therefore, the minimum of $g(a)$ is attained in $[x_1, x_N]$.
  - $g(a)$ is continuous, piece wise linear function on $[x_1, x_N]$.
  - $g(a)$ is decreasing if $2k \leq N$ and increasing if $2k \geq N$.

- If $N = 2m$ is even, the minimum of $g(a)$ is attained in the central sub-interval $[x_m, x_{m+1}]$.
  - $g(a)$ is constant on this sub-interval.
  - Any value of $[x_m, x_{m+1}]$ is a minimizer of $g(a)$.
  - $a = \frac{x_m + x_{m+1}}{2}$ the conventional median in this case.

- If $N = 2m+1$, $g(a)$ is decreasing in one of the central sub-interval $[x_m, x_{m+1}]$ and increasing in another central sub-interval $[x_{m+1}, x_{m+2}]$.
  - Then x_{m+1} is the minimizer, which is the median in this case.

Proof of (c)

- Let

$$
g(a) = \max_{1 \leq i \leq N} |x_i - a|. \tag{130}
$$

- Then it is easy to verify that

$$
g(a) = \begin{cases} x_N - a, & \text{if } a < x_1; \\ x_N - a, & \text{if } x_1 \leq a < \frac{x_1 + x_N}{2}; \\ a - x_1, & \text{if } \frac{x_1 + x_N}{2} \leq a \leq x_N; \\ a - x_1, & \text{if } x_N < a. \end{cases} \tag{131}
$$

  The conclusion follows immediately.

14.2 Edge detectors

14.2.1 Edge: what is it?

Edge

- Edge detectors are important local image pre-processing methods to locate (sharp) changes in images.
- Edges are pixels where the brightness function changes abruptly.
- Neurological and psychophysical study suggests that locations in the image in which the inensity value changes abruptly are important for image perception.
- Edges are to a certain degree invariant to changes of illumination and viewpoint.

Edge

- If only edge elements with strong magnitude (edgels) are considered, such information often suffices for image understanding.

Figure 17: Siesta by Pablo Picasso, 1919

.

- 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 by Gradient

- Calculus describes changes of continuous functions using derivatives.
- An image function depends on two variables — so operators describing edges are expressed using partial derivatives.
- A change of the image function can be described by a gradient that points in the direction of the largest growth of the image function.

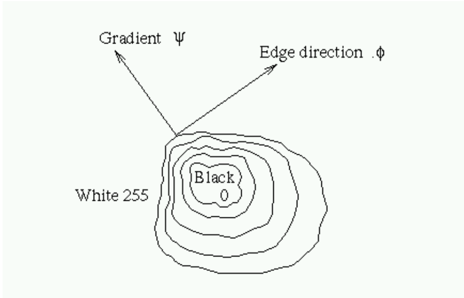
- An edge is a (local) property attached to an individual pixel and is calculated from the image function in a neighborhood of the pixel.
- It is a `vector variable` with two components
  - magnitude of the gradient;
  - and direction ϕ is rotated with respect to the gradient direction ψ by -90^o.

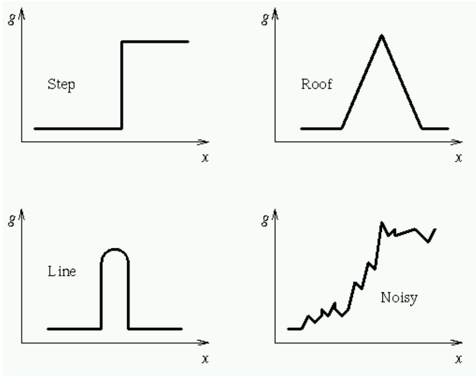
Edge and Gradient

- The gradient direction gives the direction of maximal growth of the function, e.g., from black ($g(i,j) = 0$) to white ($f(i,j) = 255$).
- This is illustrated below; closed contour lines are lines of the same brightness; the orientation 0^o points East.

Edges and Boundaries

- Edges are often used in image analysis for finding region boundaries.
- Boundary is at the pixels where the image function varies and consists of pixels with `high edge magnitude`.
- Boundary and its parts (edges) are perpendicular to the direction of the gradient.





Typical edge profiles.

Some Edge Profiles

- The following figure shows several typical standard edge profiles.
- Roof edges are typical for objects corresponding to thin lines in the image.
- Edge detectors are usually tuned for some type of edge profile.

Laplacian as an Edge Detector

- Sometimes we are interested only in changing magnitude without regard to the changing orientation.
- A linear differential operator called the Laplacian may be used.
- The Laplacian has the same properties in all directions and is therefore invariant to rotation in the image.

$$\Delta g(x, y) = \frac{\partial^2 g(x, y)}{\partial x^2} + \frac{\partial^2 g(x, y)}{\partial y^2} \tag{132}$$

14.2.2 Finite Gradient

Gradient

- The gradient magnitude and gradient direction are image functions,

$$|\mathrm{grad} g(x, y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2} \tag{133}$$

$$\psi = \arg(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}) \tag{134}$$

where $\arg(u, v) = \arctan(\frac{v}{u})$ is the angle (in radians) from the x-axis to the point (u, v).

- In practice, for fast computation, the magnitude is approximated by

$$|\text{grad}g(x,y)| = |\frac{\partial g}{\partial x}| + |\frac{\partial g}{\partial y}| \tag{135}$$

or

$$|\text{grad}g(x,y)| = \max\{|\frac{\partial g}{\partial x}|, |\frac{\partial g}{\partial y}|\} \tag{136}$$

Gradient by Finite Differences

- Derivatives must be approximated by finite differences for digital images.
- The first order differences of the image g can be approximated by backward difference

$$\Delta_i g(i,j) = \frac{g(i,j) - g(i-n,j)}{n} \tag{137}$$

$$\Delta_j g(i,j) = \frac{g(i,j) - g(i,j-n)}{n} \tag{138}$$

or by forward difference

$$\Delta_i g(i,j) = \frac{g(i+n,j) - g(i,j)}{n} \tag{139}$$

$$\Delta_j g(i,j) = \frac{g(i,j+n) - g(i,j)}{n} \tag{140}$$

- n is a small integer, usually 1.
- The value n should be chosen small enough to provide a good approximation to the derivative, but large enough to neglect unimportant changes in the image function.
- Central differences, are not usually used because they neglect the impact of the pixel (i,j) itself.

