Histograms

Histogram Equalization

Properties of histograms

- Integrated optical density

$$IOD = \int_0^\infty D\,H(D)\,dD$$

- Mean grey level

$$MGL = IOD / area$$

Image statistics

IOD

Mean Grey Level = MGL

- MEAN $\mu = \dfrac{\sum_{y=0}^{M-1}\sum_{x=0}^{N-1} f(x,y)}{N * M}$

- VARIANCE $\sigma^2 = \dfrac{\sum_{y=0}^{M-1}\sum_{x=0}^{N-1} (f(x,y)-\mu)^2}{N * M}$

- STANDARD DEVIATION $\sigma = \sqrt{\mathrm{var}\,iance}$

Definition: What is a histogram?

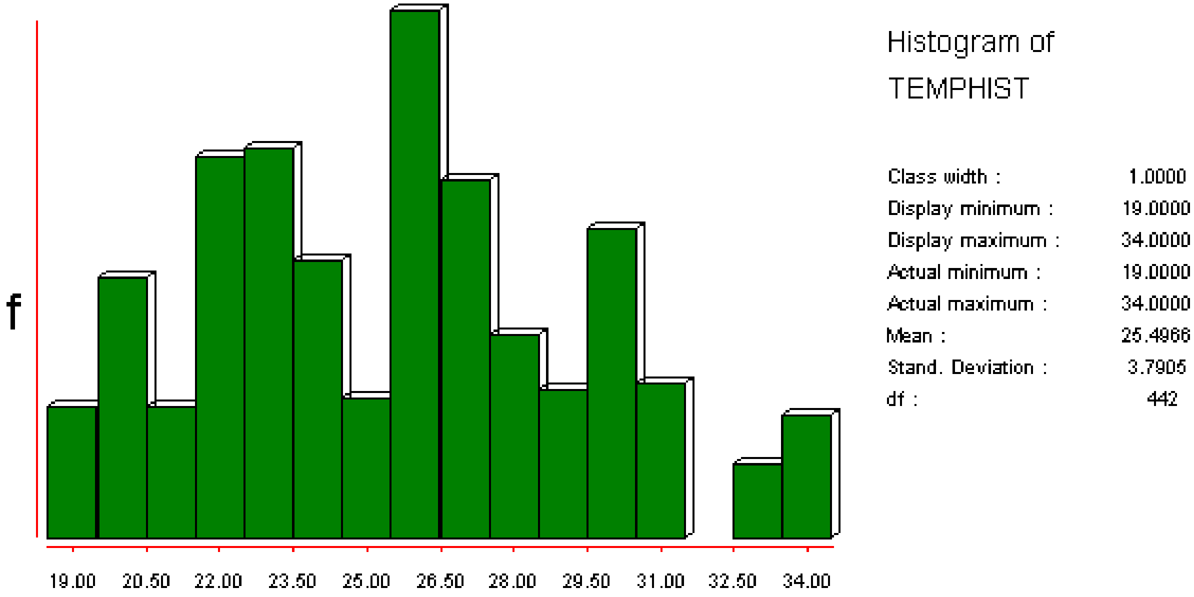
Histograms count the number of occurrences of each possible value

$$H(D)$$

Count — Grey level

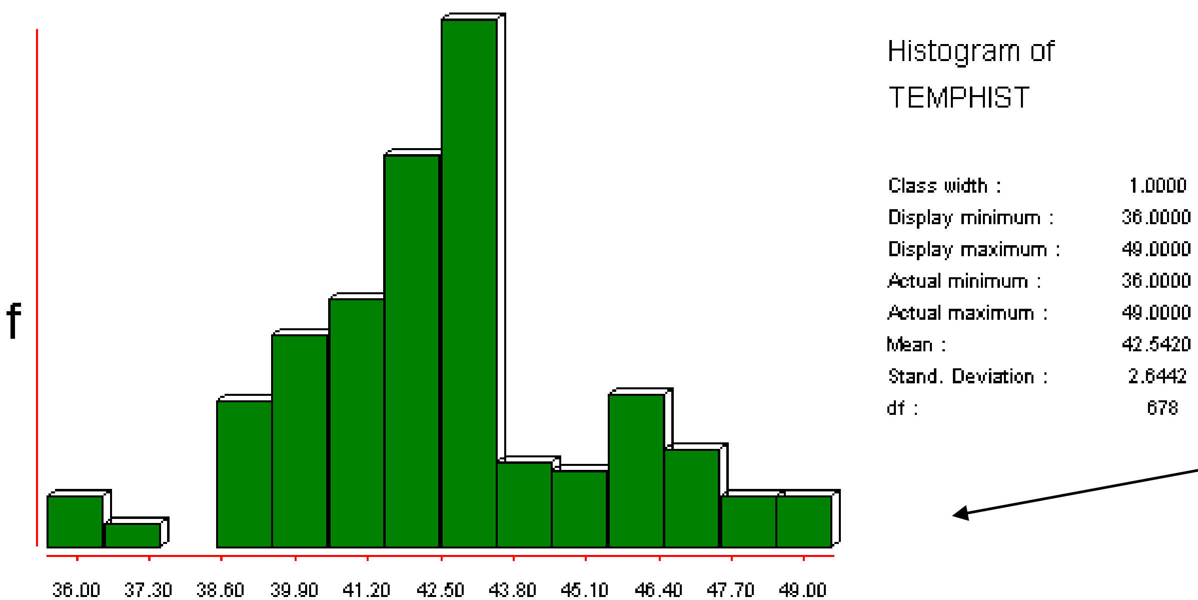
Example: Processing of aerial images

Examples of histograms



Grass marsh

Histograms of band 75x training sites



Tree swamp

Properties of histograms

- Sum of all values in the histogram equals the total number of pixels

$$\int_0^\infty H(D)\,dD = \text{image area}$$

…obvious because….. because every pixel has only one value in histogram….

Properties of histograms

- Sum of all values between *a* and *b* equals the area of all objects in that range

$$\int_a^b H(D)\,dD = \text{area of all parts}\; a \le I \le b$$

….also obvious…..

What are the Applications of Histograms?

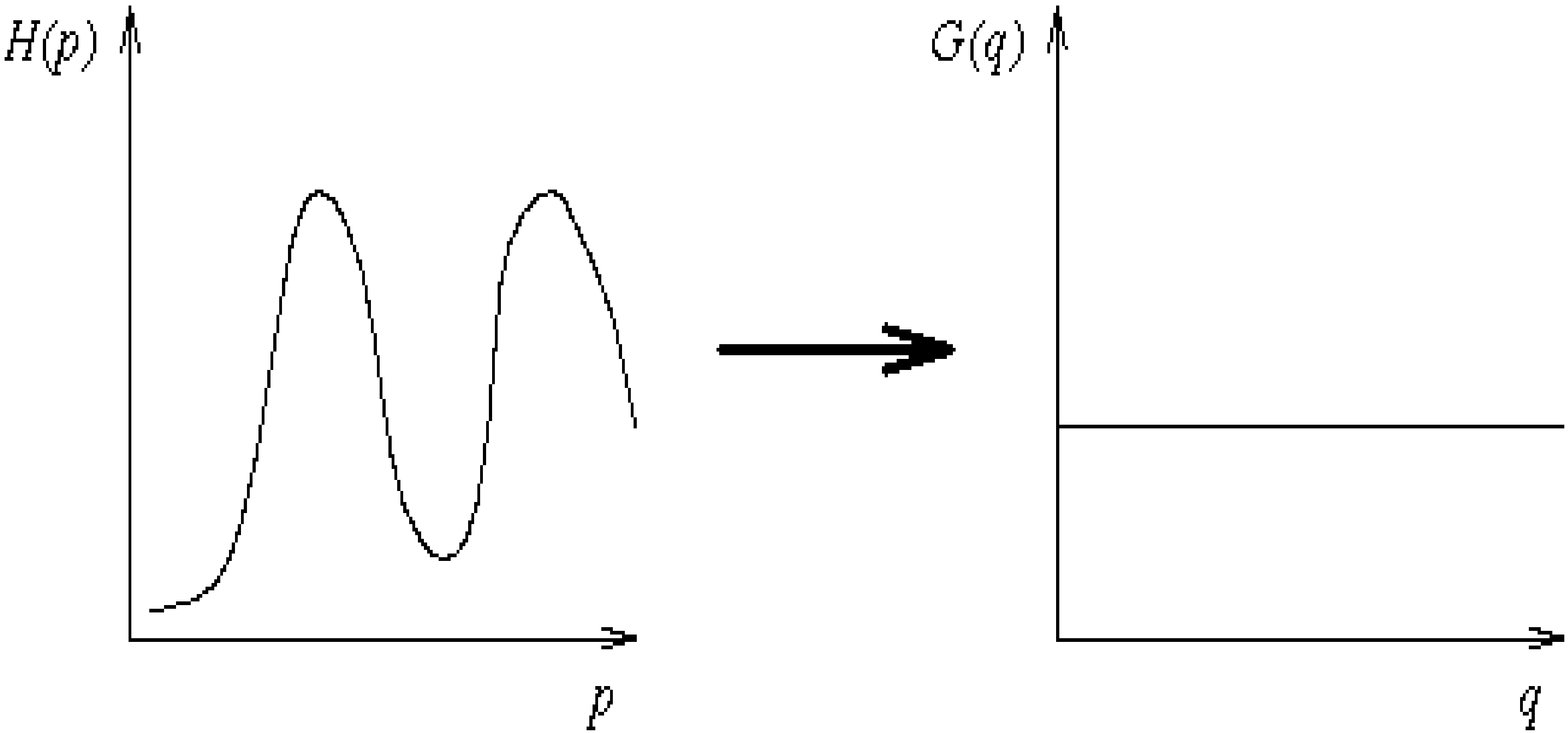
- Image Equalization
- Image Enhancement
- Adjusting Camera Parameters
- Histogram Normalization
- Logarithmic Contrast Enhancement
- Log histogramming for edge detection

What is Histogram Equalization?

Pixel intervals are also called classes.

- Usually in image you have equal intervals but various number of pixels in each interval.
- Histogram Equalization:
  - creates new intervals
  - places equal number of pixels in each of the new intervals
- Resulting histogram will have unequal intervals, but equal number of pixels in each class
- It can be done automatically or aided by a human.

