Introduction to Video Processing

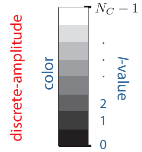
Dr. Deepa Kundur

University of Toronto

Digital Images

Recall:

- discrete-space and discrete-amplitude
- $m = 0, 1, \ldots, N_x - 1$ and $n = 0, 1, \ldots, N_y - 1$
- image consisting of grayscale colors from a finite set $\mathcal{C}$ and indexed via the set: $\{0, 1, 2, \ldots, N_C - 1\}$
- Example: $N_C = 8$ and grayscale values linearly distributed in intensity between black (0) and white ($N_C - 1$)



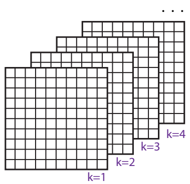
$$I = \begin{bmatrix} 0 & 1 & 2 & 2 & 2 & 2 & 3 & 5 & 7 & 7 \\ 0 & 0 & 1 & 2 & 2 & 3 & 5 & 7 & 7 & 7 \\ 0 & 0 & 0 & 1 & 2 & 3 & 5 & 7 & 7 & 7 \\ 0 & 0 & 0 & 2 & 2 & 3 & 6 & 7 & 7 & 7 \\ 0 & 0 & 0 & 2 & 2 & 3 & 7 & 7 & 7 & 7 \\ 0 & 0 & 0 & 0 & 1 & 2 & 5 & 6 & 7 & 7 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 5 & 6 & 7 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 5 & 6 & 7 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 4 & 5 & 7 \\ 0 & 0 & 1 & 1 & 1 & 1 & 2 & 3 & 4 & 6 \end{bmatrix}$$

Digital Video

- ordered sequence of digital image frames played in succession at a given rate:

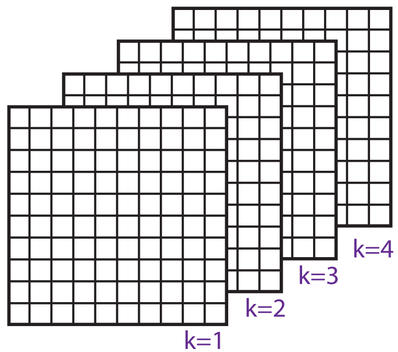
$$I_1, I_2, I_3, \ldots, I_{N_F}$$

$$I_1 = \begin{bmatrix} 0 & 1 & 2 & 2 & 2 & 2 & 3 & 5 & 7 & 7 \\ 0 & 0 & 1 & 2 & 2 & 3 & 5 & 7 & 7 & 7 \\ 0 & 0 & 0 & 1 & 2 & 3 & 5 & 7 & 7 & 7 \\ 0 & 0 & 0 & 2 & 2 & 3 & 6 & 7 & 7 & 7 \\ 0 & 0 & 0 & 2 & 2 & 3 & 7 & 7 & 7 & 7 \\ 0 & 0 & 0 & 0 & 1 & 2 & 5 & 6 & 7 & 7 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 5 & 6 & 7 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 5 & 6 & 7 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 4 & 5 & 7 \\ 0 & 0 & 1 & 1 & 1 & 1 & 2 & 3 & 4 & 6 \end{bmatrix}, I_2 = \begin{bmatrix} 0 & 1 & 2 & 2 & 2 & 2 & 3 & 5 & 7 & 7 \\ 0 & 0 & 0 & 2 & 2 & 3 & 5 & 7 & 7 & 7 \\ 0 & 0 & 0 & 0 & 2 & 3 & 5 & 7 & 7 & 7 \\ 0 & 0 & 0 & 0 & 2 & 3 & 6 & 7 & 7 & 7 \\ 0 & 0 & 0 & 0 & 2 & 3 & 7 & 7 & 7 & 7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 5 & 6 & 7 & 7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 5 & 6 & 7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 5 & 6 & 7 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 4 & 5 & 7 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 3 & 4 & 6 \end{bmatrix}$$



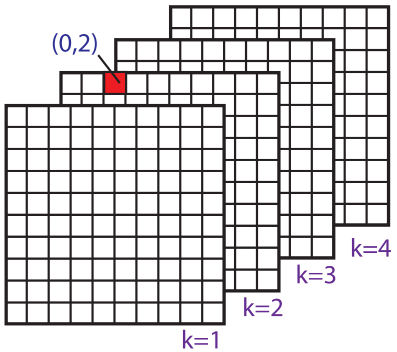
Digital Video

- discrete-space, discrete-amplitude and discrete-time
- $m = 0, 1, \ldots, N_x - 1$ and $n = 0, 1, \ldots, N_y - 1$
- image consisting of grayscale colors from a finite set $\mathcal{C}$ and indexed via the set: $\{0, 1, 2, \ldots, N_C - 1\}$
- $k = 1, 2, \ldots, N_F$



