A **digital image** is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels. A digital image a[m,n] described in a 2D discrete space is derived from an analog image a(x,y) in a 2D continuous space through a sampling process that is frequently referred to as digitization

Fourier Transformation

- The Fourier transform decomposes a function of time (a signal) into the frequencies that make it up, similarly to how a musical chord can be expressed as the amplitude (or loudness) of its constituent notes.
- It states that any signal can be represented by the integral of sine and cosine multiplied by weighing function.
- In image processing, it is very important to decompose an image in its sine and cosine components.

Discrete Fourier Transformation

- We are only concerned about images, we apply Discrete Fourier Transform (DFT).
- DFT is the sampled Fourier Transform.
- Does not contain all frequencies forming an image, but only set of samples which is large enough to fully describe the spatial domain image.
- Number of frequencies corresponds to the number of pixels in spatial domain image, i.e. the image in the spatial and Fourier domain are of the same size.
- We are only concerned about images, we apply Discrete Fourier Transform(DFT).
- DFT is the sampled Fourier Transform.
- Does not contain all frequencies forming an image, but only set of samples which is large enough to fully describe the spatial domain image.
- Number of frequencies corresponds to the number of pixels in spatial domain image, i.e. the image in the spatial and Fourier domain are of the same size.

Fast Fourier

- A fast Fourier transform (FFT) algorithm computes the discrete Fourier transform (DFT) of a sequence, or its inverse.
- The DFT is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fields but computing it directly from the definition is often too slow to be practical.
- An FFT is a way to compute the same result more quickly: computing the DFT of N points in the naive way, using the definition, takes O(N2) arithmetical operations, while an FFT can compute the same DFT in only O(N log N) operations.
- The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to N / log N.
- The best-known FFT algorithms depend upon the factorization of N, but there are FFTs with O(N log N) complexity for all N, even for prime N

Properties of hadamard transform

- The Hadamard transform H is real, symmetric, and orthogonal, that is,
  - $H=H^{-1}=H^{T}=H^{*}$
- The Hadamard transform is a fast transform.
- The Hadamard transform has good to very good energy compaction for highly correlated images.

Hadamard transform is faster than sinusoidal transforms, since no multiplication is required; useful for digital hardware implementation of image processing algorithms. Easy to simulate but difficult to analyze. Applications in image data compression, filtering and design of codes. Has good energy compaction for images.

STEPS

Step 1: image acquisition

– Acquire a digital image using an image sensor

• a monochrome or color TV camera: produces an entire image of the problem domain every 1/30 second

• a line-scan camera: produces a single image line at a time, motion past the camera produces a 2-dimensional image

– If not digital, an *analog-to-digital* conversion process is required

– The nature of the image sensor (and the produced image) are determined by the application

• Mail reading applications rely greatly on line-scan cameras

• CCD and CMOS imaging sensors are very common in many applications

Step 2: Preprocessing

– Key function: improve the image in ways that increase the chance for success of the other processes

– In the mail example, may deal with contrast enhancement, removing noise, and isolating regions whose texture indicates a likelihood of alphanumeric information

Step 3: segmentation

– Broadly defined: breaking an image into its constituent parts

– In general, one of the most difficult tasks in image processing

• Good segmentation simplifies the rest of the problem

• Poor segmentation make the task impossible

– Output is usually raw pixel data: may represent region boundaries, points in the region itself, etc.

• Boundary representation can be useful when the focus is on external shape characteristics (e.g. corners, rounded edges, etc.)

• Region representation is appropriate when the focus is on internal properties (e.g. texture or skeletal shape)

– For the mail problem (character recognition) both representations can be necessary

Step 4: representation & description

– Representation: transforming raw data into a form

suitable for computer processing

– Description (also called feature extraction) deals with extracting features that result in some quantitative information of interest or features which are basic for differentiating one class of objects from another

– In terms of character recognition, *descriptors* such as lakes (holes) and bays help differentiate one part of the alphabet from another

Step 5: recognition & interpretation

– Recognition: The process which assigns a label to an object based on the information provided by its descriptors

