Image Analysis

Inel 5046
Prof. Vidya Manian
What is an image?

References:

- Pattern Recognition and Image Analysis, Gose, Johnsonbaugh, Jost.
- Digital Image Processing, R. C. Gonzalez, R. E. Woodsd.

Matlab: Image Processing Toolbox.
Computer Vision

Object → Camera → A/D Converter → CPU
Electromagnetic Wavelength (Light)
Illumination and Reflection

\[ I(\lambda) = \rho(\lambda)L(\lambda) \]

\[ L(\lambda) \]
Digitization of an Image

Image → Sampling → Quantification → A/D Converter → Digital Computer
Sampling

Analog Image

Samples of the Image
Quantizer
Black and White Image

BW256 =

\[
\begin{bmatrix}
0 & 0 & 256 & 0 & 0 \\
0 & 256 & 256 & 256 & 0 \\
256 & 256 & 256 & 256 & 256 \\
0 & 256 & 256 & 256 & 0 \\
0 & 0 & 256 & 0 & 0 \\
\end{bmatrix}
\]

imagesc(BW256)
colormap(gray(256))
Black and White Image

\[
\text{BW} = \\
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 \\
\end{bmatrix}
\]

imagesc(BW)
colormap(gray(256))
Image Grayscale
0-255

\[
X =
\begin{bmatrix}
0 & 25 & 100 \\
115 & 150 & 200 \\
215 & 250 & 255
\end{bmatrix}
\]

n = 3
m = 3

figure('Position', [100 100 n m])
imagesc(X)

colormap(gray(256))
Gray Scale Image
load trees
Image (indexed Image)
First 5 columns and rows

\[
X_{\text{sub}} = 
\begin{bmatrix}
109 & 117 & 99 & 109 & 88 \\
113 & 113 & 113 & 106 & 113 \\
113 & 113 & 106 & 113 & 113 \\
106 & 106 & 106 & 106 & 106 \\
113 & 106 & 113 & 106 & 106 \\
\end{bmatrix}
\]

\[
\text{n} = 5 \\
\text{m} = 5
\]

```matlab
figure('Position', [100 100 n m])
imagesc(X)
colormap(gray(256))
image(Xsub)
colormap(gray(256))
```
Visualization in MATLAB

load trees
I = ind2gray(X,map);
imshow(I)

What is I? (Intensity Image)

Isub =

0.7232    0.8245    0.6599    0.7232    0.6003
0.7745    0.7745    0.7745    0.7025    0.7745
0.7745    0.7745    0.7745    0.7025    0.7745
0.7745    0.7025    0.7745    0.7025    0.7025
0.7025    0.7025    0.7025    0.7025    0.7025
Matlab Image classes

- Double- floating number ("a number with decimals") between 0 (black) and 1 (white) to each pixel
- Uint8- assigns an integer between 0 and 255
- requires roughly 1/8 of the storage compared to the class double
- Indexed image – 2 matrices, first matrix has the same size as the image and one number for each pixel. second matrix is called the *color map* and its size may be different from the image.
- The numbers in the first matrix is an instruction of what number to use in the color map matrix.
Image format conversions

<table>
<thead>
<tr>
<th>Operation</th>
<th>Matlab command:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convert between intensity/indexed/RGB format to binary format.</td>
<td>dither()</td>
</tr>
<tr>
<td>Convert between intensity format to indexed format.</td>
<td>gray2ind()</td>
</tr>
<tr>
<td>Convert between indexed format to intensity format.</td>
<td>ind2gray()</td>
</tr>
<tr>
<td>Convert between indexed format to RGB format.</td>
<td>ind2rgb()</td>
</tr>
<tr>
<td>Convert a regular matrix to intensity format by scaling.</td>
<td>mat2gray()</td>
</tr>
<tr>
<td>Convert between RGB format to intensity format.</td>
<td>rgb2gray()</td>
</tr>
<tr>
<td>Convert between RGB format to indexed format.</td>
<td>rgb2ind()</td>
</tr>
</tbody>
</table>
\( I = \text{im2double}(I); \) converts an image named \( I \) from uint8 to double.

