

**Spatial Resolution**

* Determined by how sampling was carried out
* Pixel size, dots per inch (DPI)
* Smaller image size > blur

**Intensity / Gray Level Resolution**

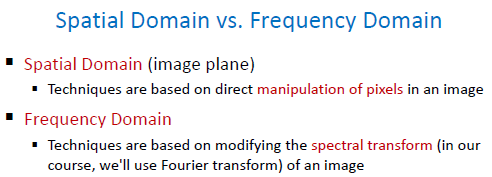
* Number of intensity levels used to represent the image
* More intensity levels used, finer the level of detail
* Number of bits used to store each intensity level (eg. 8 bits > 2^8 no. of intensity lv.)
* Lower grey levels, lesser choice of colours used in image

**Image Enhancement**

* Make image more visually appealing (Gamma correction > boost dark values > brighter)
* Highlight interesting detail in images
* Remove noise from images (Median filter > Remove Salt and Pepper Noise)

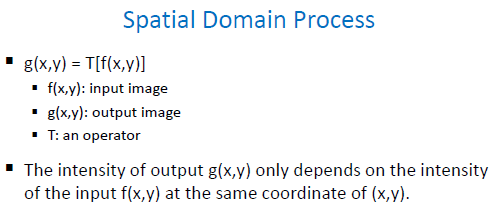
**CMYK:** In its color space, the brightness and chroma are distributed over each of the three components

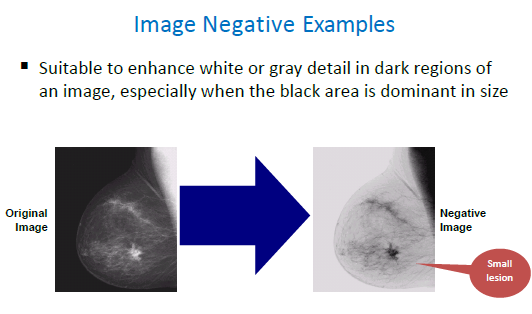
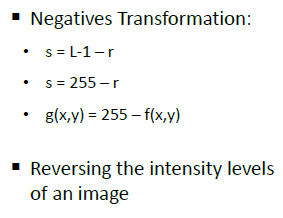
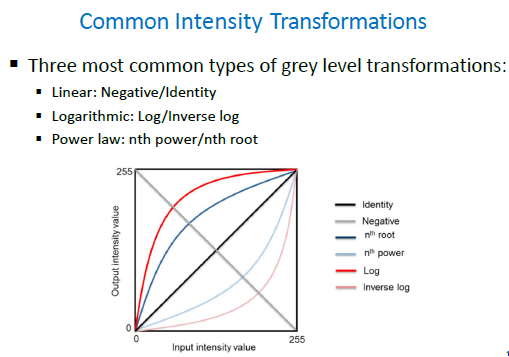
**Image reverse operation: NOT** applied for enhancing images with mainly brighter grayscale

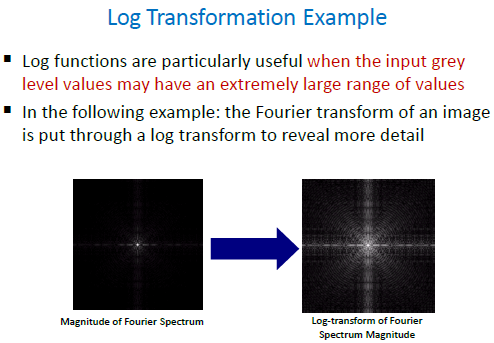
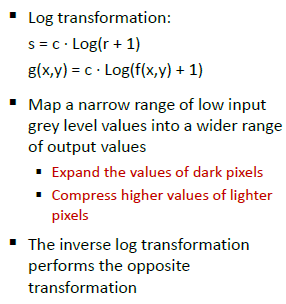


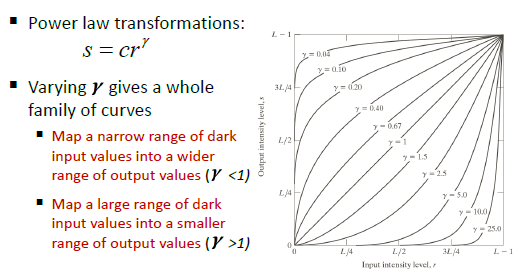
**Image Enhancement: Spatial Domain (ch 2)**