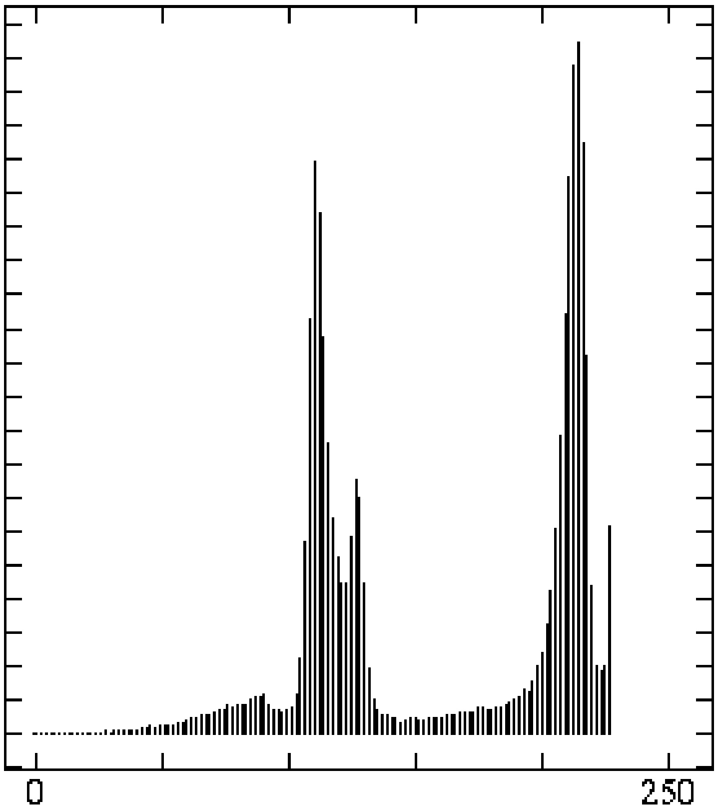
Histogram equalisation



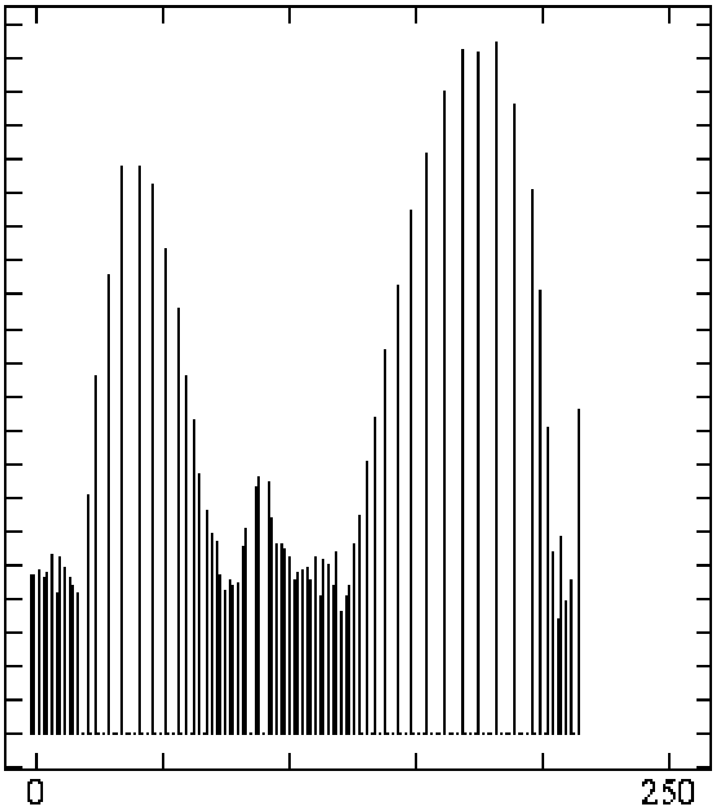
Typical histogram

Ideal histogram

Histogram equalization example



Typical histogram



After histogram equalization

Algorithm for Histogram Equalization

Probability Density Functions, p(l)

- Limits 0 < p(l) < 1
- p(l) = h(l) / n
- n = N*M (total number of pixels)
- $$\sum_{l=0}^{MAX} p(l) = 1$$

Histograms, h(l)

- Counts the number of occurrences of each grey level in an image
- l = 0,1,2,… L-1
- l = grey level, intensity level
- L = maximum grey level, typically 256

$$\sum_{l=0}^{MAX} h(l) =$$

- Area under histogram
- Total number of pixels N*M
  - unimodal, bimodal, multi-modal, dark, light, low contrast, high contrast

Histogram Equalization, E(l)

- Increases dynamic range of an image
- Enhances contrast of image to cover all possible grey levels
- Ideal histogram = flat
  - same no. of pixels at each grey level
- *Ideal no. of pixels* at each grey level = $$i = \frac{N*M}{L}$$

E(l) Algorithm

- Allocate pixel with lowest grey level in old image to 0 in new image
- If new grey level 0 has less than *ideal no. of pixels*, allocate pixels at next lowest grey level in old image <u>also</u> to grey level 0 in new image
- When grey level 0 in new image has > *ideal no. of pixels* move up to next grey level and use same algorithm
- Start with any unallocated pixels that have the lowest grey level in the old image
- If earlier allocation of pixels already gives grey level 0 in new image **TWICE** its fair share of pixels, it means it has also used up its quota for grey level 1 in new image
- Therefore, ignore new grey level one and start at grey level 2 …..

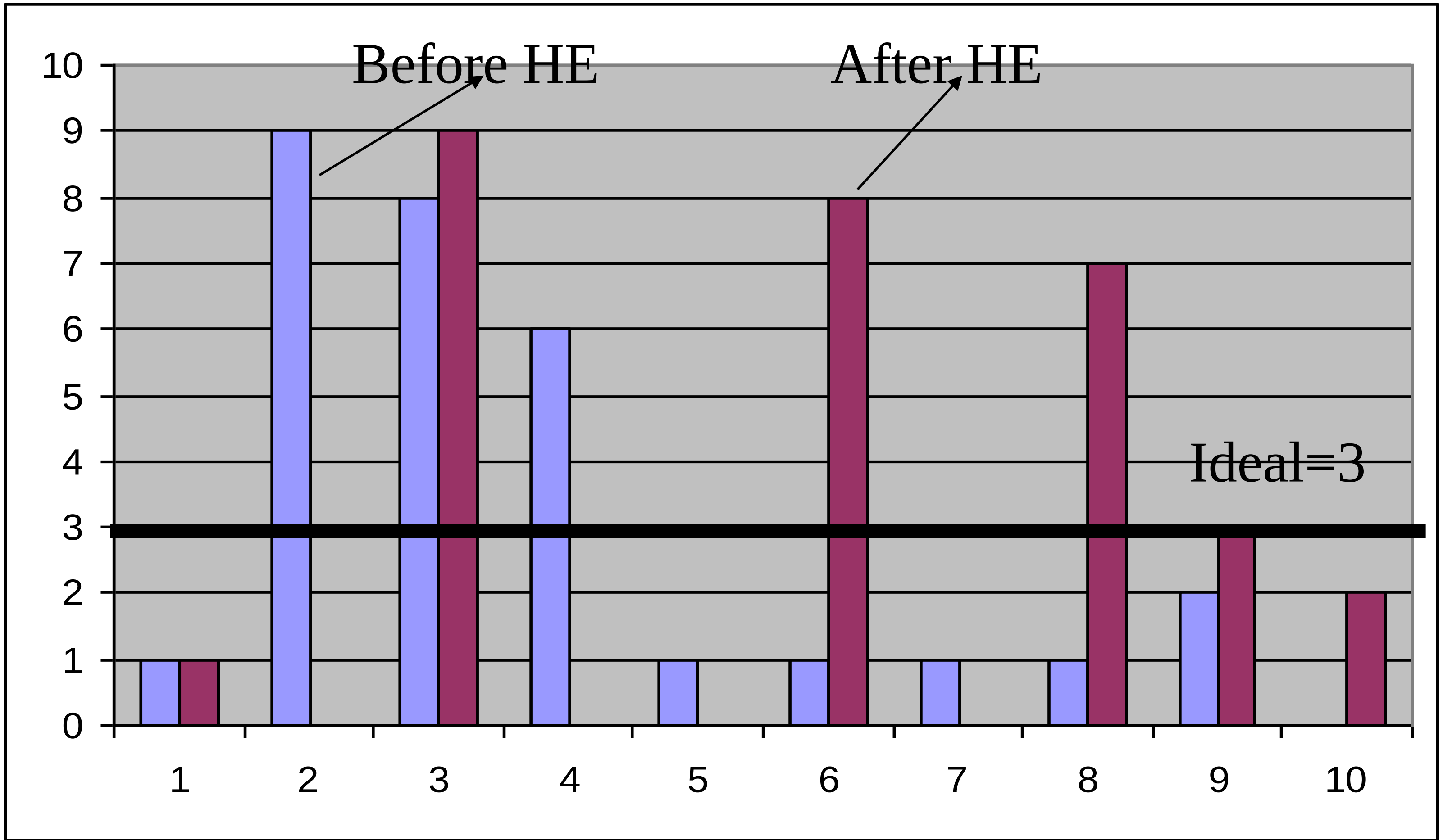
Simplified Formula for Histogram Equalization

$$E(l) = \max(o, round((\frac{L}{N*M})*t(l)) - 1)$$

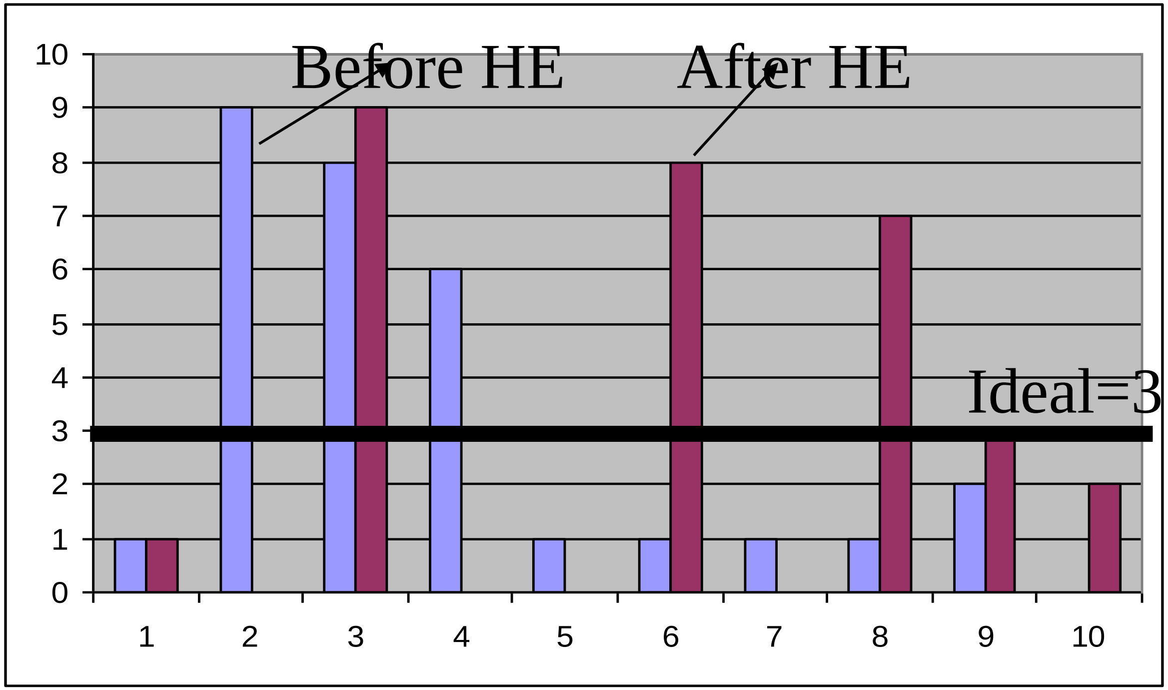
- **E(l)** → equalized function
- **max** → maximum dynamic range
- **round** → round to the nearest integer (up or down)
- **L** → no. of grey levels
- **N*M** → size of image
- **t(l)** → accumulated frequencies

Practical Histogram Equalization example

$$E(l) = \max(o, round((\frac{L}{N*M})*t(l))-1)$$



Histogram Equalization example



One pixel with value 1

9 pixels with value 2

Total number of intervals = 10

Total number of pixels = 3

Ideal (average) value = 3 = 30 /10

Original interval

Accumulated value

New interval

g	h(g)	t(g)	e(g)	New hist
1	1	1	1	1
2	9	10	3	0
3	8	18	6	9
4	6	24	8	0
5	1	25	8	0
6	1	26	9	8
7	1	27	9	0
8	1	28	9	7
9	2	30	10	3
10	0	30	10	2

1+9+8+7+3+2=30, as before

Where is Histogram Equalization Used?

