

ECE 178: Image Processing REVIEW

Lecture #20
March 13, 2003

Course Outline

- Introduction
- Digital Images
- Image Transforms
- Sampling and Quantization
- Image Enhancement
- Image/Video Coding
 - JPEG
 - MPEG

Relationship between pixels

■ Neighborhoods

- 4-neighbors (N,S,W,E pixels) == $N_4(p)$.
- Diagonal neighbors == $N_D(p)$.
- 8-neighbors: include diagonal pixels == $N_8(p)$.

■ Connectivity -> to trace contours, define object boundaries, segmentation.

- V : Set of gray level values used to define connectivity; e.g., $V = \{1\}$.

Connected components

- If p and q are in S , p is connected to q in S if there is a path from p to q entirely in S .
- Connected component: Set of pixels in S that are connected; There can be more than one such set within a given S .
- Labelling of connected components.

Distance Measures

■ What is a Distance Metric?

For pixels p, q , and z , with coordinates (x, y) , (s, t) , and (u, v) , respectively:

$$D(p, q) \geq 0 \quad (D(p, q) = 0 \text{ iff } p = q)$$

$$D(p, q) = D(q, p)$$

$$D(p, z) \leq D(p, q) + D(q, z)$$

Linear systems-review

2D impulse function

Line function

Step function

Linear systems and Shift invariance

Impulse Response of LSI Systems

2-D Convolution

2-D Digital Transforms

Week II-III

2D FT: Properties

Linearity: $a f(x,y) + b g(x,y) \longleftrightarrow a F(u,v) + b G(u,v)$

Convolution: $f(x,y) \star g(x,y) = F(u,v) G(u,v)$

Multiplication: $f(x,y) g(x,y) = F(u,v) \star G(u,v)$

Separable functions: Suppose $f(x,y) = g(x) h(y)$, Then
 $F(u,v) = G(u)H(v)$

Shifting: $f(x \pm x_0, y \pm y_0) \longleftrightarrow \exp[2\pi (x_0 u \pm y_0 v)] F(u,v)$

2-D DFT

Often it is convenient to consider a symmetric transform:

$$v(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} u(n) W_N^{kn} \quad \text{and}$$
$$u(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} v(k) W_N^{-kn}$$

In 2-D:
consider a
NXN image

$$v(k, l) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m, n) W_N^{km} W_N^{ln},$$
$$u(m, n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k, l) W_N^{-km-ln}$$

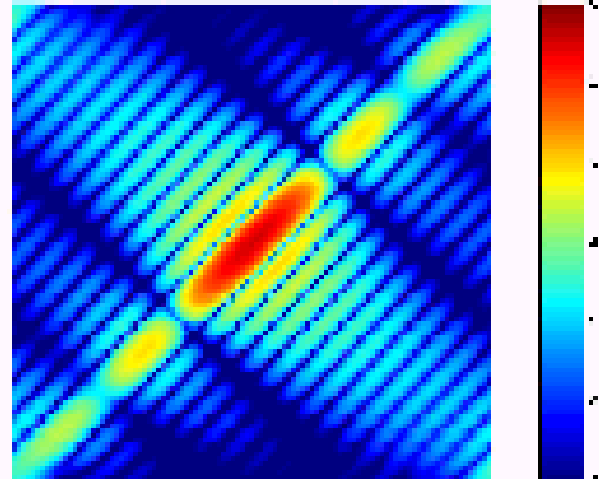
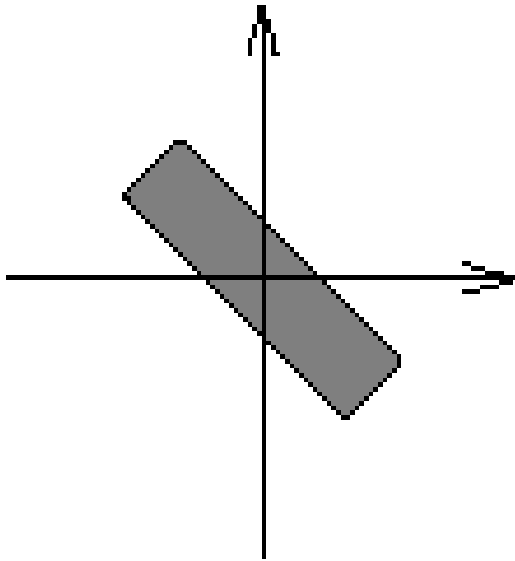
2D DFT Properties: Periodicity & Conjugate Symmetry