1. **Point processing**
   * Gray values change without any knowledge of its surroundings (Grey level transformation)
     + **Log (spectrum)**
     + **Power-law**
     + **Piecewise linear**
       - Contrast stretching
         * Increase the contrast
       - Intensity level slicing
         * Highlighting
     + **Binaryzation**
     + **Histogram Equalization (pdf -> cdf -> transformation)**
       - Not for smoothing
       - After point processing, the histogram of the processed image is **undeterminable** when compared with the original one
       - The corresponding relations between an image and its gray histogram is **many to one**
2. **Neighbourhood processing (filtering)**
   * Gray values change depending on the grey values in a small neighbourhood of pixels around the given pixel
     + **Basic spatial filters**
       - Linear spatial filters (see 2B p.4-6)
     + **Smoothing filters** (remove fine details, blurring and noise reduction)
       - (Neighbourhood) Averaging filters (linear)
       - Weighted smoothing filters
       - Order Statistics filters (non-linear) (Go to page 10)
         * Median filters (**better than average filter in removing noise**, see 2B p.16)
         * Max filters
         * Min filters
     + **Sharpening filters** (highlight fine details, remove blurring and highlight edges > spatial differentiation)
       - Laplacian filters (Original – Laplacian = Sharpened)
       - Sobel filters (Gradient mask)





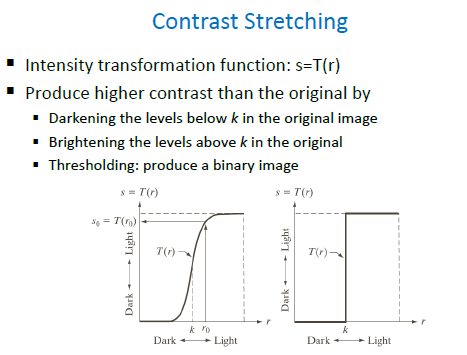
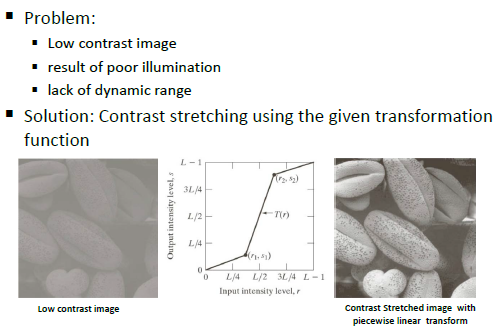


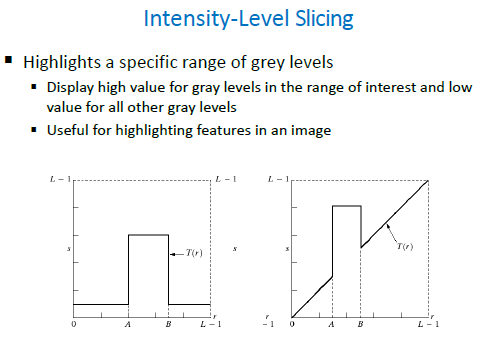
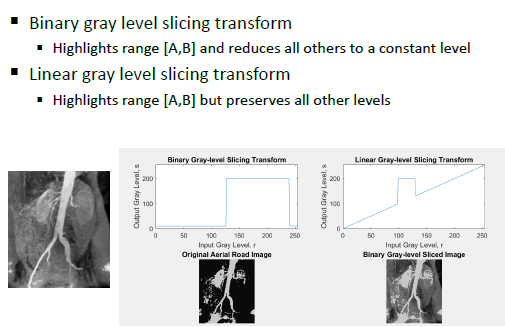
**Power Law Transformation**

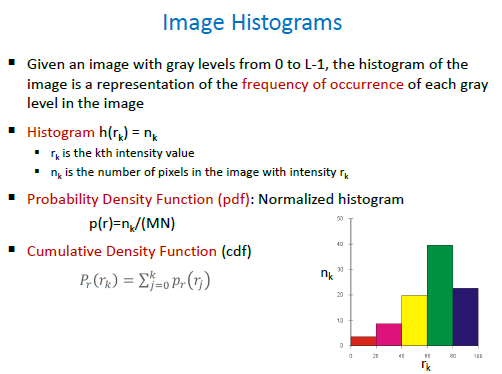
Gamma < 1 : brighter, less dark input values, expansion of lower levels

