Machine Intelligence and Brain Research (MIBR)
Course No: ID-7123:

Module:

Classical MACHINE VISION

Dr. Sukhendu Das
Deptt. of Computer Science and Engg.,
Indian Institute of Technology, Madras
Chennai – 600036, India.

http://www.cse.iitm.ac.in/~sdas
Email: sdas@cse.iitm.ac.in

Contents Covered:

- Edge Detection
- Local Feature Detectors and Descriptors
- Segmentation
- Video Object Tracking

Other Interesting Items (not covered):

- Image Filtering and Enhancement;
- Stereo and Depth;
- Object detection and Recognition;
- SIFT, SURF, HOG, BOW, ….
- Scene Modeling, Augmented Reality;
- Image Compression

Concepts in

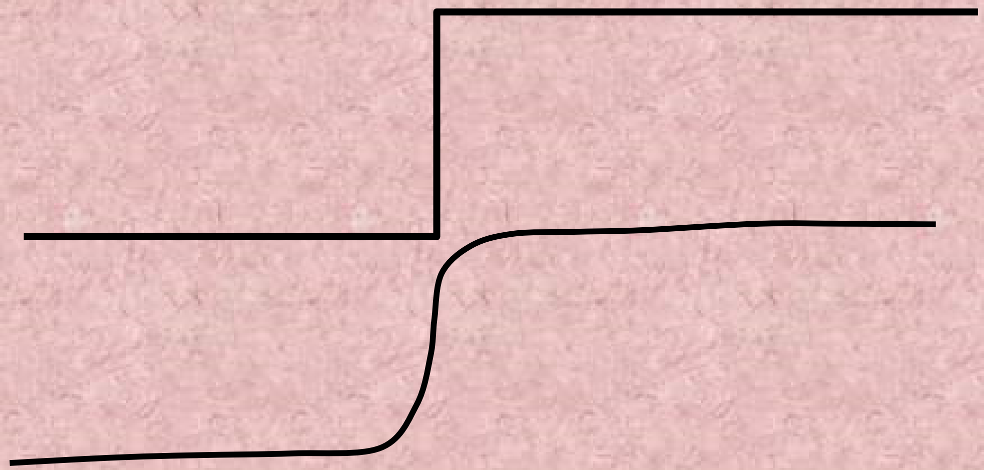
Edge Detection

Edge Detection

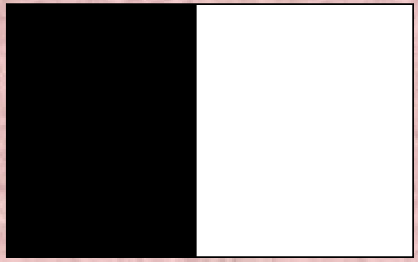
Edge is a boundary between two homogeneous regions. The gray level properties of the two regions on either side of an edge are distinct and exhibit some local uniformity or homogeneity among themselves.

An edge is typically extracted by computing the derivative of the image intensity function. This consists of two parts:

- Magnitude of the derivative: measure of the strength/contrast of the edge
- Direction of the derivative vector: edge orientation



Ideal Step edge in 1-D

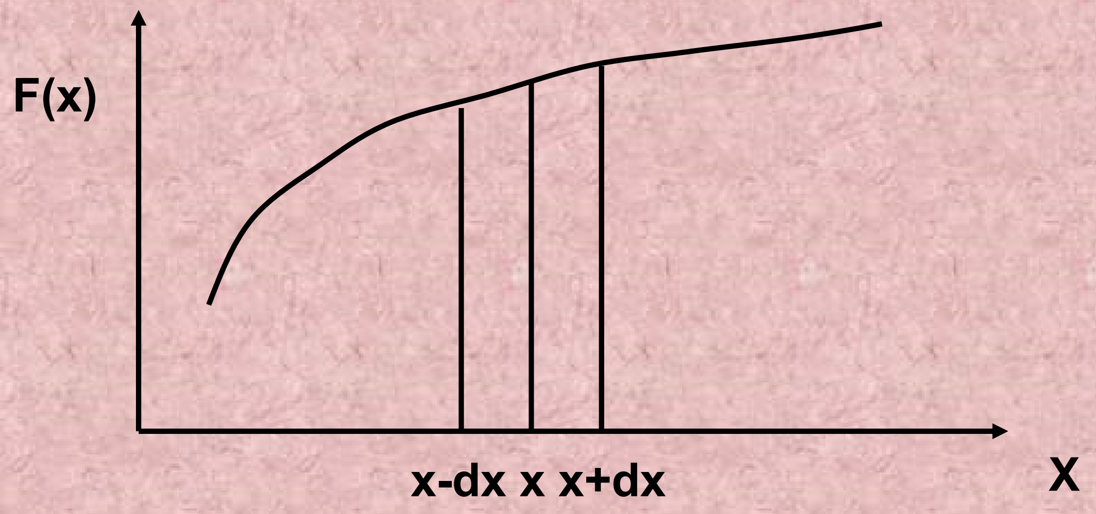


Step edge in 2-D

Computing the derivative: **Finite difference in 1-D**

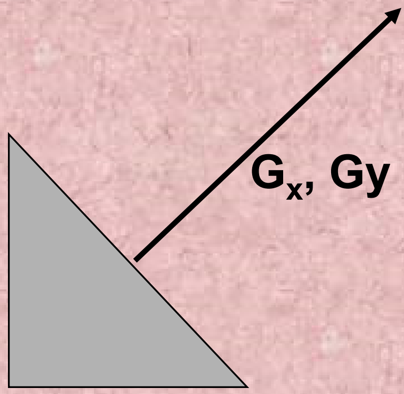
$$\frac{df}{dx} \approx \frac{f(x+dx)-f(x)}{dx} \approx \frac{f(x+dx)-f(x-dx)}{2dx}$$

$$\frac{d^2 f}{d^2 x} \approx \frac{f(x+dx)-2f(x)+f(x-dx)}{dx^2}$$



Original Image **Horizontal derivative** **Vertical derivative**

Two components of the edge values computed are:



Gradient values: $G_x = \delta f/\delta x$; $Gy = \delta f/\delta y$.