$$u(m,n) \xleftrightarrow{F} v(k,l)$$

$$v(k,l) = v(k+N, l) = v(k, l+N) = v(k+N, l+N)$$

If $u(m,n)$ is real, $v(k,l)$ also exhibits conjugate symmetry
 $v(k,l) = v^* (-k, -l)$ or $|v(k,l)| = |v(-k, -l)|$

Rotation



Linear Convolution using DFT

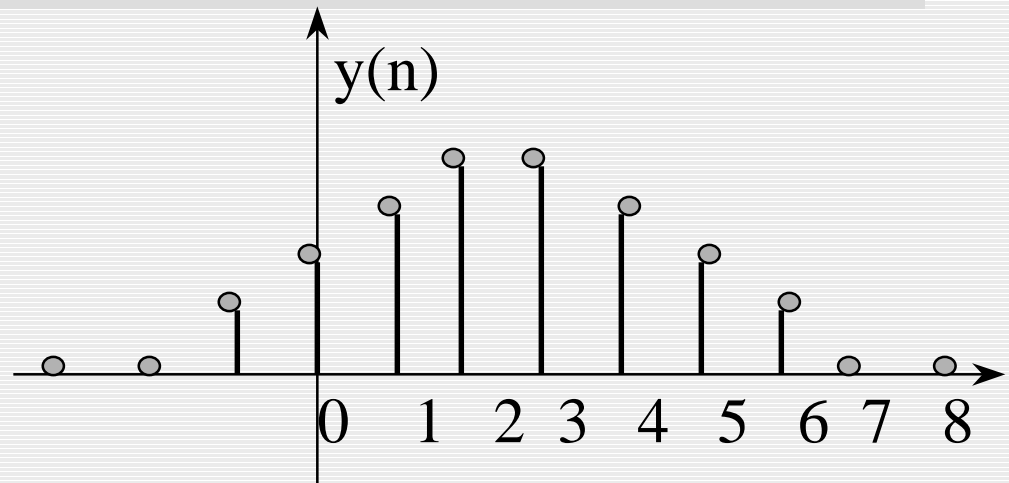
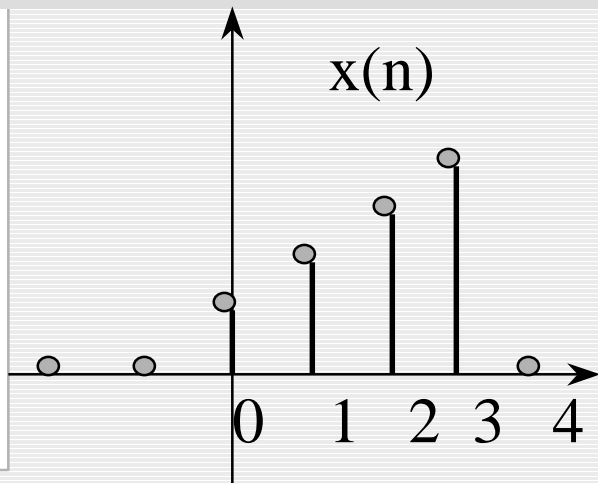
Extended sequences
Circular convolution

Discrete Cosine Transform

Consider 1 - D first; Let $x(n)$ be a N point sequence $0 \leq n \leq N - 1$.

$$x(n) \leftrightarrow \begin{matrix} 2 - N \text{ point} \\ y(n) \end{matrix} \xleftrightarrow{DFT} \begin{matrix} 2 - N \text{ point} \\ Y(u) \end{matrix} \leftrightarrow \begin{matrix} N - \text{point} \\ C(u) \end{matrix}$$

$$y(n) = x(n) + x(2N - 1 - n) = \begin{cases} x(n), & 0 \leq n \leq N - 1 \\ x(2N - 1 - n), & N \leq n \leq 2N - 1 \end{cases}$$



Why DCT?