- robot vision,
- photographics,
- aerial images

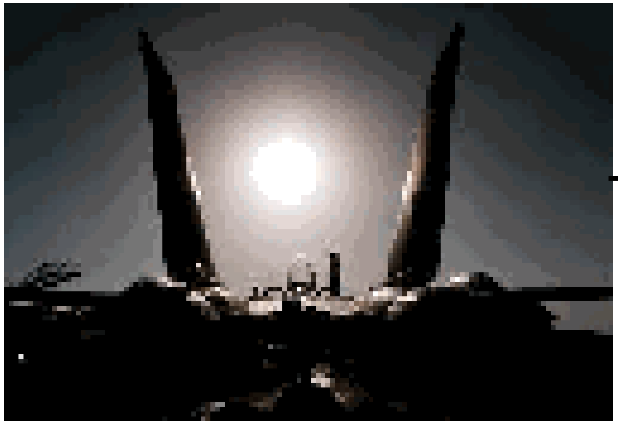
Normalizing Histograms

- Probability density function = histogram normalized by area

$$p(D) = \frac{1}{A} H(D)$$

Color Image histogram Equalization in Matlab

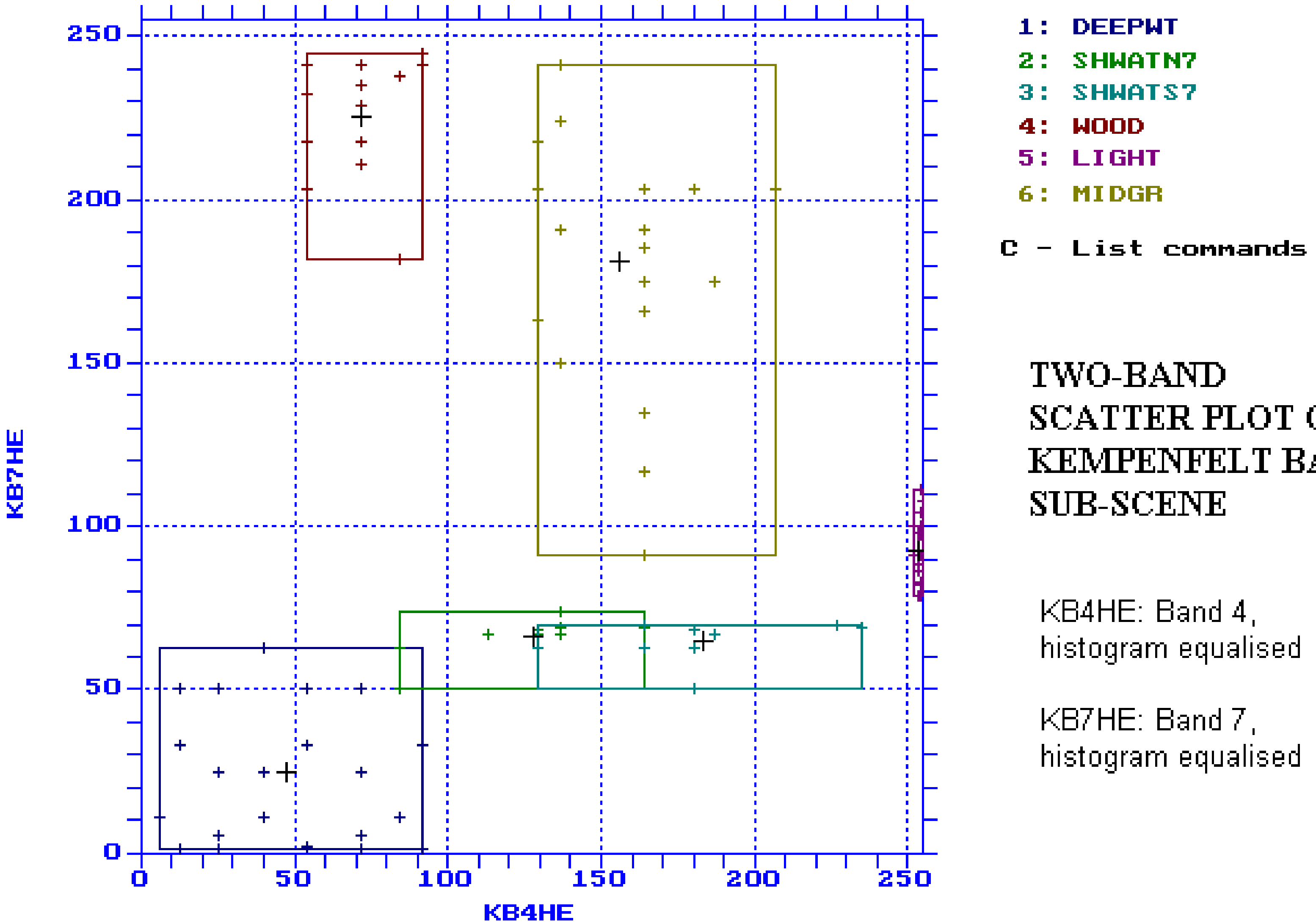
Image



```
Image = imread(‘Wings.jpg’);
HSVImage=rgb2hsv(Image);
VImage=HSVImage(:,:,3);
EImage=histeq(VImage,256);
HSVI(:,:,3)=EImage;
OutImage=hsv2rgb(HSVI);
```

OutImage

Two Band Scatter Plot



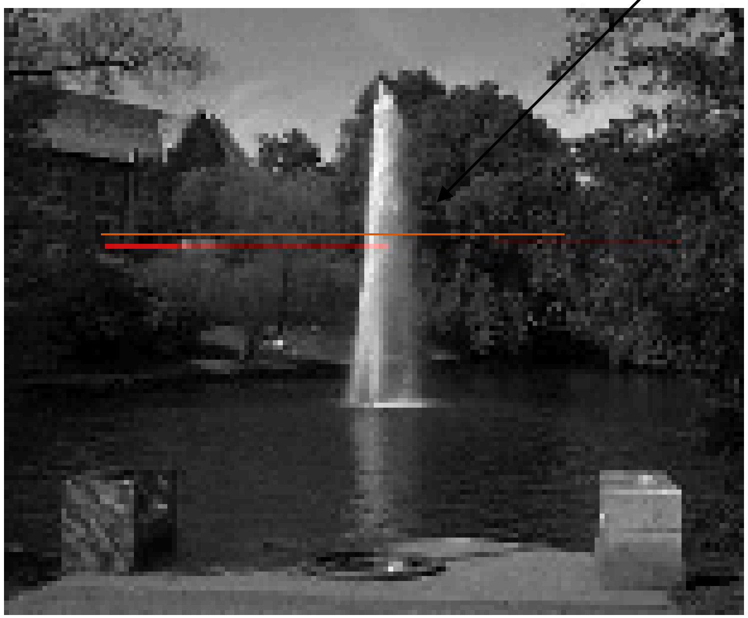
TWO-BAND SCATTER PLOT OF KEMPENFELT BAY SUB-SCENE

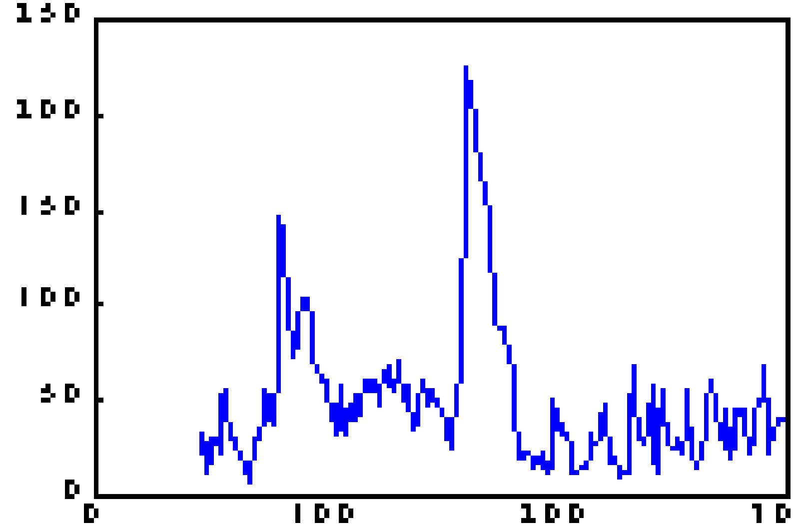
KB4HE: Band 4, histogram equalised

KB7HE: Band 7, histogram equalised

Improfile = image profile

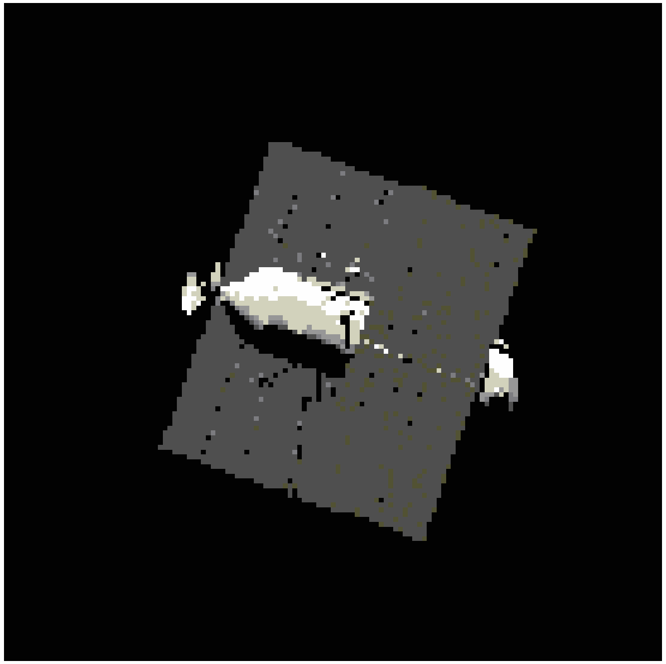
Profile taken

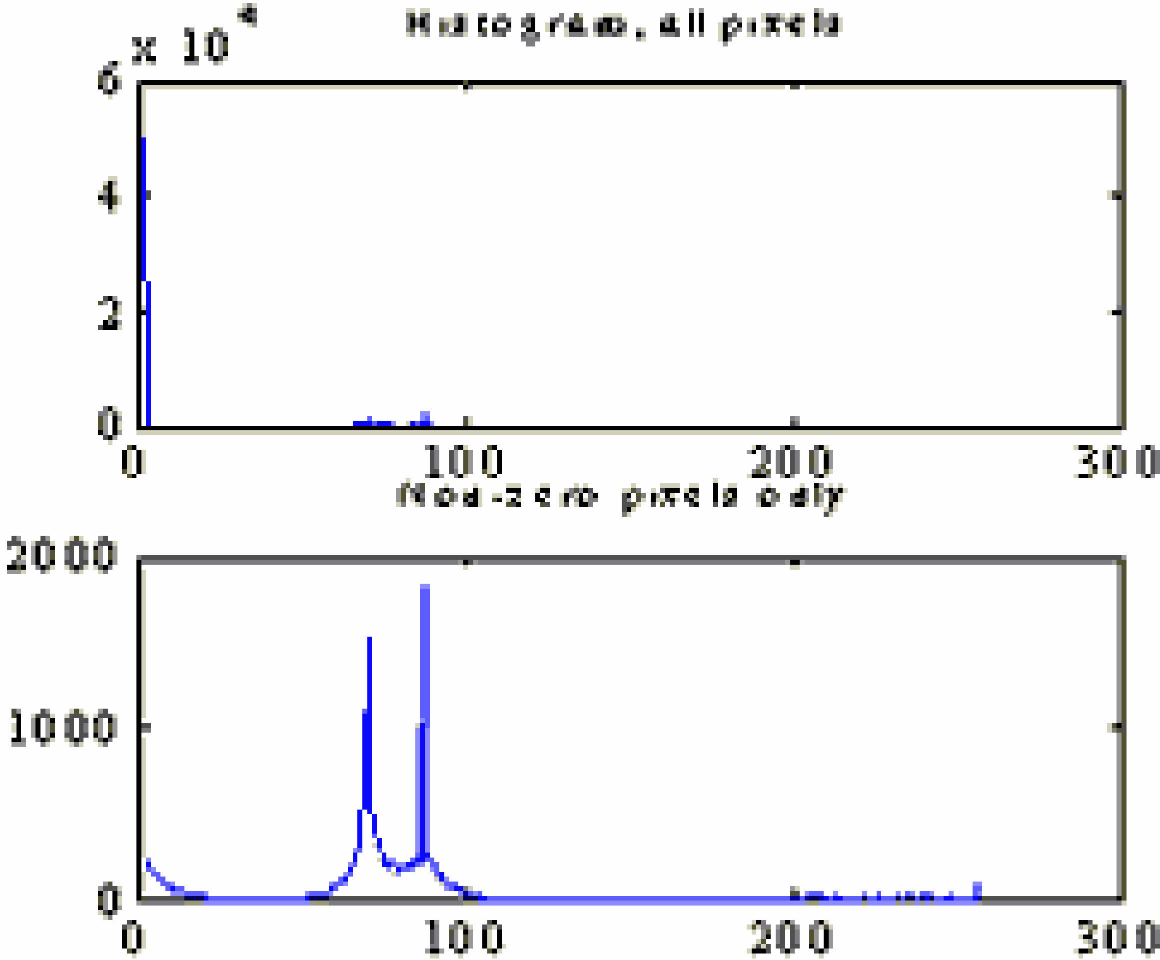




```
>> [cx, cy,  c] = improfile;
>> plot(cx, c);
```