The ***magnitude*** of the edge is calculated as:

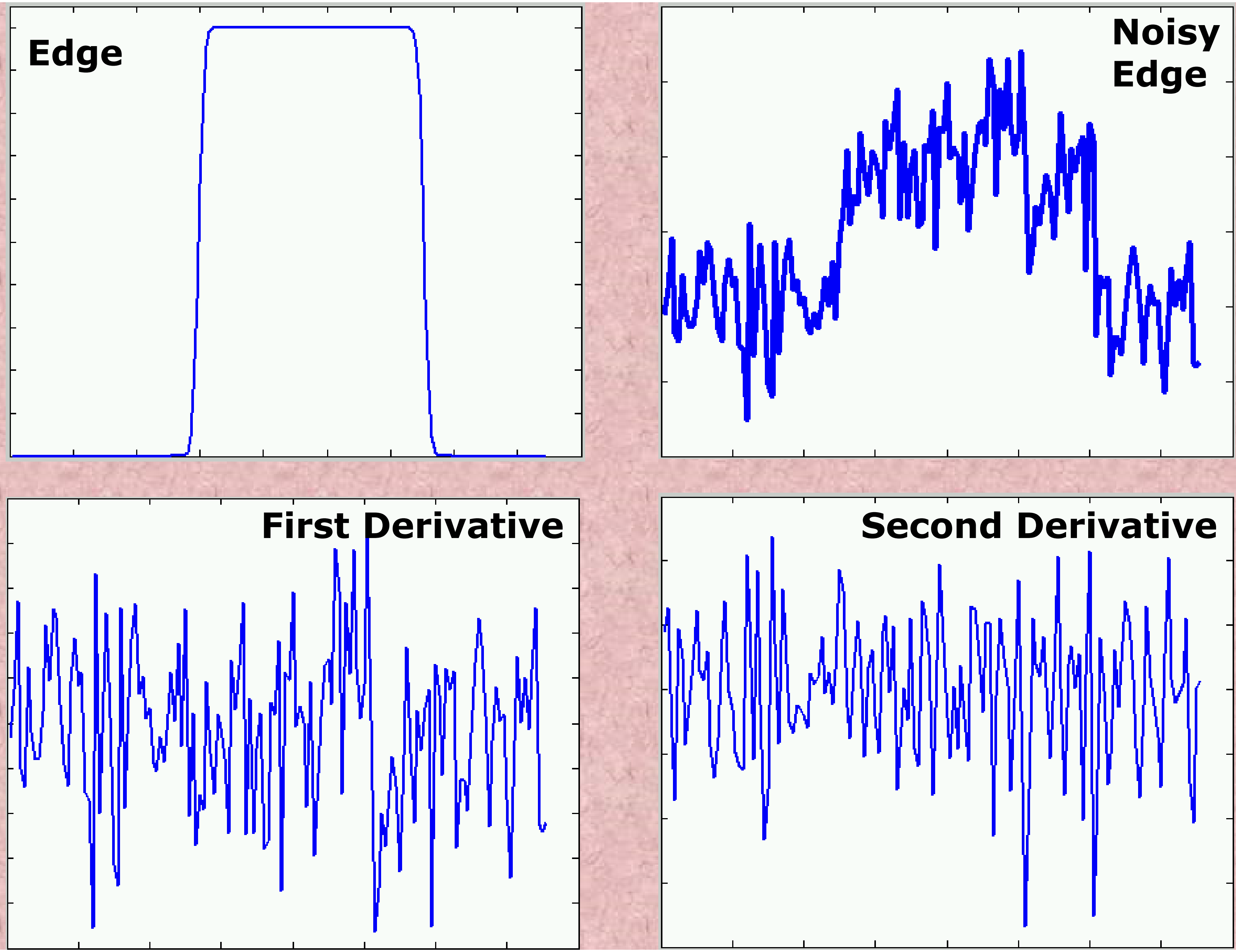
$|G| = [G_x^2 + G_y^2]^{1/2}$

and ***orientation*** as:

$\theta = \arctan(G_y/G_x)$

Most of these partial derivative operators are sensitive to noise. Use of edge operators/masks results in thick edges or boundaries, in addition to spurious edge pixels due to noise.

Laplacian mask is highly sensitive to spike noise. Use of noise smoothing became mandatory before edge detection, specifically for noisy images. But noise smoothing, typically by the use of a ***Gaussian*** function, caused a blurring or smearing of the edge information or gradient values.


Edge
Noisy Edge
First Derivative
Second Derivative

Canny in 1986 suggested an optimal operator, which uses the Gaussian smoothing and the derivative function together. He proved that the first derivative of the Gaussian function, as shown below, is a good approximation to his optimal operator.

It combines both the derivative and smoothing properties in a nice way to obtain good edges. Canny also talks of a hysteresis based thresholding strategy for marking the edges from the gradient values.

Smoothing and derivative when applied separately, were not producing good results under noisy conditions. This is because, one opposes the other. Whereas, when combined together produces the desired output.

Expression of Canny (1-D operator is):

$$c(x) = g'(x) = \left(\frac{-x}{\sqrt{2\pi}\sigma^3}\right)\exp\left(\frac{-x^2}{2\sigma^2}\right)$$

Canny's algorithm for edge detection:

Detect an edge, where simultaneously the following conditions are satisfied:

$\nabla^2 G*f = 0$ and
$\nabla G*f$ reaches a maximum.