Digital Video

- basic unit of processing is called the pixel denoted by its location in the 3-D video sequence using k, m, n.
- $I_k(m,n) \in \mathcal{C}$.
- Note: $I_2(0,2)$ is shown below:



Digital Video Processing

- Often makes use of the correlation amongst pixels:
  - spatially
  - temporally

Spatial Correlation

Assuming real images:

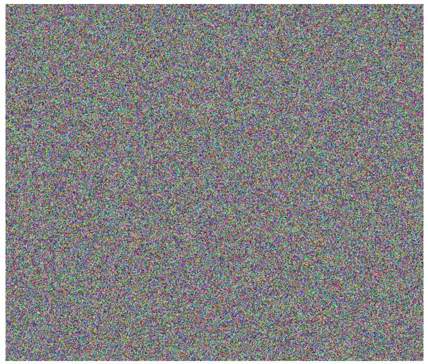
$$R_I(i,j) = \sum_{m,n} I(m,n)I(m-i,n-j)$$

- "high" correlation can mean that pixels within a neighborhood have similar colors
- "zero" correlation can mean that adjacent pixels are unrelated in color

Spatial Correlation

natural image with positive spatial correlation

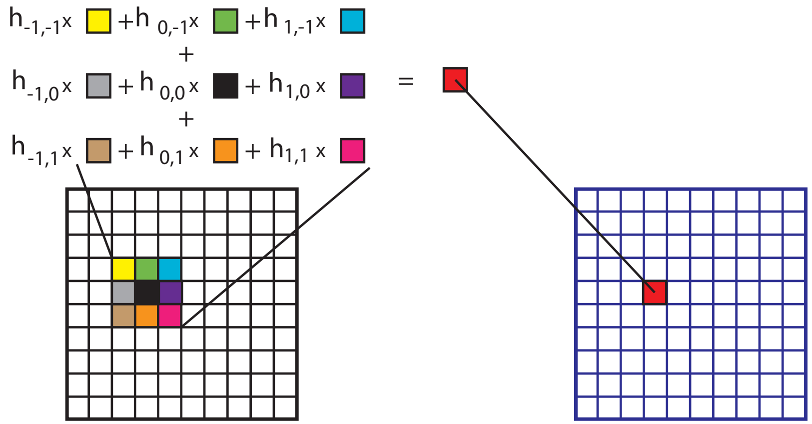
random image with zero correlation



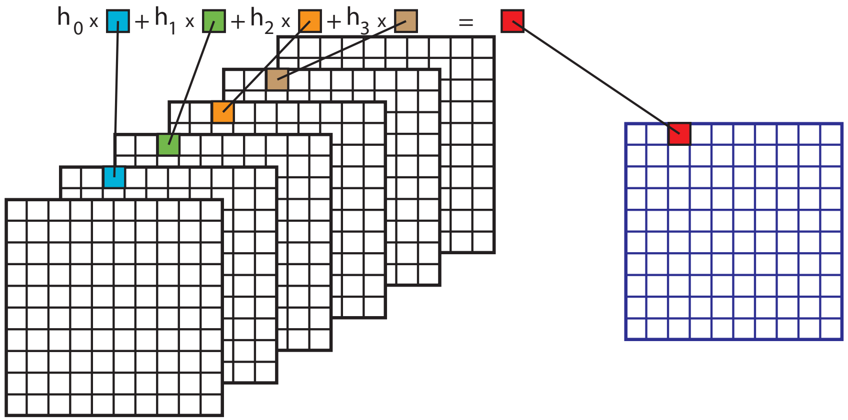
Spatial Correlation

maximum correlation maximum correlation

Spatial Processing



Temporal Processing



Temporal Correlation

maximum correlation uncorrelated

Types of Digital Video Processing

- FIR filtering
- denoising
- enhancing
- restoration
- object identification
- motion detection
- compression
- ...

Edge Detection

- process of identifying "sharp" changes in brightness
- Why is this important?
  - detects changes in properties of the world
  - the image formation process in many applications (seismology, photography, radar, microscopy, ultrasound, magnetic resonance imaging, etc.) is such that changes in brightness can relate to:
    - severe changes in depth/distance
    - discontinuities in surface orientation
    - changes in material properties
    - variations in scene illumination
    - boundaries of an object

Edge Detection

Q: Why conduct edge detection?