A may be the alphanumeric character A

– Interpretation: Assigning meaning to an ensemble of recognized objects

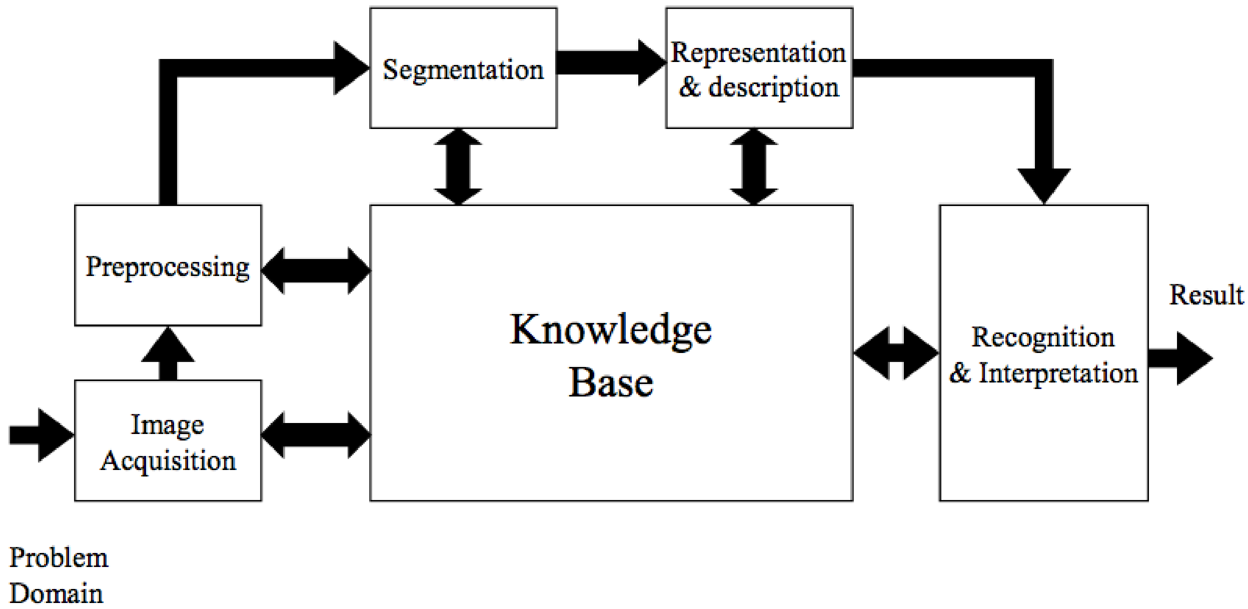


Image Compression

The objective of **image compression** is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form.

Image compression may be lossy or lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. Lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. Lossy compression that produces negligible differences may be called visually lossless.

Methods for lossless image compression are:

- Run-length encoding – used in default method in PCX and as one of possible in BMP, TGA, TIFF

- Area image compression
- DPCM and Predictive Coding
- Entropy encoding
- Adaptive dictionary algorithms such as LZW – used in GIF and TIFF
- Deflation – used in PNG, MNG, and TIFF
- Chain codes

Methods for lossy compression:

- Reducing the color space to the most common colors in the image. The selected colors are specified in the color palette in the header of the compressed image. Each pixel just references the index of a color in the color palette, this method can be combined with dithering to avoid posterization.
- Chroma subsampling. This takes advantage of the fact that the human eye perceives spatial changes of brightness more sharply than those of color, by averaging or dropping some of the chrominance information in the image.
- Transform coding. This is the most commonly used method. In particular, a Fourier-related transform such as the Discrete Cosine Transform (DCT) is widely used: N. Ahmed, T. Natarajan and K.R.Rao, "Discrete Cosine Transform," *IEEE Trans. Computers*, 90-93, Jan. 1974. The DCT is sometimes referred to as "DCT-II" in the context of a family of discrete cosine transforms; e.g., see discrete cosine transform. The more recently developed wavelet transform is also used extensively, followed by quantization and entropy coding.
- Fractal compression.

Contrast Stretching

Contrast Stretching (**Image Processing**) The**contrast** of an **image** is a measure of its dynamic range, or the "spread" of its histogram. The dynamic range of an **image** is defined to be the entire range of intensity values contained within an **image**, or put a simpler way, the maximum pixel value minus the minimum pixel value.