Gamma > 1 : darker, compression of higher gray levels

**\*Will not affect the image to become blurred or details look much clear\***



Dark > low side of grey scale Light > high side of grey scale

Low contrast > narrow and centred towards the middle of grey scale High contrast > broad range of grey scale (evenly spaced histogram)

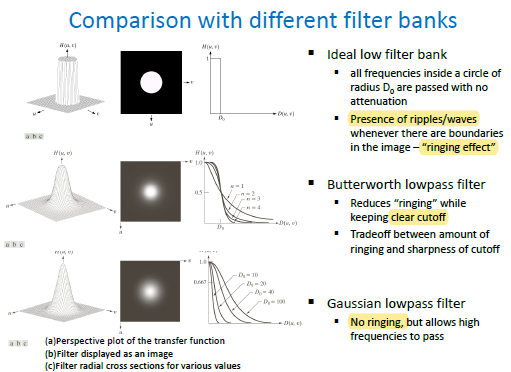
**Image Enhancement: Frequency Domain (ch 3)**

**Fourier Transform Spectrum (see 3A p.24-28)**

* **Low freq.:** Center of spectrum, Uniform grey values
* **High freq.:** Boundary of spectrum, Edge and shape transitions in grey values (Small image details and shape edges), **NOT** responsible for overall gray level display in smooth areas
* After image translation, the amplitude and phase characteristics of the Fourier transform are **NOT** unchanged
* **Features of Fourier Transform:**
  + Original data can be fully recovered from the result of transform
  + Optimal in a mean square error sense
  + Has the plural operation
  + Does not have the concept of frequency domain

**Filtering in Frequency Domain**

* **Ideal filtering**
  + Has ringing effect in images
* **Low-Pass filtering** (Ideal / Butterworth / Gaussian)
  + Remove high freq.
  + Remove noise
  + Blur image 🡪 Achieve smoothing
  + **Butterworth better than ideal** (reduce “ringing” while keeping clear cut-off)
  + Gaussian (**No ringing** but allows high freq. to pass)

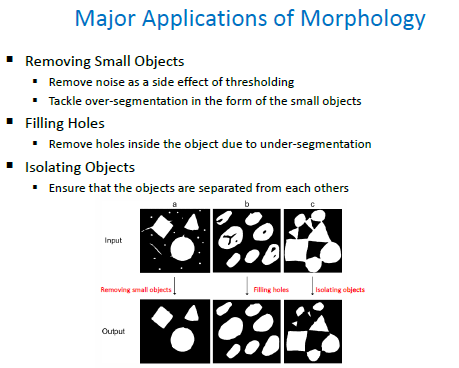


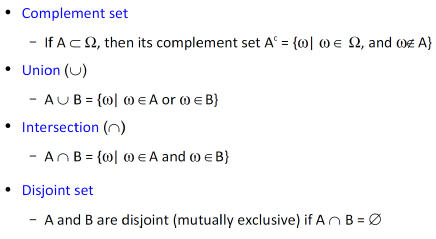
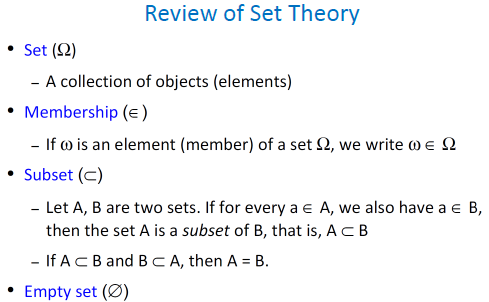
* **High-Pass filtering** (Ideal / Butterworth / Gaussian)
  + Remove low freq.
  + Sharpening, edge enhancement
  + Cannot denoise
* **Laplacian filtering**
  + Sharpening, better than Laplacian in Spatial
* **Homomorphic filtering** (see 3B p.36-40)
  + Illumination: decrease contribution made by low freq.
  + Reflection: amplify contribution made by high freq.
* **Selective filtering**
  + Bandpass / Band-reject
    - Process specific bands
    - **Band-reject:** remove periodic noise
  + Notch filters
    - Process small regions of the freq. rectangle
    - Rejects / passes specific frequencies
    - Remove periodic noise (eg. image mosaics, scan line noise, halftoning noise)

