ELIC 629
Digital Image Processing

Winter 2005

Digital Image Fundamentals: Image Sampling (cont.) & Basic Relationships Between Pixels

Bill Kapralos

Tuesday, January 24 2006

ELIC 629, Winter 2006, Bill Kapralos

Overview (1):

- **Before We Begin**
  - Administrative details
  - Review → some questions to consider
- **Image Sampling and Quantization (cont…)**
  - Spatial and gray-level resolution
  - Aliasing
- **Zooming and Shrinking of Images**
  - Image zooming
  - Image shrinking

Overview (1):

- **Basic Relationships Amongst Pixels**
  - Pixel neighborhoods and adjacency
  - Connectivity
  - Regions
  - Boundaries
- **Linear and Non-Linear Operations**
  - Linear operators
  - Non-linear operators

Before We Begin

Administrative Details (1):

- **Miscellaneous Notes (Reminder Once Again)**
  - No access to the lab and its equipment other than during our regularly scheduled lab hours
    - Even if lab is open, no one else can provide you access to the camera equipment
    - Shouldn't be a problem completing labs during your lab hours

---

Administrative Details (2):

- **Lab Two**
  - Will involve the use of a digital camera which I will distribute (along with appropriate equipment) during the lab
  - You can ignore "Procedure C" completely
  - A fairly "straight-forward" lab but may take some time getting used to LabView if you haven't don't so already!
 - Lab report required for this lab

---

Some Questions to Consider (1):

- Why is visual perception important to DIP ?
- List and briefly describe a few of the components of the eye (e.g., retina, iris diaphragm, lens, blind-spot…)
- What is the range of intensities to which the eye is sensitive too ?
- What is brightness adaptation ?
- What is a CCD (briefly describe) ?
- What is sampling and what is quantization ?
- Describe some of the sampling techniques
- How does the color of an object we see come about ?

Image Sampling and Quantization (cont…)

Spatial & Gray-Level Resolution (1):

- **Spatial Resolution**
  - Smallest detail we can perceive ("see") in an image
  - Determined by sampling
  - Common definition → Vertical lines of width W with spacing between lines also W
    - Line and space is referred to as a line pair
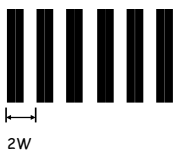
Spatial resolution → smallest number of line pairs per unit distance we can perceive ("see") before they are perceived as one

2W

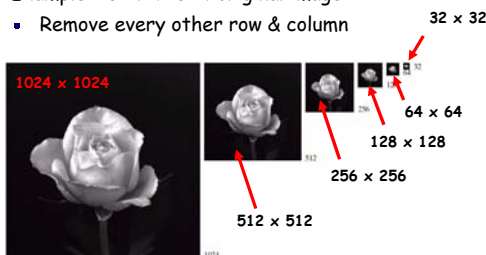## Spatial & Gray-Level Resolution (2):

- **Gray-Level Resolution**
  - Smallest change in gray-level (intensity) we can perceive
  - Remember → number of gray levels typically integer power of two due to hardware considerations
  - Common number of gray levels → 2^8, 2^{16}

Spatial & Gray-Level Resolution (3):

- **Spatial Sub-Sampling**
  - Deleting the appropriate number of rows/columns
  - Example: 1024 x 1024 original image
    - Remove every other row & column

1024 x 1024
512 x 512
256 x 256
128 x 128
64 x 64
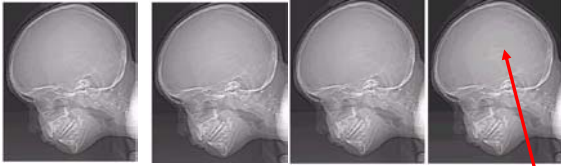32 x 32

Spatial & Gray-Level Resolution (4):

- **Sub-Sampling (cont…)**
  - Many practical computer vision applications
  - Increase performance
  - Can sub-sample to "any" spatial resolution
  - Detail is obviously lost when rows/columns are deleted!

