Image Enhancement

Outline

• Introduction

• Intensity domain methods
  • Histogram modification
  • Histogram equalization
  • Local enhancement

• Spatial domain methods
  • Mean and median filters
  • Gaussian smoothing
  • Sharpening filters

• Frequency domain methods
  • Low-pass filtering
  • High-pass filtering

• Applications
  • Biomedical
  • Visual Inspection
  • Document processing
Image Enhancement

Introduction

Image enhancement deals with the improvement of visual appearance of the scene, to improve the detectability of objects to be used by either a machine vision system or a human observer.

Sources of image deterioration

- Noise
- Low Resolution
- Quantization levels

Image

• Attenuate the effects of sub-sampling
• Attenuate quantization effects
• Remove noise and simultaneously preserve edges and image details
• Avoid aliasing effects
• Attenuate the blockiness effect
• Enhance special features to be more easily detected by a machine or a human observer.

Enhancement System

Machine vision system
Human observer
Resolution, quantization levels and noise are the main characteristics we need to look into to understand the degradation and to select an appropriate method to its removal or attenuation. The effect of noise was illustrated in the Introduction.
Image Enhancement

Introduction

A scarce representation or a lossy compression may demand the use of a post-processing enhancement operation.

Quadtree representation

DCT compression

Original Pout Image

Reconstructed Image

Error Image

The MSE (with images normalized) is 0.00076.
Intensity domain methods

Histogram Modification

A histogram of a digital image $h(g_k)$ is

$$h(g_k) = \frac{n_k}{n}$$

$g_k$ - $k$th grey level

$n_k$ - number of pixel with grey level $g_k$

$n$ - total number of pixels

Histogram Stretch

$$g' = \frac{g'_{\text{max}} - g'_{\text{min}}}{g_{\text{max}} - g_{\text{min}}} (g - g_{\text{max}}) + g'_{\text{min}}$$
Intensity domain methods

Histogram Modification

Histogram Equalization

\[ g'_k = T(g_k) = \sum_{j=0}^{k} h(g_k) \]

and \( k = 0, 1, \ldots, L - 1 \)

\( T \) is the mapping function. It is the discrete cumulative distribution function.

This operation is called hist. equalization because the corresponding continuous operation gives rise to a uniform histogram.
Gaussian smoothing

Gaussian filters are a class of smoothing filters where the kernel values have a 2D Gaussian shape.

### 3×3 kernel

\[
\begin{bmatrix}
1 & 2 & 1 \\
2 & 4 & 2 \\
1 & 2 & 1 \\
\end{bmatrix}
\]

\[\frac{1}{16}\]

### 5×5 kernel

\[
\begin{bmatrix}
1 & 4 & 6 & 4 & 1 \\
4 & 16 & 24 & 16 & 4 \\
6 & 24 & 36 & 24 & 6 \\
4 & 16 & 24 & 16 & 4 \\
1 & 4 & 6 & 4 & 1 \\
\end{bmatrix}
\]

\[\frac{1}{256}\]
Gaussian smoothing
Changing the kernel size

Input

Output

3×3 kernel

5×5 kernel

Image

Intensity profile
(line 128)

Histogram

Image Processing Applications

Image Enhancement
Gaussian smoothing
Designing gaussian kernels

I. Sampling the gaussian function

Compute directly the kernel values from the gaussian function
\[ g(i, j) = c e^{-\frac{(i^2 + j^2)}{2\sigma^2}} \]

