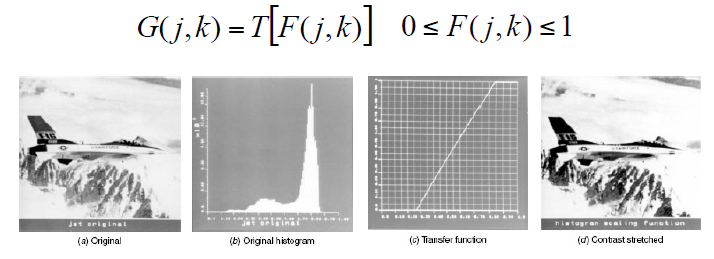
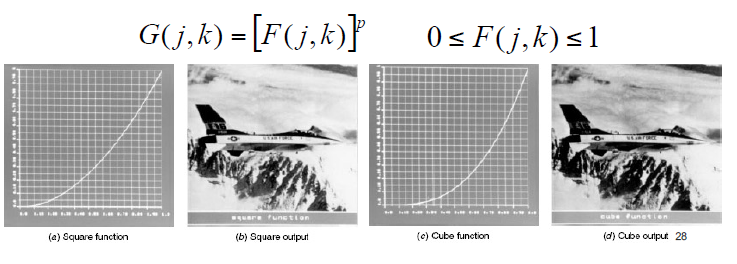
## **1. transfer function**

**Linear**

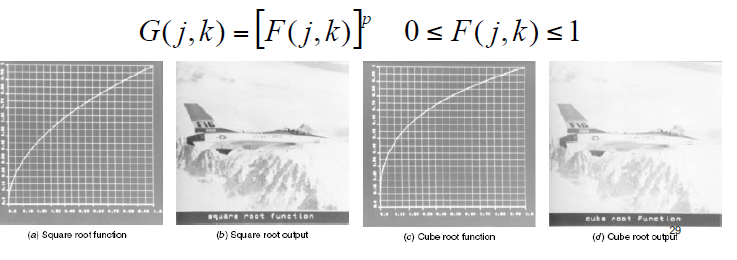


**Nonlinear : Power-Law**

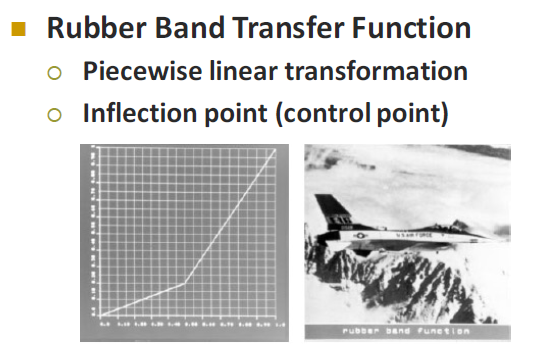
**Low contrast is suppression, but high contrast dynamic range of grey level is extended.**



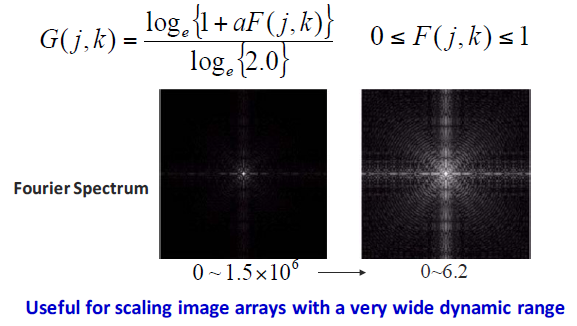
Power law : the low contrast is extended, but the high contrast is supporession.