Contrast stretching (often called normalization) is a simple **image** enhancement technique that attempts to improve the **contrast** in an **image** by **`stretching**' the range of intensity values it contains to span a desired range of values, e.g. the the full range of pixel values that the **image** type concerned allows

```
MP = 2^q-1                    (e.g. 255 for 8-bpp image)
a = min(I)
b = max(I)
R = b-a

foreach (pixel p in I)
  p = [(p-a)/R]MP             (apply linear scaling function)
  p = round(p)
end
```

Convolution:

Convolution is a simple mathematical operation which is fundamental to many common image processing operators. Convolution provides a way of `multiplying together' two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality. This can be used in image processing to implement operators whose output pixel values are simple linear combinations of certain input pixel values.

Properties

Associativity

THEOREM 1: Associative Law

$f_1(t)*(f_2(t)*f_3(t))=(f_1(t)*f_2(t))*f_3(t)$

Commutativity

THEOREM 2: Commutative Law

$$\begin{aligned} y(t) &= f(t)*h(t) \\ &= h(t)*f(t) \end{aligned}$$

Distribution

THEOREM 3: Distributive Law

$f_1(t)*(f_2(t)+f_3(t))=f_1(t)*f_2(t)+f_1(t)*f_3(t)$**(6)**

Time Shift

THEOREM 4: Shift Property

For $c(t)=f(t)*h(t)$, then

$c(t-T)=f(t-T)*h(t)$**(7)**

and

$c(t-T)=f(t)*h(t-T)$

Width

In continuous time, if Duration$(f_1)=T_1$ and Duration$(f_2)=T_2$, then

Duration$(f_1*f_2)=T_1+T_2$

Causality

If f and h are both causal, then $f*h$ is also causal.

Thresholding

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images**.** The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T or a white pixel if the image intensity is greater than that constant. In the example image on the right, this results in the dark tree becoming completely black, and the white snow becoming completely white.

Categorizing thresholding Methods

To make thresholding completely automated, it is necessary for the computer to automatically select the threshold. Thresholding methods is categorize into the following six groups based on the information the algorithm manipulates.

- **Histogram shape**-based methods, where, for example, the peaks, valleys and curvatures of the smoothed histogram are analyzed

- **Clustering**-based methods, where the gray-level samples are clustered in two parts as background and foreground (object), or alternately are modeled as a mixture of two Gaussians
- **Entropy**-based methods result in algorithms that use the entropy of the foreground and background regions, the cross-entropy between the original and binarized image, etc.[1]
- **Object Attribute**-based methods search a measure of similarity between the gray-level and the binarized images, such as fuzzy shape similarity, edge coincidence, etc.
- **Spatial** methods [that] use higher-order probability distribution and/or correlation between pixels
- **Local** methods adapt the threshold value on each pixel to the local image characteristics. In these methods, a different T is selected for each pixel in the image.

Edge Detection

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*. The same problem of finding discontinuities in 1D signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

Gradient Filter

The most common type of edge detection process uses a gradient operator, of which there have been several variations. Mathematically, for an image function, f(x,y), the gradient magnitude, g(x,y) and the gradient direction, θ(x,y) are computed as

$$g(x,y) \cong (\Delta x^2 + \Delta y^2)^{\frac{1}{2}}$$

and,

$$\theta(x,y) \cong atan\ (\frac{\Delta y}{\Delta x})$$

where,

$$\Delta x = f(x+n,y) - f(x-n,y) \quad and\ \Delta y = f(x,y+n) - f(x,y-n)$$

and n is a small integer, usually unity. For example, the simplest implementation of this would be to convolve the following mask with the image data, aligning the mask with the x and y axes to compute the values of Δx and Δy.

-1	0	1

Variations on this theme have included the Roberts, Prewitt and Sobel operators. The Sobel operator was by far the most extensively used until those based on the Gaussian function and its derivatives were recognised widely. In this case the masks are extended to a 3 by 3 neighbourhood, rather than the 3 by 1 neighbourhood given above. The x and y masks given below are first convolved with the image to compute the values of Δx and Δy. Then the magnitude and angle of the edges are computed from these values and stored (usually) as two separate image frames.

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

Types of derivative filter

3x3 filters

Introduction:

The simplest mathematical situation is represented when using a 3x3 filter, however these filters may not be exact enough to calculate slopes. A 3x3 filter uses the 9 input values to calculate a value for the center pixel in the output map.

To do calculations with a 3x3 filter, a local coordinate system (x,y) is defined around the current center pixel, as:

(-1,1)	(0,1)	(1,1)
(-1,0)	(0,0)	(1,0)
(-1,-1)	(0,-1)	(1,-1)

Both x and y can have the values -1, 0, and 1.