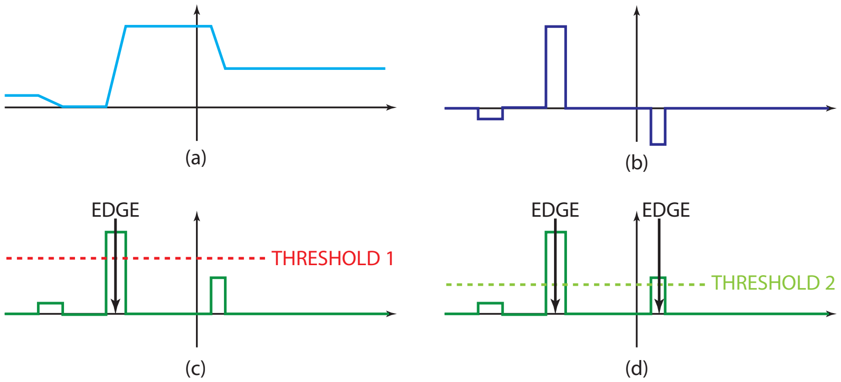
A: Applying edge detection

- identifies discontinuities or sharp boundaries within an image which can significantly help in later interpretation of the image
- reduces the volume of data making subsequent processing (often involving image analysis or computer vision) simpler

Gradient-Based Edge Detection

1. Compute gradient of image to characterize color changes amongst neighboring pixels
   - a gradient measures the difference in values amongst adjacent pixels
2. Compute edge strength by taking the magnitude of the gradient
   - a gradient is more pronounced when the difference amongst adjacent pixel values is larger
3. Threshold to identify where the edges exist
   - a smaller/larger threshold will detect fewer/more edges
   - this "cleans up" the image resulting in a binary result
4. Edge thinning
   - results in edges that are only one pixel in width making subsequent interpretation easier

One-Dimensional Example



(a) Signal, (b) Signal Gradient, (c) Absolute Value of Signal Gradient with Edge Detection using Threshold 1, (d) Absolute Value of Signal Gradient with Edge Detection using Threshold 2.

Sobel Edge Detection

- Assume an intensity (i.e., grayscale) image.
- Two gradients are computed in the horizontal and vertical directions denoted G_x and G_y, respectively.
- The gradients are computed as follows:

$$\begin{aligned} G_x(m,n) &= M_x(m,n) * I_k(m,n) \\ G_y(m,n) &= M_y(m,n) * I_k(m,n) \end{aligned}$$

where $*$ represents 2-D convolution of the image I_k with the masks:

$$M_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} M_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

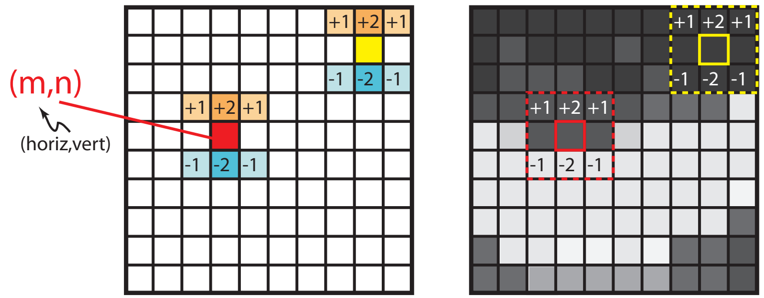
- M_x and M_y represent 2-D FIR filters.

Sobel Edge Detection

Consider $G_x(m,n)$:

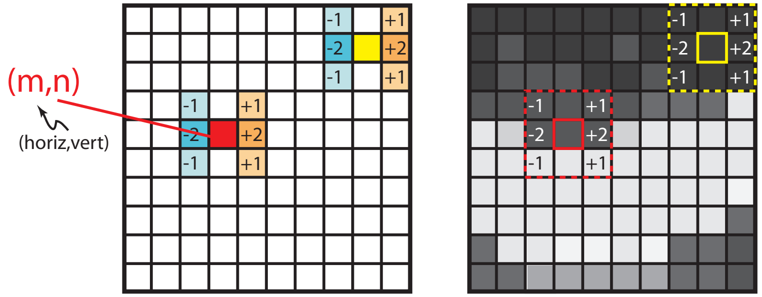
$$\begin{aligned} G_x(m,n) &= M_x(m,n) * I_k(m,n) \\ &= \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} M_x(i,j) I_k(m-i, n-k) \\ &= -1 \cdot I_k(m+1,n+1) + 1 \cdot I_k(m+1,n-1) \\ &\quad -2 \cdot I_k(m,n+1) + 2 \cdot I_k(m,n-1) \\ &\quad -1 \cdot I_k(m-1,n+1) + 1 \cdot I_k(m-1,n-1) \\ &= [I_k(m+1,n-1) + 2I_k(m,n-1) + I_k(m-1,n-1)] \\ &\quad - [I_k(m+1,n+1) + 2I_k(m,n+1) + I_k(m-1,n+1)] \end{aligned}$$

Computation of G_x



$|G_x(m,n)|$ is large if (m,n) lies on the boundary of a horizontal edge.

Computation of G_y



Similarly $|G_y(m,n)|$ is large if (m,n) lies on the boundary of a vertical edge.

Magnitude and Thresholding

Compute $|G(m,n)| = \sqrt{G_x{}^2(m,n) + G_y{}^2(m,n)}$

- If $|G(m,n)| > \mathcal{T}$ then (m,n) lies at an edge boundary.
- Otherwise, (m,n) is not at a boundary.

Magnitude and Thresholding