- Blocking artifacts less pronounced in DCT than in DFT.
- Good approximation to the Karhunen-Loeve Transform (KLT) but with basis vectors fixed.
- DCT is used in JPEG image compression standard.

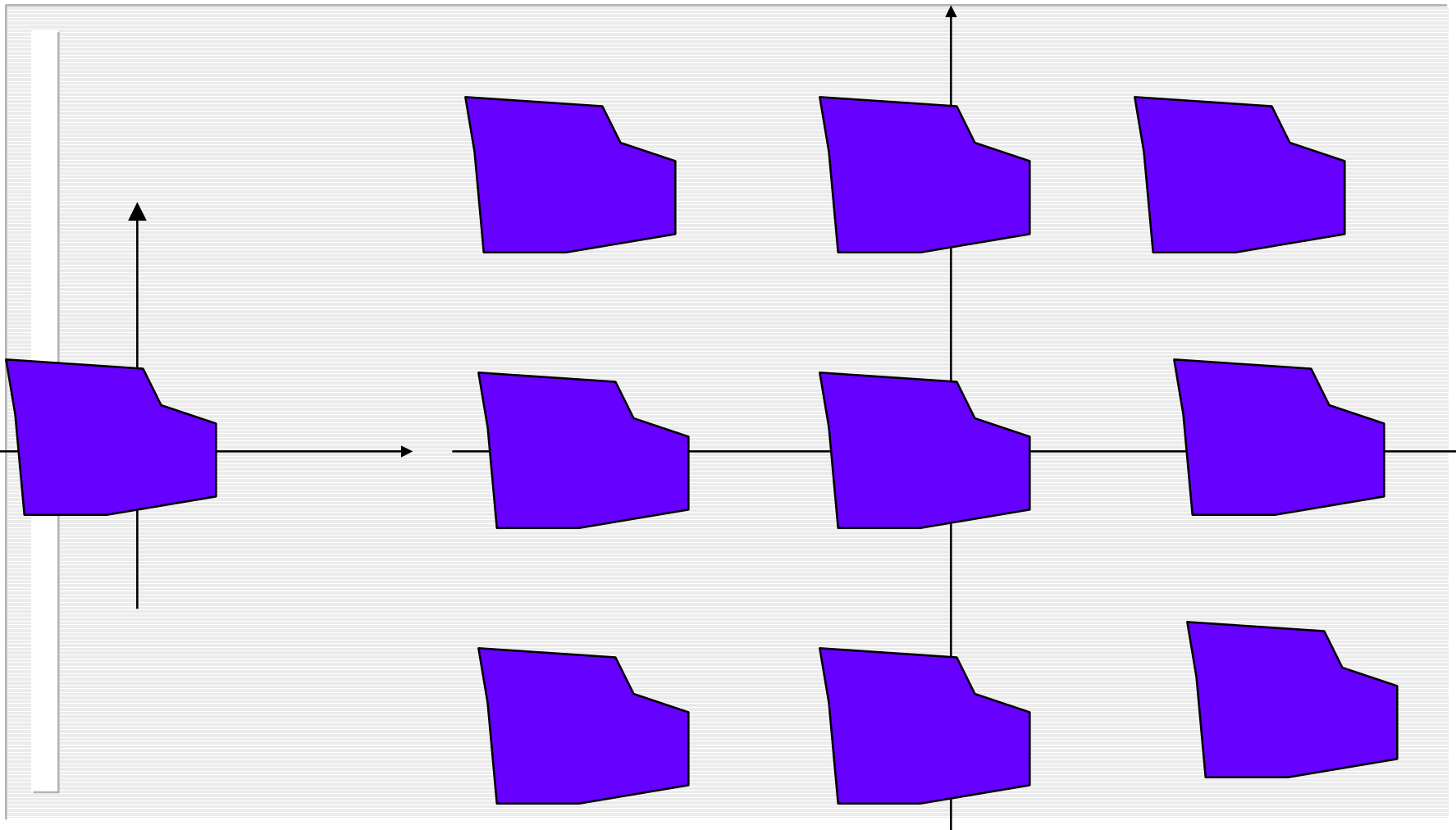