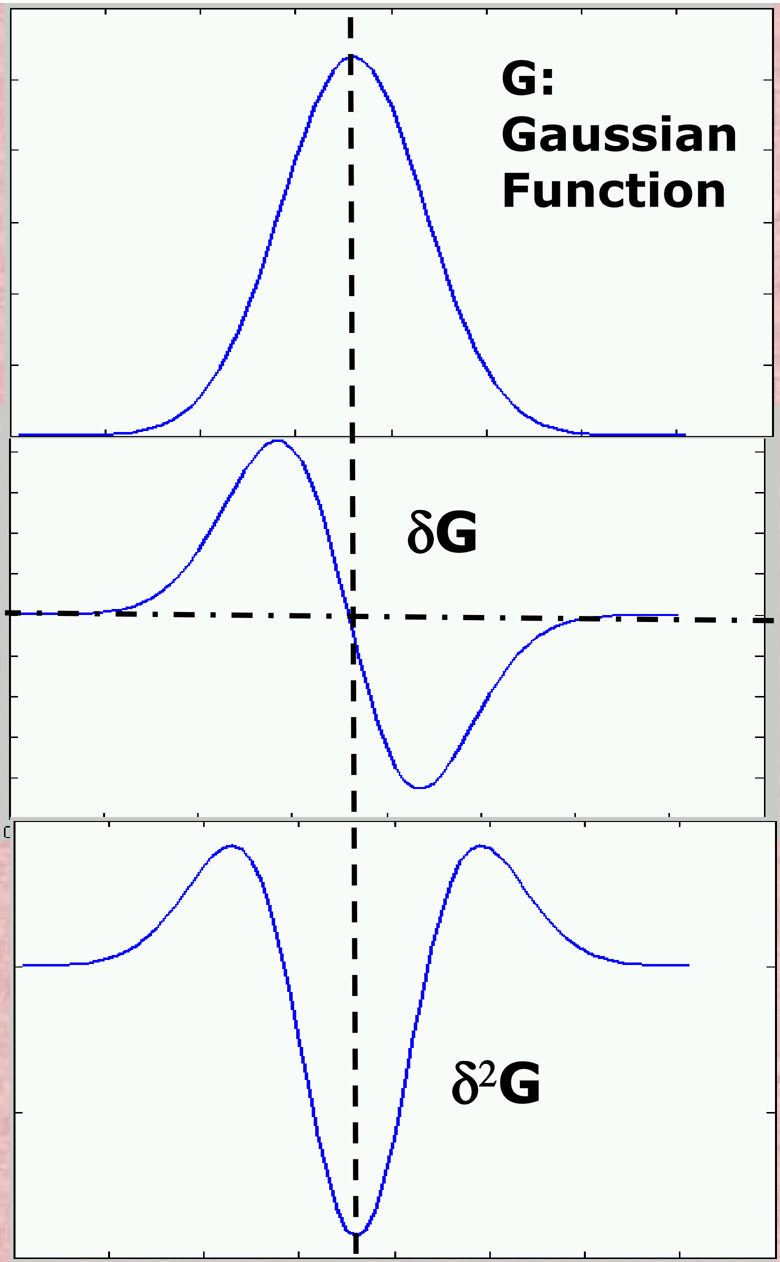
∇G is the first derivative of the Gausian defined (in 1-D) as:

$$\nabla G(x) = \frac{-x}{\sqrt{2\pi}\sigma_2^3} \exp\left(-\frac{x^2}{2\sigma_2^2}\right)$$

and

$\nabla^2 G$ in two-dimension is given by (also known as the *"Laplacian of the Gaussian" or LOG operator*):

$$\nabla^2 G(r) = \left(\frac{1}{\pi\sigma^4}\right)\left(r^2/2\,\sigma^2 - 1\right)\exp\left(\frac{-r^2}{2\sigma^2}\right)$$



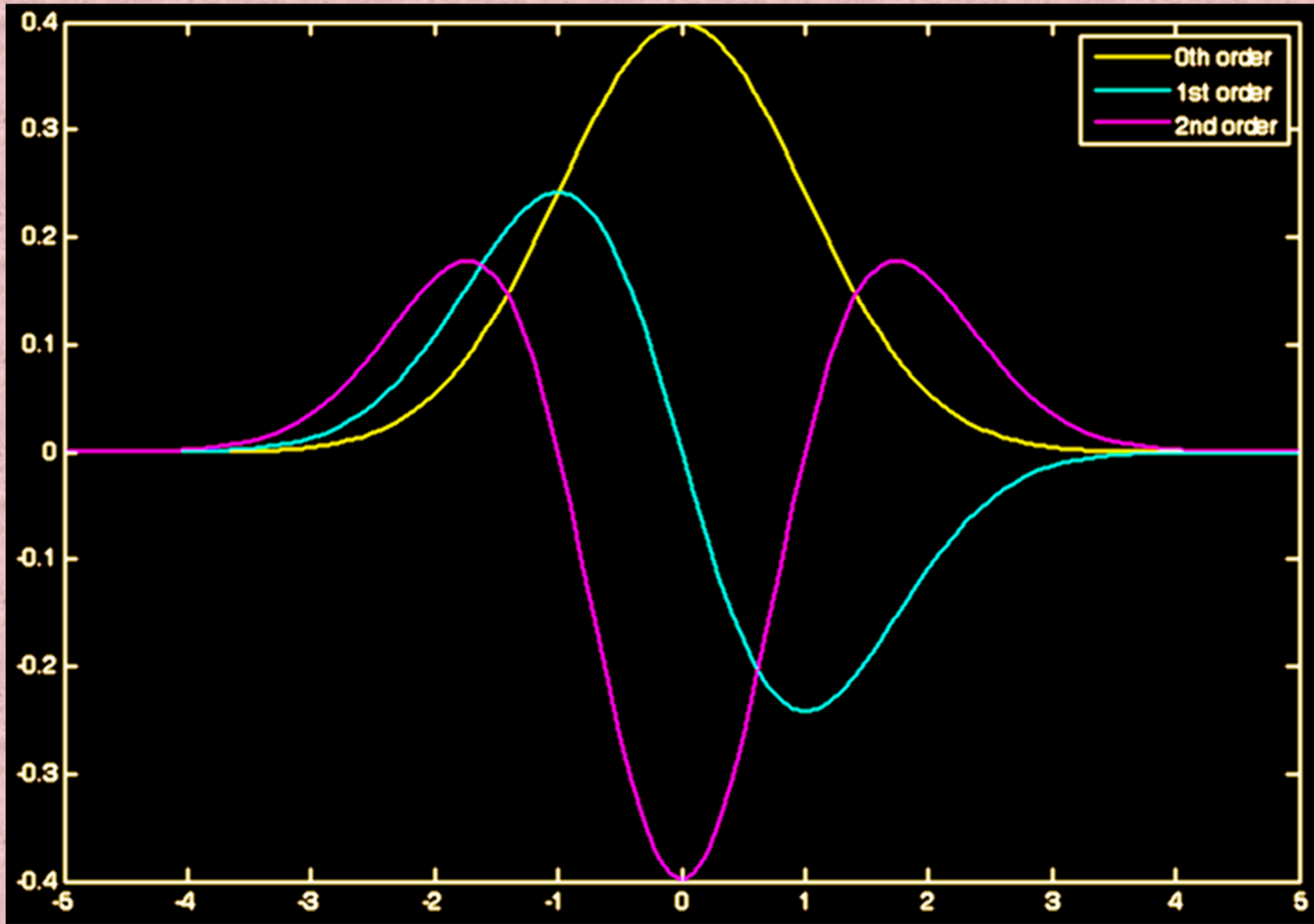
$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$\nabla G(x) =$$