Imhist = Image Histogram





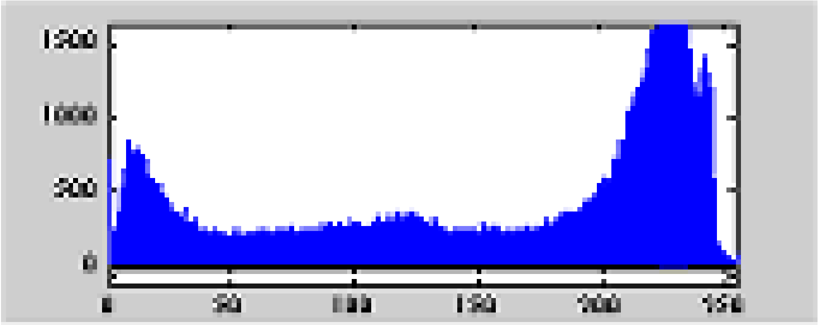
Histogram Stretching in Matlab

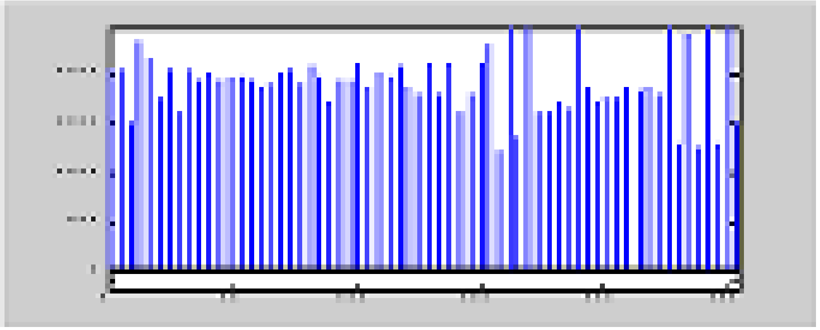
Image A

Image B

```
>> A = imread('tankfire.jpg');
>> B = imadjust(A, [0 75/255],[]);
```

Histogram Equalization in Matlab





```
>> A = imread('BigLogo.jpg');
>> B = histeq(A,64);
```

Can improve contrast

Used in preprocessing

Application: Adjusting Camera Parameters

- Too bright - lots of pixels at 255 (or max)
- Too dark - lots of pixels at 0
- Gain too low - not enough of the range used

Example of image enhancement

Application: Image Segmentation

- Can be used to separate bright objects from dark background (or vice versa)

Cumulative Histograms

- Counts pixels with values up to and including the specified value

$$C(a) = \int_0^a H(D)\,dD$$

Cumulative Density Functions

- Normalized cumulative histograms

$$P(a) = \int_0^a p(D)\,dD = \frac{1}{A}C(a)$$

IMAGE ENHANCEMENT

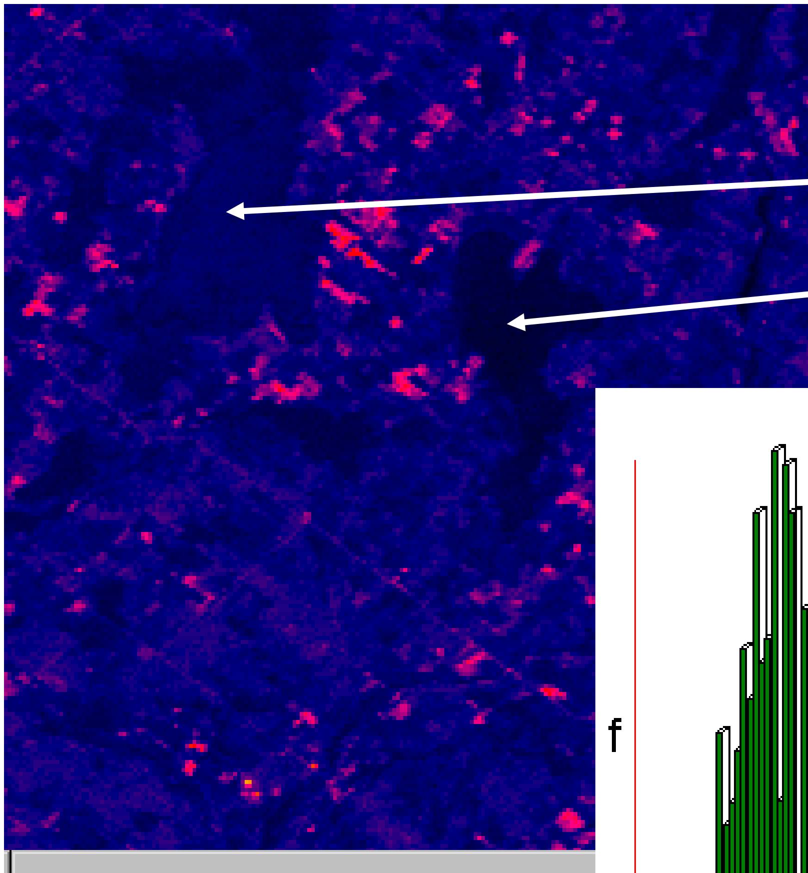
IMAGE ENHANCEMENT

- Due to the fact that original pixel values are integer values, and that frequency of the values varies with each class, the result will be an actual number of pixels in each class which only *approximates* the equalized percentage
- Alternative explanation which incorporates **probability** and a transformation function:
  - note the difference in the two histograms, original and equalized

IMAGE ENHANCEMENT

ORIGINAL MSS BAND 5 DATA

CAMDEN, ONT. area



Camden Lake

Varty Lake

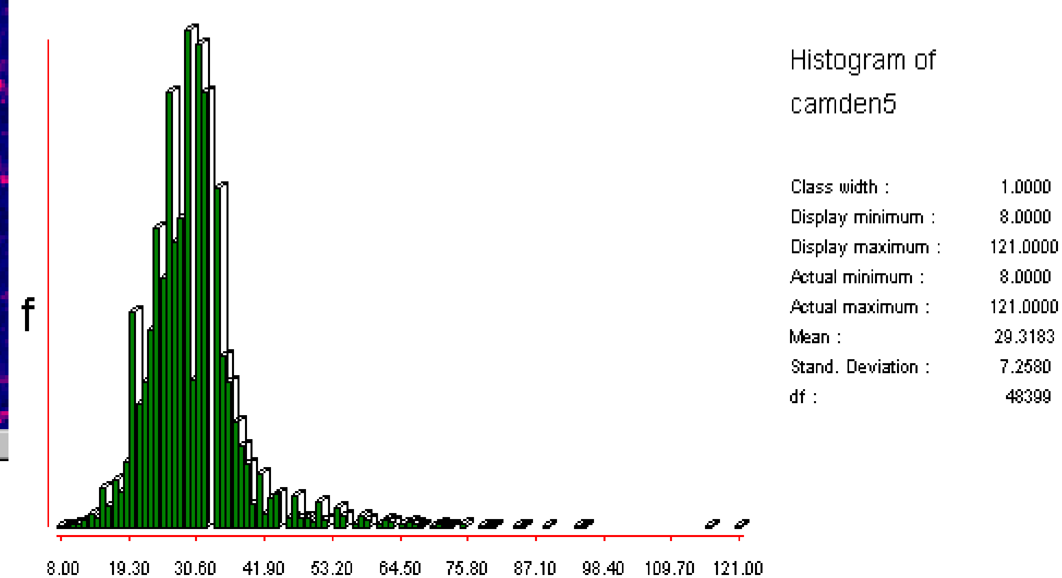
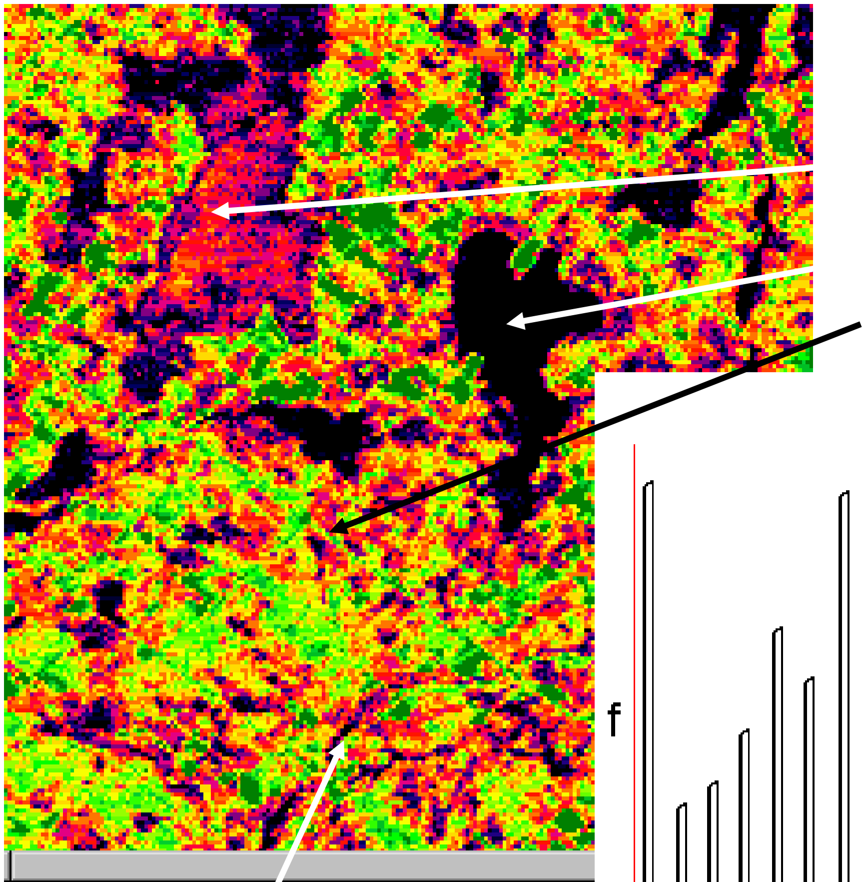


IMAGE ENHANCEMENT

LINEAR STRETCHED MSS BAND 5 DATA

CAMDEN, ONT. area



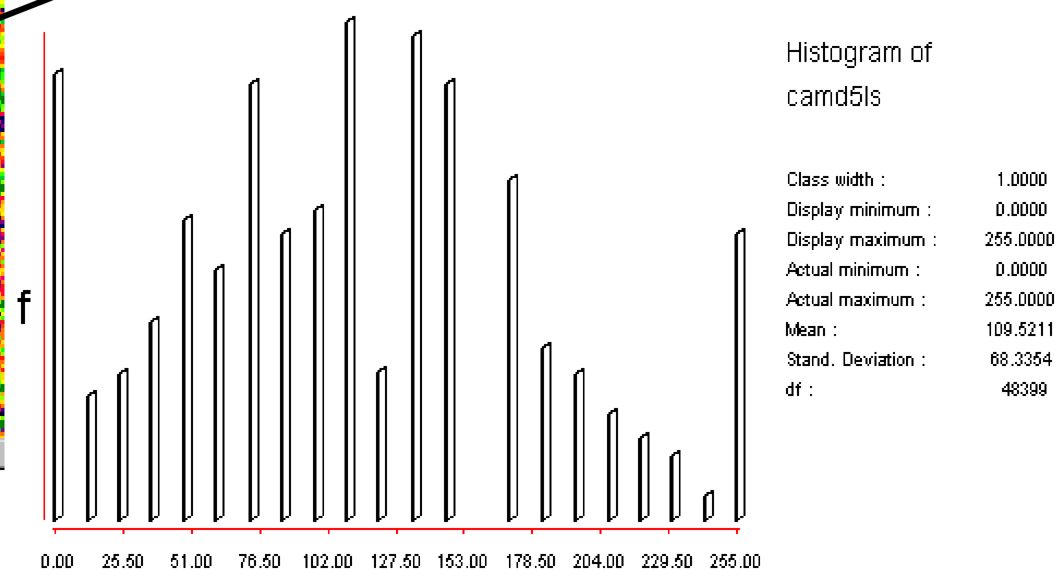


IMAGE ENHANCEMENT

SPATIAL FILTERING

- ***Spatial frequency***: "The number of changes in brightness value per unit distance for any particular part of an image" (Jensen)
  - ***Few changes*? Low frequency**
  - ***Many changes*? High frequency**