Sampling and Quantization

WEEK IV

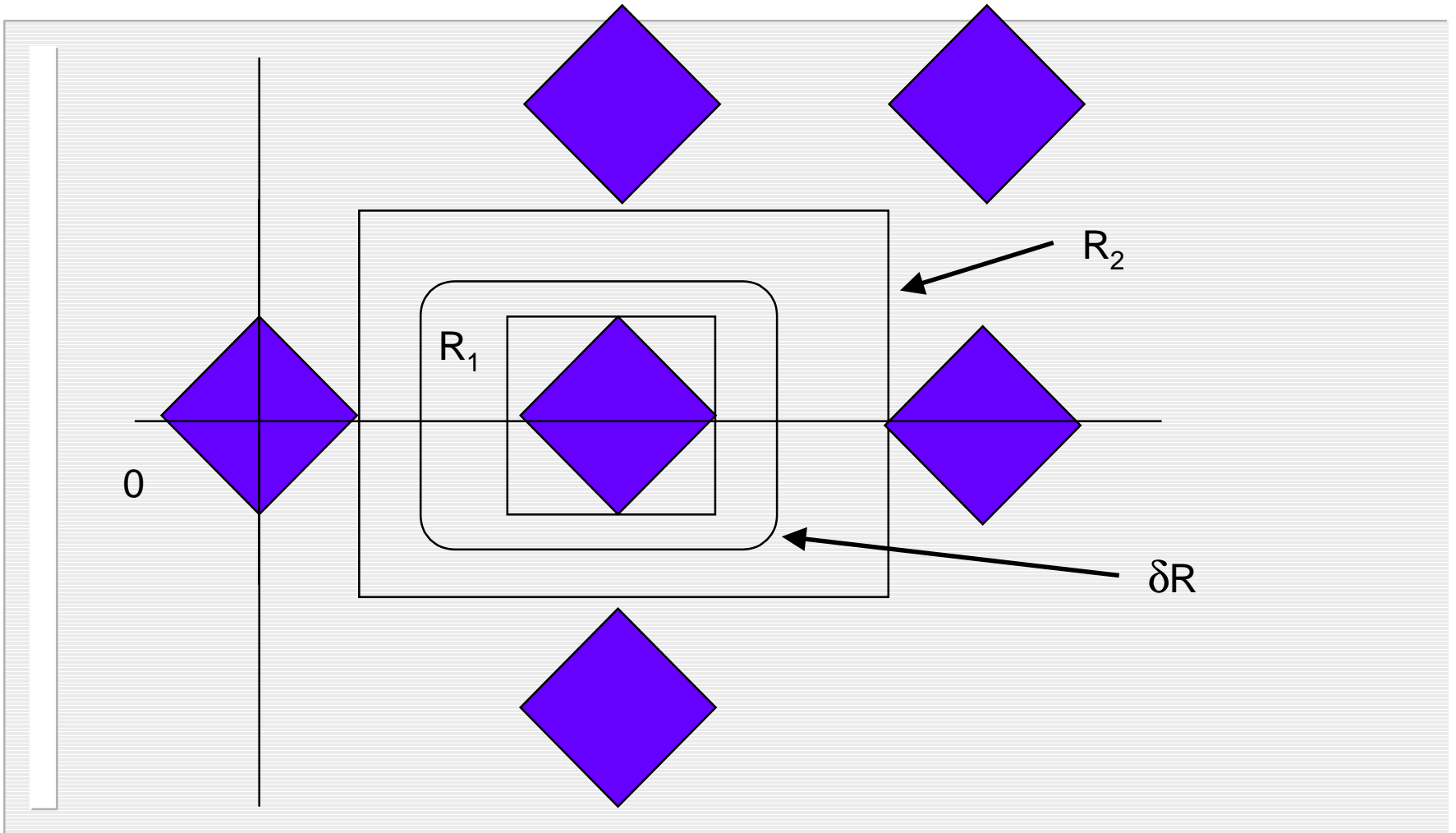
Sampling and Quantization

- 2D sampling theorem
- Optimal quantizer
 - Uniform quantizer

Sampled Spectrum: Example



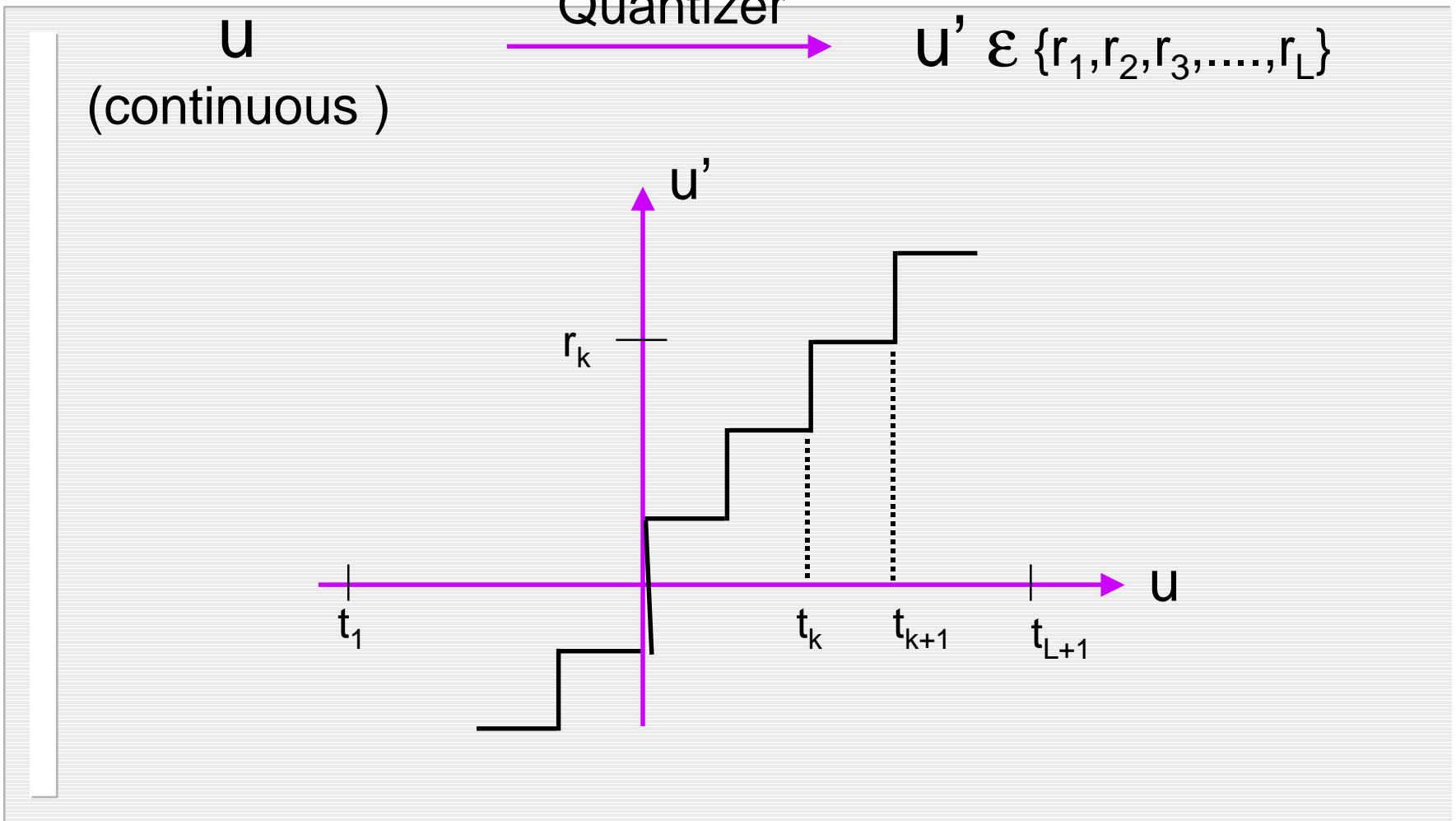
Reconstruction



Quantization

- Optimal quantizer
- Uniform Quantizer

Image Quantization



MMSE Quantizer

Minimise the mean squared error, MSE = Expected value of $(u-u')^2$ given the number of quantization levels L.

