Cartesian image ---- Log-Polar representation ---- Retinal representation
Rough Idea: Ideal Case

Dirac Delta Function 2D "Comb"

\[ \delta(x,y) = 0 \text{ for } x \neq 0, y \neq 0 \]
\[ \int \int \delta(x,y) \, dx \, dy = 1 \]
\[ \int \int f(x,y)\delta(x-a,y-b) \, dx \, dy = f(a,b) \]
\[ \delta(x-ns,y-ns) \text{ for } n = 1\ldots32 \text{ (e.g.)} \]
Rough Idea: Actual Case

- Can't realize an ideal point function in real equipment
- "Delta function" equivalent has an area
- Value returned is the average over this area
Image irradiance is the average of the scene radiance over the area of the surface intersecting the solid angle!
Introduction to Computer Vision

Mixed Pixel Problem
Goal: determine a mapping from a continuous signal (e.g. analog video signal) to one of $K$ discrete (digital) levels.

$I(x,y) = .1583$ volts

= ???? Digital value
I(x, y) = continuous signal: \( 0 \leq I \leq M \)

- Want to quantize to \( K \) values 0, 1, ..., \( K-1 \)
- \( K \) usually chosen to be a power of 2:

<table>
<thead>
<tr>
<th>( K )</th>
<th>#Levels</th>
<th>#Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>128</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

- Mapping from input signal to output signal is to be determined.
- Several types of mappings: uniform, logarithmic, etc.
Introduction to Computer Vision

Choice of K

Original

Linear Ramp

K=2

K=4

K=16

K=32
Introduction to Computer Vision

Choice of $K$

- $K=2$ (each color)
- $K=4$ (each color)
Introduction to Computer Vision

Digital X-rays
Introduction to Computer Vision

Digital X-rays: 8 is enough?
Digital X-rays: 1 bit
Introduction to Computer Vision

Digital X-rays: 2 bits
Introduction to Computer Vision

Digital X-rays: 3 bit
Introduction to Computer Vision

Digital X-rays: 8 is enough?
More gray levels can be simulated with more resolution.

A “gray” pixel:

- Doubling the resolution in each direction adds at least four new gray levels. But maybe more?
Introduction to Computer Vision

Pseudocolor
Introduction to Computer Vision

Digital X-rays: 8 is enough?
Uniform sampling divides the signal range [0-M] into K equal-sized intervals.

The integers 0,...K-1 are assigned to these intervals.

All signal values within an interval are represented by the associated integer value.

Defines a mapping:
Signal is log \( I(x,y) \).

Effect is:

- Detail enhanced in the low signal values at expense of detail in high signal values.
Introduction to Computer Vision

Histogram Equalization

An unequalized image

Corresponding histogram

Same image after histogram equalization

Corresponding histogram
Two methods:

- Change the data (histogram equalization)
- Use a look up table (brightness or color remapping)
Maps Brightness Value -> RGB Color

- 0 -> (1, 0, 0)
- 1 -> (0, 1, 0)
- 2 -> (0, 0, 1)
- 3 -> (0 , 1, 1)
- ...  
- 255 -> (1, 1, 1)
Two methods:
- Change the data.
- Use a look up table.
Maps Brightness Value -> RGB Color
- 0 -> (0, 0, 0)
- 1 -> (0, 0, 0)
- 2 -> (0, 0, 0)
- 3 -> (0, 0, 0)
- ...
- 130 -> (0, 0, 0)
- 131 -> (.01, .01, .01)
- 132 -> (.02, .02, .02)
- ...
- 200 -> (1, 1, 1)
- 201 -> (1, 1, 1)
- ...
- 255 -> (1, 1, 1)
Introduction to Computer Vision

Brightness Equalization

An unequalized image

An equalized image
Introduction to Computer Vision

Tessellation Patterns

Hexagonal

Triangular

Rectangular

Typical
Introduction to Computer Vision

Spatial Frequencies

Image

Fourier Power Spectrum

one “unit” of distance

(5,0)
Introduction to Computer Vision

Spatial Frequencies

Fourier Power Spectrum

(2,0)
Introduction to Computer Vision

Spatial Frequencies

Fourier Power Spectrum

(0,5)
Introduction to Computer Vision

Spatial Frequencies

Fourier Power Spectrum

(0,5)

(5,0)
Introduction to Computer Vision

Spatial Frequencies

Fourier Power Spectrum

(8,8)
Every sampling scheme captures some spatial frequencies but not others:

- Low frequency sampling doesn’t capture the picket fence
- High frequency does.

Which two-dimensional sampling scheme is most “efficient”? 
Introduction to Computer Vision

Tesselation Patterns

Hexagonal

Triangular

Rectangular

Typical
Introduction to Computer Vision

Sampling Grids

Rectangular sampling

Hexagonal sampling
Introduction to Computer Vision

Retina

Cones in the fovea

Moving outward from fovea

All of them are cones!
Introduction to Computer Vision

Digital Geometry

- Neighborhood
- Connectedness
- Distance Metrics

Pixel value \( I(i,j) \) =

- 0,1 Binary Image
- 0 - K-1 Gray Scale Image
- Vector: Multispectral Image

Picture Element or Pixel
- Binary image with multiple 'objects'
- Separate 'objects' must be labeled individually

6 Connected Components
Two points in an image are 'connected' if a path can be found for which the value of the image function is the same all along the path.

- $P_1$ connected to $P_2$
- $P_3$ connected to $P_4$
- $P_1$ not connected to $P_3$ or $P_4$
- $P_2$ not connected to $P_3$ or $P_4$
- $P_3$ not connected to $P_1$ or $P_2$
- $P_4$ not connected to $P_1$ or $P_2$
- Pick any pixel in the image and assign it a label
- Assign same label to any neighbor pixel with the same value of the image function
- Continue labeling neighbors until no neighbors can be assigned this label
- Choose another label and another pixel not already labeled and continue
- If no more unlabeled image points, stop.

Who's my neighbor?
Consider the definition of the term 'neighbor'

Two common definitions:

- Consider what happens with a closed curve.
- One would expect a closed curve to partition the plane into two connected regions.
Neither neighborhood definition satisfactory!
Possible Solutions

- Use 4-neighborhood for object and 8-neighborhood for background
  - requires a-priori knowledge about which pixels are object and which are background
- Use a six-connected neighborhood:
Alternate distance metrics for digital images

- **Euclidean Distance**
  
  \[ d = \sqrt{(i-n)^2 + (j-m)^2} \]

- **City Block Distance**
  
  \[ d = |i-n| + |j-m| \]

- **Chessboard Distance**
  
  \[ d = \max[ |i-n|, |j-m| ] \]