1. Normalize the function and compute the values
\[ g(i, j) = \frac{1}{\sum_{i} \sum_{j}} g(i, j) \]

2. Values for a 7x7 window and \( \sigma^2 \)

<table>
<thead>
<tr>
<th>(i,j)</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>0.011</td>
<td>0.039</td>
<td>0.082</td>
<td>0.105</td>
<td>0.082</td>
<td>0.039</td>
<td>0.011</td>
</tr>
<tr>
<td>-2</td>
<td>0.039</td>
<td>0.135</td>
<td>0.287</td>
<td>0.368</td>
<td>0.287</td>
<td>0.135</td>
<td>0.039</td>
</tr>
<tr>
<td>-1</td>
<td>0.082</td>
<td>0.287</td>
<td>0.607</td>
<td>0.779</td>
<td>0.607</td>
<td>0.287</td>
<td>0.082</td>
</tr>
<tr>
<td>0</td>
<td>0.105</td>
<td>0.368</td>
<td>0.779</td>
<td>1.000</td>
<td>0.779</td>
<td>0.368</td>
<td>0.105</td>
</tr>
<tr>
<td>1</td>
<td>0.082</td>
<td>0.287</td>
<td>0.607</td>
<td>0.779</td>
<td>0.607</td>
<td>0.287</td>
<td>0.082</td>
</tr>
<tr>
<td>2</td>
<td>0.039</td>
<td>0.135</td>
<td>0.287</td>
<td>0.368</td>
<td>0.287</td>
<td>0.135</td>
<td>0.039</td>
</tr>
<tr>
<td>3</td>
<td>0.011</td>
<td>0.039</td>
<td>0.082</td>
<td>0.105</td>
<td>0.082</td>
<td>0.039</td>
<td>0.011</td>
</tr>
</tbody>
</table>

3. Normalization by \( g(3,3) \)

<table>
<thead>
<tr>
<th>(i,j)</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>9</td>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>-2</td>
<td>3</td>
<td>12</td>
<td>26</td>
<td>33</td>
<td>26</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>-1</td>
<td>7</td>
<td>26</td>
<td>55</td>
<td>70</td>
<td>55</td>
<td>26</td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>9</td>
<td>33</td>
<td>70</td>
<td>90</td>
<td>70</td>
<td>33</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>26</td>
<td>55</td>
<td>70</td>
<td>55</td>
<td>26</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>12</td>
<td>26</td>
<td>33</td>
<td>26</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>9</td>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

4. Divide by \( \Sigma_i \Sigma_j = 1105 \)
Gaussian smoothing
Designing gaussian kernels

II. Use the Pascal’s triangle

\[(1 + x)^n = \binom{n}{0} + \binom{n}{1}x + \binom{n}{2}x^2 + \cdots + \binom{n}{n}x^n\]

The row \(n\) of the Pascal triangle’s is the 1-D, \(n\)-point approximation to a Gaussian filter. For example, for \(n = 5\) the horizontal mask is

\[[1\ 4\ 6\ 4\ 1]\]

A 2-D Gaussian filter can be implemented as the successive convolutions of two, 1-D Gaussian, one in the horizontal direction and the other in the vertical.
Gaussian smoothing
Gaussian kernels properties

• In two dimensions Gaussian functions are rotationally symmetric. Thus, the amount of smoothing is independent of the direction. This property implies that no bias is introduced for a specific direction of an edge.
• The Gaussian function is uni-modal. As a consequence, the weight given to a neighbour decreases with the distance from the central pixel.
• The Fourier transform of a Gaussian function is still Gaussian. The single lobe in the Fourier transform means that the smoothed image will not be corrupted by contributions from un-wanted high-frequency signal, while keeping the desirable signals.
• The degree of smoothing is parametrized by the standard deviation of the filter.
• Gaussian functions are separable. As a result a Gaussian convolution can be implemented by a 1D horizontal convolution followed by a 1D vertical convolution. Using this decompositions the number of operations decreases significantly.
Sharpening filters

**Objective:** enhance a particular feature or detail in an image.

**High frequency emphasis**
A high pass filter can be obtained by the difference between the original image and a low pass version, e.g. \( g_H(i,j) = f(i,j) - f_L(i,j) \)

\[
\begin{align*}
g_E(i,j) &= \alpha f(i,j) - f_L(i,j) \\
g(i,j) &= (\alpha - 1)f(i,j) + f(i,j) - f_L(i,j) \\
g(i,j) &= (\alpha - 1)f(i,j) + g_H(i,j)
\end{align*}
\]

\( \alpha = 1 \), high pass result
\( \alpha > 1 \), part of the original is added back to the high pass result, restoring the low-frequency components.