When calculating the first derivative only in the x-direction, y remains 0, and x can have value -1, 0, or 1.

To calculate derivatives, a continuous function is needed. The input pixel values are to be described by a function f_x where:

f_0 = input pixel value of the center pixel; x=0
f_{-1} = input pixel value of the pixel to the left of the center pixel; x=-1
f_1 = input pixel value of the pixel to the right of the center pixel; x=1

5x5 filters

Calculation of matrix coefficients for 5x5 filters follows the same method as for 3x3 filters. Although 5x5 filters are a little bit more complicated, they will produce more accurate results.

Again a local coordinate system is used around the current center pixel as:

(-2,2)	(-1,2)	(0,2)	(1,2)	(2,2)
(-2,1)	(-1,1)	(0,1)	(1,1)	(2,1)
(-2,0)	(-1,0)	(0,0)	(1,0)	(2,0)
(-2,-1)	(-1,-1)	(0,-1)	(1,-1)	(2,-1)
(-2,-2)	(-1,-2)	(0,-2)	(1,-2)	(2,-2)

Both x and y can have the values -2, -1, 0, 1, and 2.

Smoothing Operations

These algorithms are applied in order to reduce noise and/or to prepare images for further processing such as segmentation. We distinguish between linear and non-linear algorithms where the former are amenable to analysis in the Fourier domain and the latter are not. We also distinguish between implementations based on a rectangular support for the filter and implementations based on a circular support for the filter.

Smoothing may be distinguished from the related and partially overlapping concept of curve fitting in the following ways:

- curve fitting often involves the use of an explicit function form for the result, whereas the immediate results from smoothing are the "smoothed" values with no later use made of a functional form if there is one;
- the aim of smoothing is to give a general idea of relatively slow changes of value with little attention paid to the close matching of data values, while curve fitting concentrates on achieving as close a match as possible.
- smoothing methods often have an associated tuning parameter which is used to control the extent of smoothing. Curve fitting will adjust any number of parameters of the function to obtain the 'best' fit.

Mean Filter

Mean filtering is a simple, intuitive and easy to implement method of *smoothing* images, *i.e.* reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images.

How It Works

The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used, as shown in Figure 1, although larger kernels (*e.g.* 5×5 squares) can be used for more severe smoothing. (Note that a small kernel can be applied more than once in order to produce a similar but not identical effect as a single pass with a large kernel.)

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Figure 1 3×3 averaging kernel often used in mean filtering

Computing the straightforward convolution of an image with this kernel carries out the mean filtering process.

Histograms

An **image histogram** is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance. The horizontal axis of the graph represents the tonal variations, while the vertical axis represents the number of pixels in that particular tone. The left side of the horizontal axis represents the black and dark areas, the middle represents medium grey and the right hand side represents light and pure white areas. The vertical axis represents the size of the area that is captured in each one of these zones. Thus, the histogram for a very dark image will have the majority of its data points on the left side and center of the graph. Conversely, the histogram for a very bright image with few dark areas and/or shadows will have most of its data points on the right side and center of the graph.

- Image enhancements
- Image statistics
- Image compression
- Image segmentation
- Simple to calculate in software
- Economic hardware implementations