Gradient by Convolution

- Individual gradient operators that examine small local neighborhoods are in fact linear space-invariant operators.
- They are hence are equivalent to convolutions, cf. (114), and can be expressed by convolution masks.
- Each convolution mask corresponds to a derivative in one certain direction, if the masks induces edge orientation information.

14.2.3 More on Gradient Operators

Gradient Operators

- Gradient operators as a measure of edge can be divided into three categories
  1. Operators approximating derivatives using finite differences:
     - Some are rotationally invariant (e.g., the Laplacian) and direction independent and thus need one convolution mask only.
     - Others approximate first derivatives using several masks.
  2. Operators based on the zero crossings of the second derivatives (e.g., Marr-Hildreth or Canny edge detector).
  3. Operators which attempt to match an image function to a parametric model of edges.
- It may be difficult to select the optimal edge detection strategy.

Gradient Operators with Multiple Masks

- The direction of the gradient is given by the mask giving maximal response.
- The absolute value of the response on that mask is the magnitude.
- Operators approximating first derivative of an image function are sometimes called `compass operators` because of its ability to determine gradient direction.

14.2.4 Roberts operator

Roberts Operator

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

$$h_1 = \begin{bmatrix} h_1(0,0) & h_1(1,0) \\ h_1(0,1) & h_1(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{141}$$

- The magnitude of the edge is computed as

$$|g(i,j) - g(i+1,j+1)| + |g(i,j+1) - g(i+1,j)| \tag{142}$$

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

14.2.5 Discrete Laplacian

Discrete Laplacian I

- The Laplace operator is a very popular operator approximating the second derivative which gives the gradient magnitude only.
- The Laplacian (132) is approximated in digital images by a convolution sum.
- A 3×3 mask h_4 is often used.
- For 4-neighborhoods, it is defined as

$$h_{4,1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \text{or} \qquad h_{4,2} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{143}$$

Discrete Laplacian II

- For 8-neighborhoods, it is defined as

$$h_{8,1} = \frac{h_{4,1} + 2h_{4,2}}{3} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{144}$$

or

$$h_{8,2} = \frac{4h_{4,2} - h_{4,1}}{3} = \frac{1}{3}\begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix} \tag{145}$$

or

$$h_{8,3} = 3h_{4,1} - 2h_{4,2} = \begin{bmatrix} -1 & 3 & -1 \\ 3 & -8 & 3 \\ -1 & 3 & -1 \end{bmatrix} \tag{146}$$

- A Laplacian operator with stressed significance of the the central pixel or its neighborhood is sometimes used.

14.2.6 Prewitt Operator

Prewitt Operator

- The gradient is estimated in eight (for a 3×3 convolution mask) possible directions.
- Larger masks are possible.
- We present only the first three 3×3 masks for each operator; the others can be created by simple repeated clockwise 45^o rotation.

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

14.2.7 Sobel Operator

Sobel Operator

- $$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$
$$h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- The Sobel operator is often used as a simple detector of horizontality and verticality of edges. In this case only masks h_1 and h_3 are used.

14.2.8 Robinson Operator

Robinson Operator

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

14.2.9 Kirsch Operator

Kirsch Operator

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

Discussions

- Visually, the edge images produced by the foregoing edge operators (Roberts, Prewitt, Sobel, Kirsch, Robinson operators) appears rather similar.
- The Roberts operator, being 2×2, responds best on sharp transitions in low-noise images.
- The other operators, being 3×3, handle more gradual transition and noisier images better.

14.2.10 Image Sharpening

Image Sharpening

- Image sharpening makes edges steeper.
- The sharpened output image f is obtained from the input image g as

$$f(i,j) = g(i,j) - CS(i,j) \tag{147}$$

- C is a positive coefficient which gives the strength of sharpening and $S(i,j)$ is a measure of the image function sheerness that is calculated using a gradient operator.
- The Laplacian is very often used to estimate $S(i,j)$.
- Image sharpening/edge detection can be interpreted in the frequency domain as well.
- The result of the Fourier transform is a combination of harmonic functions.
- The derivative of the harmonic function $\sin(nx)$ is $n\cos(nx)$; thus the higher the frequency, the higher the magnitude of its derivative.

Image Sharpening: 1D illustration

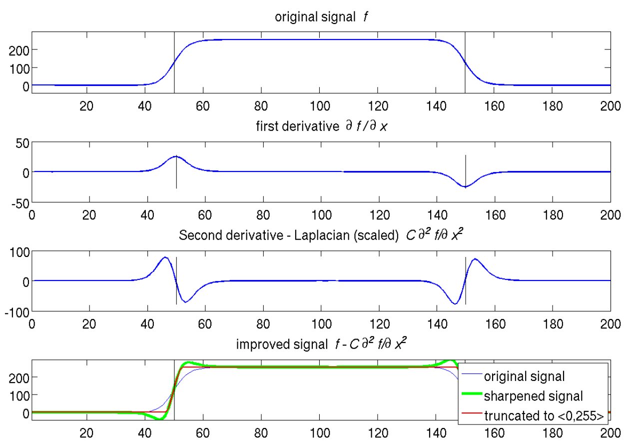


Figure 18: Image Sharpenging

Image Sharpening: Example I

Image Sharpening: Example II

- The matlab script from `visionbook` is is visionbook/05Preproc/imsharpen_demo.m.

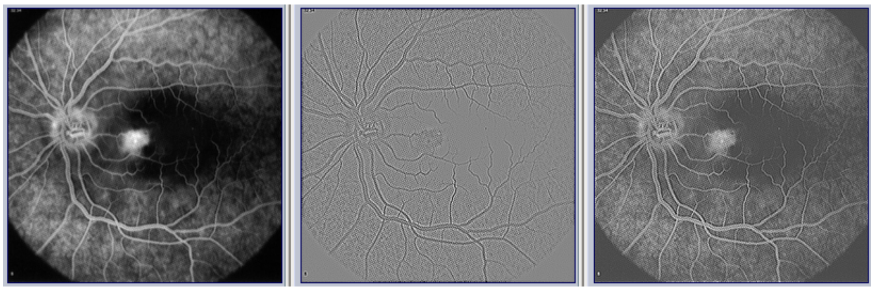


Figure 19: Image Sharpenging

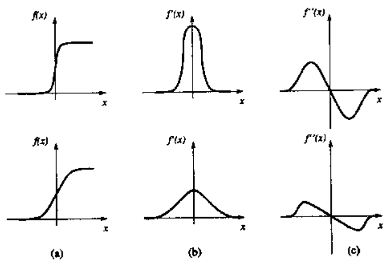
14.3 Zero-crossings of the Second Derivative

Marr's Theory

- In the 1970's, Marr's theory conclude from neurophysiological experiments that object boundaries are the most important cues that link an intensity image with its interpretation.
- Edge detection techniques at that time like the Kirsch, Sobel, Prewitt 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.

