Computer Vision

Lec 4: Image Filtering (contd.), Canny Edge Detector

Dr. Pratik Mazumder

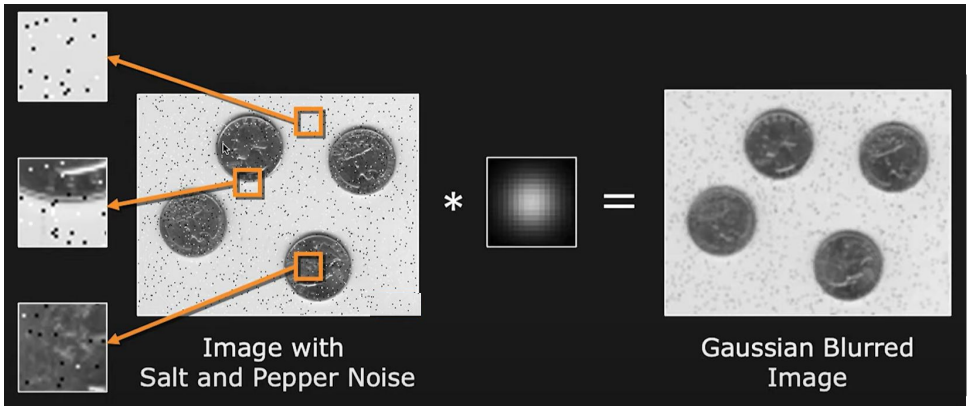
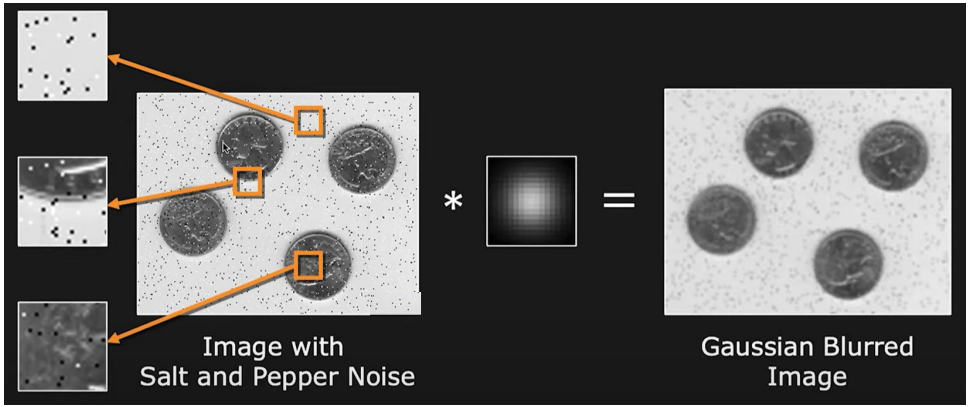


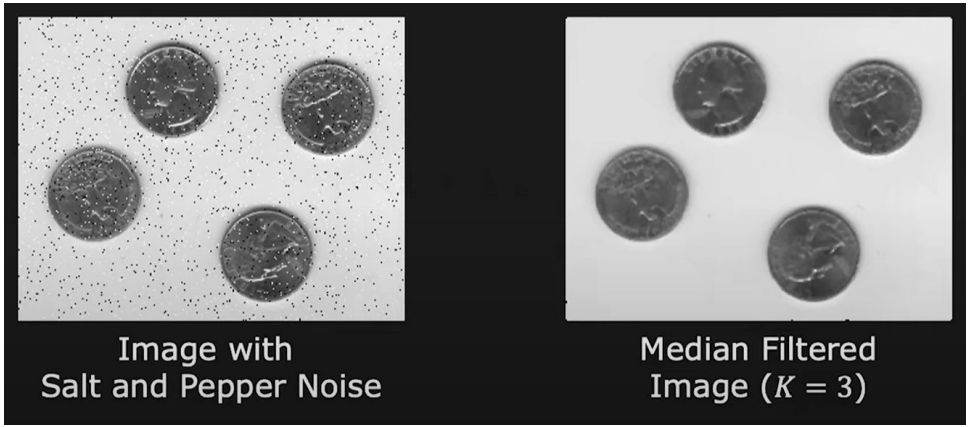
Image with
Salt and Pepper Noise
*
=
Gaussian Blurred
Image

Non Linear Filters: Median Filter

- For a region of size KxK centered around each pixel in the image (except those where a KxK region is not possible)
    - Sort the K^2 values in the region of the image
    - The output for this filter is the middle value of the sorted list [MEDIAN]