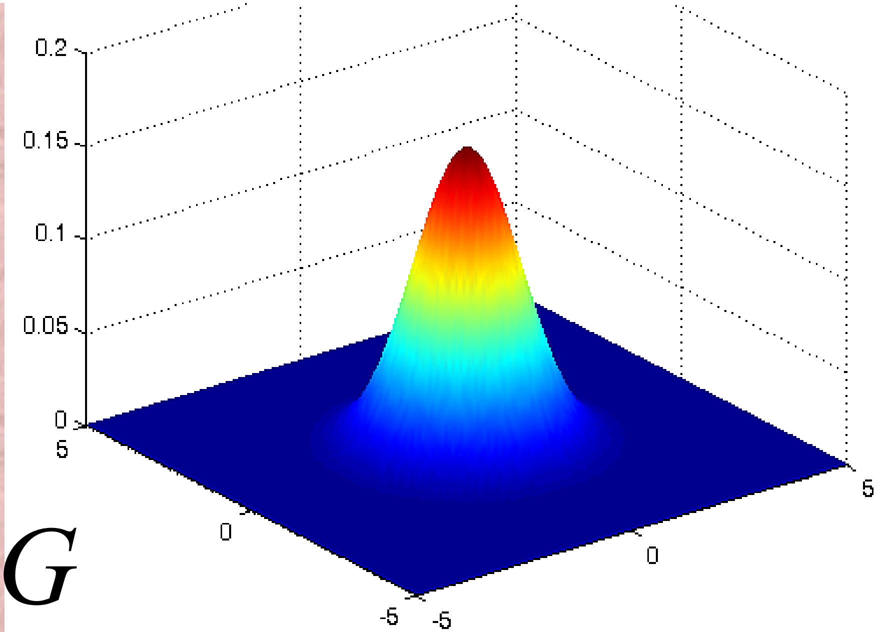
$$\frac{-x}{\sqrt{2\pi}\sigma_2^3} \exp(-\frac{x^2}{2\sigma_2^2})$$

$$\nabla^2 G(x) =$$

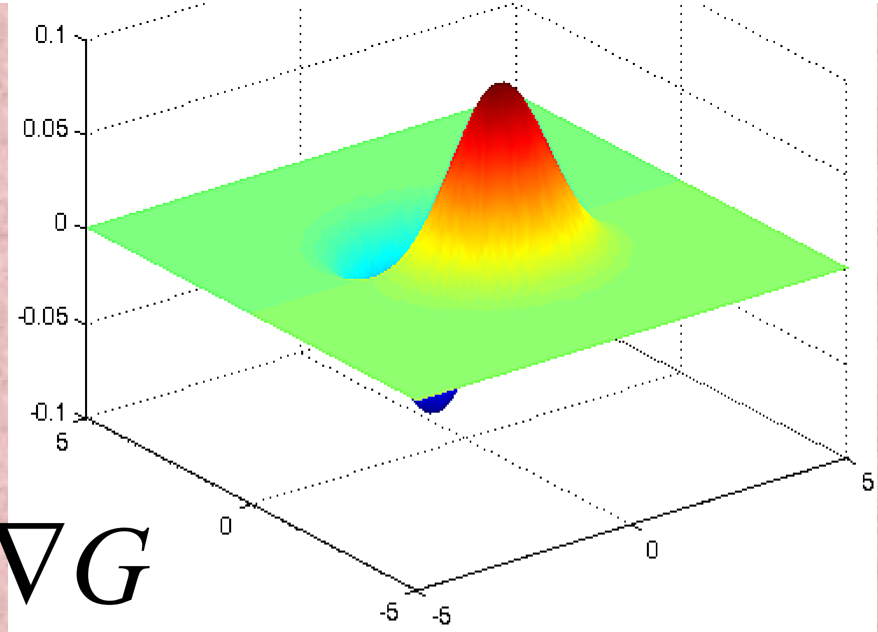
$$\frac{-[\left(x/\sigma\right)^2 - 1]}{\sqrt{2\pi}\sigma_2^3} \exp(-\frac{x^2}{2\sigma_2^2})$$



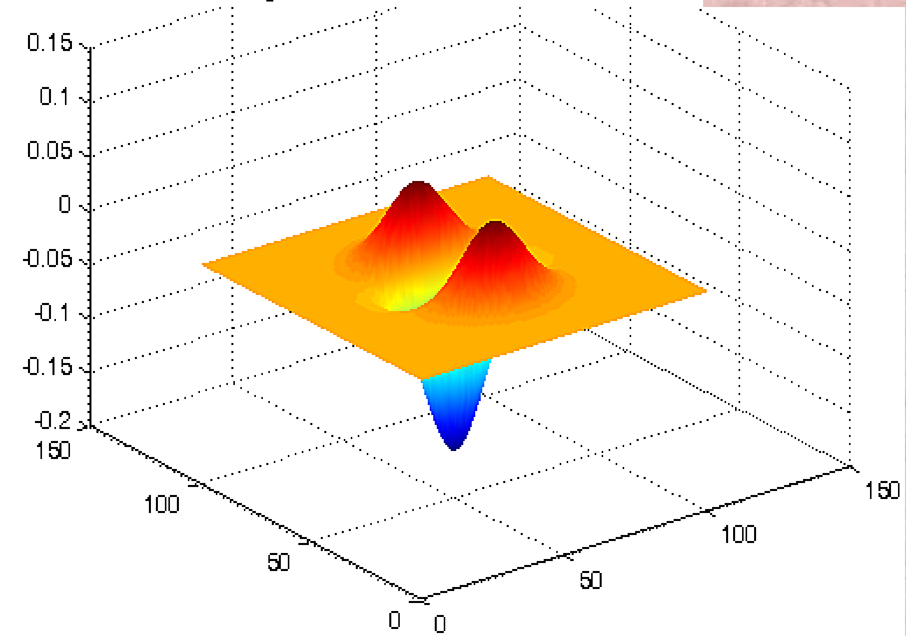
0th order
1st order
2nd order
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-5
-4
-3
-2
-1
0
1
2
3
4
5