Zero-crossings

- An edge detection technique, based on the `zero crossings` of the second derivative (in its original form, the `Marr-Hildreth edge detector`) 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 extreme at the position corresponding to the edge in the image;
  - the second derivative should be zero at the same position.
- It is much easier and more precise to find a zero crossing position than an extreme - see the follwoing figure.



Smoothing Filter

- The crucial question is how to compute the the 2nd derivative robustly.
- 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, [Marr and Hildreth, 1980].

1. 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.
2. The constraint of spatial localization requires the response of a filter to be from nearby points in the image.

Uncertainty Principle

- These two criteria are conflicting — `uncertainty principle`.
- *A nonzero function and its Fourier transform cannot both be sharply localized*, [Folland and Sitaram, 1997, p. 207].
- But they can be optimized simultaneously using a Gaussian distribution.

Uncertainty Principle

- ([Folland and Sitaram, 1997, Theorem 1.1]) For any $f \in L^2(\mathbf{R})$ and any $a \in \mathbf{R}$ and $b \in \mathbf{R}$,

$$\int (x-a)^2 |f(x)|^2 \, dx \int (\xi - b)^2 |\hat{f}(\xi)|^2 \, d\xi \geq \frac{\|f\|_2^4}{16\pi^2}. \tag{148}$$

Equality holds if and only if $f = c\mathbf{e}^{2\pi i bx - \gamma(x-a)^2}$ for some $c \in \mathbf{C}$ and $\gamma > 0$.

Uncertainty Principle

- [Folland and Sitaram, 1997, Theorem 7.6] For a, $b > 0$, let $E(a, b)$ be the space of all measurable functions f on $\mathbf{R}$ such that

$$|f(x)| \leq c\mathbf{e}^{-a\pi x^2}, \tag{149}$$

$$|\hat{f}(\xi)| \leq c\mathbf{e}^{-b\pi\xi^2}, \tag{150}$$

for some $c > 0$. Then

(1) If $ab < 1$, $\dim E(a, b) = \infty$;
(2) If $ab = 1$, $E(a, b) = \mathbf{C}\mathbf{e}^{-a\pi x^2}$;
(3) If $ab > 1$, $E(a, b) = \{0\}$;

Uncertainty Principle

There is a well known joke:

- "Heisenberg is pulled over by a policeman whilst driving down a motorway, the policeman gets out of his car, walks towards Heisenberg's window and motions with his hand for Heisenberg to wind the window down, which he does. The policeman then says 'Do you know what speed you were driving at sir?', to which Heisenberg responds 'No, but I knew exactly where I was."'[2]

2D Gaussian Smoothing

- The `2D Gaussian smoothing operator` $G(x, y)$

$$G(x, y) = \frac{1}{2\pi\sigma^2} \mathbf{e}^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{151}$$

where x and y are the image co-ordinates and σ is the standard deviation of the associated probability distribution.

- The standard deviation σ is the only parameter of the Gaussian filter — it is proportional to the size of neighborhood on which the filter operates.
- Pixels more distant from the center of the operator have smaller influence, and pixels further than 3σ from the center have negligible influence.

[2]This is copy-edited from http://en.wikipedia.org/wiki/Uncertainty_principle.

LoG: Laplacian of Gaussian

- Our goal is to get a second derivative of a smoothed 2D function $f(x, y)$.
- The Laplacian operator gives the second derivative, and is non-directional (isotropic).

- Consider then the Laplacian of an image $f(x, y)$ smoothed by a Gaussian.
- This operator is abbreviated by some authors as `LoG`, from `Laplacian of Gaussian`:

$$\Delta[G(x, y) * f(x, y)] \tag{152}$$

- The order of differentiation and convolution can be interchanged due to linearity of the operations:

$$[\Delta G(x, y)] * f(x, y) \tag{153}$$

LoG: Laplacian of Gaussian

- The derivative of the Gaussian filter can be precomputed analytically

$$\Delta G(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^6} \mathbf{e}^{-\frac{r^2}{2\sigma^2}} \tag{154}$$

- Because its shape, the LoG operator is commonly called a Mexican hat.

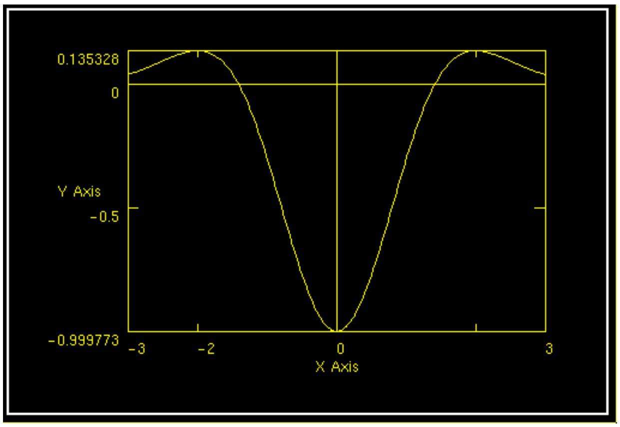


Figure 20: LoG filter.

LoG: Discussions

- Gaussian smoothing effectively suppresses the influence of the pixels that are up to a distance 3σ from the current pixel; then the Laplace operator is an efficient and stable measure of changes in the image.
- The location in the LoG image where the zero level is crossed corresponds to the position of the edges.
- The advantage of this approach compared to classical edge operators of small size is that a larger area surrounding the current pixel is taken into account; the influence of more distant points decreases according to the variance σ of the Gaussian.

LoG to DoG

- Convolution masks become large for larger σ.
- Fortunately, there is a separable decomposition of the ΔG operator that can speed up computation considerable.