Spatial & Gray-Level Resolution (5):

- **Gray Level Sub-Sampling**
  - Reducing the number of gray-levels (intensities)
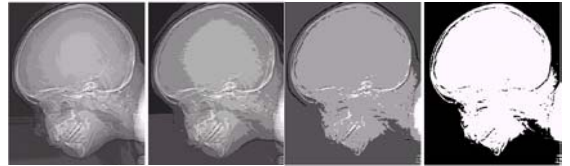

| 256 levels | 128 levels | 64 levels | 32 levels |

- 128 and 64 → very little difference
- 32 → "artifacts" start to become apparent especially in smooth gray level areas

Spatial & Gray-Level Resolution (6):

- **Gray Level Sub-Sampling (cont…)**
  - Further examples

| 16 levels | 8 levels | 4 levels | 2 levels (binary) |

- Difference is now evident!

Aliasing (1):

- **What is Aliasing ?**
  - Occurs when we "under-sample" an image
  - Review → Fourier analysis gives frequency content of a signal, including an image (image is also a signal!)
    - Nyquist theorem → sample signal at a rate twice the highest frequency component of signal in order to reconstruct original signal
    - Under-sampling → not sampling at a high enough rate (e.g., sample frequency too low)

Aliasing (2):

- **What is Aliasing ? (cont…)**
  - Ambiguity → cannot un-ambiguously determine all frequency components of our image
  - One or more frequency components have (incorrectly) been classified (aliased) to another frequency component
    - View it as the introduction of "new" additional frequency components which may not necessarily be present in the original (analog) image

Zooming and Shrinking

Zooming & Shrinking Images (1):

- **Related to Sampling & Quantization**
  - Zooming → over-sampling e.g., enlarging image
  - Shrinking → under-sampling e.g., smaller image
  - Difference → zooming and shrinking applied to digital image unlike sampling and quantization

Zooming (1):

- **Zooming – Two Steps**
  1. Creation of new pixel locations
  2. Assignment of gray-levels to these new areas

  - Example → original image 500 x 500 pixels
    - "Zoom" so that image is 750 x 750 (1.5 times)
    - Concept
      1. imagine overlaying 750 x 750 grid on original image (spacing in grid less than one pixel)
      2. To assign gray levels to new image pixels, look at pixel in original image closest to new pixel location and take its intensity value
   3. Expand "new" image to actual e.g., 750 x 750

Zooming (2):

- **Zooming – Two Steps (cont…)**
  - Example presented is an example of "neighbor" interpolation or pixel replication
    - Interpolation → to introduce information (data etc.) between parts (samples)
    - Valid when we want to increase image size by integer factors e.g., 500 x 500 to 750 x 750
    - For example, to double image size → duplicate each column then duplicate each row
    - Although simple to implement and perform, not a good approach especially for large zoom factors
    - Produces checkerboard effect

Zooming (3):

- **Zooming – Two Steps (cont…)**
  - Example of the checkerboard effect
  - New (zoomed) image size is 1024 x 1024
    - Original image sizes:
    - a) 128 x 128 b) 64 x 64 c) 32 x 32
  - Notice the checkerboard effect !
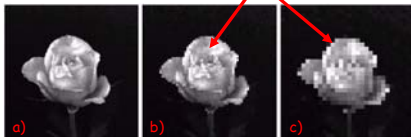
Zooming (4):

- **Zooming – Bi-linear Interpolation**
  - More sophisticated interpolation method
  - Uses information from the four nearest pixel neighbors as opposed to one in previous technique
  - Mathematically:

    $(x', y') \rightarrow$ zoomed image coordinates

    $v(x', y') \rightarrow$ gray-level assignment to zoomed image coordinates

    $$v(x', y') = ax + by + cxy + d$$
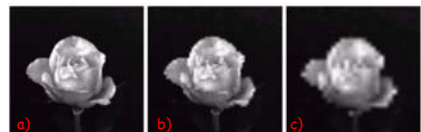
Zooming (5):

- **Zooming – Bi-linear Interpolation (cont…)**
  - Four coefficients a,b,c,d determined from the four equations in four unknowns using the four nearest neighbors of (x, y)

Zooming (6):

- **Zooming – Bi-linear Interpolation (cont…)**
  - Example: New (zoomed) image size is 1024 x 1024
    - Original image sizes:
    - a) 128 x 128 b) 64 x 64 c) 32 x 32
  - Less checkerboard effect than "naïve" (simple) interpolation method