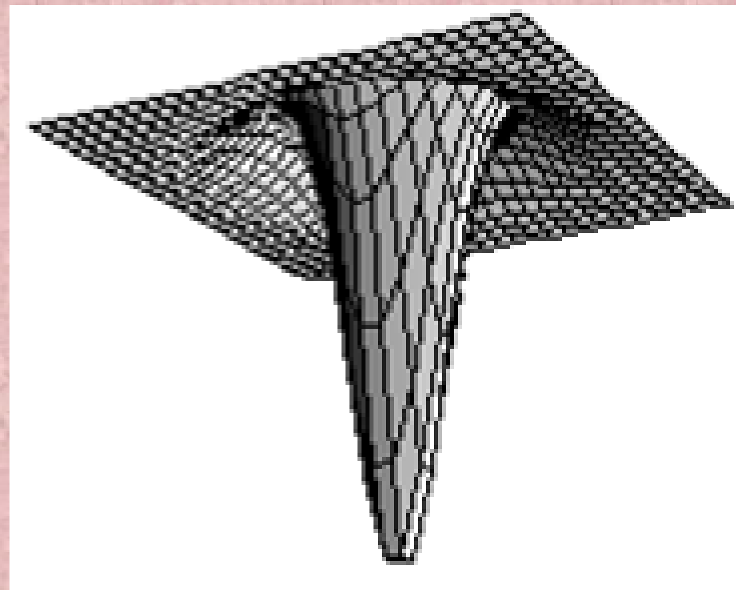
G



∇G



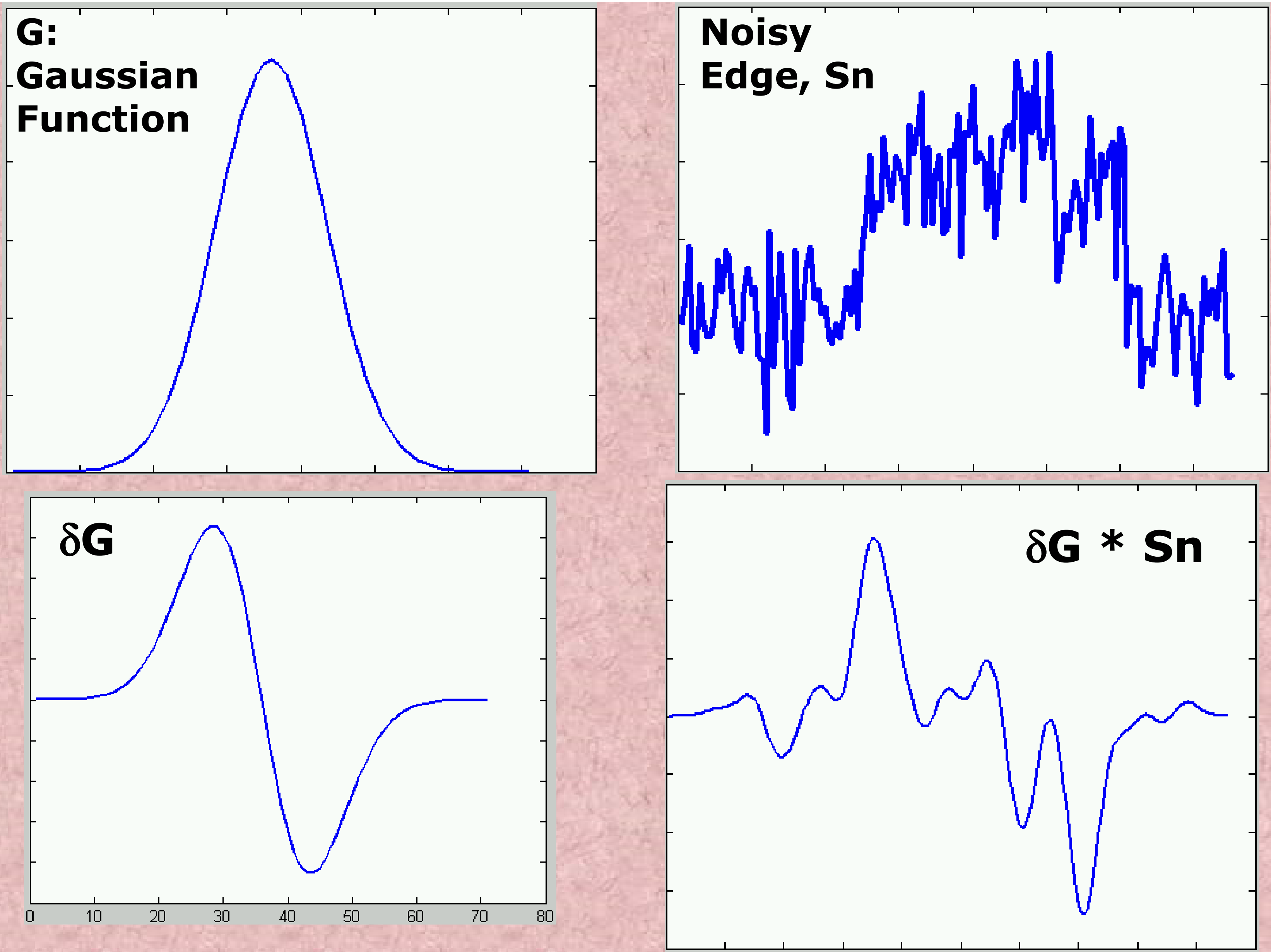
$\nabla^2 G$

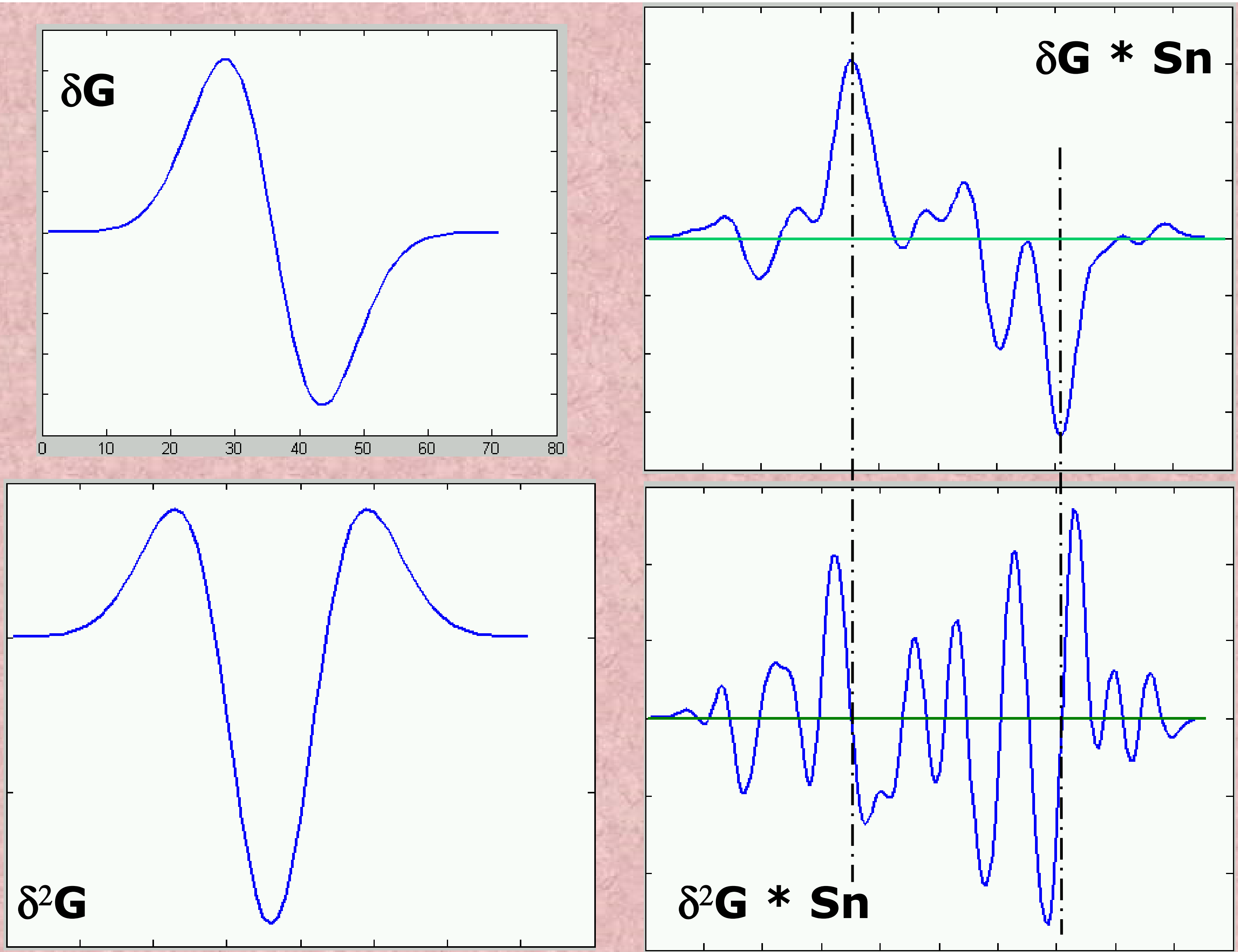


1. **Detection:**
   The probability of detecting real edge points should be maximized while the probability of falsely detecting non-edge points should be minimized. This corresponds to maximizing the signal-to-noise ratio (SNR).
   (Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible).

2. **Localization:**
   The detected edges should be as close as possible to the real edges.
   (The edge point detected from the operator should accurately localize on the center of the edge).

3. **Number of responses:**
   Minimize the number of local maxima around the true edge;
   One real edge should not result in more than one detected edge
   (a given edge in the image should only be marked once, and where possible, image noise should not create false edges).