- If only globally significant edges are required, the standard deviation of the Gaussian smoothing filter may be increased, having the effect of suppressing less significant evidence.
- The LoG operator can be very effectively approximated by convolution with a mask that is the difference of two Gaussian averaging masks with substantially different — this method is called the `Difference of Gaussians` — `DoG`.
- Even coarser approximations to LoG are sometimes used — the image is filtered twice by an averaging operator with smoothing masks of different size and the difference image is produced.

Implementation

- When implementing a zero-crossing edge detector, trying to detect zeros in the LoG or DoG image will inevitably fail,
- while naive approaches of thresholding the LoG/DoG image and defining the zero-crossings in some interval of values close to zero give piecewise disconnected edges at best.

- Many other approaches improving zero-crossing performance can be found in the literature.

Disadvantages

- The traditional second-derivative zero-crossing has disadvantages as well:
  - it smooths the shape too much; for example, sharp corners are lost.
  - it tends to create closed loops of edges (nicknamed the 'plate of spaghetti' effect).

- Neurophysiological experiments provide evidence that the human retina operation on image can be described analytically as the convolution of the image with the ΔG operator.

14.4 Scale in Image Processing

Scale and Object Size

- Many image processing techniques work locally, theoretically at the level of individual pixels — edge detection methods are examples.
- The essential problem in such computation is `scale`.

- Edges correspond to the gradient of the image function that is computed as a difference among pixels in some neighborhood.
- There is seldom a sound reason for choosing a particular size of neighborhood:
  - The 'right' size depends on the size of the objects under investigation.
  - To know what the objects are assumes that it is clear how to interpret an image and this is not in general known at the pre-processing stage.

Scale and Representations

- The phenomenon under investigation can be expressed at different resolutions of the description, and a formal model is created at each resolution.
- Then the qualitative behavior of the model is studied under changing resolution of the description.
- Such a methodology enables the deduction of meta-knowledge about the phenomenon that is not seen at the individual description.
- Different description levels are easily interpreted as different `scales` in the domain of digital images.

Scale and Marr's Edge Detection

- The idea of scale is fundamental to Marr's edge detection technique, introduced in the previous sub-section, where different scales are provided by different sizes of Gaussian filter masks.
- The aim there was not only to eliminate fine scale noise but also to separate events at different scales arising from distinct physical processes [Marr, 1982].

Representation by Scales

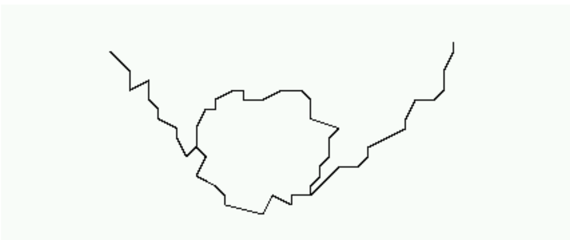
- Assume that a signal has been smoothed with several masks of variable sizes.
- Every setting of the scale parameters implies a different description, but it is not known which one is correct.
- For many tasks, no one scale is categorically correct.
- If the ambiguity introduced by the scale is inescapable, the goal of scale-independent description is to reduce this ambiguity as much as possible.

Scale Space

- Many publications tackle `scale-space` problems.
- Here we shall consider just three examples of the application of multiple scale description to image analysis.
- There are other approaches involving non-linear partial differential equations to generate non-linear scale-space descriptions.

Curve Example I

- The first approach aimed to process planar noisy curves at a range of scales — the segment of curve that represents the underlying structure of the scene needs to be found.
- The problem is illustrated in by an example of two noisy curves in the following figure.



Curve Example II

- One of these may be interpreted as a closed (perhaps circular) curve, while the other could be described as two intersecting straight lines.
- Local tangent direction and curvature of the curve are significant only with some scales after the curve is smoothed by Gaussian filter with varying standard deviations.
- After smoothing using the Gaussian filter with varying standard deviations, the significant segments of the original curve can be found.

Scale Space Filtering I

- The second approach, `scale space filtering`, is to describe signals qualitatively with respect to scale.
- The problem is formulated for 1D signals $f(x)$, but it could be generalized to images.
- The original 1D signal f(x) is smoothed by convolution with a 1D Gaussian

$$f(x, \sigma) = G(x, \sigma) * f(x) \tag{155}$$

- If σ is changed, the function $f(x, \sigma)$ represents a surface on the (x, σ) plane that is called the `scale-space image`.
- Inflection points of the curve $f(x, \sigma_0)$ for a distinct value σ_0

$$\frac{\partial^2 f(x, \sigma_0)}{\partial x^2} = 0, \qquad \frac{\partial^3 f(x, \sigma_0)}{\partial x^3} \neq 0. \tag{156}$$

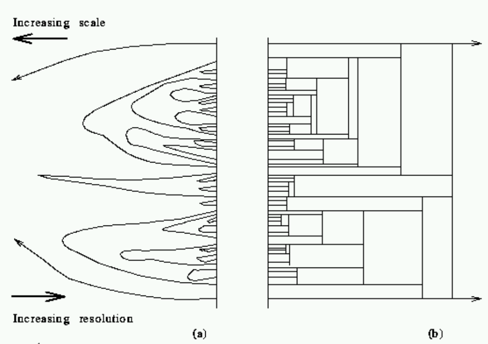
  describe the curve $f(x)$ qualitatively.

Scale Space Filtering II

- The positions of inflection points can be drawn as a set of curves

$$\Sigma(x, \sigma_0) \tag{157}$$

  in (x, σ) co-ordinates



Scale-space image:
(a) Varying number and locations of curve segmentation points as a function of scale,
(b) curve representation by an interval tree.

Scale Space Filtering III

- Coarse to fine analysis of the curves corresponding to inflection points, i.e., in the direction of the decreasing value of the σ, localizes events at different scales.
- The qualitative information contained in the scale-space image can be transformed into a simple `interval tree` that expresses the structure of the signal $f(x)$ over all (observed) scales.
- The interval tree is built from the root that corresponds to the largest scale.
- Then the scale-space image is searched in the direction of decreasing σ.
- The interval tree branches at those points where new curves corresponding inflection points appears.

Canny Edge Detector

- The third example of the application of scale — Canny edge detector, discussed in the next subsection.

14.5 Canny Edge Detection

Canny Edge Detector

- `Canny edge detector` is optimal for `step edges` corrupted by white noise.
- The optimality of it is related to three criteria:
  - The `detection criterion` expresses that fact that important edges should not be missed, and that there should be no spurious responses.
  - The `localization criterion` requires 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 (also partly covered by the first criterion, since when there are two responses to a single edge one of them should be considered as false).