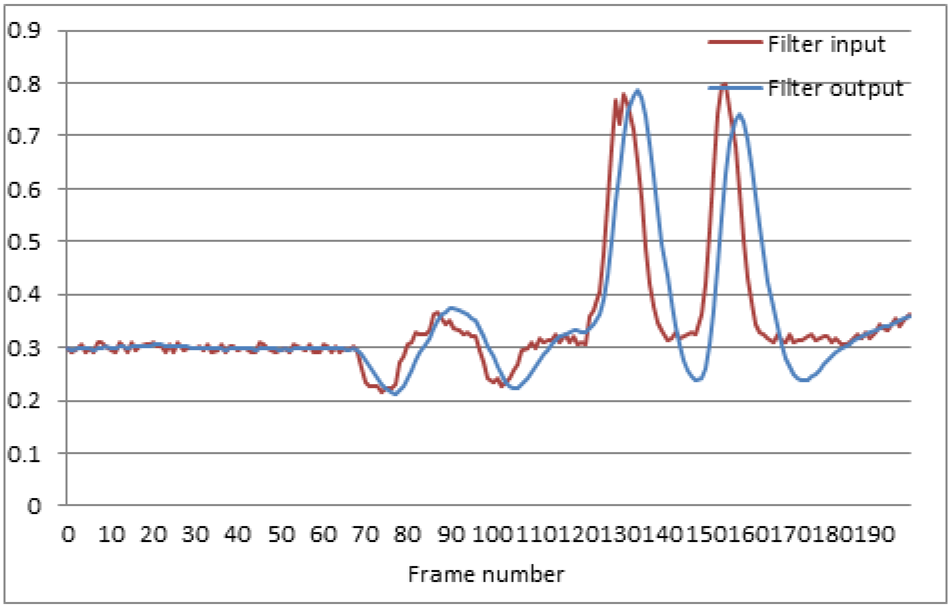
Skeleton Filtering

The skeletal tracking (ST) system of the Natural User Interface (NUI) provides joint positions of tracked persons' skeletons. These joint positions are the data consumed as position and pose, and they are used for many purposes, such as gesture detection, navigating user interfaces, and so on.

Measurement errors and noise are by-products of almost any system that measures a physical quantity via a sensor. The characteristics of this error are usually described by accuracy and precision of the system, where *accuracy* is defined as the degree of closeness of measured quantity to its actual value, and *precision* is defined as the degree to which repeated measurements are close to each other. An accurate system does not have any systematic error in measurements and therefore does not add a systematic bias. A precise system results in measurements close to each other when the measurement is repeated

Latency can be defined as the time it takes from when a person makes a move, till the time at which the person sees the response to his or her body movement on the screen. Latency degrades the experience as soon as people start to notice there is a delay in response to their movements. User research shows that 72% of people start noticing this delay when latency is more than 100 msec, and therefore, it is suggested that developers aim for an overall latency of 100 msec.

The *joint filtering latency* is how much time it takes for filter output to catch up to the actual joint position when there is a movement in a joint. This is shown in Figure , which shows that filter output is lagging behind input when there are changes in input. It is important to note that the latency introduced by joint filtering is not the CPU time it takes for the filtering routine to execute.



Neural Networks

In many industrial, medical, and scientific image-processing applications, feature- and pattern-recognition techniques such as normalized correlation are used to match specific features in an image with known templates. Despite the capabilities of these techniques, some applications require the simultaneous analysis of multiple, complex, and irregular features within images. For example, in semiconductor-wafer inspection, discovered defects are often complex and irregular and demand more "human-like" inspection techniques. By incorporating neural-network techniques, developers can train such imaging systems with hundreds or thousands of images until the system eventually "learns" to recognize irregularities.

Unlike conventional image-processing systems, neural networks are based on models of the human brain and are composed of a large number of simple processing elements that operate in parallel. In the human brain, more than a billion elements or cells, called neurons, are individually connected to tens of thousands of other neurons. Although each cell operates like a simple processor, the massive interaction among them makes vision, memory, and thought processes possible

The backpropagation algorithm

The backpropagation equations provide us with a way of computing the gradient of the cost function. Let's explicitly write this out in the form of an algorithm:

1. **Input** $x x$**:** Set the corresponding activation $a^1 a1$ for the input layer.
2. **Feedforward:** For each $l=2,3,\ldots,L l=2,3,\ldots,L$ compute $z^l=w^l a^{l-1}+b^l$ $zl=wlal-1+bl$ and $a^l=\sigma(z^l)$ $al=\sigma(zl)$.
3. **Output error** $\delta^L \delta L$**:** Compute the vector $\delta^L=\nabla_a C \odot \sigma'(z^L)$ $\delta L=\nabla aC \odot \sigma'(zL)$.
4. **Backpropagate the error:** For each $l=L-1,L-2,\ldots,2$ $l=L-1,L-2,\ldots,2$ compute $\delta^l=((w^{l+1})^T\delta^{l+1})\odot\sigma'(z^l)$ $\delta l=((wl+1)T\delta l+1)\odot\sigma'(zl)$.
5. **Output:** The gradient of the cost function is given by $\frac{\partial C}{\partial w^l_{jk}}=a^{l-1}_k\delta^l_j$ $\partial C\partial wjkl=akl-1\delta jl$ and $\frac{\partial C}{\partial b^l_j}=\delta^l_j$ $\partial C\partial bjl=\delta jl$.