**Morphological Processing (ch 4)**

**Morphology**

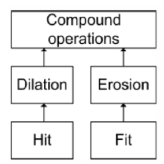
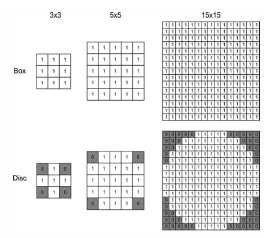
* Extract image components
* Applications:





**Structuring Elements**

* Box-shaped SE
  + Preserve sharp object corners
* Disk-shaped SE
  + Round the corners of the objects

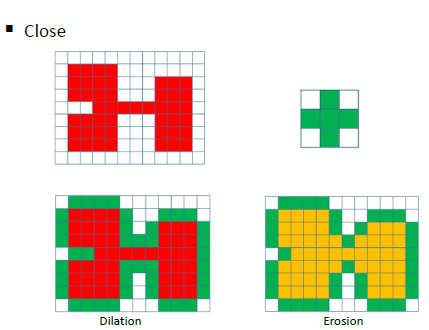
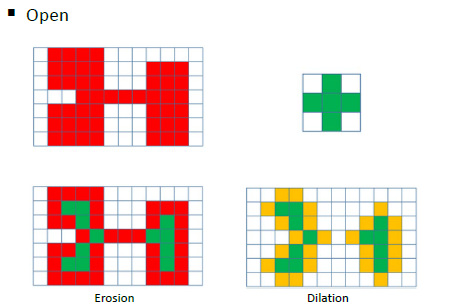
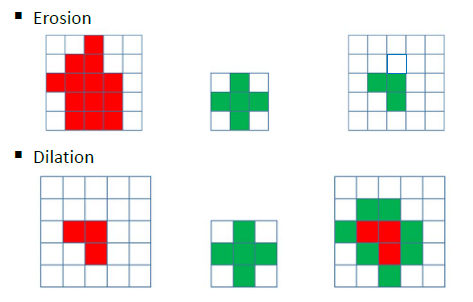


**Basic Morphological Methods**

* **Fit** (ALL of the ‘1’s in SE is covered by image)
  + **Erosion** (applying fit to entire image) (operator: Θ)
    - Contrasts an object
* **Hit** (any ONE of the ‘1’s in SE is covered by image)
  + **Dilation** (applying hit to entire image) (operator: ⊕)
    - Expands an object
    - Remember to reflect the SE (along x then along y) before applying dilation
* **Opening** (Erosion followed by Dilation)
  + Smoothens contours
  + Enlarges narrow gaps
  + Eliminates thin protrusions and ridges
  + **Cannot** remove holes
  + “Smoothing from inside”
* **Closing** (Dilation followed by Erosion)
  + Fills narrow gaps, holes and small breaks
  + Preserves objects’ size
  + “Smoothing from outside”

**\*Bigger the SE, bigger the effect in the image**

**\*Combination of opening and closing:** Image with fingerprints

****

**Morphological Algorithms**

* **Hit or Miss Transform**
  + Detecting specific shapes
* **Boundary Extraction**
* **Hole Filling**
* **Connected Components** (similar to hole filling)
* **Skeletons**