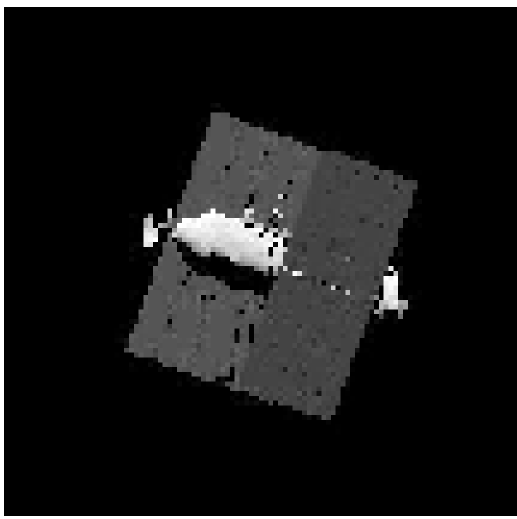
IMAGE ENHANCEMENT

- ***The principles?*** Pixel values along a single scan line can be represented by a complex curve which comprises many simple curves, each with its own constant wavelength

- The **complex curve** can be separated into its component wavelengths by mathematical process of filtering

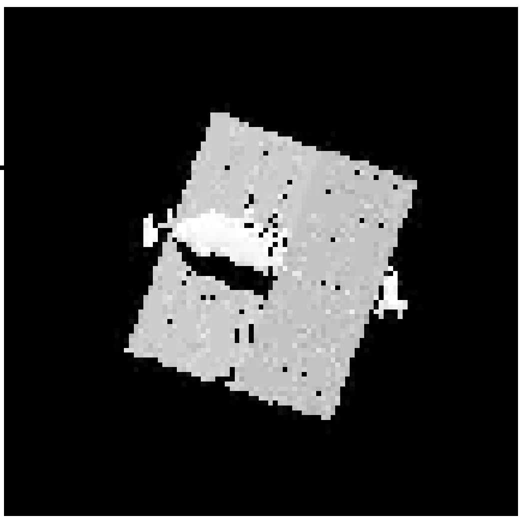
Logarithmic Contrast Enhancement in Matlab

Image



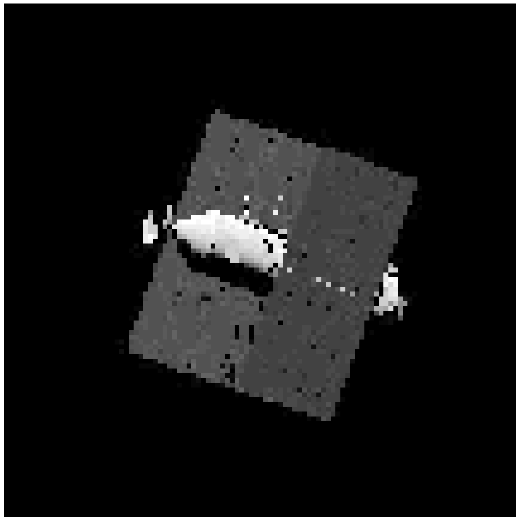
```
LogImage = log10(Image+1);
```

LogImage



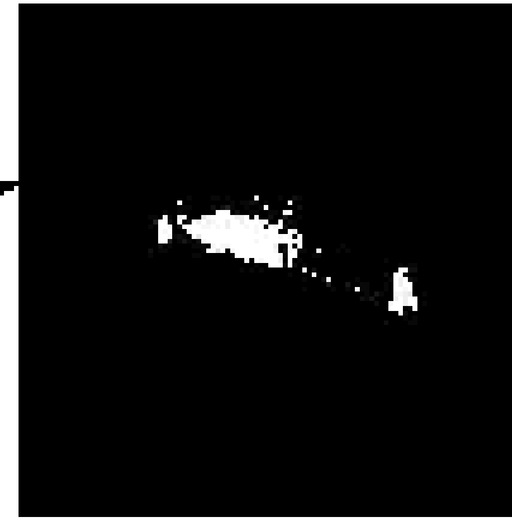
Thresholding in Matlab

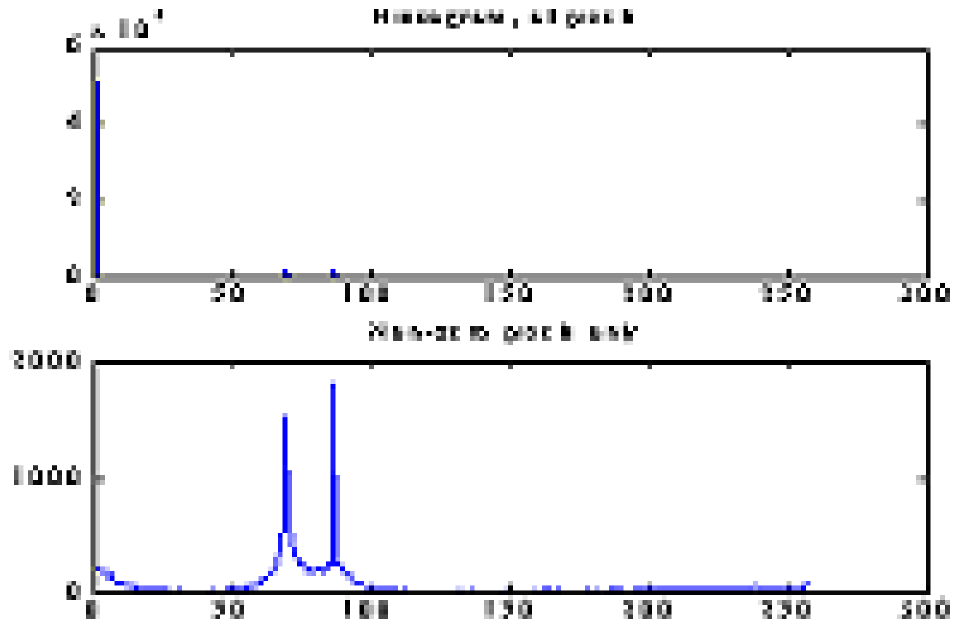
Image



im2bw(Image,0.54)

ThresholdImage





Edge Detection using gradient operators in Matlab

```
SobelEdge=edge(image,'sobel');
```

```
PrewittEdge=edge(image,'prewitt');
```

For Comparison, Edge Detection using the LOG

Combines LoG histogramming and convolution filtering

```
Edge1=edge(image,'log',0.006);
```

```
Edge2=edge(image,'log',0.001,4);
```

POINT OPERATIONS

- *Operates on ONLY 1 pixel at a time without considering neighboring values*
- *Thresholding*
- *Contrast stretching*
- *Temporal image smoothing*
- *Image difference*
- *Color adjustment or selection*

Thresholding

- ***Creates binary image from grayscale image***
- ***Image histogram***
- ***Determining threshold***

Temporal smoothing

- ***Noisy images, e.g. pictures from the moon***
- ***Several frames of the same scene***
- ***Take average of the same pixel value over time***
- ***Standard deviation of noise decreases on averaging***

Image difference over time

- ***Static camera is assumed***
- ***Compute I(t) - I(t+dt)***
- ***Threshold to eliminate small differences***
- ***Still scene ==> Nothing in difference***
- ***Moving object ==> object detected before & after motion***

Color adjustment/selection

- ***Hue, saturation, intensity***
- ***Red, green, blue***
- ***Selecting sky***
- ***Selecting grass***

Color models

- Color models for images : RGB, CMY
- Color models for video : YIQ, YUV (YCbCr)

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YIQ from RGB

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YCbCr from RGB

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YUV from RGB

$$C_b = B - Y \qquad U = 0.565(B - Y)$$

$$C_r = R - Y \qquad V = 0.713(R - Y)$$

Region and segmentation

- Region ($P_1, P_2, \ldots, P_k$)
  - A subset of an image
- Segmentation
  - Grouping of pixels into regions such that

$$\bigcup_{i=1}^{k} P_i = \text{Entire image}$$

$$P_i \bigcap P_j = \emptyset , \quad i \neq j$$

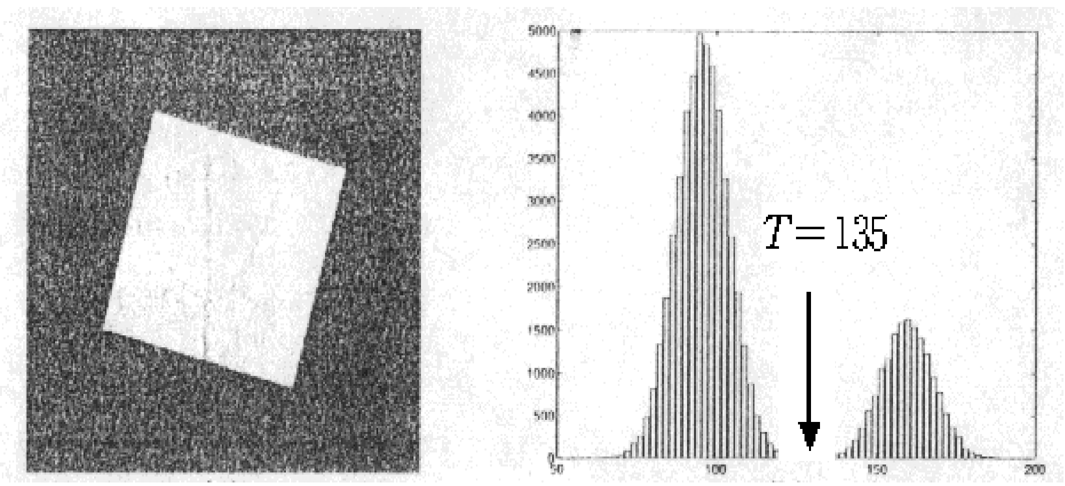
Thresholding

- Thresholding :
  - A method to convert a gray scale image into a binary image for object-background separation
- $F_T[i,j]$: Thresholded gray image
  - Obtained using a threshold ***T*** for the original gray image $F[i,j]$
- $B[i,j]$: Binary image = $F_T[i,j]$
- Two types of thresholding