\( I = \text{im2uint8}(I); \) converts an image named \( I \) from double to uint8.

<table>
<thead>
<tr>
<th>Reading and writing images</th>
<th>Matlab command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read an image ('filename')</td>
<td>\textit{imread()}</td>
</tr>
<tr>
<td>Write an image to a file ('filename', format)</td>
<td>\textit{imwrite( , )}</td>
</tr>
</tbody>
</table>
Fundamentals of Image Processing

- Image Processing: \( \text{image in } \rightarrow \text{image out} \)
- Image Analysis: \( \text{image in } \rightarrow \text{measurements out} \)
- Image Understanding: \( \text{image in } \rightarrow \text{high-level description out} \)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Typical values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows</td>
<td>(N)</td>
<td>256, 512, 525, 625, 1024, 1035</td>
</tr>
<tr>
<td>Columns</td>
<td>(M)</td>
<td>256, 512, 768, 1024, 1320</td>
</tr>
<tr>
<td>Gray Levels</td>
<td>(L)</td>
<td>2, 64, 256, 1024, 4096, 16384</td>
</tr>
</tbody>
</table>

**Table 1:** Common values of digital image parameters

Grey levels, \( L = 2^B \), where \( B \) is the number of bits in the Binary representation of the brightness levels. \( B > 1 \) grey level Image, \( B = 1 \) then binary image
Value of pixel at Row=3, column=10 is 110

Value = a(x, y, z, λ, t)
Characteristics of Image Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Characterization</th>
<th>Generic Complexity/Pixel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>– the output value at a specific coordinate is dependent only on the input value at that same coordinate.</td>
<td><em>constant</em></td>
</tr>
<tr>
<td>Local</td>
<td>– the output value at a specific coordinate is dependent on the input values in the <em>neighborhood</em> of that same coordinate.</td>
<td><em>p</em>^2</td>
</tr>
<tr>
<td>Global</td>
<td>– the output value at a specific coordinate is dependent on all the values in the input image.</td>
<td><em>N</em>^2</td>
</tr>
</tbody>
</table>

*Table 2:* Types of image operations. Image size = *N* × *N*; neighborhood size = *P* × *P*. Note that the complexity is specified in operations *per pixel.*
Illustration of various types of image operations

- **Point**
  - a
  - b

- **Local**
  - a
  - b

- **Global**
  - a
  - b

\( \bullet = [m=m_0, n=n_0] \)
Types of neighborhoods

- restrict to rectangular sampling, due to hardware/software considerations

**Figure 3a**
Rectangular sampling
4-connected

**Figure 3b**
Rectangular sampling
8-connected

**Figure 3c**
Hexagonal sampling
6-connected
Tools-convolution

\[ c = a \otimes b = a \ast b \]

In 2D continuous space:

\[ c(x,y) = a(x,y) \otimes b(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} a(\chi,\zeta) b(x - \chi, y - \zeta) d\chi d\zeta \]

In 2D discrete space:

\[ c[m,n] = a[m,n] \otimes b[m,n] = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} a[j,k] b[m-j,n-k] \]

Properties of convolution: commutative, associative and distributive
Tools-Fourier transform

• FT represents 2D signal as a weighted sum of sines and cosines.

\[ e^{jq} = \cos(q) + jsin(q) \]

Forward FT

\[ A = F \{ a \} \]

Inverse FT

\[ a = F^{-1} \{ A \} \]

Fourier transform is a unique and invertible operations

\[ a = F^{-1} \{ F \{ a \} \} \quad \text{and} \quad A = F \{ F^{-1} \{ A \} \} \]
In 2D discrete space:

**Forward**

\[
A(\Omega, \Psi) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} a[m, n] e^{-j(\Omega m + \Psi n)}
\]