Non Linear Filters: Median Filter

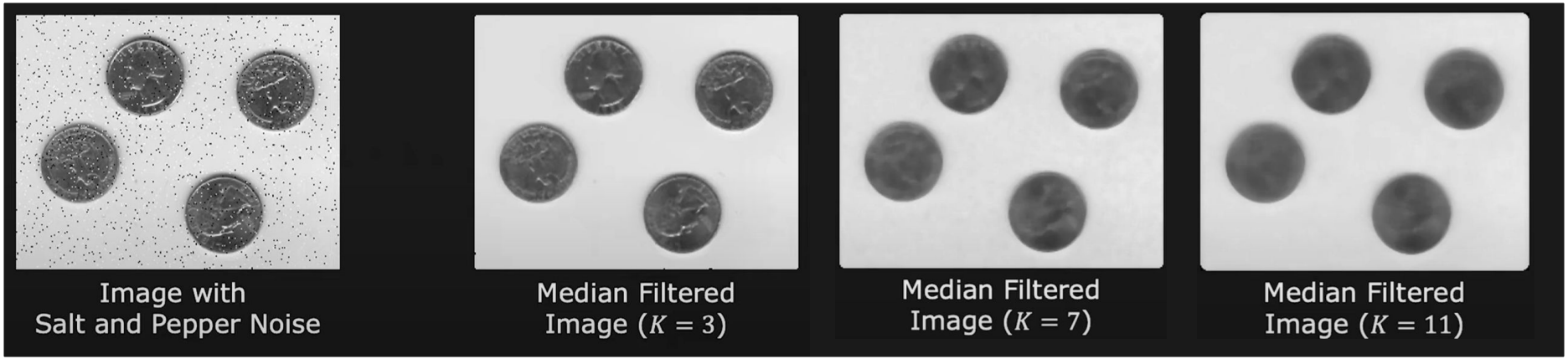




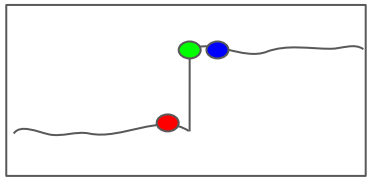
Non Linear Filters

- A Median Filter operates over a window by selecting the median intensity in the window.
- A Min Filter operates over a window by selecting the minimum intensity in the window.
- A Max Filter operates over a window by selecting the maximum intensity in the window.
- Cannot be implemented using convolution/correlation.

