Local Enhancement

- Local Enhancement
- Median filtering

Sometimes Local Enhancement is Preferred.

MATLAB: BlkProc operation for block processing.

Left: original "tire" image.

Histogram equalized

Local histogram equalized

Local Contrast Enhancement

- Enhancing local contrast
  \[ g(x,y) = A(x,y) \cdot [ f(x,y) - m(x,y) ] + m(x,y) \]

  \[ A(x,y) = k \cdot \frac{M}{\sigma(x,y)} \quad 0 < k < 1 \]

  \[ M: \text{Global mean} \]
  \[ m(x,y), \sigma(x,y): \text{Local mean and standard dev.} \]

  Areas with low contrast \( \Rightarrow \) Larger gain \( A(x,y) \)

Fig 3.23: Another example

Local Contrast Enhancement

F = histeq;
I = imread('tire.tif');
J = blkproc(I,[20 20], F);
Image Subtraction

\[ g(x,y) = f(x,y) - h(x,y) \]

- Application in medical imaging—"mask mode radiography"
- \( h(x,y) \) is the mask, e.g., an X-ray image of part of a body; \( f(x,y) \) —incoming image after injecting a contrast medium.
Averaging

\[ g(x,y) = f(x,y) + \eta(x,y) \]
\[ \overline{g(x,y)} = \frac{1}{M} \sum_{i=1}^{M} g_i(x,y) \]
\[ E(\overline{g(x,y)}) = f(x,y) \quad \text{and} \quad \sigma_{\overline{g}}^2 = \frac{1}{M} \sigma^2 \eta(x,y) \]
\[ \eta(x,y) \rightarrow \text{Uncorrelated zero mean} \]
\[ \sigma_{\overline{g}}^2(x,y) \rightarrow \text{Reduces the noise variance} \]

Another example

Images with additive Gaussian Noise:
Independent Samples.

Averaged image

Left: averaged image (10 samples);
Right: original image

Spatial filtering

Smoothing (Low Pass) Filtering

Replace \( f(x,y) \) with \( f(x,y) = \sum \omega_j f_j \)
Linear filter

LPF: reduces additive noise \( \Rightarrow \) blurs the image
\( \Rightarrow \) sharpness details are lost
(Example: Local averaging)
Figure 3.35: smoothing

Figure 3.36: another example

**Median filtering**

Replace \( f(x,y) \) with \( \text{median}\{f(x',y')\} \)

\((x',y') \in \text{neighbourhood}\)

- Useful in eliminating intensity spikes. (salt & pepper noise)
- Better at preserving edges.

Example:

\[
\begin{array}{cccccccccc}
10 & 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \\
20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \\
25 & 25 & 25 & 20 & 20 & 20 & 100 & 100 & 100 & 100 \\
\end{array}
\]

Median = 20

So replace (15) with (20)

**Median Filter: Root Signal**

Repeated applications of median filter to a signal results in an invariant signal called the “root signal”.

A root signal is invariant to further application of the median filter.

Example: 1-D signal: Median filter length = 3

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 1 & 2 & 1 & 2 & 1 \\
0 & 0 & 0 & 1 & 1 & 2 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

\(00011111000\) root signal

**Invariant Signals**

Invariant signals to a median filter:

- Constant
- Monotonically increasing
- Monotonically decreasing

length?
Media Filter: another example

Original and with salt & pepper noise

```
imnoise(image, 'salt & pepper');
```

Donoised images

Local averaging

```
K=fspecial('average',3);image)/255.
```

Median filtered

```
L=medfilt2(image, [3 3]);
```

Sharpening Filters

- Enhance finer image details (such as edges)
- Detect region/object boundaries.

Example:

```
-1 -1 -1
-1 8 8 8
-1 -1 -1 -1
```

Edges (Fig 3.38)

Unsharp Masking

Subtract Low pass filtered version from the original
emphasizes high frequency information

```
I' = A ( Original) - Low pass
HP = O - LP  A > 1
I' = (A - 1) O + HP
A = 1 => I' = HP
A > 1 => LF components added back.
```

Fig 3.43 –example of unsharp masking
**Derivative Filters**

\[
\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]^T
\]

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

**Edge Detection**

Gradient based methods

\[
\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}
\]

**Digital edge detectors**

- Robert’s operator
  \[
  |\nabla f| = \left[ (z_5-z_9)^2 + (z_5-z_6)^2 \right]^{1/2}
  \]

- Prewitt
  \[
  |\nabla f| = |z_5-z_9| + |z_5-z_6|
  \]

- Sobel’s
  \[
  |\nabla f| = |z_5-z_9| + |z_5-z_6|
  \]

**Laplacian based edge detectors**

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

- Rotationally symmetric, linear operator
- Check for the zero crossings to detect edges
- Second derivatives => sensitive to noise.

**Fig 3.45: Sobel edge detector**

**Fig 3.40: an example**