Canny Edge Detector in 1D

- Canny's edge detector is based on several ideas:
  1. The edge detector was expressed for a 1D 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 by the first derivative of a Gaussian smoothing filter with standard deviation σ;
     - the reason for doing this is for an effective implementation.
     - There is a strong similarity here to the Marr-Hildreth edge detector (Laplacian of a Gaussian).

Canny Edge Detector in 2D I

- The detector is then generalized to two dimension.
- 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 Laplacian filter.

Canny Edge Detector in 2D II

- Suppose G is a 2D Gaussian (151) and assume we wish to convolute the image with an operator $G_{\mathbf{n}}$ which is the first derivative of G in the direction $\mathbf{n}$.

$$G_{\mathbf{n}} = \frac{\partial G}{\partial \mathbf{n}} = \mathbf{n} \cdot \nabla G. \quad (158)$$

- The direction $\mathbf{n}$ should be oriented perpendicular to the edge
  - this direction is not known in advance
  - however, a robust estimate of it based on the smoothed gradient direction is available
  - if g is the image, the normal to the edge is estimated as

$$\mathbf{n} = \frac{\nabla(G * g)}{|\nabla(G * g)|}. \quad (159)$$

Canny Edge Detector in 2D III

- The edge location is then at the local maximum in the direction $\mathbf{n}$ of the operator $G_{\mathbf{n}}$ convoluted with the image g

$$\frac{\partial}{\partial \mathbf{n}} G_{\mathbf{n}} * g = 0. \quad (160)$$

- Substituting in (160) for $G_{\mathbf{n}}$ from equation (158), we get

$$\frac{\partial^2}{\partial \mathbf{n}^2} G * g = 0. \quad (161)$$

- This equation (161) shows how to find local maxima in the direction perpendicular to the edge;
- this operation is often referred to as `non-maximal suppression`.

Canny Edge Detector in 2D III

- As the convolution and derivative are associative operations in (161), we can
  - first convolute an image g with a symmetric Gaussian G;
  - then compute the directional second derivative using an estimate of the direction $\mathbf{n}$ computed according to equation (159).
- The strength of the edge (magnitude of the gradient of the image intensity function g) is measured as

$$|G_{\mathbf{n}} * g| = |\nabla(G * g)|. \quad (162)$$

Canny Edge Detector: Noise and Spurious Response

- Spurious responses to the single edge caused by noise usually create a so called 'streaking' problem that is very common in edge detection in general.
  - Output of an edge detector is usually thresholded to decide which edges are significant.
  - Streaking means breaking up of the edge contour caused by the operator fluctuating above and below the threshold.

Canny Edge Detector: Hysteresis Thresholding

- Streaking can be eliminated by `thresholding with hysteresis`.
  - If any edge response is above a high threshold, those pixels constitute definite output of the edge 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.
  - Please refer to Canny's original paper for detailed discussions.

Example: Hysteresis Thresholding

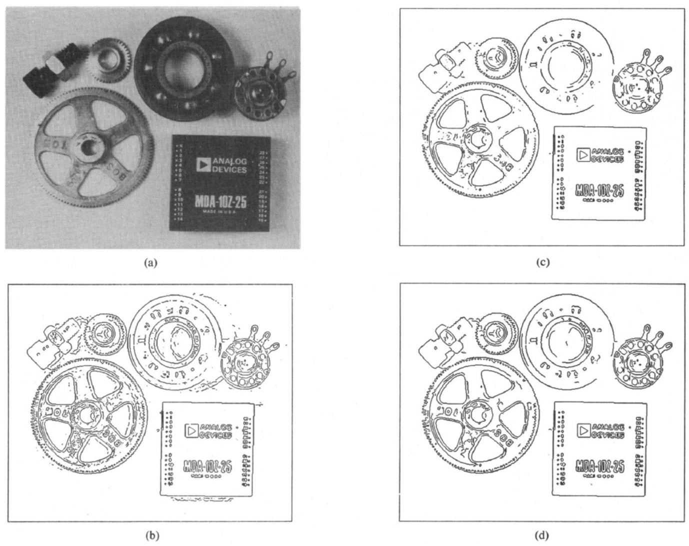


Figure 21: (a) Parts image, 576 by 454 pixels. (b) Image thesholded at T_1. (c) Image thresholded at $2T_1$. (d) Image thresholded with hysteresis using both the thresholds in (b) and (c). [Canny, 1986].

Canny Edge Detector with Scales

- 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 scale for the Canny detector is represented by different standard deviations σ 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 Edge Detector: Feature Synthesis

- Canny proposed a `Feature synthesis` approach.
- All significant edges from the operator with the smallest scale are marked first.

- Edges of a hypothetical operator with larger σ are synthesized from them (i.e., a prediction is made of how the larger σ should perform on the evidence gleaned from the smaller σ).
- Then the synthesized edge response is compared with the actual edge response for larger σ.
- Additional edges are marked only if they have significantly stronger response than that predicted from synthetic output.
- This procedure may be repeated for a sequence of scales, a cumulative edge map is built by adding those edges that were not identified at smaller scales.

Example: Feature Synthesis

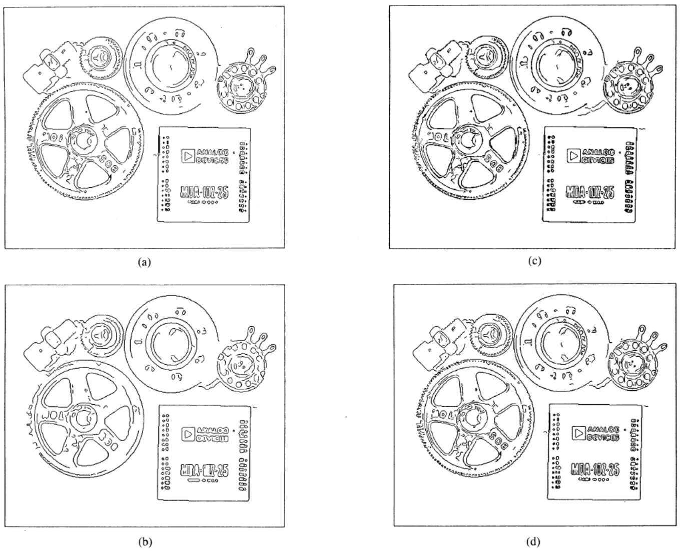


Figure 22: (a) Edges from parts image at $\sigma = 1.0$. (b) Edges at $\sigma = 2.0$. (c) Superposition of the edges. (d) Edges combined using feature synthesis. [Canny, 1986]

Canny Edge Detector

1. Repeat steps (2) till (6) for ascending values of the standard deviation.
2. Convolve an image g with a Gaussian of scale σ.
3. Estimate local edge normal directions $\mathbf{n}$ using equation (159) for each pixel in the image.
4. Find the location of the edges using equation (161) (non-maximal suppression).
5. Compute the magnitude of the edge using equation (162).
6. Threshold edges in the image with hysteresis to eliminate spurious responses.
7. Aggregate the final information about edges at multiple scale using the "feature synthesis" approach.