**Image Restoration (ch 6)**

* **Degradation & Restoration Process Models**
  + If the image is degraded, it **should** first make a restoration process, further enhancement
  + The degradation caused by blurring will **NOT** decrease the spatial resolution of the image
  + **Degradation function & Noise function (principle)**
* **Noise Models**
  + **Source of Noise**
    - Digital image acquisition process
    - Image transmission
  + **Models**
    - **Gaussian**
      * Electronic circuit noise, sensor noise (poor illumination / high temperature)
    - **Rayleigh**
      * Model noise in range imaging
    - **Erlang (Gamma)**
      * Laser imaging
    - **Exponential**
      * Laser imaging
    - **Uniform**
    - **Impulse (Salt and Pepper)**
      * Find in quick transients (eg. faulty switches)
* **Restoration in the presence of noise only (spatial filtering)**
  + **Mean filters**
    - **Arithmetic Mean**
    - **Geometric Mean**
      * Multiply all values in the filter ^ (1/MN)
      * Ineffective in the presence of pepper noise
      * Lose less detail than arithmetic
    - **Harmonic Mean**
      * MN / summation of all (1 / x)
      * Works well for salt noise, Gaussian noise
      * Retains detail better than arithmetic
    - **Contraharmonic Mean**
      * Sum of x^Q+1 / Sum of x^Q
      * + Q: eliminating pepper noise
      * - Q: eliminating salt noise
  + **Order-statistics filters**
    - **Order filter**
      * Arranges all pixels in sequential order based on grey level value in ascending order
    - **Minimum filter**
      * Works best for salt-type noise (high value)
      * Replace the centre value with the smallest value around it
    - **Maximum filter**
      * Works best for pepper-type noise (low value)
      * Replace the centre value with the largest value around it
    - **Median filter**
      * Works best for both salt and pepper noise (high and low values)
      * Better noise remover than Averaging filter
      * Preserve line structures
      * Replace the centre value with the median value around it
    - **Midpoint filter**
      * Works best for Gaussian and uniform noise
      * Not good for handling salt and pepper noise
      * Replace the centre value with (min + max) / 2
    - **Alpha-trimmed mean filter**
      * Works best for combination noise such as salt and pepper with Gaussian noise
      * Remove d / 2 highest and lowest values, then sum the remaining values / (MN – d)
  + **Adaptive mean filters**
    - Depends on the characteristics of image inside the filter region
    - Superior performance but increase in filter complexity
    - **Preserve edges**
* **Estimation of the degradation function**
  + Image observation
  + Experimentation
  + Mathematical modelling
    - Atmospheric blur
    - Motion blur
      * Taking images when your hand is not stable
* **Inverse filtering**
  + Numerical problem
    - If any points in H(u, v) are zero (division by zero)
      * **Solutions:** Do not take zero points of H(u, v) into account
    - In practice the denominator H(u, v) may have too small magnitude in the high frequency part, making the inverse filter highly unstable.
      * **Solutions:** To prevent this, we can (1) confine the inverse filter operation only to the low frequency part; or (2) add a small constant in the denominator of the inverse filter process, similar to Wiener filter.
* **Wiener filtering (Minimum Mean Square Error filtering)**
  + **Objectives:** Find an estimate of the uncorrupted image such that the mean square error between them is minimum
  + If the noise is zero, the Wiener filter reduces to the inverse filter
  + Noise power spectrum is constant when dealing with spectrally white noise
  + **Differences:** When the image is blurred by a known lowpass filter, it is possible to recover the image by inverse filtering. However, **inverse filtering is very sensitive to additive noise.** The approach of reducing one degradation at a time allows us to develop a restoration algorithm for each type of degradation and simply combine them. The Wiener filtering executes an optimal tradeoff between inverse filtering and noise smoothing. **It removes the additive noise and inverts the blurring simultaneously.**

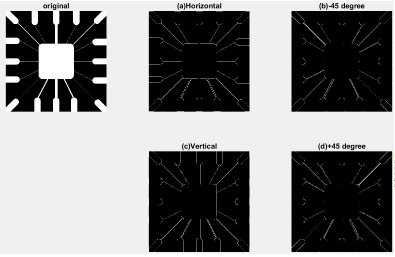
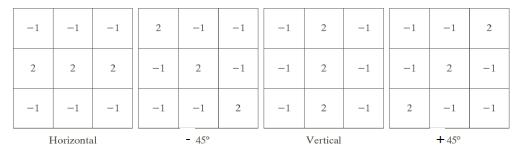
**Segmentation – Categories (ch 8)**

**Segmentation**

* Split/separate/subdivide an image into regions or objects
* To facilitate recognition, understand region of interest
* **Challenges:**
  + The definition of a region/object is problem-dependent
  + One of the most difficult tasks in image processing
  + Accuracy determines success or failure of application