Assume that the density function $\mathbf{p}_u(\mathbf{u})$ is known (or can be approximated by a normalised histogram).

Note that for images, u ==image intensity. $\mathbf{p}_u(\mathbf{u})$ is the image intensity ditribution.

Optimum transition/reconst.

- (1) Optimal transition levels lie halfway between the optimum reconstruction levels.
- (2) Optimum reconstruction levels lie at the center of mass of the probability density in between the transition levels.
- (3) A and B are simultaneous non-linear equations (in general)
→ Closed form solutions normally dont exist → use numerical techniques

Image Enhancement

WEEK V-VI

Image Enhancement

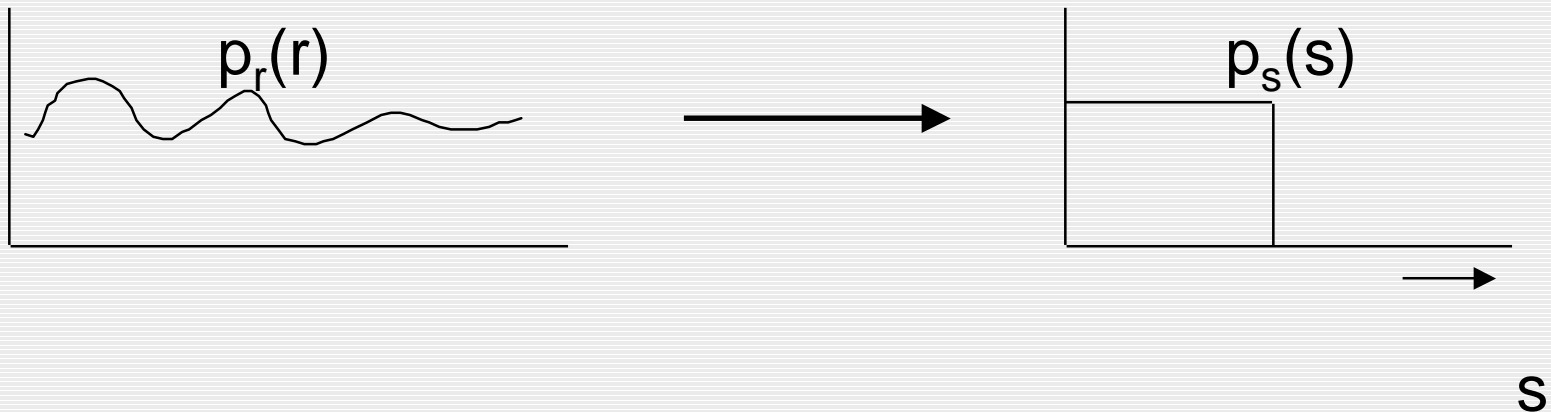
- Point Operations
- Histogram Modification
- Average and Median Filtering
- Frequency domain operations
- Homomorphic Filtering
- Edge enhancement

Histogram Modification

- Histogram Equalization
- Histogram Specification

Equalization (contd.)

We are interested in obtaining a transformation function $T(\cdot)$ which transforms an arbitrary p.d.f. to a uniform distribution



Histogram specification

Suppose $s = T(r) = \int_0^r p_r(w) dw$

$p_r(r) \rightarrow$ Original histogram ; $p_z(z) \rightarrow$ Desired histogram

Let $v = G(z) = \int_0^z p_z(w) dw$ and $z = G^{-1}(v)$

But s and v are identical p.d.f.

$\therefore z = G^{-1}(v) = G^{-1}(s) = G^{-1}(T(r))$

Steps:

- (1) Equalize the levels of original image
- (2) Specify the desired $p_z(z)$ and obtain $G(z)$
- (3) Apply $z=G^{-1}(s)$ to the levels s obtained in step 1

Example Problem

Which transformation should be applied in order to change the

histogram $h(b) = \frac{3}{2} \sqrt{b}$ into $t(b) = \frac{\pi}{2} \sin(\pi b)$?