Discussions

- Canny's detector represents a complicated but major contribution to edge detection.
- Its full implementation is unusual, it being common to find implementations that omit feature synthesis — that is, just steps 2 — 6 of algorithm.
- Reference for this section is Canny's paper [Canny, 1986].

Example

- The matlab script file is ../program/canny.m.
- Compare the result with the edge detector with other filters.
- Note the improvement at the top part of the hat.
- Study the matlab code. Note the "thin" code.

Homework

- This is an important homework. The total score is 20. The homework includes:
  1. (20 points) Implement median filter with variable window size and shape;
  2. (20 points) Implement another local smoothing filter with possible variable parameters;
  3. (40) Implement Canny edge detector; what is your extension to step 7? (Without step 7, you can only get at most 30 points.).
  4. (30) Implement another edge detector and compare it with Canny detector.

14.6 Parametric Edge Models

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.
- While the continuous image function is not known, it can be estimated from the available discrete image intensity function and image properties can be determined from this estimate, possibly with sub-pixel precision.
- Piecewise continuous function estimate called `facets` are used to represent (a neighborhood) image pixel.
- Such image representation is called `facet model`.

Parametric Edge Models

- 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 applications is given in [Haralic and Shapiro, 1992].

Bi-cubic Facet Model I

- Consider a bi-cubic facet model

$$g(x, y) = c_1 + c_2 x + c_3 y + c_4 x^2 + c_5 xy + c_6 y^2 + c_7 x^3 + c_8 x^2 y + c_9 xy^2 + c_{10} y^3. \quad (163)$$

- The parameters are estimated from a pixel neighborhood(the central pixel is at $(0, 0)$).
- A least-squares method with singular-value decomposition may be used.

Bi-cubic Facet Model II

- 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 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.
- However, their computational requirements are much higher.

14.7 Edges in Multi-spectral Images

Edge Detection in Multi-spectral Images

- The first is to detect edges separately in individual image spectral components using the ordinary edge detectors.
  - Individual images of edges can be combined to get the resulting edge image;
  - The value corresponding to edge magnitude and direction is the maximal edge value from all spectral components.
  - A linear combination of edge spectral components can also be used, and other combination techniques are possible.
- A second possibility is to use the brightness difference of the same pixel in two different spectral components.
  - The ratio instead of the difference can also be used as well, although it is necessary to assume that pixel values are not zero in this case.

Multi-spectral Edge Detector

- A third possibility is to create a multi-spectral edge detector which uses brightness information from all n spectral bands.
  - An edge detector of this kind was proposed in [Cervenka and Charvat, 1987].
  - The neighborhood used has size $2 \times 2 \times n$ pixels, where the 2×2 neighborhood is similar to that of the Roberts gradient, (141).
  - The coefficients weighting the influence of the component pixels are similar to the correlation correlation coefficients.
  - Let $\bar{f}(i,j)$ denote the arithmetic mean of the brightness corresponding to the pixel with the same co-ordinates (i,j) in all n spectral component images and f_r be the brightness of the r^{th} spectral component.

Multi-spectral Edge Detector

- The edge detector result in pixel (i,j) is given as the minimum of the following expressions:

$$\frac{\sum_{r=1}^{n} d_r(i,j) d_r(i+1,j+1)}{\sqrt{\sum_{r=1}^{n} \left(d_r(i,j)\right)^2 \sum_{r=1}^{n} \left(d_r(i+1,j+1)\right)^2}} \times \frac{\sum_{r=1}^{n} d_r(i+1,j) d_r(i,j+1)}{\sqrt{\sum_{r=1}^{n} \left(d_r(i+1,j)\right)^2 \sum_{r=1}^{n} \left(d_r(i,j+1)\right)^2}}$$

  where $d_r(i,j) = f_r(i,j) - \bar{f}(i,j)$.

- This multi-spectral edge detector gives very good (?) results on remotely sensed images.

14.8 Line Detection by Local Pre-processing Operators

Other Local Operators

- Several other local operations exists which do not belong to the taxonomy in § 14, as they are used for different purposes.
- `Line finding`, `line thinning`, and `line filling` operators are among them.
- 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`.

Why?

- 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.

14.8.1 Line Detection

Line Detection I

- `Line finding operators` aim to find very thin curves in the image, e.g., roads in satellite images.
- It is assumed that curves do not bend sharply.
- Such curves and straight lines are called `lines` for the purpose of describing this technique.
- Lines are modeled by a roof profile among edges, Fig. 375.
- We assume that the width of the lines is approximately one or two pixels.
- Lines in the image can be detected by a number of local convolution operators h_k, which serve as line patterns [Cervenka and Charvat, 1987].
- The output value in pixel (i, j) is given by

$$f(i,j) = \max\left\{0, \max_k (f * h_k)\right\} . \tag{164}$$

Line Detection II

- Four masks of size 3×3, able to detect lines rotated modulo the angle 45^o:
- A similar construction can be applied to bigger masks.

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

$$h_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$h_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad h_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$h_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad h_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$h_7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad h_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Line Detection III

- Convolution masks of size 5×5; 14 orientations;
- Only the first eight are shown, as the others are obvious by rotation.
- Such line detectors may produce more lines than needed, and other non-linear constraints should be added to reduce this number.