Piecewise



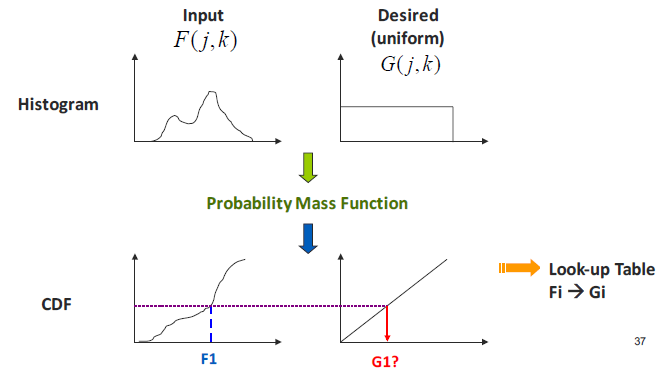
**Logarithmic Point Transformation**



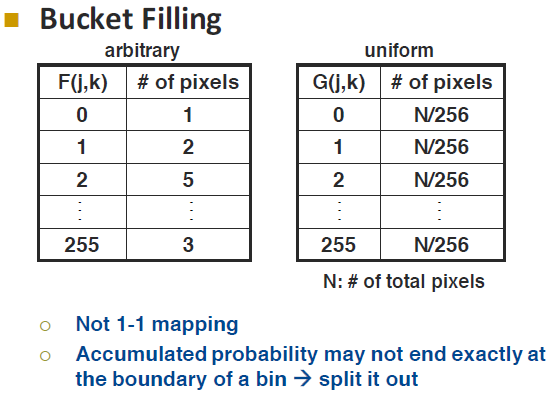
## **2. histogram equalization:**

PMF -> uniform, CDF->

**Output histogram is not really uniformly distributed.**



**Bucket filling : each grey level has the same pixels.**



## **3. noise removal/cleaning**

Two types of noise

**Uniform Noise**

Additive uniform noise, Gaussian noise

**Impulse Noise**

Salt and pepper noise

**Solutions**

Uniform Noise : **low-pass** filtering, mean filter

Impulse Noise : **non-linear** filtering like Median filter, pseudomedian filter **PMED**

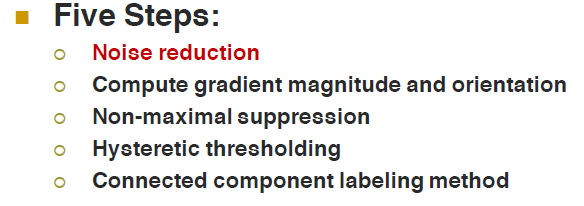
## **4. how to detect edge :**

first order, sobel,

2nd order derivative, zero-crossing,

homogenous is also zero-crossing,

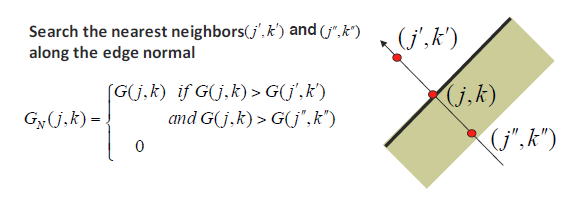
canny edge detection :

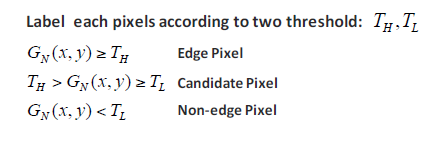


1. noise removal by gaussian filter

2. Gradient magnitude G(j,k)

3. Non-maximal suppression







**GEOMETRICAL IMAGE MODIFICATION**

warping: translation, scaling, rotation

[z,y] => [x, y, 1] **homogenous coordination** to help LA calculation,

**Lecture 4 :morphology**

dilation(close), erosion(open) ,

Shrinking, thinning, skeleton,

# Lecture 6 Digital halftoning : con, pro, how?

## **Patterning : 4 steps**

1. Read in the given grey level image
2. Quantization from 256 level to n\*n+1 level
3. Design the patterning table n\*n
4. Map each pixel to its corresponding pattern

**Pro** : simplest way

**Con** : The generated image has higher spatial resolution than the source image.

## **dithering** :

Create an image with the same number of dots as the number of pixels in the source image.

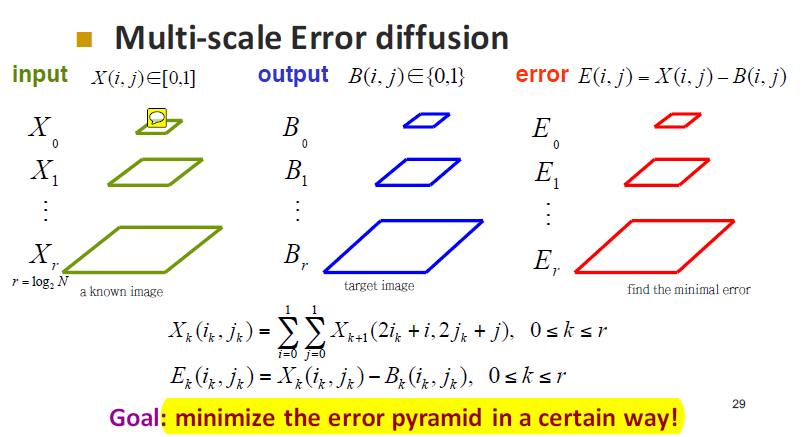
## 

## matrix, order, why?

threshold matrix ??