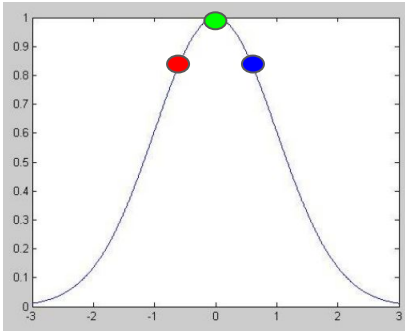
Non Linear Filters: Median Filter

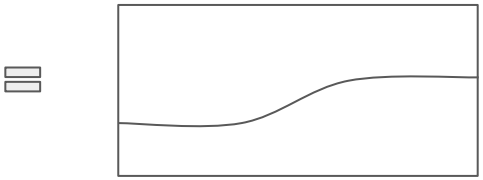


Gaussian Blurring Effect



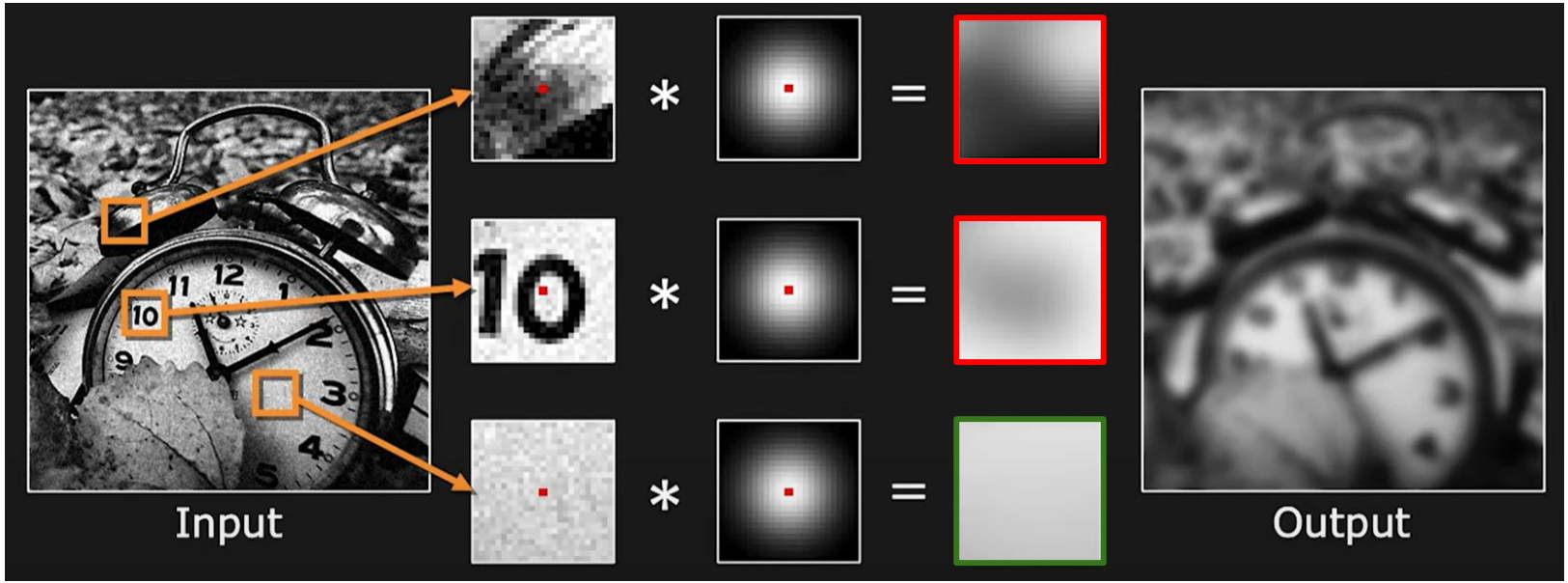
Noisy Input





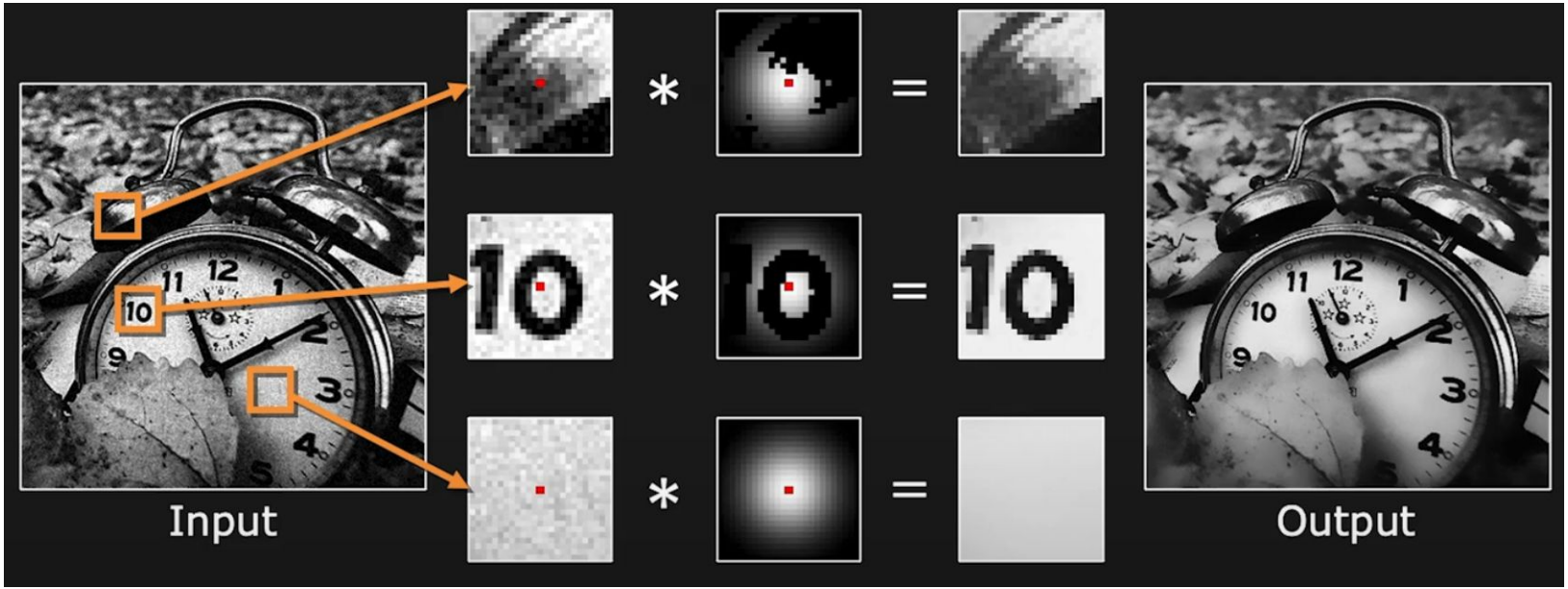
Gaussian blurring often blurs out the edges

Gaussian Blurring Effect



Gaussian smoothing: Same Filter Applied at all Positions

Gaussian Blurring: A slight modification



Modified Gaussian Blurring: Pixels in the window very different from the central pixel in terms of intensity are given a lower weight, effectively restricting the operation to similar pixels.

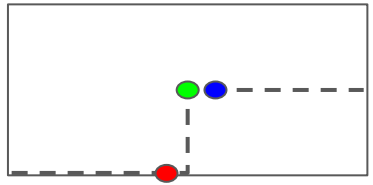
Bilateral Filter

- The bilateral filter is also defined as a weighted average of nearby pixels, in a manner very similar to Gaussian smoothing.

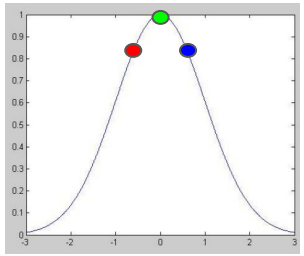
- The difference is that the bilateral filter takes into account the difference in value with the neighbors to preserve edges while smoothing.

- The key idea of the bilateral filter is that for a pixel to influence another pixel, it should not only occupy a nearby location but also have a similar value.