14.8.2 Line Thinning

Line Thinning

- Thresholded edges are usually wider than one pixel, and line thinning techniques may give a better result.
- One line thinning method uses knowledge about orientation and in this case edges are thinned before thresholding.
  - Edge magnitude and directions provided by some gradient operator are used as input, and the edge magnitude of two neighboring pixels perpendicular to the edge direction are examined for each pixel in the image.
  - If at least one 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.
  - The technique is called `non-maximal suppression` and is similar to the idea in Canny edge detector.
- There are many line thinning methods which we do not present here.
- In most cases the best line thinning is achieved using mathematical morphology methods (?).

14.8.3 Edge Filling

Edge Filling

- Edge points after thresholding do not create contiguous boundaries and the edge filling method tries to recover edge pixels on the potential object boundary which are missing.

- The following is a very simple local edge filling technique.
- The local edge filling procedure [Cervenka and Charvat, 1987] checks the 3×3 neighborhood of the current pixel matches one of the following masks If so, the central pixel of the mask is changed

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

from zero to one.

Discussions

- These simple methods for edge thinning and filling do not guarantee that the width of the lines will be equal to one and the contiguity of the edges are is not certain either.
- Note that local thinning and filling operators can be treated as special cases of mathematical morphology operators.

14.9 Detection of Corners (interesting points)

Corresponding Points

- In many cases it is of advantage to find pairs of `corresponding points` in two similar images.
- E.g., when finding geometric transforms, knowing the position of corresponding points enables the estimation of geometric transforms from live data.
- Finding corresponding points is also a core problem in the analysis of moving images and for recovering depth information form pairs of stereo images.

Interest Points

- In general, all possible pairs of points should be examined to solve this `correspondence problem`, and this is very computation expensive.
- If two images have n pixels each, the complexity is $O(n^2)$.
- This process might be simplified if the correspondence is examined among a much smaller number of points, called `interest points`.
- An interest point should have some typical local property.
- E.g., if square objects are present in the image, then `corners` are very good interest points.

Corners

- Corners serve better than lines for the correspondence problem.
- 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. They provide ground for unique matching, cf. the following figure for illustration.

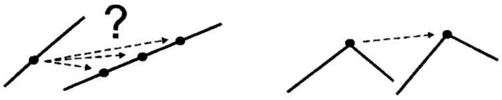


Figure 23: Ambiguity of lines for matching and unambiguity of corners.

Corner Detectors

- Edge detectors themselves are not stable at corners.
- This is natural as the gradient at the tip of the corner is ambiguous.
- Near the corner there is a discontinuity in the gradient direction.

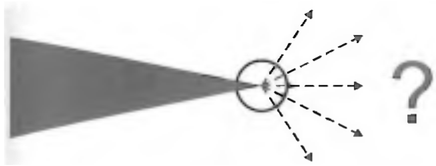


Figure 24: Ambiguity of lines for matching and unambiguity of corners.

- This observation is used in corner detectors.

Corners

- The corner in the image can be defined as a pixel in whose small neighborhood there 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 with 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

- 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 necessary for estimating a transformation sought between these images.

Moravec Detector

- The simplest corner detector is the `Moravec detector` which is maximal in pixels with high contrast.
- These points are on corners and sharp edges.
- The Moravec operator MO is given by

$$\mathrm{MO}(i,j) = \frac{1}{8} \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} |g(k,l) - g(i,j)|. \tag{165}$$

Corner Detectors by Facet Model

- Better results are based on the facet model, (163).
- The image g is approximated by a bi-cubic polynomial

$$g(x,y) = c_1 + c_2 x + c_3 y + c_4 x^2 + c_5 xy + c_6 y^2 + c_7 x^3 + c_8 x^2 y + c_9 x y^2 + c_{10} y^3. \quad (166)$$

- The Zuniga-Haralick operator ZH is given by

$$\mathrm{ZH}(i,j) = \frac{-\left(c_2^2 c_6 - 2c_2 c_3 c_5 + c_3^2 c_4\right)}{(c_2^2 + c_3^2)^{\frac{3}{2}}}, \quad (167)$$

  which is the curvature of $g(x,y) = \text{const}$.
- The Kitchen-Rosenfeld KR operator is given by

$$\mathrm{KR}(i,j) = \frac{-\left(c_2^2 c_6 - 2c_2 c_3 c_5 + c_3^2 c_4\right)}{c_2^2 + c_3^2}, \quad (168)$$

  which is the second order derivative along the direction of edge.
- The ZH operator has been shown to outperform the KR detector in test images.

Harris Corner Detector I

- The Harris corner detector improved upon Moravec's by considering the differential of the corner score (sum of square differences).
- Consider a 2D gray-scale image f.
- An image patch W in f is taken and is shifted by Δx and Δy.
- The sum S of square differences between values of the image f given by the patch W and its shifted variant by Δx and Δy is given by:

$$S_W(\Delta x, \Delta y) = \sum_{(x_i,y_i)\in W} \left(f(x_i,y_i) - f(x_i - \Delta x, y_i - \Delta y)\right)^2. \quad (169)$$

- A corner point not suffering from the aperture problem must have a high response of $S_W(\Delta x, \Delta y)$ for all Δx and Δy.

Harris Corner Detector II

- By the first-order Taylor expansion

$$f(x_i - \Delta x, y_i - \Delta y) = f(x_i,y_i) - \frac{\partial f}{\partial x}(x_i,y_i)\Delta x - \frac{\partial f}{\partial y}(x_i,y_i)\Delta y, \quad (170)$$

  the minimum of $S_W(\Delta x, \Delta y)$ can be obtained analytically.
- Substituting (170) into (169),

$$\begin{aligned} S_W(\Delta x, \Delta y) &= \sum_{(x_i,y_i)\in W} \left(\frac{\partial f}{\partial x}(x_i,y_i)\Delta x + \frac{\partial f}{\partial y}(x_i,y_i)\Delta y,\right)^2 \\ &= \begin{pmatrix}\Delta x & \Delta y\end{pmatrix} A_W(x,y) \begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix} \end{aligned}$$

  where the Harris matrix

$$A_W(x,y) = \sum_{(x_i,y_i)\in W} \begin{pmatrix} \left(\frac{\partial f}{\partial x}(x_i,y_i)\right)^2 & \frac{\partial f}{\partial x}(x_i,y_i)\frac{\partial f}{\partial y}(x_i,y_i) \\ \frac{\partial f}{\partial x}(x_i,y_i)\frac{\partial f}{\partial y}(x_i,y_i) & \left(\frac{\partial f}{\partial y}(x_i,y_i)\right)^2 \end{pmatrix}. \quad (171)$$

Harris Corner Detector III

- The Harris matrix A_W is symmetric and positive semi-definite.
- Its main modes of variation correspond to partial derivatives in orthogonal directions and are reflected in eigenvalues λ_1 and λ_2 of the matrix A_W.
- Three distinct cases can appear:
  1. Both eigenvalues are small. This means that image f is flat in the examined pixel. There are no edges or corners in this location.
  2. One eigenvalue is small and the second one large. The local neighborhood is ridge-shaped. Significant change of image f occurs if a small movement is made perpendicularly to the ridge.
  3. Both eigenvalues are rather large. A small shift in any direction causes significant change of the image f. A corner is found.