Examining the algorithm you can see why it's called*back*propagation. We compute the error vectors $\delta^l \delta l$ backward, starting from the final layer. It may seem peculiar that we're going through the network backward. But if you think about the proof of backpropagation, the backward movement is a consequence of the fact that the cost is a function of outputs from the network. To understand how the cost varies with earlier weights and biases we need to repeatedly apply the chain rule, working backward through the layers to obtain usable expressions.

Perception

Many image processing applications are intended to produce images that are to be viewed by human observers. It is therefore important to understand the characteristics and limitations of the human visual system to understand the “receiver” of the 2D signals. At the outset it is important to realise that

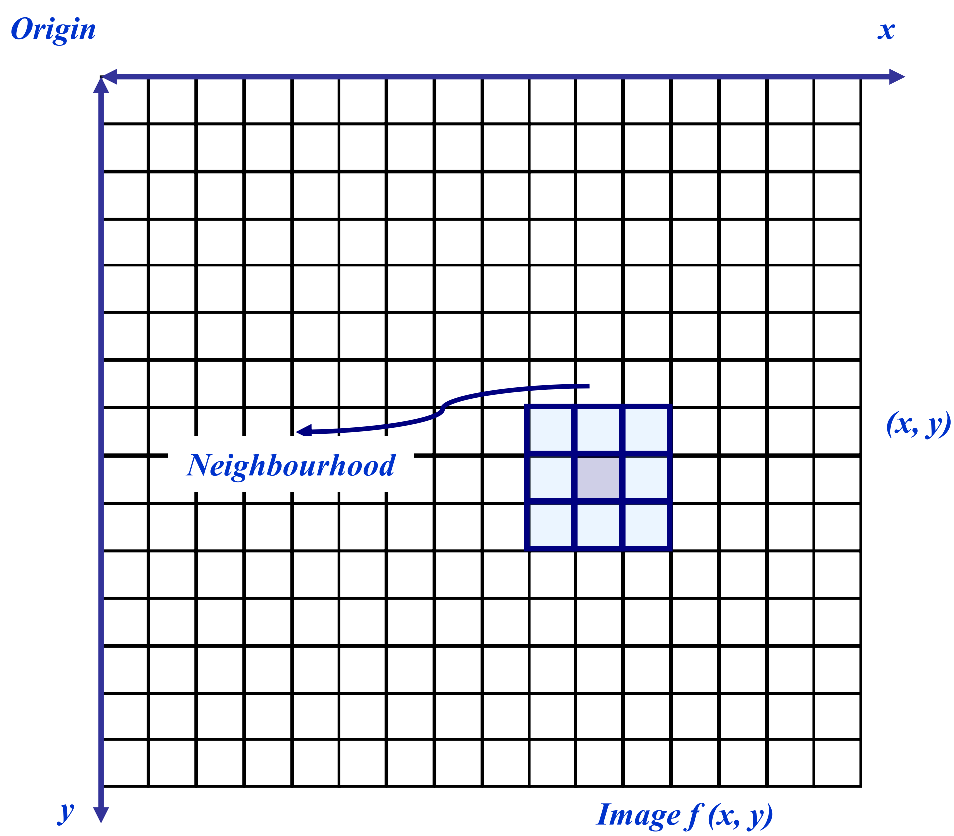
(1) human visual system (HVS) is not well understood;

(2) no objective measure exists for judging the quality of an image that corresponds to human assessment of image quality,

(3) the typical human observer does not exist Nevertheless, research in perceptual psychology has provided some important insights into the visual system [stock ham].

Neighbourhood

Neighbourhood operations simply operate on a larger neighbourhood of pixels than point operations. Neighbourhoods are mostly a rectangle around a central pixel. Any size rectangle and any shape filter are possible.



Some simple neighbourhood operations include:

Min: Set the pixel value to the minimum in the neighbourhood

Max: Set the pixel value to the maximum in the neighbourhood

Median: The median value of a set of numbers is the midpoint value in that set (e.g. from the set [1, 7, 15, 18, 24] 15 is the median). Sometimes the median works better than the average.