Shrinking (1):

- **Image Shrinking**
  - Very similar to image zooming except we now delete rows and columns instead of adding new ones
  - Using grid example, we now expand the grid and place it over original image and after we choose the appropriate gray levels, we then shrink to desired size
- **Finally**
  - Many, more complicated interpolation techniques exist which use even more pixel neighbors

Basic Pixel Relationships and Common Definitions

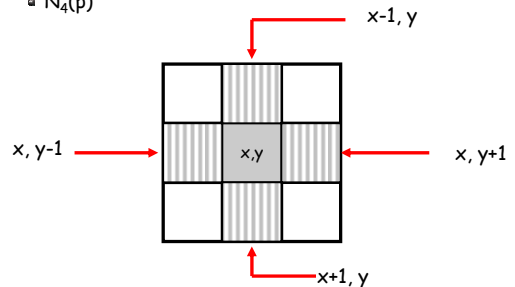## Neighbors & Adjacency (1):

- **Neighbors of Pixel at Coords (x,y)**
  - Pixels at a unit distance from (x,y)
  - Be careful at image borders!

x-1, y
x-1, y-1
x-1, y+1
x, y-1
x,y
x, y+1
x+1, y-1
x+1, y+1
x+1, y
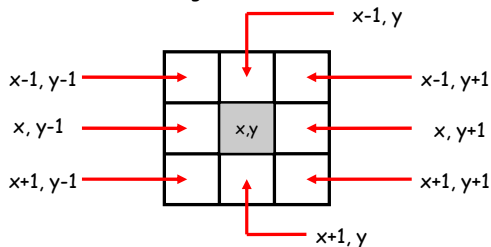
Neighbors and Adjacency (2):

- **Neighbors of Pixel at Coords (x,y) (cont..)**
  - Horizontal neighbors → 4-neighbors of pixel x,y
  - $N_4(p)$

x-1, y
x, y-1
x,y
x, y+1
x+1, y

Neighbors and Adjacency (3):

- **Neighbors of Pixel at Coords (x,y) (cont…)**
  - Diagonal neighbors $N_D(p)$
  - 8-neighbors of pixel x,y → diagonal neighbors + horizontal neighbors → again, careful at borders!

x-1, y-1
x-1, y+1
x,y
x+1, y-1
x+1, y+1
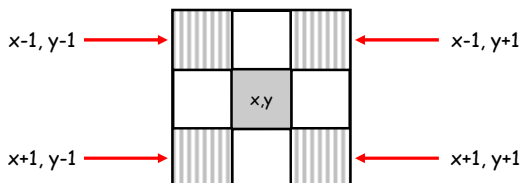
Neighbors and Adjacency (4):

- **Adjacency**
  - Adding an additional constraint to the "neighbors"
    - Intensity of neighbors must now also satisfy some criterion as well (e.g., same gray level)
    - Idea → two pixels p,q, will be adjacent if they are neighbors and their intensity values satisfy some criterion
    - Let V denote the set of gray values used to denote adjacency
      Binary image → V = {1}
      Gray-scale image → range of gray values from 0 to 255

Neighbors and Adjacency (5):

- **Three Types of Adjacency**
  - 4-adjacency
    - Two pixels p,q, with gray levels from V that are also 4-neighbors
  - 8-adjacency
    - Two pixels p,q, with gray levels from V that are also 8-neighbors
  - m-adjacency
    - Two pixels p,q, with gray levels from V that are
      a) 4-neighbors or
      b) "diagonal" neighbors & $N_4(p) \cap N_4(q)$ has no pixels values that are from V

Path (1):

- **Definition**
  - A sequence of indices (x_0, y_0), (x_1, y_1) … (x_n, y_n) beginning from pixel (x_0, y_0) and ending at pixel (x_n, y_n) such that the pixel at (x_i, y_i) is adjacent to pixel (x_{i+1}, y_{i+1}) for all i s.t. $0 \le k \le n$
    - If $(x_0, y_0) = (x_n, y_n)$ then its known as a closed path
    - 8-path → all 8-adjacency
    - 4-path → all 4-adjacency

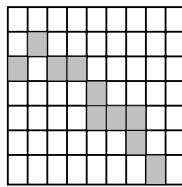## Path (2):

