CS 548: Computer Vision and Image Processing
Digital Image Basics
Spring 2016
Dr. Michael J. Reale
HUMAN VISION
Introduction

- In Computer Vision, we are ultimately trying to equal (or surpass) the human vision system

- Therefore, it is beneficial to have an understanding of how the human vision system works
  - Especially in contrast to cameras
Human Vision System: The Human Eye

Region of maximum visual acuity, densely populated by cones

Outside the fovea, the retina is densely populated by rods
Human Vision System: Cones and Rods

- Two types of receptors
  - Cones
    - Sensitive to brightness and color
    - 7 million cones
    - Most located in fovea
      - Highest visual acuity and true “center” of vision
    - Cone-vision = photopic, bright-light vision
  - Rod (cell):
    - Sensitive to low-level illumination (no color)
    - 100 million rods
    - Rod-vision = scotopic, dim-light vision
Image Perception and Formation
Image Perception and Formation: Eye vs. Camera

- **Camera**
  - **Variable lens-to-sensor distance**
    - I.e., lens moves forwards or backwards to focus image on sensor/film
  - **Lens does not change shape**

- **Eye**
  - **Fixed lens-to-sensor distance**
  - **Lens changes thickness** to change focus
Vision Properties: Brightness and Contrast

- Human vision system dynamically adjusts perceived brightness and contrast based on average local intensity

**Example:**
- (a) Center blocks have *same intensity*, BUT *appear different* (left one seems brighter)
- (b) Centers blocks *appear* to have the *same* intensity, BUT have *different intensities* (left one is darker)
Vision Properties: Brightness Adaptation

- Brightness adaptation
  - Range of intensity levels that human eye can adapt to:
    - Photopic (cones) $\rightarrow 10^{-3}$ mL $- 10^3$ mL
    - Scopotic (rods) $\rightarrow 10^{-3}$ mL $- 10^{-1}$ mL
    - mL = millilumens
  - Cannot adapt the whole range *simultaneously*
Vision Properties:
Brightness Discrimination

- **Brightness discrimination** = the ability to discriminate different intensity level
  - **Weber ratio** = just noticeable difference of intensity versus the background intensity

- The intensity defined in a digital image is *not* the real intensity. It is a contrast scale (e.g., gray scale).
Vision Properties: Contrast

• Types of Contrast:

  ◦ Absolute contrast: $C = \frac{B_{\text{max}}}{B_{\text{min}}}$
    - $B_{\text{max}} = \text{maximum brightness intensity}$
    - $B_{\text{min}} = \text{minimum brightness intensity}$

  ◦ Relative contrast: $C_r = \frac{(B - B_0)}{B_0}$
    - $B = \text{brightness of object}$
    - $B_0 = \text{background brightness}$
Vision Properties: Mach Band Effect

- “Mach Band” effect
  - Over-shooting effect because of perceived increased contrast on edges
  - Gives “scalloped” appearance
Vision Properties: Spatial Discrimination (SD)

- Spatial discrimination (SD)
  - Minimum view angle which can discriminate two points on the object being viewed
  - \( \frac{d}{2 \pi L} = \frac{\theta}{360} \)
Vision Properties: What Affects Spatial Discrimination (SD)

- Low illumination $\rightarrow$ SD decreases
- Low contrast $\rightarrow$ SD decreases
- Too high illumination $\rightarrow$ SD does not increase too much

- SD of color is weaker than SD of intensity

- Projection on fovea $\rightarrow$ SD increases
Human vision model can be modeled with $g(x, y) = T \left[ f(x, y) \right]$

- $T$ = transforms input optical scene $f(x,y)$ to output image $g(x,y)$
  - Can be linear or non-linear transform
  - $H(u,v)$ = low pass filter (e.g., limited discrimination, linear)
  - Log response to the brightness (e.g., non-linear)
  - Time-delay effect (e.g., “persistence of vision”)
Electromagnetic Spectrum

- Electromagnetic spectrum
  - Varies in wavelength
  - Visible light only a fraction of spectrum
IMAGE ACQUISITION
Image Acquisition: Sensor Mechanics

- Principle of imaging sensor
  - Transform (illumination energy) → (digital image)
  - Output voltage waveform proportional to light
Image Acquisition: Image Digitizing

- **Sampling** = digitizing the *coordinate* values (spatially)
  - Ideally, sampling at **Nyquist Rate**: $2*F_{\text{max}}$
    - $F_{\text{max}}$ is the maximum frequency
  - Limited by the number of sensors
  - May be uniform or non-uniform (e.g., fovea-based, “fish-eye” lens)

- **Quantization** = digitizing the *amplitude* values
  - May be uniform
    - I.e., evenly divide range of intensity in N slots
  - May be non-uniform
    - Based on image characteristics
Image Sensing: Types of Sensors