G:
Gaussian
Function
Noisy
Edge, Sn
δG
0
10
20
30
40
50
60
70
80
δG * Sn


δG
0
10
20
30
40
50
60
70
80
δG * Sn
δ²G
δ²G * Sn

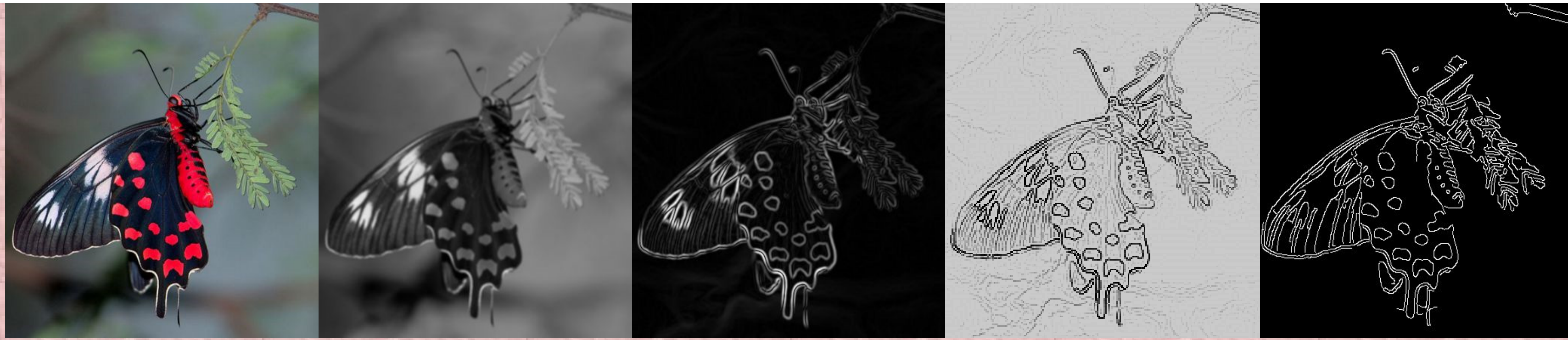
Before Non-max Suppression

After non-max suppression

Hysteresis thresholding

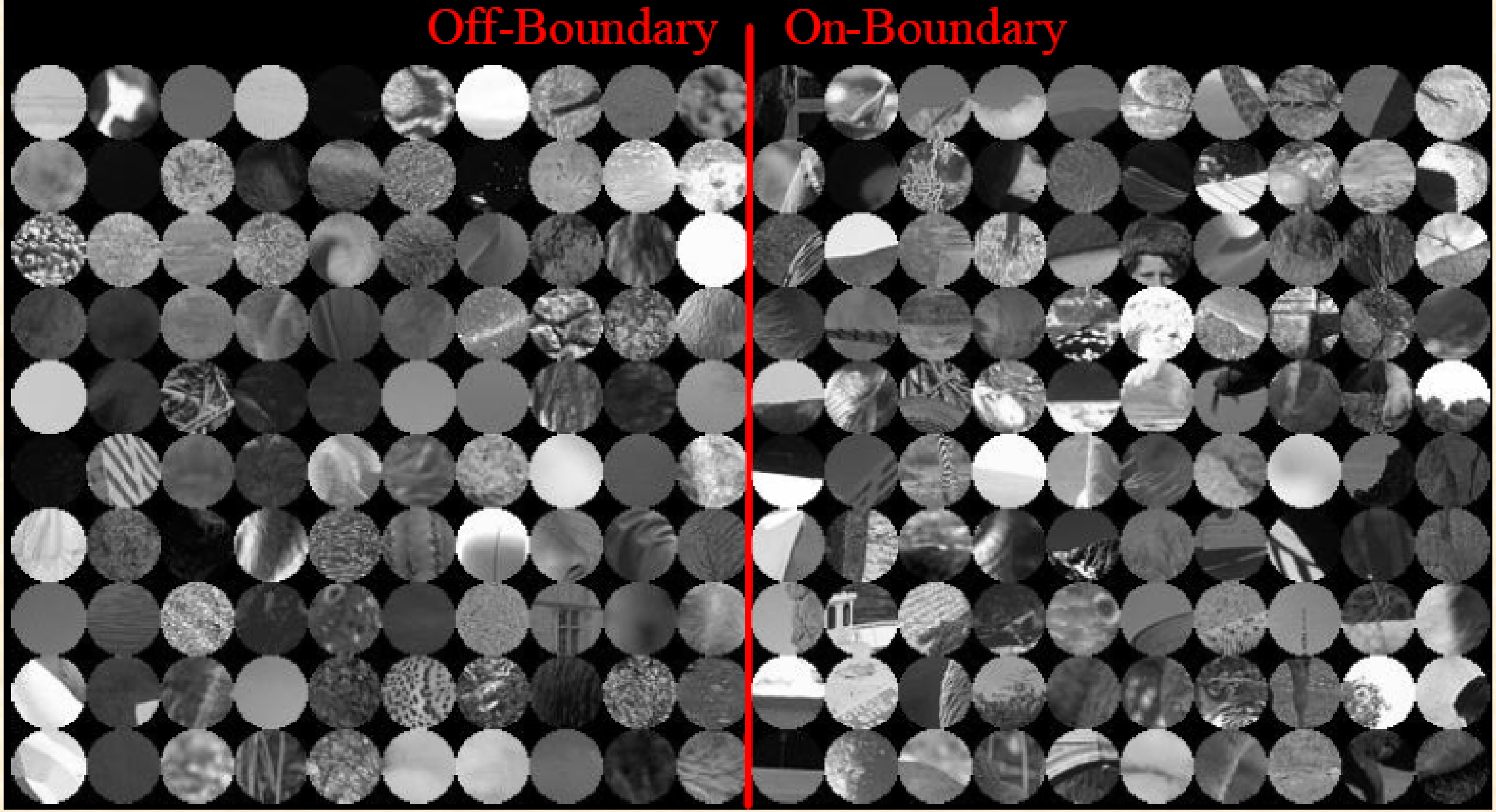
- Threshold at low/high levels to get weak/strong edge pixels
- Do connected components, starting from strong edge pixels

Final Canny Edges



Original Image, Presmoothed Image, Gradient Image, Non-maximum Suppressed Image, Final Result

How good are humans locally?



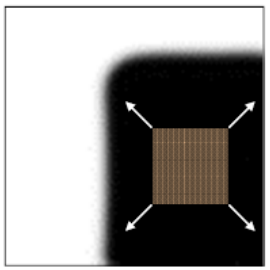
Classical MACHINE VISION

Local Feature Detectors
and
Descriptors

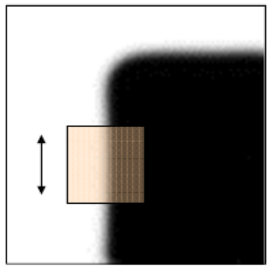
Some popular detectors

- Hessian/ Harris corner detection
- Laplacian of Gaussian (LOG) detector
- Difference of Gaussian (DOG) detector
- Hessian/ Harris Laplacian detector
- Hessian/ Harris Affine detector
- Maximally Stable Extremal Regions (MSER)
- Many others ….

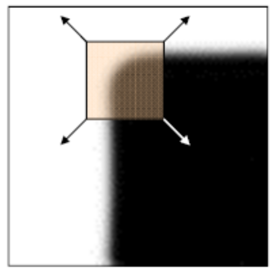
Looks for change in image gradient in two direction - CORNERS



No change in any direction



Change in one direction only



Change in both the directions

Slide credit: Fei Fei Li

Hessian Corner Detector

[Beaudet, 1978]

Searches for image locations which have strong change in gradient along both the orthogonal direction.