- **Graphical Illustration**

Shaded pixels comprise the path

Two pixels along the path comprise a link
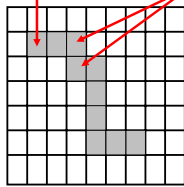
4-path   8-path

Connected (1):

- **Definition**
  - Let S be a subset of pixels in an image
  - Two pixels p and q are connected if there exists a path between them consisting of only pixels within S
  - Connected component (very important!)
    - For any pixel in S, the set of pixels connected to it
    - Connected set → when it contains only one connected component

Boundary and Region (1):

- **Region**
  - A connected set is also known as a region
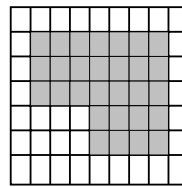- **Boundary (or Border or Contour) of Region R**
  - The set of pixels in the region that have one or more neighbors that are not in R
  - Normally, pixels of the boundary are actually part of the region
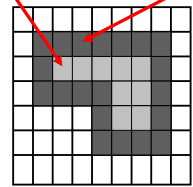 - Inner pixels → pixels of region not on boundary

Region and Boundary (2):

- **Graphical Illustration**

Inner pixels   Boundary

Region   Region with boundary

Distance Measure (1):

- **In Many Applications, Necessary to Find Distance d(p,q) Between Two Pixels p, q**
    - No unique method of defining distance between two pixels (or two components) in image
    - Many different distance measures exist and used
    - Each distance metric MUST satisfy the following otherwise not a distance measure! For pixels p,q,r
        1. $d(p,q) \geq 0$ & $d(p,q) = 0$ iff $p = q$
        2. $d(p, q) = d(q,p)$
        3. $d(p,r) \leq d(p,q) + d(q,r)$

Distance Measure (2):

- **Some Common Distance Measures**
    - Euclidean
    $$D_{Euclid} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
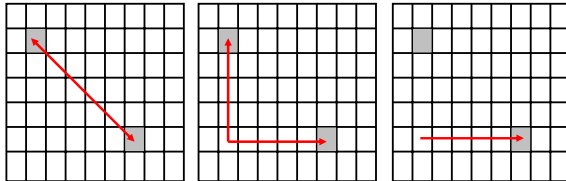   - City-block (or D_4)
    $$D_{City} = |x_2 - x_1| + |y_2 - y_1|$$
    - City-block, chessboard (or D_4)
    $$D_{Chess} = \max(|x_2 - x_1|, |y_2 - y_1|)$$

Distance Measure (3):

- **Some Common Distance Measures (cont…)**
    - Graphical illustration

Euclidean   City-block   Chessboard (max)

Distance Measure (4):

- **Notes Regarding Distance Measures**
    - Euclidean distance is closest to "actual" (continuous) distance but also the most expensive to compute given its square root requirement
        - Results in non-integer output

Linear & Non-Linear Operators

Linear Operators (1):

- **Mathematical Definition**
    - For any two images "f" and "g", two scalars "a" and "b" and operator "H" (whose input and output is an image)

    $$H(af + bg) = aH(f) + bH(g)$$

    - In words → Result of applying operator H to sum of two integers that have been multiplied by some scalar is equal to applying the operator to each image separately, multiplying each image by the appropriate scalar and summing the results

Linear Operators (2):

- **Very Important to Digital Image Processing**
  - Based on well understood theoretical and practical results
  - Usually predictable
  - Can "break" an operator down to several "sub-operators", perform each sub-operator separately and then combine results
    - e.g., useful in networked environments or multi-threaded applications where we can "simultaneously" solve the sub-operators and then combine

Non-Linear Operators (1):

- **Definition**
  - Any operator which does not satisfy the linear operator constraint
  - Any operator that is not linear!
  - Can be more computationally efficient as opposed to linear operators
  - Not well understood theoretically
  - Example: H → absolute value of the difference of two images

$$H(f, g) = |f- g| \neq |H(f)| + |H(g)|$$

Non-Linear Operators (2):

- **Further Examples/Comments**
  - How about the following:
    - H(f) = f^2 → linear or non-linear ??
    - Can you think of any others ?
    - How do we show (prove) an operator is linear or non-linear ?