- **Single sensor**
  - Must move sensor/object in at least two dimensions OR use mirrors
  - “Infinite” possible resolution

- **Line sensors**
  - Must move sensor/object in at least one dimension
  - Sensor strips (scanners) and CT/MRI
  - Finite resolution in one dimension; “infinite” in others

- **Array of sensors**
  - 2D array CCD
  - Finite resolution in all dimensions
Image Sensing: Sensor Motion

- Linear motion
- Rotation
- Sensing ring to create cross-sectional images
  - Example: CT scan with X-rays

"UPMCEast CTscan" by daveynin from United States - New UPMC East Uploaded by crazypaco. Licensed under CC BY 2.0 via Wikimedia Commons - https://commons.wikimedia.org/wiki/File:UPMCEast_CTScan.jpg#/media/File:UPMCEast_CTScan.jpg
Image Acquisition: How much space to represent a digital image?

- Given a digital image with:
  - M rows (y coordinate)
  - N columns (x coordinate)
  - k bits per pixel

- Number of pixels total = M*N
- Number of possible gray levels per pixel = 2^k

- Total size = M*N*k
Image Acquisition: Image Data Size

- **Example:**
  - \( N = 640, M = 480, k = 8 \)
  - Possible gray levels = \( 2^8 = 256 = [0,255] \)

- **Total size =**
  - \( 640 \times 480 \times 8 = \)
  - 2457600 bits =
  - 307200 Bytes =
  - 0.3 MB
Image Acquisition: Image Coordinates vs. World Coordinates

- For an image $f(x,y)$, the coordinates $(x,y)$ are **image coordinates**
  - They are NOT the same as world coordinates (that is, the coordinates of the object in the real world)
  - Concept often encountered in computer graphics (projection plane becomes the image drawn to the screen)

[Image](http://upload.wikimedia.org/wikipedia/commons/0/0b/Blender3D_ViewCoordinatesProjectionPlane.jpg)
Image Acquisition: Gray Level vs. Illumination Values

- Likewise, the gray level value from $f(x,y)$ does NOT necessarily refer to the true illumination value
  - Gray level value is an INDEX to an illumination value
  - May have non-uniform mapping from illumination value to gray level value

Example: intensity mapped to height above sea level
Image Acquisition: Spatial Resolution

- Spatial resolution can be measured by:
  - Number of pixels with respect to the image size
    - E.g., DPI = dots per inch
  - Number of line pairs = smallest discernible detail per unit distance in an image
    - E.g., 100 lp/mm
Image Acquisition: Spatial (N) vs. Gray Level (K) Resolutions

• Based on human perception, given an image where the spatial resolution (N) and the gray level resolution (K) are varied independently:
  
  ◦ Increasing N and K $\rightarrow$ improves quality
  
  ◦ Increasing N but decreasing K $\rightarrow$ increases contrast
    • To subjects, can look like improved quality
Image Acquisition: Spatial (N) vs. Gray Level (K) Resolutions

- For images with lots of detail, K does not much affect perceived quality
Image Quality

- Image quality can be assessed *subjectively* or *objectively*
  - Subjective
    - Humans rate image quality on some kind of scale (e.g., 1-5)
    - Often used for evaluation of image enhancement, effects, restoration, compression, etc.
  - Objective
    - Compare each pixel against reference image (ground truth)
    - Mean square error E, then use $-10 \log(E)$ to get final score
    - Often used in image coding (encoding then decoding), etc.
Image Transformation: Size Change

• Size change
  ◦ “Zoom in” – take piece of image and make it larger (low to high resolution)
    • Problem: how do you fill in gaps?
      • Pixel replication → nearest neighbor per pixel
      • Pixel interpolation → look at multiple neighbors per pixel and get weighted average
      • “Super-resolution”
  ◦ “Zoom out” – take image and make it smaller (high to low resolution)
    • Problem: how do you know which pixels from old image in new image?
      • Solution is basically to think of the process in reverse → map from pixel in new, smaller image to old, larger image
Image Transformation:

- Similar issue in graphics with texturing
  - Same size as texture → life is good
  - LARGER than texture → MAGNIFICATION
  - SMALLER than texture → MINIFICATION
Image Aliasing

- Aliasing problem
  - “Jagged” or staircase effects
  - Occurs during
    - Image acquisition
    - Image resizing/zooming
    - Display in computer graphics
  - Reason:
    - Sampling/displaying resolution lower than the minimum rate $2\times F_{\text{max}}$, which is the Nyquist rate
    - $F_{\text{max}} = \text{maximum frequency}$
Image Transformation: Size Change