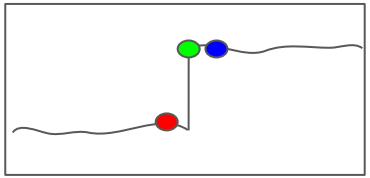
Bilateral Filter



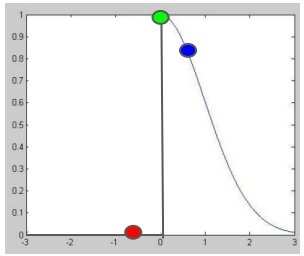
multiply



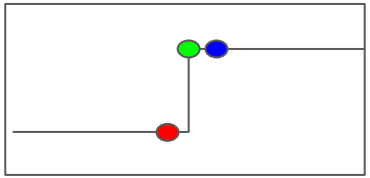
=



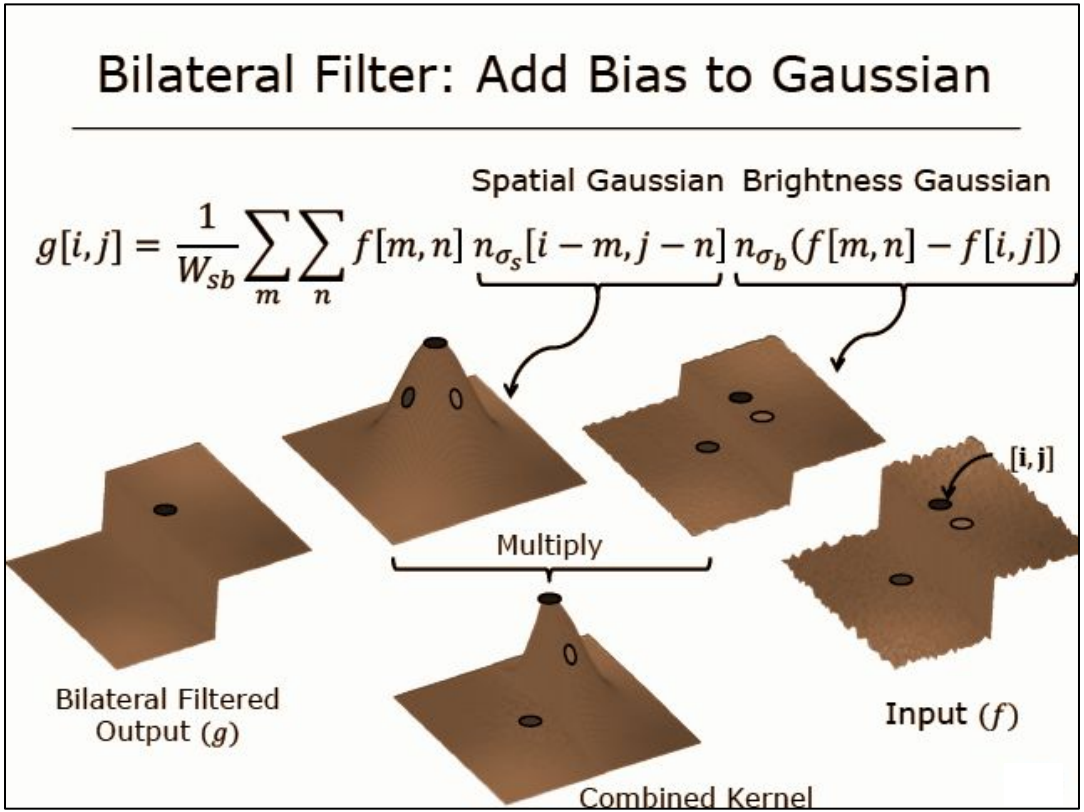
Noisy Input



=



Bilateral Filter



Bilateral Filter

$$g[i,j] = \frac{1}{W_{sb}} \sum_m \sum_n f[m,n]\, n_{\sigma_s}[i-m, j-n]\, n_{\sigma_b}(f[m,n] - f[i,j])$$

Where:

$$n_{\sigma_s}[m,n] = \frac{1}{2\pi {\sigma_s}^2} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma_s^2}\right)} \qquad n_{\sigma_b}(k) = \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{1}{2}\left(\frac{k^2}{\sigma_b^2}\right)}$$

$$W_{sb} = \sum_m \sum_n n_{\sigma_s}[i-m, j-n]\, n_{\sigma_b}(f[m,n] - f[i,j])$$

Non-linear Operation

(Cannot be implemented using convolution)