**Inverse**

\[
a[m, n] = \frac{1}{4\pi^2} \int_{-\pi}^{+\pi} \int_{-\pi}^{+\pi} A(\Omega, \Psi') e^{+j(\Omega m + \Psi n)} d\Omega d\Psi'
\]

FT can be written in terms of magnitude and phase

\[
a[m, n] = |a[m, n]| e^{j\theta[m, n]}
\]

- If a 2D signal is real, then the Fourier transform has certain symmetries.

\[
A(u, v) = A^* (-u, -v) \quad A(\Omega, \Psi) = A^* (-\Omega, -\Psi')
\]
• If a 2D signal is real and even, then the Fourier transform is real and even.

\[ A(u,v) = A(-u,-v) \quad A(\Omega,\Psi) = A(-\Omega,-\Psi) \]

• The Fourier and the inverse Fourier transforms are linear operations.

\[
F \{ w_1a + w_2b \} = F \{ w_1a \} + F \{ w_2b \} = w_1A + w_2B \\
F^{-1} \{ w_1A + w_2B \} = F^{-1} \{ w_1A \} + F^{-1} \{ w_2B \} = w_1a + w_2b
\]

• FT is periodic, with period \(2\pi\)

\[ A(\Omega + 2\pi j,\Psi + 2\pi k) = A(\Omega,\Psi) \quad j, k \text{ integers} \]

• Convolution in the spatial domain is equivalent to multiplication in the frequency domain

\[ c = a \otimes b \quad \leftrightarrow \quad C = A \cdot B \]

\[ c = a \cdot b \quad \leftrightarrow \quad C = \frac{1}{4\pi^2} A \otimes B \]
• If a two-dimensional signal \( a(x,y) \) is scaled in its spatial coordinates then:

\[
\text{If } \quad a(x,y) \quad \rightarrow \quad a\left(M_x \cdot x, M_y \cdot y\right)
\]

\[
\text{Then } \quad A(u,v) \quad \rightarrow \quad A\left(\frac{u}{M_x}, \frac{v}{M_y}\right) / |M_x \cdot M_y|
\]

• If a two-dimensional signal \( a(x,y) \) has Fourier spectrum \( A(u,v) \) then:

\[
A(u = 0, v = 0) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} a(x,y) \, dx \, dy
\]

\[
a(x = 0, y = 0) = \frac{1}{4\pi^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} A(u,v) \, dx \, dy
\]

• If a two-dimensional signal \( a(x,y) \) has Fourier spectrum \( A(u,v) \) then:

\[
\frac{\partial a(x,y)}{\partial x} \quad \mathcal{F} \leftrightarrow j\mu A(u,v)
\]

\[
\frac{\partial a(x,y)}{\partial y} \quad \mathcal{F} \leftrightarrow j\nu A(u,v)
\]

\[
\frac{\partial^2 a(x,y)}{\partial x^2} \quad \mathcal{F} \leftrightarrow -\mu^2 A(u,v)
\]

\[
\frac{\partial^2 a(x,y)}{\partial y^2} \quad \mathcal{F} \leftrightarrow -\nu^2 A(u,v)
\]
Importance of magnitude and phase

Original

$log(|A(\Omega, \Psi)|)$

$\varphi(\Omega, \Psi)$

$\varphi(\Omega, \Psi) = 0$

$|A(\Omega, \Psi)| = constant$
Image statistics

Region is the interior of the circle
Algorithms
Histogram based operations

Brightness distribution function
Brightness histogram
Contrast stretching

\[ b[m,n] = (2^B - 1) \cdot \frac{a[m,n] - \text{minimum}}{\text{maximum} - \text{minimum}} \]

Sensitive to outliers

\[ b[m,n] = \begin{cases} 
0 & a[m,n] \leq p_{\text{low}} \% \\
(2^B - 1) \cdot \frac{a[m,n] - p_{\text{low}} \%}{p_{\text{high}} \% - p_{\text{low}} \%} & p_{\text{low}} \% < a[m,n] < p_{\text{high}} \% \\
(2^B - 1) & a[m,n] \geq p_{\text{high}} \% 
\end{cases} \]