- Nearest neighbor $\rightarrow$ nearest match
- Bilinear interpolation $\rightarrow$ 4 nearest neighbors
- Bicubic interpolation $\rightarrow$ 16 nearest neighbors
**Image Transformation: Shape Change**

- **Shape change**
  - Also known as a geometric transformation

- Both transforms have similar problems/solutions as texture-mapping in computer graphics.
PIXEL NEIGHBORS AND OPERATIONS
Connected Components: Neighbors

- **Relationship of pixels**
  - Four neighbors of pixel P
    - $N_4(P) \rightarrow$ strong neighbors (North, South, East, West)
    - $N_D(P) \rightarrow$ weak neighbors (diagonals)
  - Eight neighbors of pixel P
    - $N_8(P) = N_4(P) + N_D(P)$

![Diagram showing strong and weak neighbors and their combination to form 8-neighbor set](image)
**Connected Components: Adjacency**

- **Adjacency** \((p = \text{center}; q = \text{possible neighbor})\)
  - Assume binary image = every pixel has a value of 0 or 1
  - Three types:
    - 4-adjacency
    - 8-adjacency
    - \(m\)-adjacency
Connected Components: Adjacency

- **4-adjacency** $\rightarrow$ q is in set $N_4(p)$

```
  q   p
  |   |
  |   |
  |   |
  |
```

- **8-adjacency** $\rightarrow$ q is in set $N_8(p)$

```
  q
  |
  p
  |
  |
  |
  |
  |
```
Connected Components: Adjacency

- **m-adjacency (mixed-adjacency) →**
  1. Pixel q is in set $N_4(p)$
  2. OR
  3. Pixel q is in set $N_D(p)$ AND no 1-valued pixels in intersection of $N_4(p)$ and $N_4(q)$

![Diagram of m-adjacency conditions]

- m-adjacent condition 1: q in set $N_4(p)$
- m-adjacent condition 2: q in set $N_D(p)$ AND no 1-values in $N_4(p) \cap N_4(q)$
- NOT m-adjacent q in set $N_D(p)$ BUT 1-value pixel in $N_4(p) \cap N_4(q)$
**Connected Components: Path**

- **Path** = if p and q is connected, there is a path between p and q
  - *m path*: the path between p and q based on m-connected pixels
  - *closed path*: starting p and ending q are connected
Connected Components: Definitions

- **Connected component**
  - Set of pixels which are connected
  - The set is also called a *connected set*

- **Region**
  - R is a region if R is a connected set
  - Boundary of R is a “closed path”

- **Edge**
  - Gray-level discontinuity at a point
    - I.e., perpendicular to edge, intensity changes sharply
  - Linked edge points → *edge segment*
Connected Components: Distance

- **Distance**
  - $D(p, q)$ is defined as the distance between $p$ and $q$
  - *Properties:*
    - $D(p, q) \geq 0$
    - $D(p, q) = D(q, p)$
    - $D(p, q) \leq D(p, z) + D(q, z)$
  - *Types:*
    - Euclidean ($D_e$)
    - City-block or “Manhattan” ($D_4$)
    - Chessboard ($D_8$)
    - Shortest $m$-path ($D_m$)
Connected Components: Distance

- Euclidean distance \((D_e)\)
  - Circular or disk shape
  - \(D_e(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}\)
Connected Components: Distance

- City-block or “Manhattan” ($D_4$) distance
  - Diamond shape
  - $D_4(p,q) = |(x_p - x_q)| + |(y_p - y_q)|$

```
2
2 1 2
2 1 0 1 2
2 1 2
2
```
Connected Components: Distance

- Chessboard ($D_8$) distance
  - Square shape
  - $D_8(p,q) = \max(||(x_p - x_q)||, ||(y_p - y_q)||)$

```
2 2 2 2 2
2 1 1 1 2
2 1 0 1 2
2 1 1 1 2
2 2 2 2 2
```
Connected Components: Distance

- Shortest m-path ($D_m$) distance between two points

$D_m = 8$
Pixel Operations

- Pixel operations
  - Point-wise operations
  - Can operate on one or two images
  - Treat image as an $M \times N$ matrix
  - In the slides that follow:
    - $f$ and $g =$ images
    - $(x,y) =$ the pixel in the $x$-th column and $y$-th row
    - $\ast =$ undefined value
Pixel Operations: Arithmetic

- **Condition 1 (C1)** = \( f(x,y) \neq * \) and \( g(x,y) \neq * \)

- **ADD\([f, g](x,y)\)**
  - \( = f(x,y) + g(x,y) \)  IF C1
  - \( = * \)  otherwise

- **Mult\([f,g](x,y)\)**
  - \( = f(x,y) \cdot g(x,y) \)  IF C1
  - \( = * \)  otherwise