Bilateral Filter

Noisy Image

Gaussian
$\sigma_s = 2$

BiLateral
$\sigma_s = 2, \sigma_b = 10$

Bilateral Filter

Noisy Image

Gaussian
$\sigma_s = 4$

BiLateral
$\sigma_s = 4$, $\sigma_b = 10$

Bilateral Filter

Noisy Image

Gaussian
$\sigma_s = 8$

BiLateral
$\sigma_s = 8, \sigma_b = 10$

Bilateral Filter

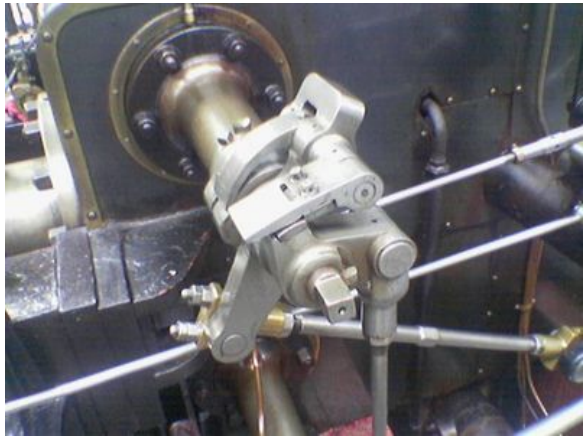
BiLateral
$\sigma_s = 6, \sigma_b = 10$

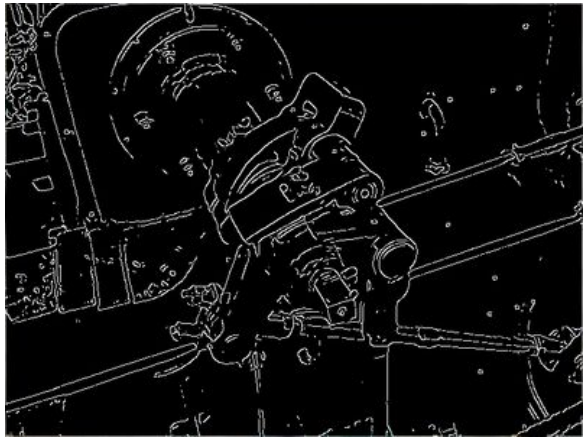
BiLateral
$\sigma_s = 6, \sigma_b = 20$

BiLateral
$\sigma_s = 6, \sigma_b = \infty$

Canny Edge Detector

- Canny edge detector is indeed one of the most widely used edge detection techniques in computer vision.
- Developed by John F. Canny in 1986.





Canny Edge Detector

- Smooth Image with Gaussian filter
- Compute Derivative of filtered image
- Find Magnitude and Orientation of gradient
- Apply Non-max suppression: To refine edges
- Apply double thresholding: To classify edges into strong, weak, and non-edges.
- Edge tracking by hysteresis: To connect weak edges to strong edges.

Canny Edge Detector

- **Smoothing**

$$S = I * g(x, y) = g(x, y) * I$$

$$g(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

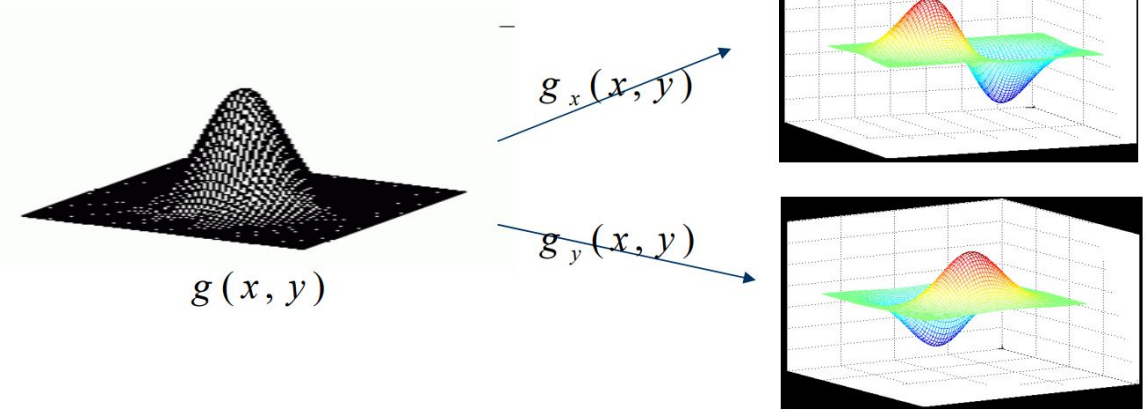
- **Derivative**

$$\nabla S = \nabla (g * I) = (\nabla g) * I$$

$$\nabla S = \begin{bmatrix} g_x \\ g_y \end{bmatrix} * I = \begin{bmatrix} g_x * I \\ g_y * I \end{bmatrix}$$

$$\nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

Canny Edge Detector



Instead of applying Gaussian Blurring and then Derivative Filtering to the Image , we can apply the Derivative Filtering to the Gaussian Kernel/Filtering first and then apply the resulting Filter to the Image