Instead of 0% and 100% use Plow=1% and Phigh=99%
In the above suppress \(2^B-1\) and normalize the brightness range to \(0 \leq b[m,n] \leq 1\)
load spine;
img=X;
[m1 n1 r1]=size(img);
img2=double(img);
% calculation of vmin and vmax-
for(k=1:r1)
    arr=sort(reshape(img2(:,:,k),m1*n1,1));
    vmin(k)=arr(ceil(0.008*m1*n1));
    vmax(k)=arr(ceil(0.992*m1*n1));
end
v_min=vmin;
v_max=vmax;
for(i=1:m1)
    for(j=1:n1)
        for(k=1:r1)
            img2(i,j,k)=255*(img2(i,j,k)-v_min(1))/(v_max(1)-v_min(1));
        end
    end
end
%-------------------
img2=uint8(img2);
figure,imshow(img),title('THIS IS THE ORIGINAL IMAGE');
figure,imshow(img2),title('THIS IS THE ENHANCED IMAGE');

Adjust contrast to the optimum level
Mathematics based operations

- **Binary operations**

  - **NOT**  \[ c = \overline{a} \]
  - **OR**  \[ c = a + b \]
  - **AND**  \[ c = a \cdot b \]
  - **XOR**  \[ c = a \oplus b = a \cdot \overline{b} + \overline{a} \cdot b \]
  - **SUB**  \[ c = a \setminus b = a - b = a \cdot \overline{b} \]

Matlab >>im2bw  %converts image to binary
### NOT

<table>
<thead>
<tr>
<th>a</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### OR

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### AND

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### XOR

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### SUB

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Examples of various binary operations

a) Image $a$  

b) Image $b$

c) NOT($b$) = $\overline{b}$

d) OR($a,b$) = $a + b$

e) AND($a,b$) = $a \cdot b$

f) XOR($a,b$) = $a \oplus b$

g) SUB($a,b$) = $a \setminus b$
# Arithmetic Based Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Definition</th>
<th>Preferred data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>( c = a + b )</td>
<td>integer</td>
</tr>
<tr>
<td>SUB</td>
<td>( c = a - b )</td>
<td>integer</td>
</tr>
<tr>
<td>MUL</td>
<td>( c = a \cdot b )</td>
<td>integer or floating point</td>
</tr>
<tr>
<td>DIV</td>
<td>( c = a / b )</td>
<td>floating point</td>
</tr>
<tr>
<td>LOG</td>
<td>( c = \log(a) )</td>
<td>floating point</td>
</tr>
<tr>
<td>EXP</td>
<td>( c = \exp(a) )</td>
<td>floating point</td>
</tr>
<tr>
<td>SQRT</td>
<td>( c = \text{sqrt}(a) )</td>
<td>floating point</td>
</tr>
<tr>
<td>TRIG.</td>
<td>( c = \sin/\cos/\tan(a) )</td>
<td>floating point</td>
</tr>
<tr>
<td>INVERT</td>
<td>( c = (2^B - 1) - a )</td>
<td>integer</td>
</tr>
</tbody>
</table>
[x,map]=imread('saturn.tif');
imshow(x,map);
[nr,nc]=size(x);
% image is of type uint8, so convert it to double
x=double(x)+1;
for p=1:nr
    for q=1:nc
        xnew(p,q)=255-x(p,q);
    end
end
imshow(xnew/max(max(xnew)));
Histogram Equalization

- Histogram normalization (linearization) - compare images on a specific basis
- Probability of occurrence of gray level $r_k$ in an image, $n$ – total number of pixels in image, $n_k$ – number of pixels that have grey level $r_k$, $s_k$ – grey level of output pixel
- $L$-total number of grey levels

$$p_r(r_k) = \frac{n_k}{k}, \text{ k}=0,1,2,...,L-1$$

- Transformation function is

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n}, \text{ k}=0,1,2,...,L-1$$
Histogram Equalization