Median filtering

Replace $f(x,y)$ with $\text{median} [f(x', y')]$
 $(x', y') \in \text{neighbourhood}$

- Useful in eliminating intensity spikes. (salt & pepper noise)
- Better at preserving edges.

Example:

10	20	20
20	15	20
25	20	100

→ (10,15,20,20,20,20,20,25,100)

Median=20

So replace (15) with (20)

Invariant Signals

Invariant signals to a median filter:

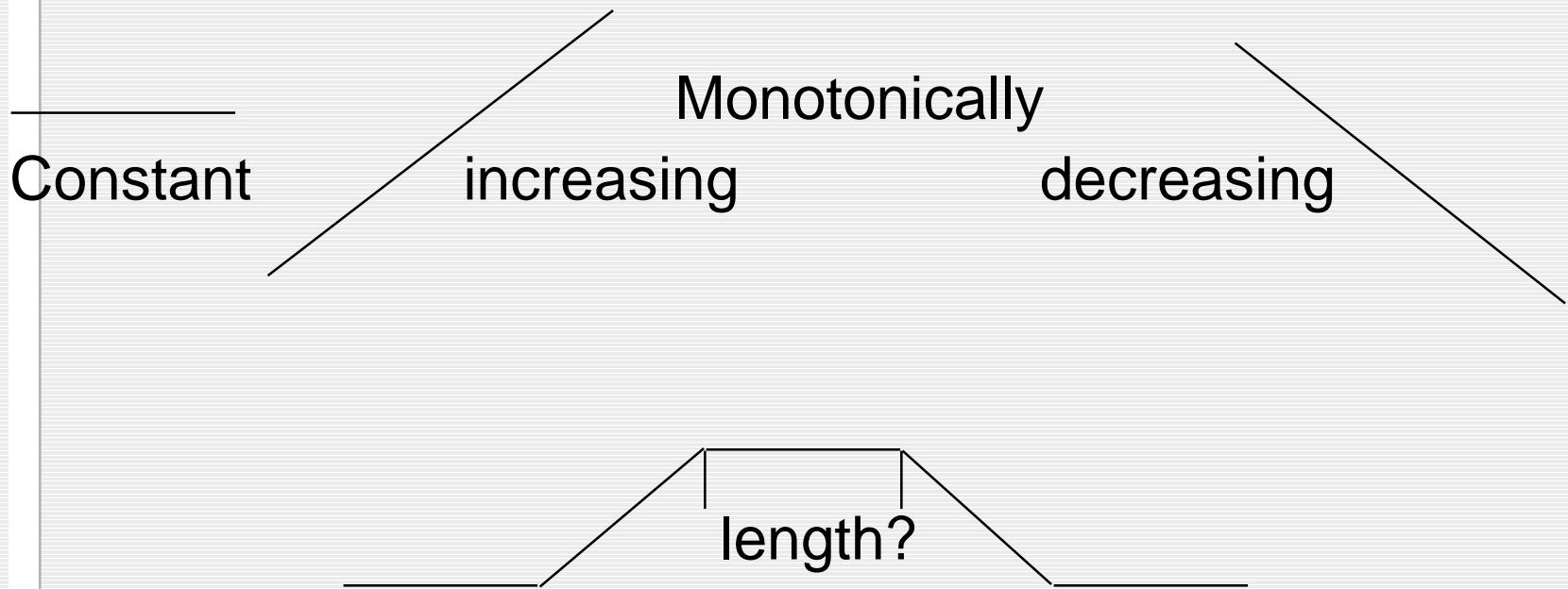


IMAGE COMPRESSION

Week VIII-IX

Image compression

Objective: To reduce the amount of data required to represent an image.

Important in data storage and transmission

- Progressive transmission of images (internet, www)
- Video coding (HDTV, Teleconferencing)
- Digital libraries and image databases
 - Medical imaging
 - Satellite images

IMAGE COMPRESSION

- Data redundancy
- Self-information and Entropy
- Error-free and lossy compression
- Huffman coding, Arithmetic coding
- Predictive coding
- Transform coding

Video Coding

- Motion compensation
- H.261, MPEG-1 and MPEG-2



The End