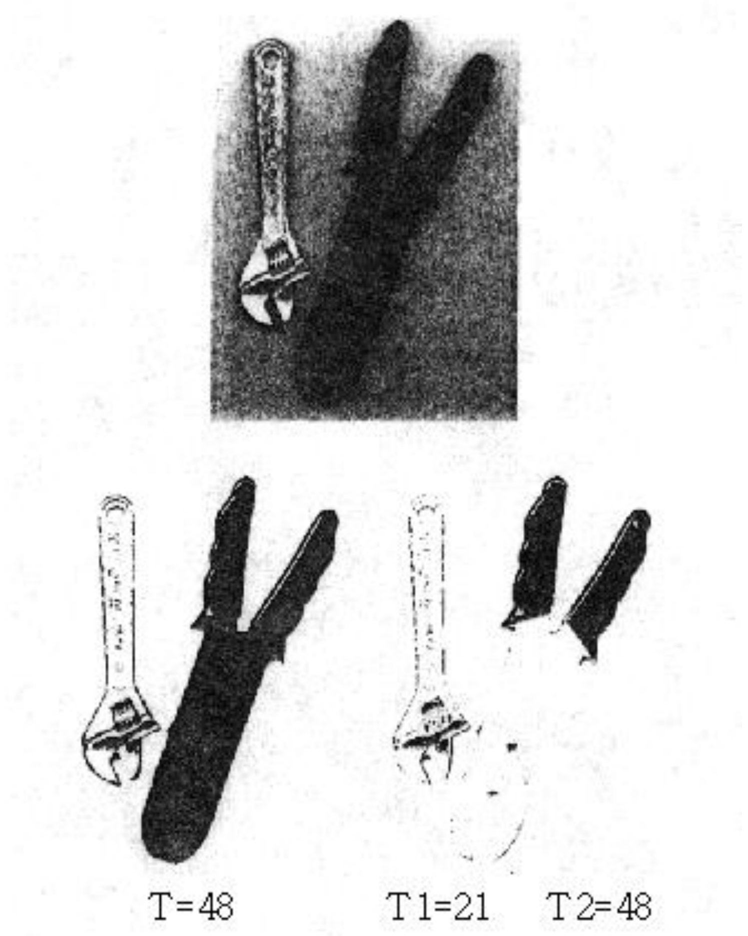
$$F_T[i,j] = \begin{cases} 1 & \text{if } F[i,j] \geq T \\ 0 & otherwise. \end{cases}$$

$$F_T[i,j] = \begin{cases} 1 & \text{if } T_1 \leq F[i,j] \leq T_2 \\ 0 & otherwise. \end{cases}$$

Thresholding



Original image and its histogram



Thresholding

Sources

- **Maja Mataric**
- **Dodds, Harvey Mudd College**
- **Damien Blond**
- **Alim Fazal**
- **Tory Richard**
- **Jim Gast**
- **Bryan S. Morse**
- **Gerald McGrath**
- **Vanessa S. Blake**
- **Matt Roach**

- **Bryan S. Morse**
- **Many WWW sources**
- **Anup Basu, Ph.D. Professor, Dept of Computing Sc. University of Alberta**
- **Professor Kim, KAIST**
- **H. Schultz, Computer science, University of Massachusetts, Web Site: www-edlab.cs.umass/cs570**

- **Many sources of slides from Internet**

http://www.cheng.cam.ac.uk/seminars/imagepro/

Sources

- 533 Text book
- http://sern.ucalgary.ca/courses/CPSC/533/W99/presentations/L2_24A_Lee_Wang/
  http://sern.ucalgary.ca/courses/CPSC/533/W99/presentations/L1_24A_Kaasten_Steller_Hoang/main.htm
  http://sern.ucalgary.ca/courses/CPSC/533/W99/presentations/L1_24_Schebywolok/index.html
  http://sern.ucalgary.ca/courses/CPSC/533/W99/presentations/L2_24B_Doering_Grenier/
- http://www.geocities.com/SoHo/Museum/3828/optical.html
- http://members.spree.com/funNgames/katbug/

Noise Reduction

Image Enhancement

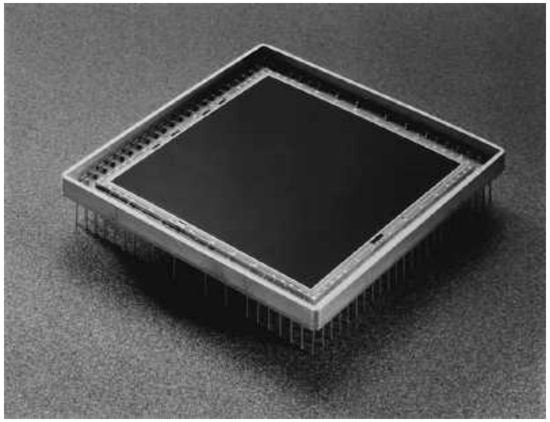
- Brightness control ✓
- Contrast enhancement ✓
- Noise reduction
- Edge enhancement
- Zooming

Objectives

- What is noise?
- How is noise reduction performed?
  - Noise reduction from first principles
  - Neighbourhood operators
    - linear filters (low pass, high pass)
    - non-linear filters (median)

Noise

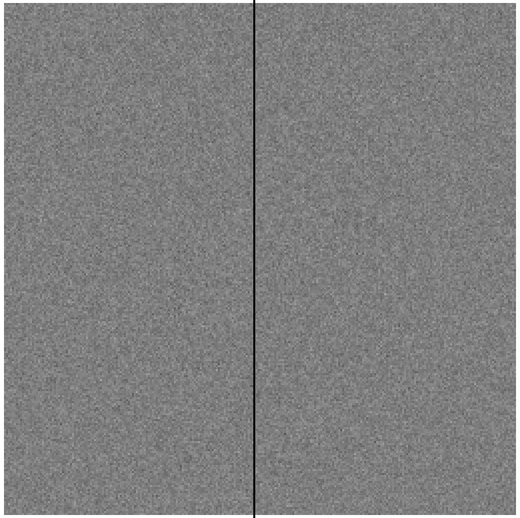
- Source of noise = CCD chip.
- Electronic signal fluctuations in detector.
- Caused by thermal energy.
- Worse for infra-red sensors.



Noise

image

+



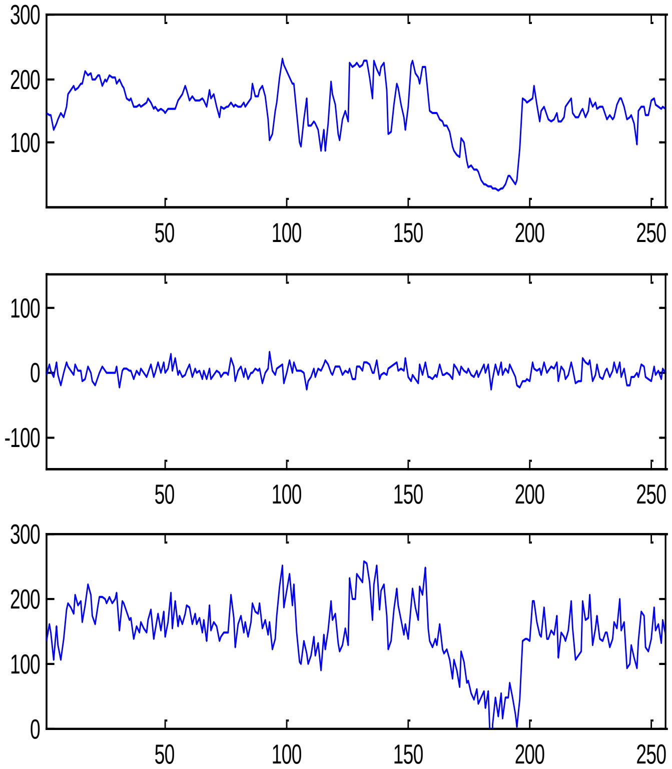
noise

=

‘grainy’
image

Noise

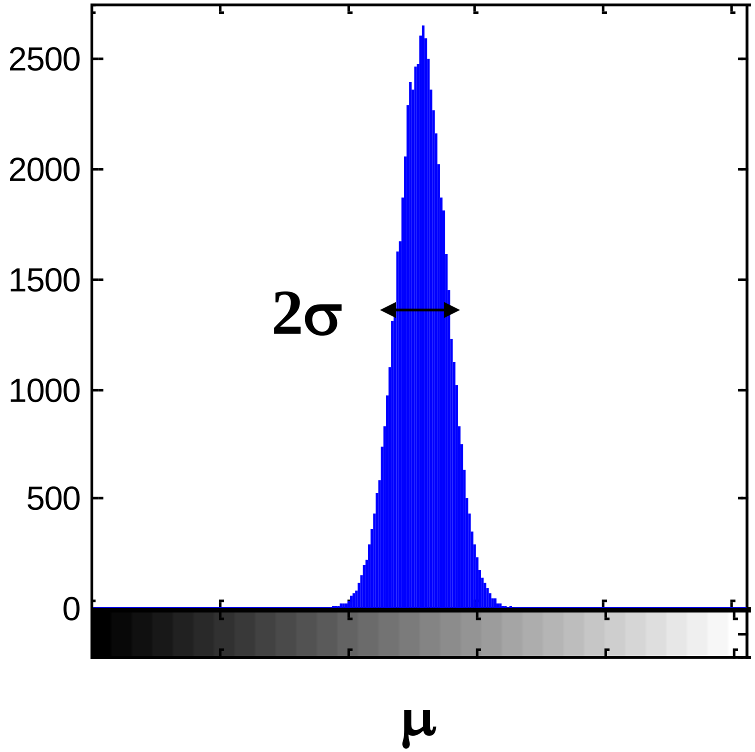
- Plot of image brightness.
- Noise is additive.
- Noise fluctuations are rapid, ie, high frequency.



Noise

- Plot noise histogram
- Histogram is called normal or Gaussian
- Mean(noise) μ = 0
- Standard deviation σ