## **error diffusion**: how?, cycle , flip

triangle, pyramid, multi-scale error diffusion, con, pros, why?



**Pro :** Preserve contrast of the original image, detail is preserved

Does not over-smooth the image

# Lecture 5 Texture Analysis:

Texture classification/segmentation

Two cases :

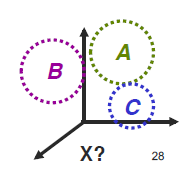
1. Each input is homogeneous that only one texture in the input;
2. Single input consists of more than one texture.

Two approaches:

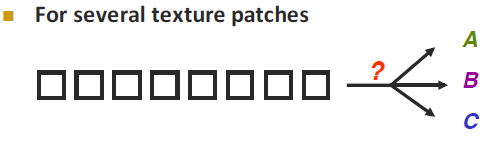
1. Supervised texture classification

For each given texture type, textureA, textureB and textureC … is a known textures which has a known center in the feature space.

Given a feature vector X, Use nearest neighbor classification rule to find the texture.



1. Un-supervised texture classification



Kmeans algorithm is the famous tool to handle unsupervised classification problem.

Good classification

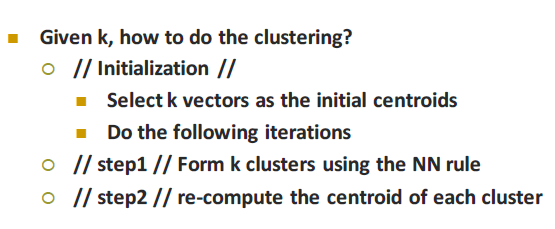
Inter-­‐clustering : largest distance

Intra-­‐clustering smallest distance

How to choose K

Depends on the inter-­‐cluster and intra-­‐cluster statistical analysis

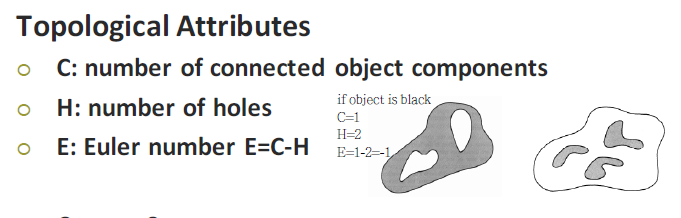
OR by the problem set-­‐up (domain knowledge)



# **OCR :**

attribute, euler number, hole, connected components, connectivity

4,8 connectivity



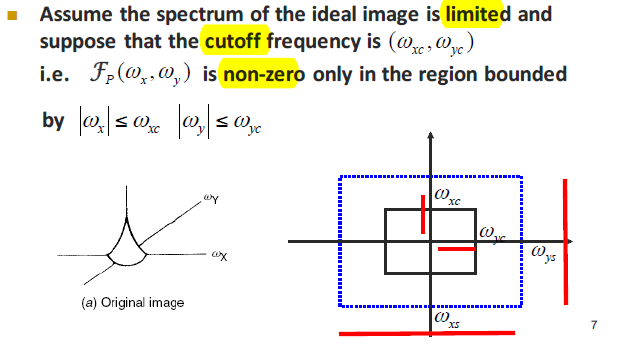
sampling, and transform, nyquist , why?

alias, overlapping

## **Sampling Theorem :** Nyquist Theorem

To sample a **band-limited signa**l, the sampling period must be no longer than one-half of the period of the finest details within the image to **avoid aliasing**. F\_s >= 2\*f\_w

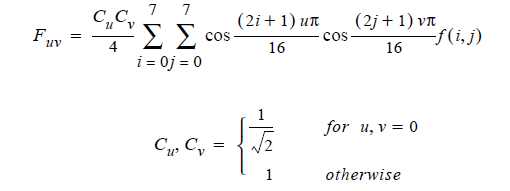
Generalization of the 1D to the 2D case



## **Why image transform?**

1. Feature extraction
2. Energy compaction : We are able to **concentrate energy distribution over a small number of transform coefficients** which can be exploited for **compression purpose**.