- Transformation function satisfies
  - (a) $T(r)$ is single valued and monotonically increasing in the interval $0 \leq r \leq 1$
  - (b) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$
Original images,
Results of histogram equalization,
corresponding histogram
Enhancement by Image Averaging

• Consider a noisy image,
  \[ g(x,y) = f(x,y) + \eta(x,y) \]

• Reduce the noise \( \eta \) adding a set of noisy images, \( \{g_i(x,y)\} \)

• Averaged image,
  \[
  \bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^{K} g_i(x, y)
  \]

  \[
  E\{\bar{g}(x, y)\} = f(x, y)
  \]

  \[
  \sigma^2 \bar{g}(x, y) = \frac{1}{K} \sigma^2 \eta(x, y)
  \]
(a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64,$ and 128 noisy images. (Original image courtesy of NASA.)
Basics of Spatial Filtering

• Neighborhood subimage-filter, mask, kernel, template, or window
• Values in mask –coefficients
• Response $R$ of linear filtering with the filter mask at a point $(x,y)$ in the image is:
  $$R = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots + w(0,0)f(x,y) + \ldots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$$
The mechanics of spatial filtering. The magnified drawing shows a $3 \times 3$ mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.
Smoothing spatial filters

\[ \frac{1}{9} \times \]
\[
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

\[ \frac{1}{16} \times \]
\[
\begin{array}{ccc}
1 & 2 & 1 \\
2 & 4 & 2 \\
1 & 2 & 1 \\
\end{array}
\]

Two 3 × 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.
(a) Original image, of size $500 \times 500$ pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15,$ and $35$, respectively. The black squares at the top are of sizes $3, 5, 9, 15, 25, 35, 45,$ and $55$ pixels, respectively; their borders are $25$ pixels apart. The letters at the bottom range in size from $10$ to $24$ points, in increments of $2$ points; the large letter at the top is $60$ points. The vertical bars are $5$ pixels wide and $100$ pixels high; their separation is $20$ pixels. The diameter of the circles is $25$ pixels, and their borders are $15$ pixels apart; their gray levels range from $0\%$ to $100\%$ black in increments of $20\%$. The background of the image is $10\%$ black. The noisy rectangles are of size $30 \times 120$ pixels.
Sharpening spatial filters

\[ \frac{\partial f}{\partial x} = f(x+1) - f(x) \]
\[ \frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \]

Change between adjacent pixels
First and second derivative
(1) Must be zero in flat areas;
(2) Non zero at the onset of gray-level step or ramp
(3) First derivative-non zero along ramps; second derivative-
Zero along ramps of constant slope
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).
The Laplacian for enhancement (the second derivative)

\[ \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \]

In the x direction

\[ \frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \]

In the y direction

\[ \frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \]

Sum min g

\[ \nabla^2 f = [f(x, y + 1) + f(x, y - 1) + f(x+1, y) + f(x-1, y)] - 4f(x, y) \]
(a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4).
(b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)
Composite Laplacian mask

(a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)
The Gradient for enhancement (the first derivative)

\[ \nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \]

\[ \nabla f = \text{mag}(\nabla f) \]

\[ = [G_x^2 + G_y^2]^{1/2} \]

\[ = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \]

\[ \nabla f \approx \left| (z_7 + 2z_8 + z_9) - (z_1 + 2z_1 + z_3) \right| + \left| (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \right| \]
A $3 \times 3$ region of an image (the $z$'s are gray-level values) and masks used to compute the gradient at point labeled $z_5$. All masks coefficients sum to zero, as expected of a derivative operator.
Optical image of contact lens (note defects on the boundary at 4 and 5 o’clock).
(b) Sobel gradient.
(Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)