1. **Edge-based**

* Finding boundary between adjacent regions
* Finding discontinuities (sharp, local changes in intensity)
* **Discontinuities in digital images (Spatial filters)**
  + Point (Laplacian)
  + Line (horizontal, -45 degree, vertical, 45 degree)
  + Edges (Roberts, Prewitt, Sobel, LOG, Canny)
* **Techniques:**
  + **Point Detection**
    - Detect an isolated point
    - Laplacian filter
      * **1st order derivatives**
        + Produce **thicker edges**
      * **2nd order derivatives**
        + **Stronger response to fine details** (thin lines, isolated points, noise)
        + Produce a double-edge response at ramp transitions in intensity
        + Sign: Determine whether a transition into an edge is from light to dark or dark to light
  + **Line Detection**
    - If each mask is moved around an image, it would respond more strongly to lines in the mentioned direction
    - 2nd derivatives have **stronger response and produce thinner lines** than 1st derivatives



* + **Edge (pixel) detection** (see ch8 p.15)
    - Edges: The boundary between two regions with relatively distinct gray-level properties
    - 3 fundamental steps (p.20)
      * Image smoothing for noise reduction
      * Detection of edge points
      * Edge localization
    - Roberts (cross gradient operators)
    - Prewitt & Sobel operators (horizontal, vertical and two diagonal masks)
    - Sobel masks (better noise suppression -> smoothing)
    - **Problems of edge detection (see p.29)**
    - **Reason of not using Laplacian to detect edges (see p.30)**
  + Edge formation from edge pixels
    - Edge Linking
      * Canny Filter
        + Use double thresholding and connectivity analysis to detect and link edges
    - **Hough Transform**
      * Motivation: Isolated points, identifying the locations and orientations of certain types of features in a digital image
      * Problem:
        + “a”approaches infinity as the line gets perpendicular to the x axis. (if a line is perpendicular to x axis, then this line is represented x=M. a->infinity)
      * Solution: 

1. **Threshold-based**

* Finding regions by grouping pixels of similar gray values

1. **Region-based**

* Finding regions directly using growing or splitting

1. **Motion-based**

* Finding regions by comparing successive frames of a video sequence to identify regions that correspond to moving objects

**Threshold & Region-based Segmentation (ch 9)**

* 1. **Threshold-based segmentation**
  + **Basic Global Thresholding**
    - **Disadvantage:** requires successive calculation of means (for both groups of pixels) at each step (refer ch9 p.19)
  + **Optimum Global Thresholding (OTSU)**
    - View thresholding as a statistical-decision theory problem
    - **Objectives:** Maximizes the between-class variance and minimize the with-in class variance
  + Multiple Thresholding
  + Variable Thresholding
  1. **Region-based segmentation**
  + Region grow (refer ch9 p.44)
  + **K-means**
    - * + Partition the data points into K clusters randomly and find centroids of each cluster
        + **Advantage:**
        + Easy for implementation and high-speed performance
        + Measurable and efficient in large data collection
        + **Disadvantage:**
        + Selection of optimal number of clusters is difficult
        + Selection of the initial centroids is random and different initializations lead to different results

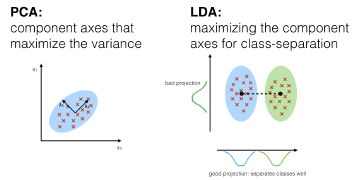
**Face Recognition (ch 10 & 11)**

1. **Principal Component Analysis (PCA)**

* Converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components
* **Purpose:** Identify the orientation with largest variance

1. **Linear Discriminant Analysis (LDA)**

* Eigenfaces **exploit the max scatter of the training images** in face space
* Fisherfaces attempt to **maximize the between class scatter, while minimizing the within class scatter**
* **Goal:** Find the best separation between two classes



1. **The Eigenfaces Algorithm**
   * Pros and Cons (see ch10 p.41)

**Deep learning in image segmentation**

* **Models**
  + Fully Convolutional Network, DeconvNet, SegNet, U-Net, PSPNet, DeepLab v1, v2, v3, v3+
* Loss functions
  + Cross entropy (CE),Weighted cross entropy, Balanced cross entropy (BCE), Focal loss, Dice loss, Tversky loss
* Able to choose suitable loss functions

**Detection (ch 12)**

* HOG
* DPM
* RCNN, Fast-RCNN, Faster-RCNN, Mask-RCNN