Canny Edge Detector

- Gradient magnitude and gradient direction

(S_x, S_y) Gradient Vector

magnitude $= \sqrt{(S_x^2 + S_y^2)}$

direction $= \theta = \tan^{-1} \frac{S_y}{S_x}$

image

gradient magnitude

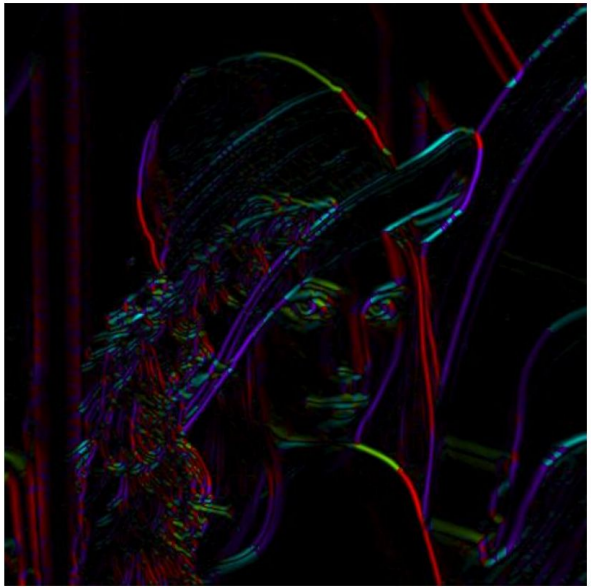
Canny Edge Detector

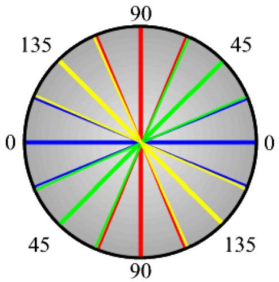
X-Derivative of Gaussian

Y-Derivative of Gaussian

Gradient Magnitude

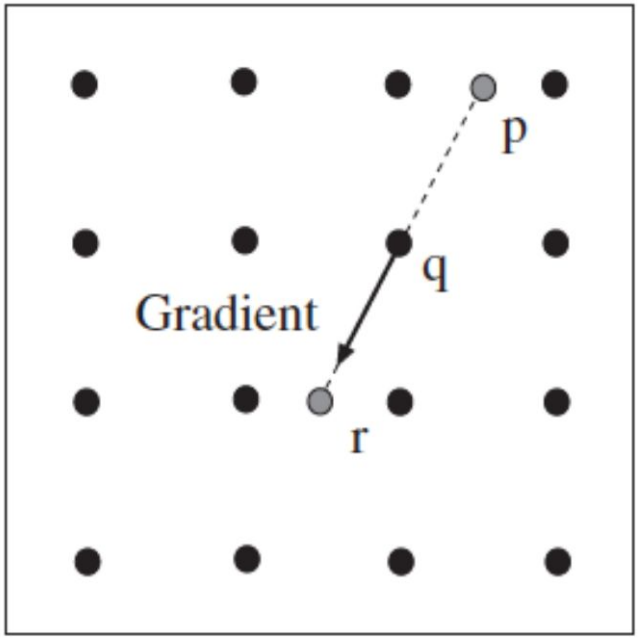
Canny Edge Detector: Gradient Orientation

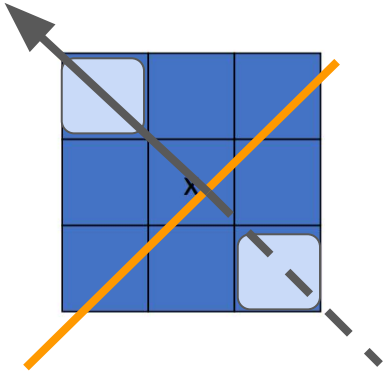


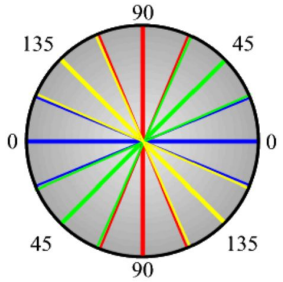


- Obtain the gradient orientation [$\tan^{-1}(g_y/g_x)$].
- For visualization, assign a color to various angles and map the orientation to these colors
- The above color wheel is just an example - may not match the orientation image
- Observe that edges having similar orientations have similar color