- **SCALAR\([t; f](x,y)\)**
  - \( = t \cdot f(x,y) \)  IF \( f(x,y) \neq * \)
  - \( = * \)  otherwise
Pixel Operations: Arithmetic

- **Max**[f,g](x,y)
  - $\text{Max}[f,g](x,y) = \max[f(x,y), g(x,y)]$ IF C1
  - $\text{Max}[f,g](x,y) = *$ otherwise

- **Min**[f,g](x,y)
  - $\text{Min}[f,g](x,y) = \min[f(x,y), g(x,y)]$ IF C1
  - $\text{Min}[f,g](x,y) = *$ otherwise

- **Sub**[f](x,y)
  - $\text{Sub}[f](x,y) = -f(x,y)$ IF $f(x,y) \neq *$
  - $\text{Sub}[f](x,y) = *$ otherwise
Pixel Operations: Arithmetic

- 
  
  \[ \text{EXTEND}[f,g](x,y) \]
  
  \begin{align*}
  \circ = f(x,y) & \quad \text{IF } f(x,y) \neq \ast \\
  \circ = g(x,y) & \quad \text{otherwise}
  \end{align*}

- 
  
  \[ \text{EXTADD}[f,g](x,y) \]
  
  \begin{align*}
  \circ = \text{ADD}[f,g](x,y) & \quad \text{IF } C1 \\
  \circ = f(x,y) & \quad \text{IF } f(x,y) \neq \ast \text{ and } g(x,y) = \ast \\
  \circ = g(x,y) & \quad \text{IF } g(x,y) \neq \ast \text{ and } f(x,y) = \ast \\
  \circ = \ast & \quad \text{Both } g \text{ and } f \text{ on undefined domain}
  \end{align*}
Pixel Operations: Arithmetic

- **THRESH**:\( [f, t](x, y) \)
  - \( = 1 \quad \text{IF} \ f(x, y) \geq t \)
  - \( = 0 \quad \text{IF} \ f(x, y) < t \)
  - \( = * \quad \text{IF} \ f(x, y) = * \)

- **TRUNC**:\( [f, t](x, y) \)
  - \( = f(x, y) \quad \text{IF} \ f(x, y) \geq t \)
  - \( = 0 \quad \text{IF} \ f(x, y) < t \)
  - \( = * \quad \text{IF} \ f(x, y) = * \)

- **Note**: \( \text{TRUNC}[f, g](x, y) = \text{Mult}[f, \text{THRESH}(f, t)] \)
Pixel Operations: Arithmetic

- **EQUAL[f,t](x,y)**
  - $= 1$ IF $f(x,y) = t$
  - $= 0$ otherwise
  - $= *$ on the undefined domain

- Similar definition for:
  - **GREATER[f,t](x,y)**
  - **BETWEEN[f, t1, t2](x,y)**

- Operation with masking: AND, OR, NOT.
Pixel Operations: Arithmetic

- \( \text{PIXSUM}(f) \) = the summation of all pixels on the defined domain

- \( \text{DOT}(f,g) = \text{SUM}[f(x,y) \times g(x,y)] \) on the common domain

- Norm\((f)\)
  - \( = \sqrt{\text{SUM}[f(x,y)^2]} \)
  - OR equivalently
  - \( = \sqrt{\text{DOT}(f,f)} \)

- REST\([f,g](x,y)\)
  - \( = f(x,y) \) IF \( g(x,y) \neq * \)
  - \( = * \) IF \( g(x,y) = * \)
Pixel Operations: Linear vs. Nonlinear

- Given:
  - H: operator
  - f, g: images
  - a, b: scale values

- For linear operations:
  - \( H(af + bg) = aH(f) + bH(g) \)

- This is NOT true for non-linear operations
  - E.g., \(|f-g|\) operation
IMAGE FORMATS
There are many image formats, but here are some of the most popular ones:

- TIFF
- GIF
- JPEG
- BMP
Image Formats: TIFF

• TIFF
  • Extension: *.TIF
  • “Tagged Image File format”
  • Uses LZW (lossless coding)
  • Very popular for commercial work and archiving images
Image Formats: GIF

- GIF
  - Extension: *.GIF
  - “Graphics Interchange Format”
  - Uses LZW (lossless coding)
  - Color quality poor (indexed color)
Image Formats: JPEG

- JPEG
  - Extension: *.JPG
  - “Joint Photographic Experts Group”
  - Very small size
  - Lossy compression
Image Formats: BMP

- BMP
  - Extension: *.BMP
  - Standard on Windows systems but supported by others
  - Different varieties (all VERY simple):
    - PBM → portable bitmap file format (binary)
    - PGM → portable greymap (grey scale)
    - PPM → portable pixmap (color)