## **DCT, discrete cosine transform**



<http://www.whydomath.org/node/wavlets/dct.html>

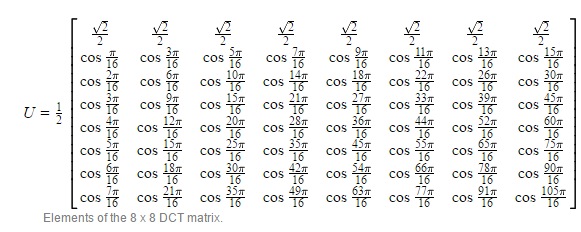
**Advantages** of the DCT

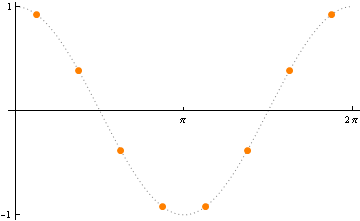
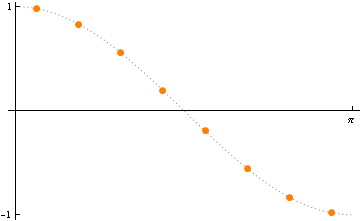
DCT is an effective tool for image compression.

1. The transformation is **orthogonal** (**inverse is transpose** and energy is preserved), fast algorithms can be used for computation,
2. The output for (near) constant matrices generally consists of a large number of (near) zero values.

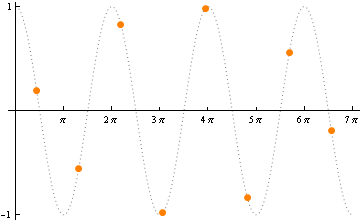
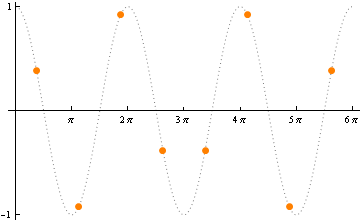
**Disadvantage** of the DCT.

1. While the input from preprocessed 8 x 8 blocks are **integer-valued**, the **output values** are typically real-valued (floating point). Thus we need a **quantization** step to each DCT block and to produce output that is **interger-valued**.





row1 row2



Row6 row7

A constant vector has all equal components. The **heights of the orange points** in any one plot **sum to zero**. That means if we take any constant **vector and dot it with any of rows 1 through 7**, the **result will be zero.** More importantly, if we **dot any of rows 1 through 7 with a near-constant vector**, the **result will be either zero or a value close to zero.**

The DCT, applied to 8 x 8 block A, is B=UAU^T . We first compute **C=UA** and this product is simply an application of U to **each column of A**. If **A is a matrix of near-constant value**s, **elements in C=UA will be (near) zero** as we proceed down the column. The final computation is B=CU^T . Here we are simply applying the columns of U^T (the rows of U ) to the rows of C. So if **elements in C are (near) constant, elements in B will be (near) zero** as we proceed to the right in each row. Putting this all together, we see that in general, the **DCT tends to store information about all 64 input values in a few values** and **"shoves" them to the upper left-hand corner of the output**. The **remaining values are either zero or approximately zero**. This makes the DCT well-suited to the JPEG standard.

## **Hough transform**

To find straight lines, circles, parabolas, ellipses, etc As the curve can be specified in a parametric form.

1. Performed after edge detection
2. All edge points in Cartesian (x,y) to (r,theta) polar coordinate, Hough transform space
3. Quantize Hough transform space, so theta and r are discrete and finite.
4. Generate an accumulator Matrix A(r, theta)
5. For all edge points in the image
   1. Compute all possible (r, theta) lines pass through the edge point from the quantized Hough space
   2. Increase A(r, theta) by one.
6. For all elements in A(r,theta)
   1. Find the maximum value which is at (r, theta)
   2. The pair (r, theta) is the colinear line of the points in image.

Advantages

Tolerate gaps between edges

Relatively unaffected by noise

Can deal with occlusion

