Motivation

Provide the user (scientist, doctor, …) with some means to:
- enhance contrast of local features
- remove noise and other artifacts
- enhance edges and boundaries
- composite multiple images for a more comprehensive view

There are two basic operations: global and local

Global operations:
- operate on the entire set of pixels at once
- examples: brightness and contrast enhancement

Local operations:
- operate only on a subset of pixels (in a pixel neighborhood)
- examples: edge detection, contouring, image sharpening, blurring

The Image Histogram

- A histogram lists the number of image pixels for each value

- The histogram reveals more insight about image contrast and brightness.
Grey Level Transformation: Basics

Problem: We only have a fixed number of grey levels (256) that can be displayed or perceived
- need to use this ‘real estate’ wisely to bring out the image features that we want

Use intensity transformations $T_p$
- enhance (remap) certain intensity ranges at the cost of compressing others

Grey Level Transformation: Enhancements

- enhance the dark areas (slope > 1)
- suppress the white areas (slope < 1)

Grey Level Transformation: Windowing

Dedicate full contrast to either bone or lungs

Grey Level Transformation: Windowing

using histogram equalization
**Histogram Equalization**

- Image contrast and brightness may be improved by modifying the histogram.
- The ‘contrast stretching’ operation requires the user to manipulate the image’s histogram.
- **Histogram equalization** is an automatic procedure to spread out the value distribution.

The discrete histogram equalization equation is:

\[
P_{\text{new}}(k) = \frac{\sum_{j=0}^{k} n(p_{\text{org}}(j))}{n_{\text{total}}} \cdot p_{\text{max}}
\]

- For example, the equalization transformation for a dark image would be:

**Color Image Processing**

- Convert the image from RGB to HSV space.
- Perform transformations of pixel H, S, V values via transfer functions.
- Convert the transformed HSV image back into RGB space and display.

**Multi-Image Operations: Noise Averaging**

Assume a pixel value \( p \) is given by: \( p = \text{signal} + \text{noise} \)

- \( E(\text{signal}) = \text{signal} \)
- \( E(\text{noise}) = 0 \), when noise is random

Thus, averaging (adding) multiple images of a steady noisy object will eliminate, or at least reduce, the noise.

**Color Image Histogram Equalization**

Equalize the V channel, and then convert back to RGB.

- **Original**
- **Hue transformed**
- **After averaging 16 subsequently acquired images**
**Multi-Image Operations: Eliminating Background**

In angiography, radio-opaque contrast agents (injected into the bloodstream) are used to enhance the perfused vessels.

An X-ray image is taken when the radio-opaque bolus of blood is coming through:
- however, the background reduces the contrast of the dye
- subtracting the (constant) background from the (dynamic) radiographic image leaves just the perfused structures (angio image)

![Image](after injection (radio-image) background (mask image) just the bolus (angio image))

**Discrete Filters**

Discrete filters since they operate on a discretized image:
- to implement discrete filters we use discrete convolution

\[
\text{for each } i, j, \quad \text{temp} = 0 \\
\text{for each } k, l, \\
\quad \text{temp} := p_{i-1,j-1} + \sum_{k=0}^{N} \sum_{l=0}^{N} w_{k,l} \quad w_{k,l} \\
\quad p_{i,j}^{\text{new}} = \text{temp}
\]

- place a weight matrix or mask at each pixel location \(p_{ij}\)
- this mask weighs the pixel's neighborhood and determines the output pixel's value
- important: do not replace the computed values into the original image, but write to an output image

**Popular Discrete Filters: Lowpass**

Smoothing (averaging, often weighted):
- also called low-passing: keeps the low frequencies, but reduces the high frequencies
- removes noise and jagged edges
- but also blurs the signal

\(S(k)\) (idealized case)

![Image](Simple averaging mask)

Smoothing:
- Averaging is the simplest form of smoothing (blurring)
- more complex functions (masks) are often used because they offer additional benefits
- for example: Gaussian (discretized)
- we shall see more on this later
**Popular Discrete Filters: Median Smoothing**

A non-linear filter, best used to remove speckle noise
- a regular smoothing filter would blur the speckles (and the signal)
- the median filter will eliminate the speckle and leave the signal as is