$$\mathrm{H}(\mathrm{x},\sigma) = \begin{bmatrix} \mathrm{I}_{\mathrm{xx}}(\mathrm{x},\sigma) & \mathrm{I}_{\mathrm{xy}}(\mathrm{x},\sigma) \\ \mathrm{I}_{\mathrm{xy}}(\mathrm{x},\sigma) & \mathrm{I}_{\mathrm{yy}}(\mathrm{x},\sigma) \end{bmatrix}$$

$$\boxed{\det(\mathrm{H}) = \mathrm{I}_{\mathrm{xx}}\mathrm{I}_{\mathrm{yy}} - \mathrm{I}_{\mathrm{xy}}^2}$$

- **Perform a non-maximum suppression using a 3*3 window.**
- **Consider points having higher value than its 8 neighbors.**

Select points where $\det(\mathrm{H}) > \theta$

Harris Corner

[Forstner and Gulch, 1987]

- Search for local neighborhoods where the image content has two main directions (eigenvectors).
- Consider 2nd moment autocorrelation matrix

$$C(x,\sigma,\tilde{\sigma}) = G(x,\tilde{\sigma}) * \begin{bmatrix} I_x^2(x,\sigma) & I_xI_y(x,\sigma) \\ I_xI_y(x,\sigma) & I_y^2(x,\sigma) \end{bmatrix} \qquad \tilde{\sigma} \approx 2\sigma$$

Gaussian sums over all the pixels in circular local neighborhood using weights accordingly.

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_xI_y \\ \sum I_xI_y & \sum I_y^2 \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

Symmetric Matrix

If λ_1 or λ_2 is about 0, the point is not a corner.

Harris Corner: Different approach

Instead of explicitly computing the eigen values, the following equivalence are used

$$\det(\mathbf{C}) = \lambda_1\lambda_2$$

$$\mathrm{trace}(\mathbf{C}) = \lambda_1 + \lambda_2$$

$$\text{If, } r = \frac{\lambda_1}{\lambda_2} (\geq 1), \quad \frac{\mathrm{trace}^2(\mathbf{C})}{\det(\mathbf{C})} =$$