Canny Edge Detector: Non-maximum suppression

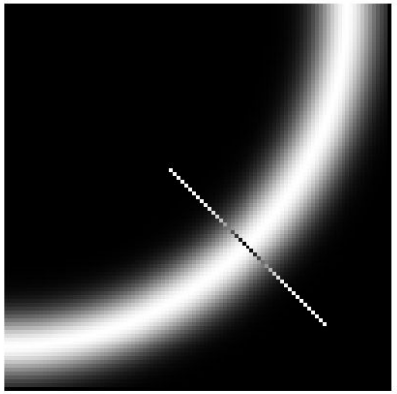


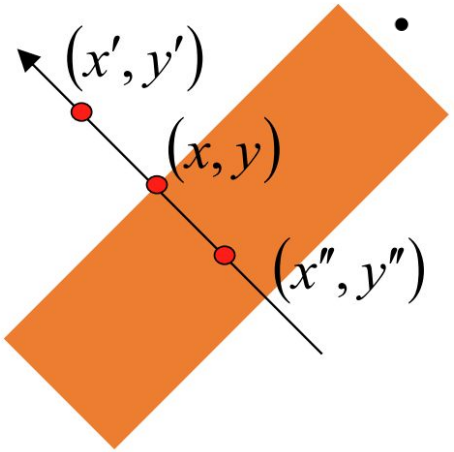


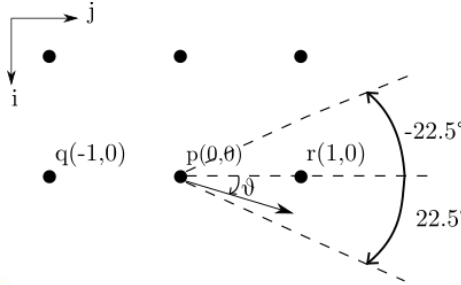


If gradient responses at r and p are smaller than q, q is an edge

Non-maximum suppression (NMS)







$$M(x,y)=\begin{cases}|\nabla S|(x,y) & \text{if } |\nabla S|(x,y)>|\Delta S|(x',y') \\ & \&\,|\Delta S|(x,y)>|\Delta S|(x'',y'') \\ 0 & \text{otherwise}\end{cases}$$

x' and **x**'' are the neighbors of **x** along normal direction to an edge

Non-maximum suppression

Edge Thinning and Localization

Before Non-Max Suppression

After Non-Max Suppression

Non-maximum suppression

By retaining only the significant local maxima, NMS plays a key role in producing clean, meaningful outputs for edge detection tasks.

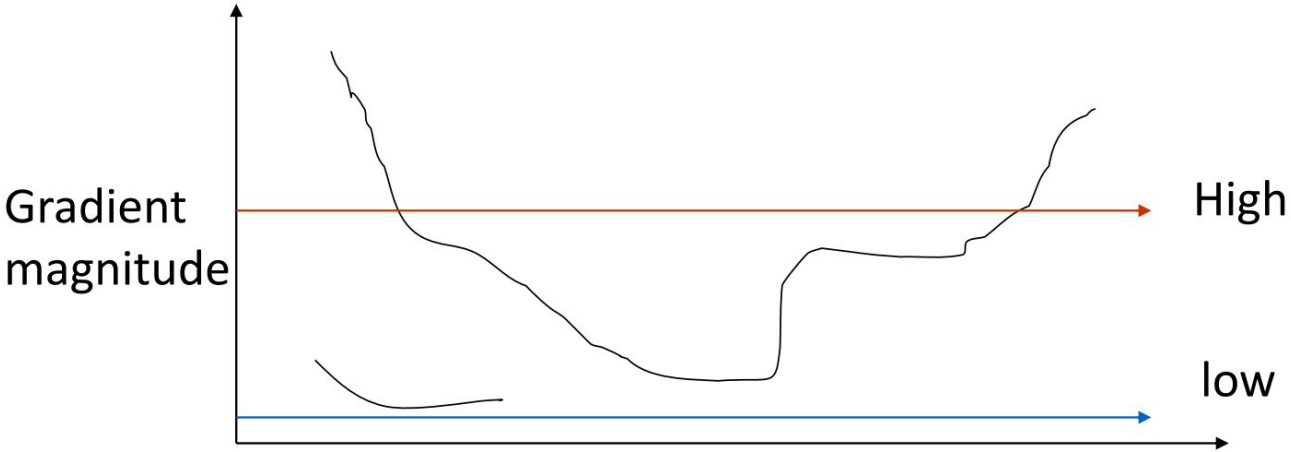
Edge Thinning and Localization

After Non-Max Suppression

Double Thresholding [L, H]

- After NMS, the remaining edge pixels provide a more accurate representation of real edges in an image.
- However, some edge pixels remain that are caused by noise and color variation.
- It is essential to filter out edge pixels with a weak gradient value and preserve edge pixels with a high gradient value.
- Achieved by selecting high and low threshold values for the gradient magnitude.
- If an edge pixel's gradient magnitude
  - is higher than the high threshold: **strong edge** pixel.
  - is smaller than the high threshold and larger than the low threshold: **weak edge** pixel.
  - is smaller than the low threshold: it will be suppressed.

Double Thresholding [L, H]



Edge tracking by Hysteresis

- The strong edge pixels are deemed to come from true edges in the image and should certainly be involved in the final edge image.

- However, there will be some debate on the weak edge pixels.
  - We want to determine whether these pixels come from a true edge or are caused by noise/color variations.