**Example**

<table>
<thead>
<tr>
<th>Laplacian</th>
<th>Sharpen</th>
<th>Generic mask</th>
</tr>
</thead>
</table>
| \[
\frac{1}{9} \times \begin{bmatrix}
-1 & -1 & -1 \\
-1 & 8 & -1 \\
-1 & -1 & -1
\end{bmatrix}
\] | \[
\frac{1}{9} \times \begin{bmatrix}
-1 & -1 & -1 \\
-1 & 9 & -1 \\
-1 & -1 & -1
\end{bmatrix}
\] | \[
\frac{1}{9} \times \begin{bmatrix}
-1 & -1 & -1 \\
-1 & w & -1 \\
-1 & -1 & -1
\end{bmatrix}
\]

\( w = 9\alpha - 1 \)
Enhancing special features

Based in a typical intensity profile, a bidimensional mask can be defined, having replicas of the typical profile along the perpendicular direction of the profile. We may observe that the vessels almost never have ideal step edges, and the intensity profile may be approximated by a Gaussian curve. Different numbers of orientations were tested, as shown in the following figure. For example, the result observed in figure b) was obtained with 18 different masks corresponding to individual rotations of 10°. The set of such 18 kernels is applied to the image, and only the maximum of their response is retained.

Masks profiles

Original

Results

a) 0°

b) 30°

c) 60°

d) 90°

e) 120°

f) 150°
Removing the background

In the analysis of retinal images one important issue is to keep the results invariant with illumination, noise and view angle. The removal of background variation due to the above factors is a possible attempt. This can be achieved by low-pass filtering using a large convolution mask (the size must be at least 2 times the vessel diameter) and then subtracting from the original.
Frequency domain methods

Frequency components

Input image
(a sinusoidal function in \( xx \) direction)

FFT magnitude
Frequency domain methods

Frequency components

Input image
(sum of two sinusoidal functions in \( xx \) and \( yy \) directions)

FFT magnitude
Frequency domain methods

Filtering

The image response of a linear system can be found either in the spatial domain or in the frequency domain, using the following relationships.

\[ f(x, y) \xrightarrow{\text{Linear Filter}} g(x, y) = h(x, y) * f(x, y) \]

\[ F(u, v) \xrightarrow{\text{Linear Filter}} G(u, v) = H(u, v) F(u, v) \]

$h(x, y)$ is the impulse response of the filter and $H(u,v)$ its Fourier transform. $f(x, y)$ and $g(x,y)$ are the input and output images, having as Fourier transforms $F(u,v)$ and $G(u,v)$. To obtain the response we must convolve (* sign) the input with the filter characteristic function $h(x,y)$, in the spatial domain, or to multiply the transfer function $H(u,v)$ by the Fourier transform of the input, $F(u,v)$. The result of the multiplication is the Fourier transform of the output, $G(u,v)$. 
Frequency domain methods

Gaussian filtering

Spatial domain

\[ f(x) = e^{-\frac{x^2}{2\sigma^2}} \]

Low-pass filter

\[ f(x) = \frac{2A}{\sqrt{2\pi \sigma}} e^{-\frac{x^2}{2\sigma^2}} \cos(2\pi u_0 x) \]

\[ \sigma^2 = \frac{1}{\nu^2} \]

Band-pass filter

\[ F(u) = A e^{-\frac{u^2}{2\nu^2}} \ast [\delta(u - u_o) + \delta(u + u_o)] \]

Frequency domain

\[ F(u) = \sqrt{2\pi \sigma} e^{-\frac{u^2}{2\nu^2}} \]
Frequency domain methods

High frequency enhancement

The transfer function of this type of filter is unity at zero frequency and increases with frequency, falling back to zero at higher frequencies. The DoG filter (Difference-of-Gaussians filter) has a high-frequency characteristic.

Frequency domain

\[
F_1(u) = A e^{-\frac{u^2}{2\nu_1^2}} \\
F_2(u) = B e^{-\frac{u^2}{2\nu_2^2}} \\
F(u) = F_1(u) - F_2(u)
\]

Spatial domain

\[
f_1(x) = \frac{A}{\sqrt{2\pi\sigma_1}} e^{-\frac{x^2}{2\sigma_1^2}} \\
f_2(x) = \frac{A}{\sqrt{2\pi\sigma_2}} e^{-\frac{x^2}{2\sigma_2^2}} \\
f(u) = f_1(u) - f_2(u)
\]
Frequency domain methods

Low pass filtering

Spatial domain

Input

Output

Frequency domain
Frequency domain methods

Gaussian High Pass Filtering

Images and intensity profiles (at the same column indicated in the input) illustrating the effects of Gaussian filtering. Gaussian 1 and Gaussian 2 are the results of a low-pass filter. The first one with a larger bandwidth. The output is obtained by subtracting Gaussian 1 from Gaussian 2.

Input  Gaussian 1  Gaussian 2  Output