Procedure:
- convolve with a mask as usual
- but this time, for each mask position, sort the values under the mask
- pick the median and write it to the output image
- the speckle pixel will be an outlier and not be selected as the median

**The Power of the Median Filter**

Superior for speckle noise

original
smoothed
median filtered

**Popular Discrete Filters: Highpass**

Edge detector / enhancer:

\[
\nabla I = \nabla h * I \quad \text{first derivative (gradient)}
\]

\[
\nabla^2 I = \nabla^2 h * I \quad \text{second derivative (Laplacian)}
\]

- also called high-passing: keeps the high frequencies, but reduces the low frequencies
- enhances edges and contrast
- but also enhances noise and jagged edges

**Highpass (2)**

The Sobel filter comes in a pair of two masks:
- one mask computes an image for the x-derivative (dx), the other for the y-derivative (dy)
- this decreases the sensitivity to noise (sharpening tends to magnify high frequency noise)
- Note that pixel values below zero will occur at edges with negative gradients

We get two images, \(img_{dx}\) and \(img_{dy}\); their pixels are combined by:

\[
img_{ren} = \left( img_{dx}^2 + img_{dy}^2 \right)^{1/2} \quad \text{or} \quad img_{ren} = |img_{dx}| + |img_{dy}|
\]
**Edge Enhancement**

Several useful effects can be achieved by subsequent filtering with different masks (kernels) and/or multi-image operations.

Subtracting a smoothed image from the original image leaves the edges (the high frequencies):

- **original**
- **smoothed**
- **original - smoothed**

**Multi-Pass Filtering: High-Pass**

**Multi-Pass Filtering: Unsharp Masking**

Places the enhanced edges on top of a smoothed original.

**Gaussian Kernel**

The Gaussian kernel is a popular filter function:

- see book for 3x3 convolution masks

$$g$$

$$\nabla_x g$$

**Gaussian**

$$\nabla_y g$$

**dy (y-gradient)**

$$\nabla^2 g$$

**Laplacian**

(difference of two Gaussians)

$$I$$

$$g * I$$

$$I - g * I$$

$$g * I + (1 + \alpha)(I - g * I)$$
Global and Local Filtering: Shortcomings (1)

Windowing enhances contrast only for a specific range of grey levels (not sensitive to edges)
- strong edges with already good contrast are further enhanced

Edge enhancement (such as sharp masking) only boosts features within a certain frequency band
- this frequency band is determined by filter size -- features outside that band are not enhanced (cannot see many scales at the same time)
- all grey value variations (within that band) are enhanced, even if they already had good contrast

Global and Local Filtering: Shortcomings (2)

One more example: digital radiograph of a foot

Original window/level operation:
- original
- edge enhanced
- window/level operation

Multi-Scale Image Enhancement: Motivation

Designed to overcome these shortcomings
- enhancements will be visible at all scales at the same time
- this requires a pyramid of detail images that are added together

Image pyramid of lowpassed images
- a hierarchy of images, repeatedly lowpassed at scales of power of 2

Multi-Scale Image Enhancement: Detail Images

We have seen detail enhancement by high-pass filtering
- the result is called a detail image

We can create an image pyramid of detail images
- constructed by subtracting the smoothed image at the corresponding pyramid level from the original: $D_i = I - I_i \ast g$
- this gives us the detail $D_i$ at scale $i$
Multi-Scale Image Enhancement: Detail Pyramid

A representation of the details occurring at multiple levels of scale is called detail pyramid.

We can reconstruct the image at level $i$ by adding the expanded image at level $(i+1)$ to the detail at level $i$:

$$ I_i = D_i + E(I_{i+1}) $$

By adding all the details we can assemble the original image:

Multi-Scale Image Enhancement: Non-Linear Mapping

Strategy:
- create pyramid of detail images $D_i$
- apply a non-linear grey-scale transformation to each of the $D_i$
- this emphasizes the low-contrast details (previously invisible)
- it de-emphasizes the high-contrast details (to just noticeable levels)

finally, re-assemble the image by adding these transformed detail images recursively

Multi-Scale Image Enhancement: Results

This strategy has been employed in the MUSICA algorithm
- developed by the company Agva Gevaert
- routinely in used in digital radiography in hospitals worldwide

edge enhanced  window/level operation  MUSICA