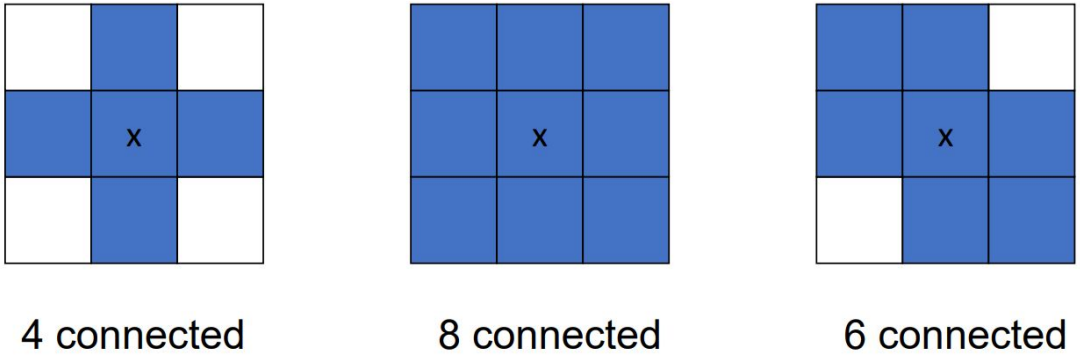
- This algorithm uses the idea that weak edge pixels from true edges will (usually) be connected to a strong edge pixel while noise responses are unconnected.

- To track the edge connection, look at the 8-connected neighborhood pixels of any edge pixel.
  - As long as there is one strong edge pixel that is involved in the blob, that weak edge point can be identified as one that should be preserved.
  - Or, in other words, all weak edge pixels in the neighborhood of a strong edge pixel are marked as strong edge pixels.

Edge tracking by Hysteresis

- Iterative Approach:
    - Identify all strong edge pixels (gradient mag $\geq T_H$) and place them in a queue.
    - For each strong edge pixel in the queue:
        - Check all its 8-connected neighbors.
        - If a neighbor is a weak edge pixel ($T_L \leq$ gradient mag $< T_H$):
            - Promote it to a strong edge pixel.
            - Add it to the queue for further processing.
    - Repeat until the queue is empty.
    - Discard all remaining weak edge pixels as noise.

Edge tracking by Hysteresis



- Threshold at low/high levels to get non/strong edge pixels
- Do connected components check, starting from strong edge pixels

Hysteresis Thresholding [L, H]

M

regular

$M \geq 25$

Hysteresis

$High = 35$

$Low = 15$

Canny Edge Detection

The original image

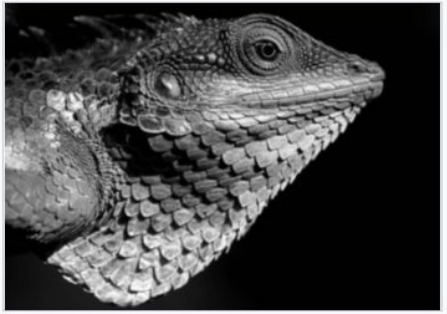
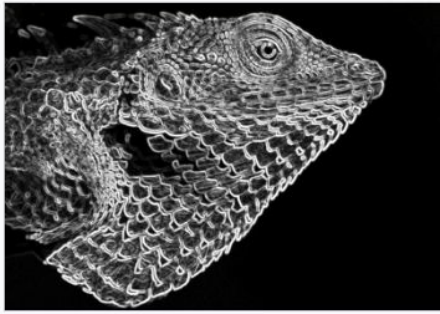
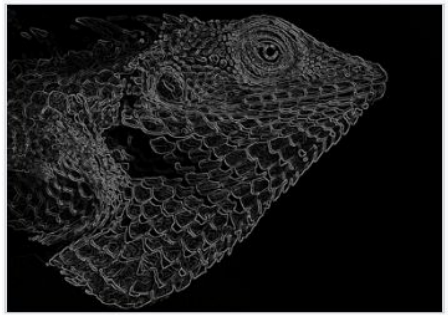


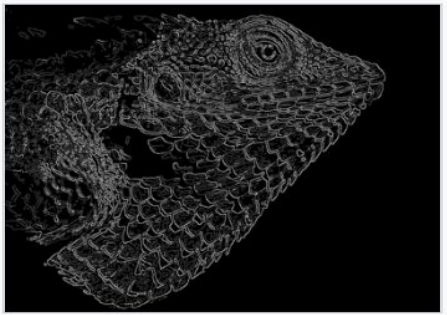
Image has been reduced to grayscale, and a 5x5 Gaussian filter with σ=1.4 has been applied.



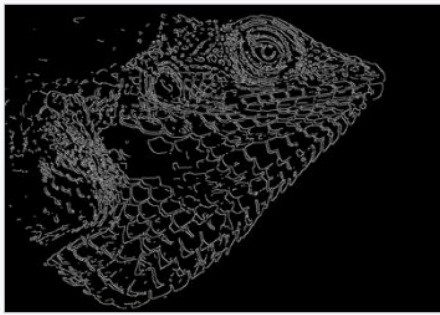
The intensity gradient of the previous image. The edges of the image have been handled by replicating.



Non-maximum suppression applied to the previous image



Double thresholding applied to the previous image. Weak pixels are those with a gradient value between 0.1 and 0.3. Strong pixels have a gradient value greater than 0.3.



Hysteresis applied to the previous image

Image Credit: Wikipedia

Final Canny Edges

Effect of Gaussian Kernel (smoothing)

original | Canny with $\sigma = 1$ | Canny with $\sigma = 2$

- The choice of depends on desired behavior
  - large **σ** detects large scale edges
  - small **σ** detects fine features