Harris Corner Detector IV

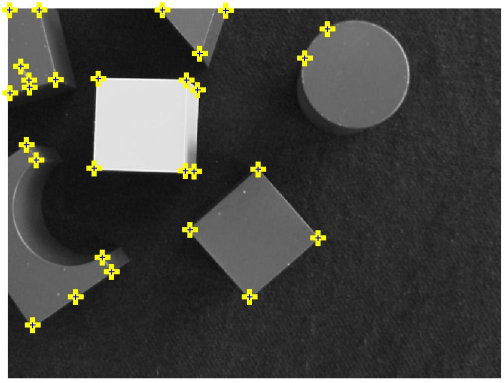
- Harris suggested that exact eigenvalue computation can be avoided by calculating the response function $R(A) = \det(A) - \kappa \text{trace}^2(A)$, where κ is a tunable parameter where values from 0.04 to 0.15 were reported in literature as appropriate.
- Harris corner detector
  1. Filter the image with a Gaussian.
  2. Estimate intensity gradient in two perpendicular directions for each pixel. This is performed by twice using a 1D convolution with the kernel approximating the derivative.
  3. For each pixel and a given neighborhood window:
     - Calculate the local structure matrix A;
     - Evaluate the response function $R(A)$.
  4. Choose the best candidates for corners by selecting a threshold on the response function $R(A)$ and perform non-maximal suppression.

Harris Corner Detector V

- The Harris corner detector has been very popular.
- Its advantages are insensitivity to 2D shift and rotation, to small illumination variations, to small viewpoint change, and its low computational requirements.
- On the other hand, it is not invariant to larger scale change, viewpoint changes and significant changes in contrast.
- Many more corner-like detectors exist, and the reader is referred to the overview papers.

Harris Corner Detector: Example

- Corner detection example with Harris corner detector.
- The matlab script from **visionbook** is visionbook/05Preproc/harris_demo.m.



References

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679 – 698.

[Cervenka and Charvat, 1987] Cervenka, V. and Charvat, K. (1987). Survey of the image processing research applicable to the thematic mapping based on aerocosmic data. Technical report, Geodetic and Carthographic Institute, Prague, Czechoslovakia. Technical Report A 12-346-811.

[D. R. Wilson and T. R. Martinez, 1997] D. R. Wilson and T. R. Martinez (1997). Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research.*

[Folland and Sitaram, 1997] Folland, G. B. and Sitaram, A. (1997). The uncertainty principle: A mathematical survey. *Journal of Fourier Analysis and Applications*, 3:207 – 238.

[Freeman, 1961] Freeman, H. (1961). On the envoding of arbitrary geometric confgiuration. *IRE Transactions on Electronic Computers*, EC-10(2):260 – 268.

[Frisby, 1979] Frisby, J. P. (1979). *Seeing — Illusion, Brain and Mind.* Oxford University Press, Oxford.

[Haralic and Shapiro, 1992] Haralic, R. M. and Shapiro, L. G. (1992). *Computer and Robot Vision.* Reading: Addision Wesley.

[Horn, 1986] Horn, B. R. (1986). *Robot Vision.* M.I.T. Press, Cambridge.

[Klette and Zamperoni, 1996] Klette, R. and Zamperoni, P. (1996). *Handbook of Image Processing Operators.* John Wiley & Sons, Inc, New York.

[Knuth, 1981] Knuth, D. E. (1981). *Seminumerical Algorithms.* Addison-Wesley, Reading, Massachusetts, 2nd edition. volume 2 of The art of Computer Programming.

[Kovalevski, 1989] Kovalevski, V. A. (1989). Finite topology as applied to image analysis. *Computer Vision, Graphics and Image Processing*, 46:141 – 161.

[Kutter, 1999] Kutter, M. (1999). *Digital Image Watermarking: hiding information in images.* ph.d dissertation, Départment d'Électricité, Ècole Polytechnique Fédérale de Lausanne.

[Marr, 1982] Marr, D. (1982). *Vision.* Freeman.

[Marr and Hildreth, 1980] Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B*, 207:187 – 217.

[Oppenheim et al., 1997] Oppenheim, A. V., Willsky, A. S., and Nawwab, S. H. (1997). *Signals and Systems.* Prentice Hall, Inc.

[Pavlidis, 1977] Pavlidis, T. (1977). *Structural Pattern Recognition.* Springer Verlag, Berlin, Heidelberg, New York.

[Pratt, 1978] Pratt, W. K. (1978). *Digital Image Processing.* John Wiley & Sons, Inc, New York.

[Press et al., 1992] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C: The Art of Scientifc Computing.* Cambridge University Press, 2nd edition.

[Rongchun Zhao, 2000] Rongchun Zhao, R. (2000). *Introduction to Digital Image Processing (in Chinese).* Northwestern Polytechnical University Press, Xi'An, Shan Xi Province.

[Russ, 1995] Russ, J. C. (1995). *The Image Processing Handbook, 2nd ed.* CRC Press.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379 – 656 and 623 – 656.

[Wandell, 1995] Wandell, B. (1995). *Foundation of Vision.* Sinauer Associates.

[Wang et al., 2004] Wang, Z., Bovik, A. C., Sheikh, H. R., Member, S., Simoncelli, E. P., and Member, S. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612.

[Wang and Wu, 1991] Wang, Z. J. and Wu, S. D. (1991). *Imaging Optics.* Science Press, Beijing. in Chinese.

[Zhao and Zhong, 1982] Zhao, K. and Zhong, X. (1982). *Optics.* Peking University Press, Beijing. 2 volumes, in Chinese.

Index