Opening and closing

- Important operation
- Derived from fundamental operation i.e dilation and erosion
- Usually applied to binary image but gray value images are also possible
- Opening and closing are dual operations

Opening

- Similar to erosion
- Spot and noise removal
- Less destructive
- Same structuring element for both operation
- Input is binary image and structuring element containing only 1's

Closing

- Similar to dilation
- Remove of holes
- Tends to enlarge regions, shrink background
- It is defined as dilation followed by erosion using the same structuring element for both operation
- Input is binary image and structuring element containing only 1's

Hopfield Network

A **Hopfield net** is a recurrent neural network having synaptic connection pattern such that there is an underlying Lyapunov function for the activity dynamics. Started in any initial state, the state of the system evolves to a final state that is a (local) minimum of the Lyapunov function.

There are two popular forms of the model:

- Binary neurons with discrete time, updated one at a time

$$V_j(t+1)=\{1,0, \text{ if } \Sigma_k T_{jk} V_k(t)+I_j>0 \text{ otherwise}$$

- Graded neurons with continuous time

$$dx_j/dt=-x_j/\tau+\Sigma_k T_{jk} g(x_k)+I_j\ .$$

Here,

- V_j denotes activity of the j-th neuron.
- x_j is the mean internal potential of the neuron.
- I_j is direct input (e.g., sensory input or bias current) to the neuron.

- T_{jk} is the strength of synaptic input from neuron k to neuron j .
- g is a monotone function that converts internal potential into firing rate output of the neuron, i.e., $V_j = g(x_j)$.

Contour Representations

When dealing with a region or object, several compact representations are available that can facilitate manipulation of and measurements on the object. Several techniques exist to represent the region or object by describing its contour.

Chain code

This representation is based upon the work of Freeman . We follow the contour in a clockwise manner and keep track of the directions as we go from one contour pixel to the next. For the standard implementation of the chain code we consider a contour pixel to be an object pixel that has a background (non-object) pixel as one or more of its 4-connected neighbors.

Chain code properties

* Even codes {0,2,4,6} correspond to horizontal and vertical directions; odd codes {1,3,5,7} correspond to the diagonal directions.

* Each code can be considered as the angular direction, in multiples of 45deg., that we must move to go from one contour pixel to the next.

* The absolute coordinates [*m*,*n*] of the first contour pixel (e.g. top, leftmost) together with the chain code of the contour represent a complete description of the discrete region contour.

* When there is a change between two consecutive chain codes, then the contour has changed direction. This point is defined as a *corner*.

Crack code

An alternative to the chain code for contour encoding is to use neither the contour pixels associated with the object nor the contour pixels associated with background but rather the line, the "crack", in between

Components of Digital Image Process System

Image Sensors

An image sensor is a device that converts an optical image into an electronic signal. It is used mostly in digital cameras, camera modules and other imaging devices. Early analog sensors were video camera tube; most currently used are digital charge coupled device (CCD) or complementary metal–oxide–semiconductor (CMOS) active pixel sensors

Specialized Image Processing Hardware

Usually consists of the digitizer, mentioned before, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images.

Processors

- RISC and CISC
- MIMD
  Multiple Instruction stream Multiple Data stream:
  Several processors execute different instructions on different data
- SIMD
  Single Instruction stream Multiple Data stream:
  Several processors execute the same instruction on different data.
- PIPELINES
  set of data processing elements connected in series, where the output of one element is the input of the next one

Computer

Mass Storage

Hardcopy devices

Image Processing software

Image Displays

Networking : is almost a default function in any computer system, in use today. Because of the large amount of data inherent in image processing applications the key consideration in image transmission is bandwidth.

Dilation

The **dilation** of an image *f* by a structuring element *s* (denoted $f \oplus s$) produces a new binary image $g = f \oplus s$ with ones in all locations (*x,y*) of a structuring element's orogin at which that structuring element *s* hits the the input image *f*, i.e. *g(x,y)* = 1 if *s* hits *f* and 0 otherwise, repeating for all pixel coordinates (*x,y*). Dilation has the opposite effect to erosion -- it adds a layer of pixels to both the inner and outer boundaries of regions.

Erosion

The **erosion** of a binary image *f* by a structuring element *s* (denoted $f \ominus s$) produces a new binary image $g = f \ominus s$ with ones in all locations (*x,y*) of a structuring element's origin at which that structuring element *s* fits the input image *f*, i.e. *g(x,y)* = 1 is *s* fits *f* and 0 otherwise, repeating for all pixel coordinates (*x,y*).