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

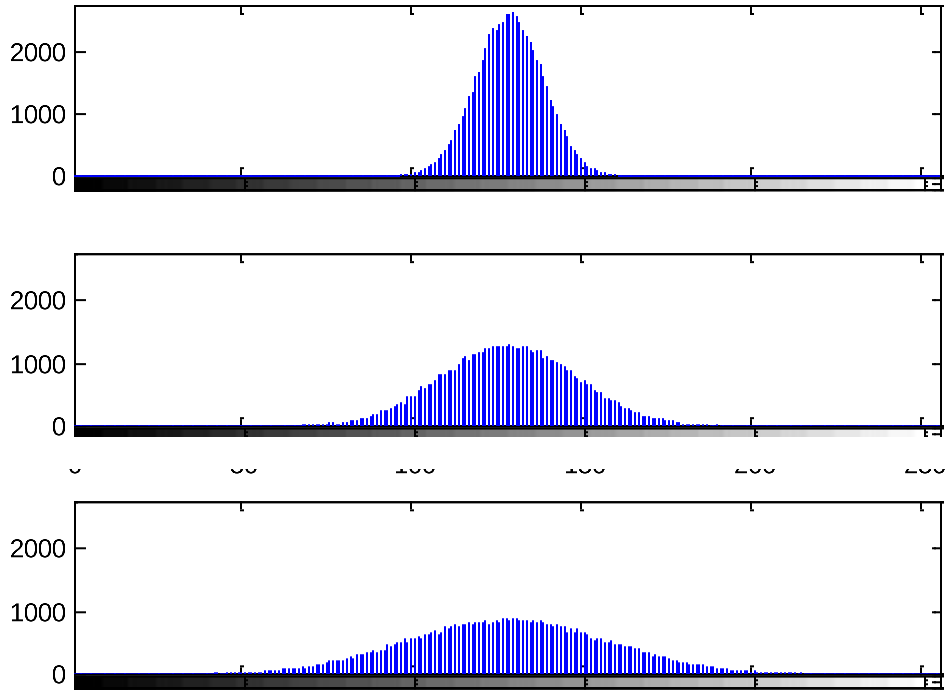


Noise

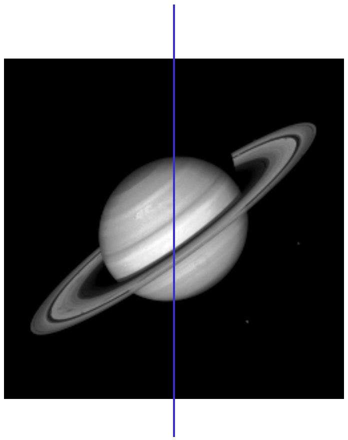
$\sigma=10$

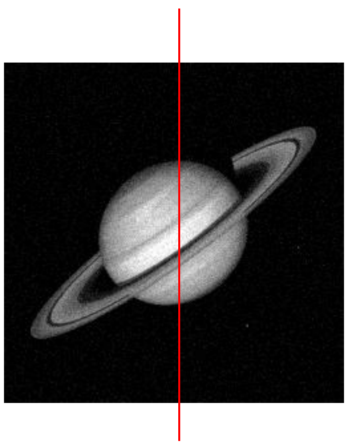
$\sigma=20$

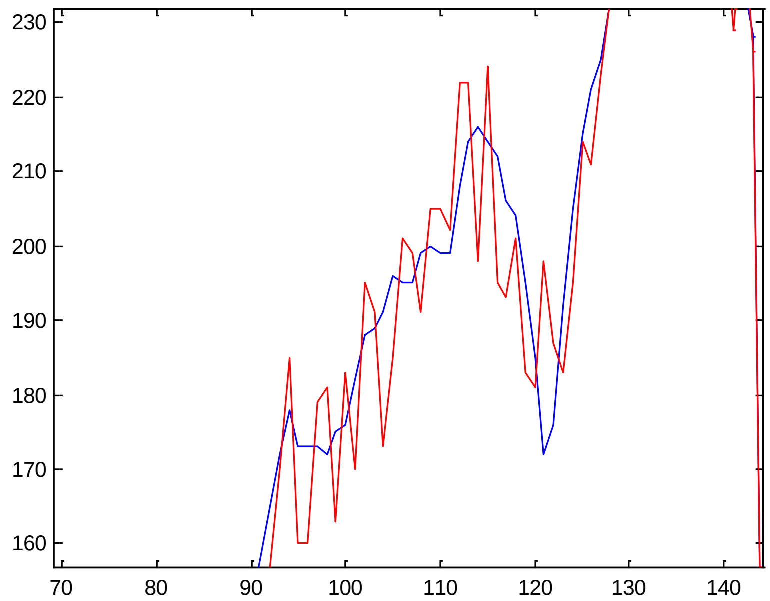
$\sigma=30$



Noise Reduction



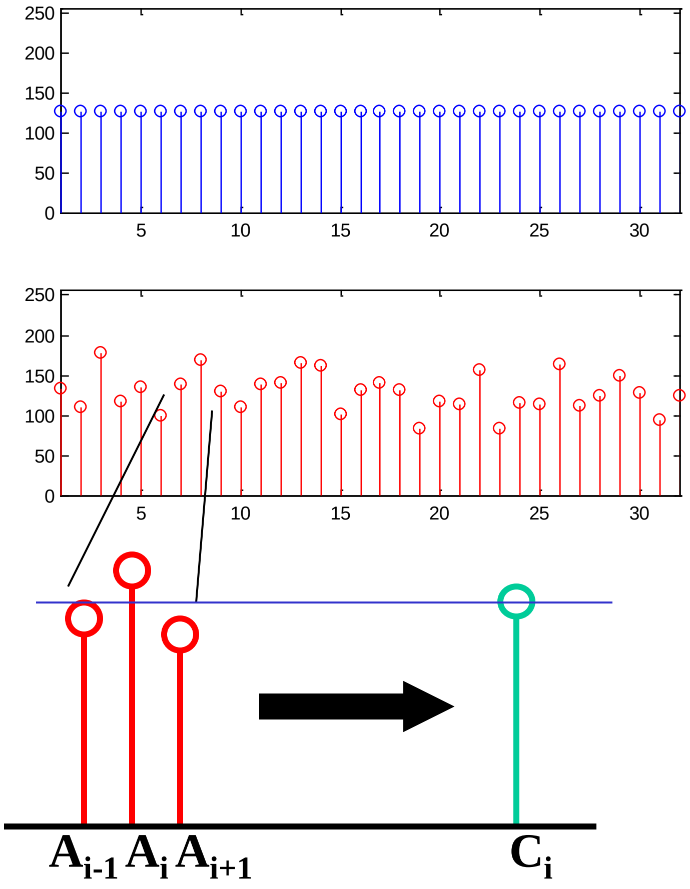




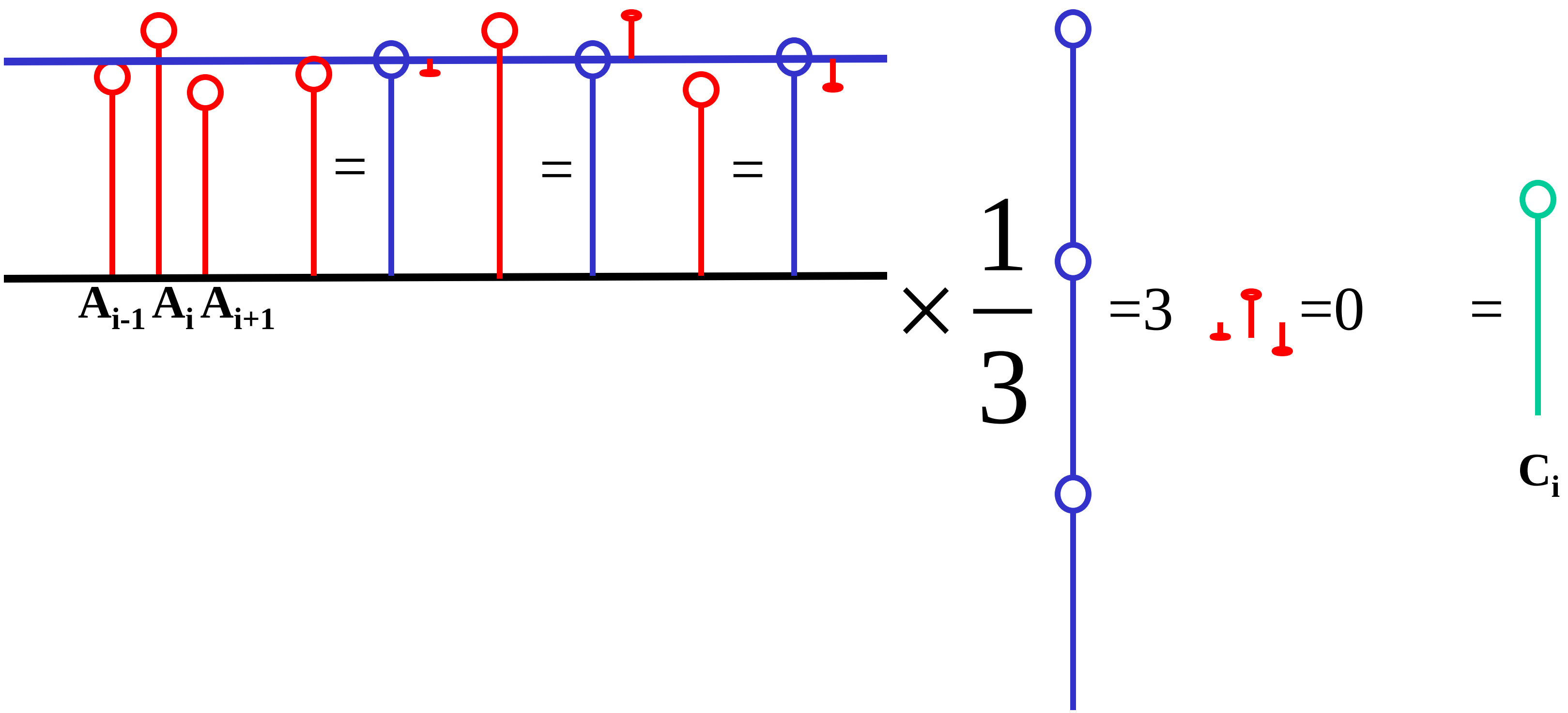
- Noise varies above and below uncorrupted image.

Noise Reduction - First Principles

- How do we reduce noise?
- Consider a uniform 1-d image and add noise.
- Focus on a pixel neighbourhood.
- Central pixel has been increased and <u>neighbouring</u> pixels have decreased.



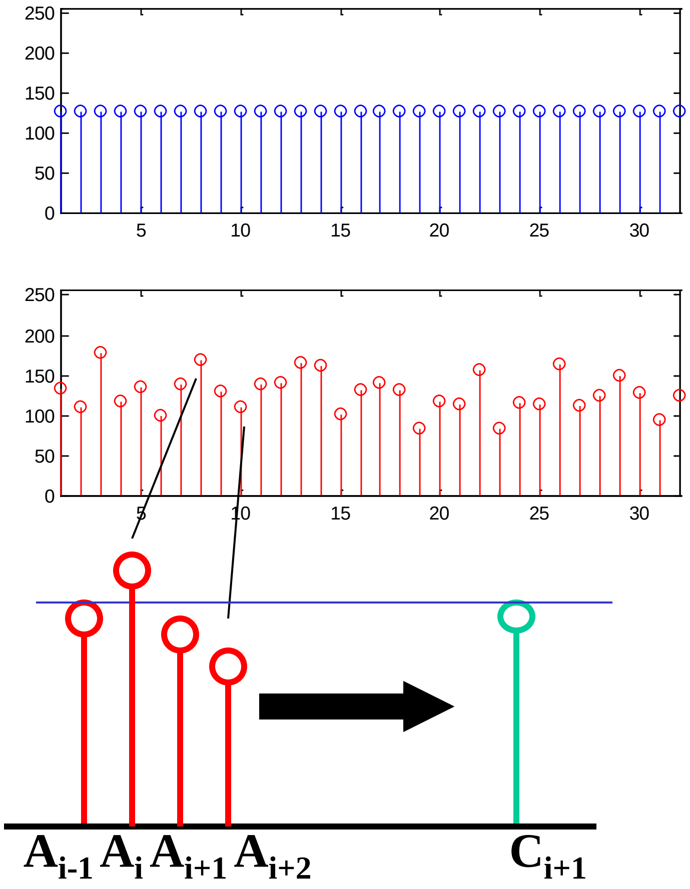
Noise Reduction - First Principles



Noise Reduction - First Principles

- Averaging 'smoothes' the noise fluctuations.
- Consider the next pixel A_{i+1}
- Repeat for remainder of pixels.

$$C_{i+1} = \frac{A_i + A_{i+1} + A_{i+2}}{3}$$



Noise Reduction - Neighbourhood operations

- All pixels can be averaged by convolving 1-d image A with mask B to give enhanced image C.
- Weights of B must equal one when added together.

$$\mathbf{C} = \mathbf{A} * \mathbf{B}$$

$$\mathbf{B} = \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix}$$

$$C_i = A_{i-1} \times B_1 + A_i \times B_2 + A_{i+1} \times B_3$$

$$B = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$C_i = \frac{A_{i-1} + A_i + A_{i+1}}{3}$$

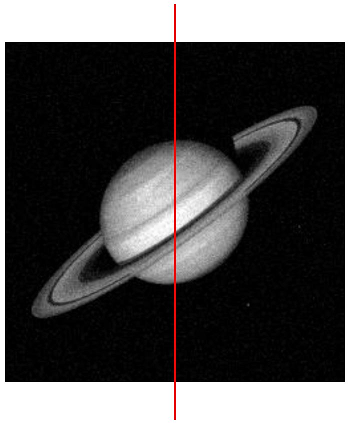
Noise Reduction - Neighbourhood operations

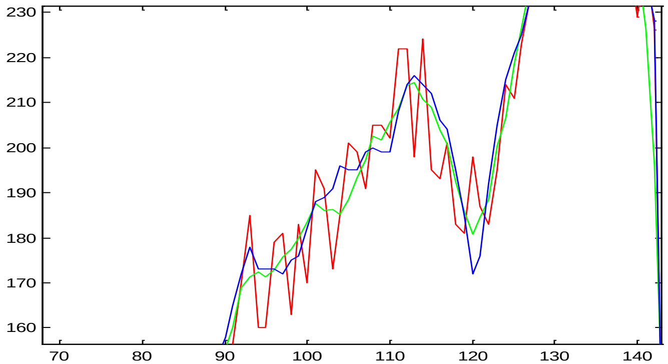
- Extend to two dimensions.

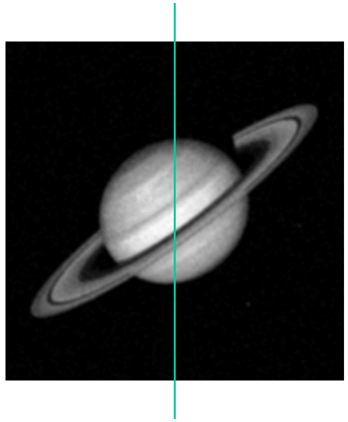
$$\mathbf{C} = \mathbf{A} * \mathbf{B}$$

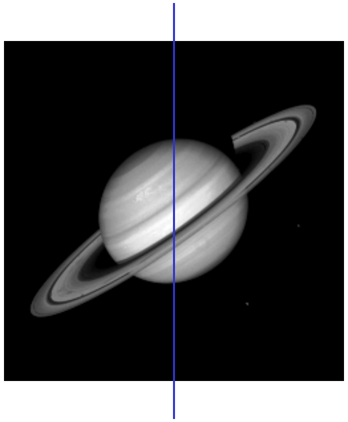
$$\mathbf{B} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Noise Reduction









Noise Reduction

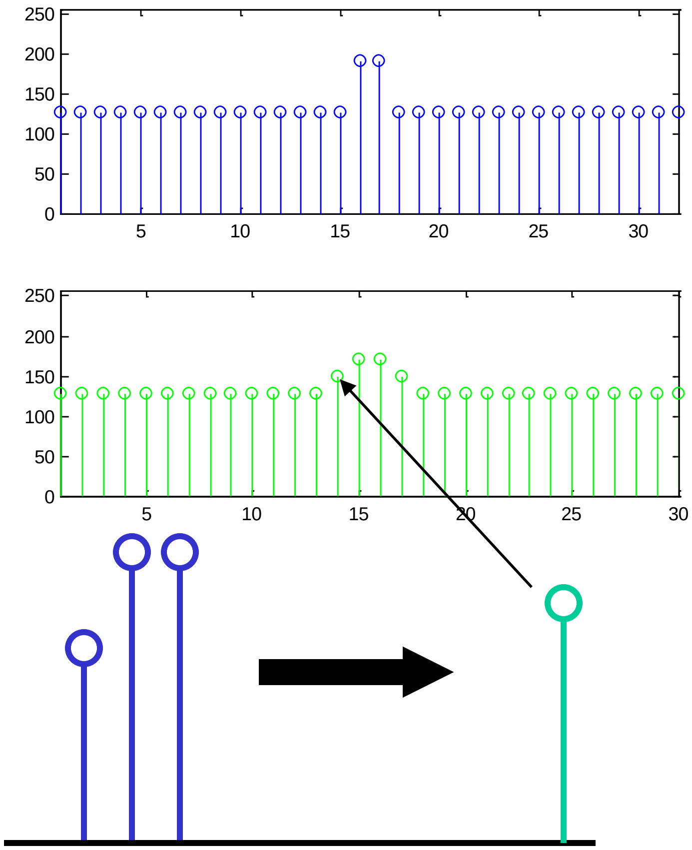
- Technique relies on high frequency noise fluctuations being ‘blocked’ by filter. Hence, low-pass filter.
- Fine detail in image may also be smoothed.
- Balance between keeping image fine detail and reducing noise.

Noise Reduction

- Saturn image coarse detail
- Boat image contains fine detail
- Noise reduced but fine detail also smoothed

Noise Reduction

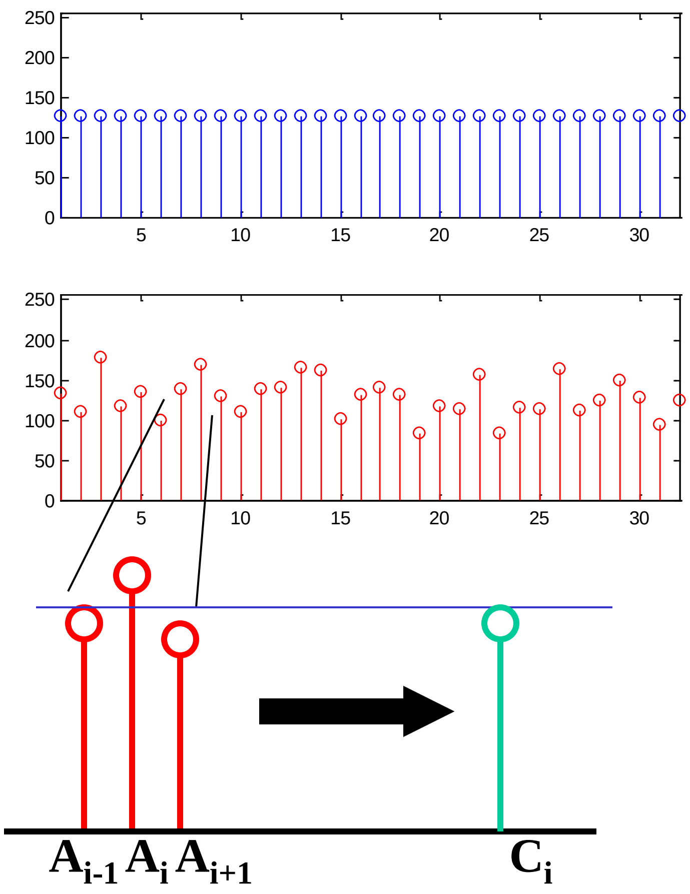
- Consider a uniform 1-d image with a step function.
- Step function corresponds to fine image detail such as an edge.
- Low-pass filter 'blurs' the edge.



Noise Reduction - First Principles

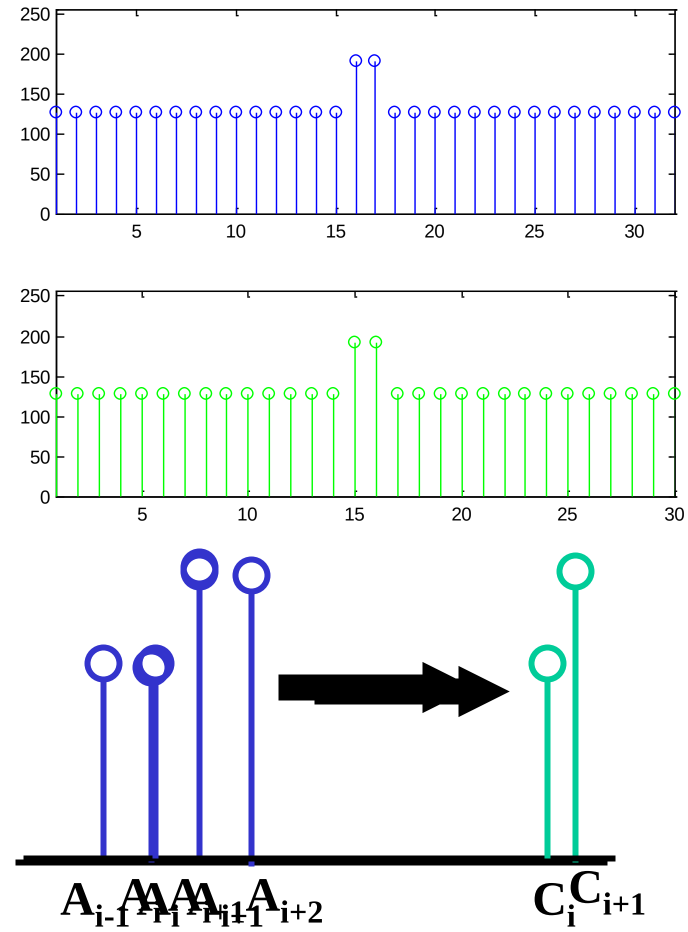
- How do we reduce noise without averaging?
- Consider a uniform 1-d image and add noise.
- Focus on a pixel neighbourhood.
- Non-linear operator?

Median filter!



Noise Reduction - First Principles

- Consider a uniform 1-d image with a step function.
- Step function corresponds to fine image detail such as an edge.
- Median filter does not 'blur' the edge.



Noise Reduction - Neighborhood operations

- All pixels can be replaced by neighbourhood median by convolving 1-d image A with median filter B to give enhanced image C.

$$\mathbf{C} = \mathbf{A} * \mathbf{B}$$

$$\mathbf{B} = \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix}$$

$$C_i = \text{median}\left\{A_{i-1} \times B_1, A_i \times B_2, A_{i+1} \times B_3\right\}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$C_i = \text{median}\left\{A_{i-1}, A_i, A_{i+1}\right\}$$

Noise Reduction - Neighborhood operations

- Extend to two dimensions.

$$C_{k,l} = \underset{i=k:k+M-1,\, j=l:l+N-1}{\text{median}} \left\{ A_{i,j} \times B_{i-k+1,\, j-l+1} \right\}$$

$$B_{i,j} = 1 \text{ for all } i,j$$

Noise Reduction

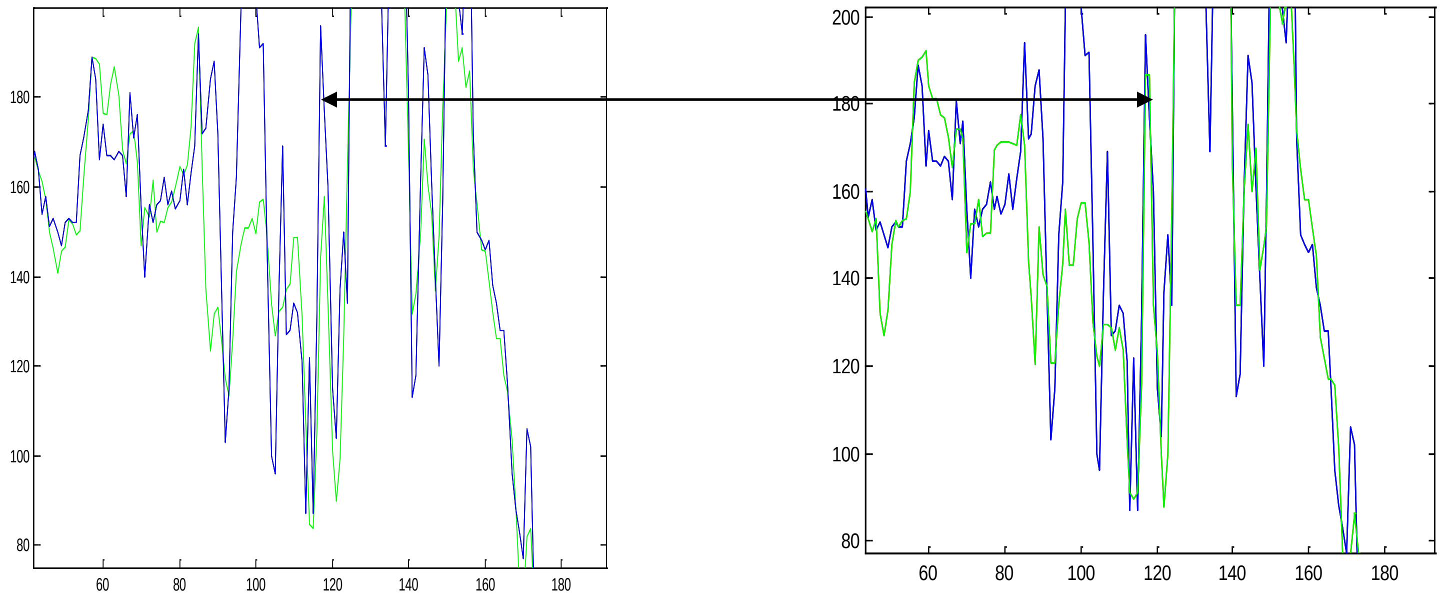
Original

Low-pass

Median

Noise Reduction

Low-pass | Median



- •Low-pass: fine detail smoothed by averaging
- •Median: fine detail passed by filter

Summary

- What is noise?
  - Gaussian distribution
- Noise reduction
  - first principles
- Neighbourhood
  - low-pass
  - median

Conclusion

- Averaging pixels corrupted by noise cancels out the noise.
- Low-pass can blur image.
- Median can retain fine image detail that may be smoothed by averaging.