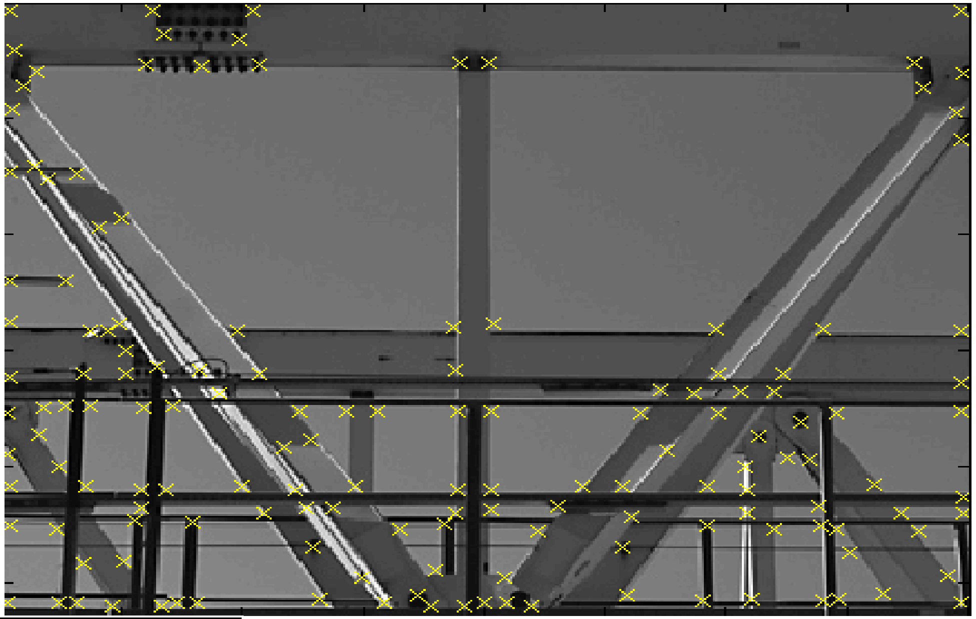
$$\Rightarrow H_c =$$

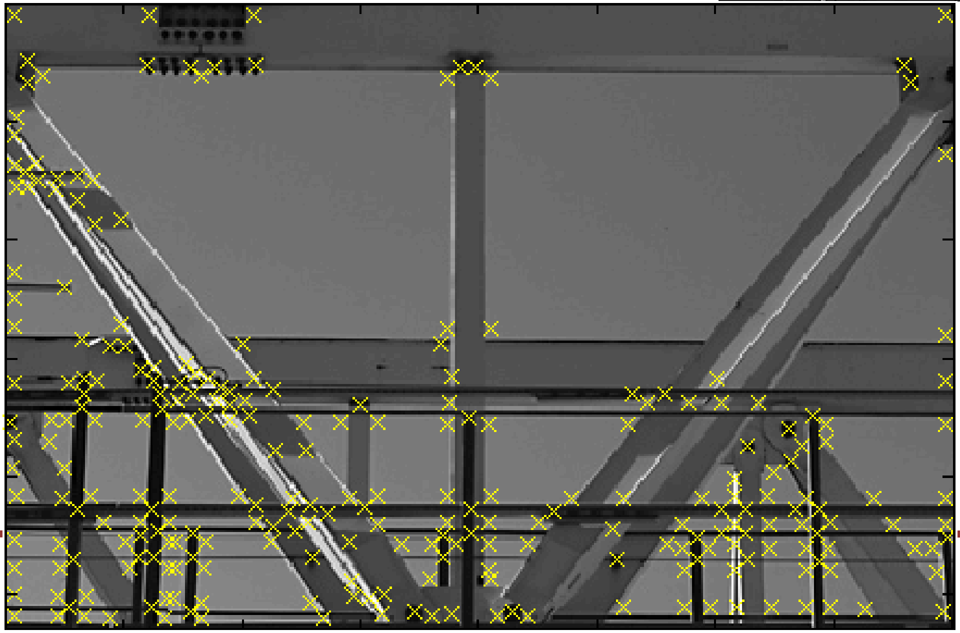
$$\det(C) - \alpha.trace^2(C) > threshold$$

Consider these points

α in the range 0.04 – 0.25, experimentally verified

Harris Corner





Hessian Detector

Segmentation of Images

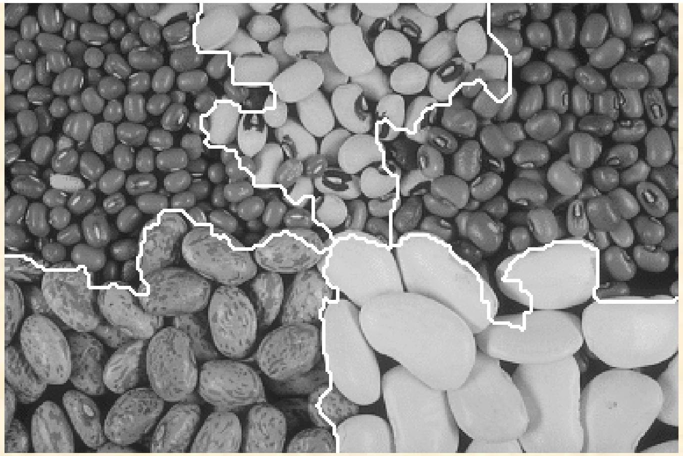
Segmentation **is a process to group pixels together into regions of similarity.**

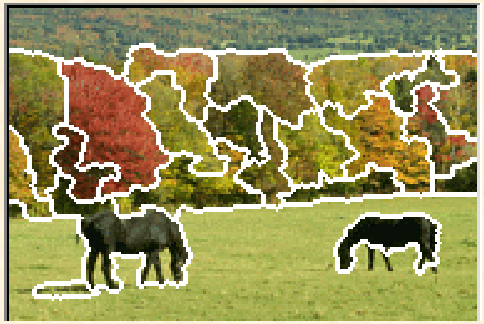
Region-based segmentation methods attempt to partition or group regions according to common image properties. These image properties consist of :

- **Intensity values from original images, or computed values based on an image operator**
- **Textures or patterns that are unique to each type of region**
- **Spectral profiles that provide multidimensional image data**

Elaborate systems may use a combination of these properties to segment images, while simpler systems may be restricted to a minimal set on properties depending of the type of data available.

Lets observe some examples from literature:




<u>Segmentation and Graph Cut</u>

- A graph can be partitioned into two disjoint sets by simply removing the edges connecting the two parts
- The degree of dissimilarity between these two pieces can be computed as total weight of the edges that have been removed
- More formally, it is called the **<u>'cut'</u>**

Weight Function for Brightness Images

- Weight measure (reflects likelihood of two pixels belonging to the same object)

$$w_{ij} = \exp - \frac{(I(i) - I(j))^2}{\sigma_I^2} * \begin{cases} \exp - \frac{\|X(i)-X(j)\|_2^2}{\sigma_X^2} & \text{if } \|X(i) - X(j)\|_2 < R \\ 0 & \text{otherwise} \end{cases}$$

For brightness images, I(*i*) represents normalized intensity level of node *I* and X(*i*) represents spatial location of node *i*.

σ_I and σ_X are parameters set to 10-20 percent of the range of their related values.

R is a parameter that controls the sparsity of the resulting graph by setting edge weights between distant pixels to 0.

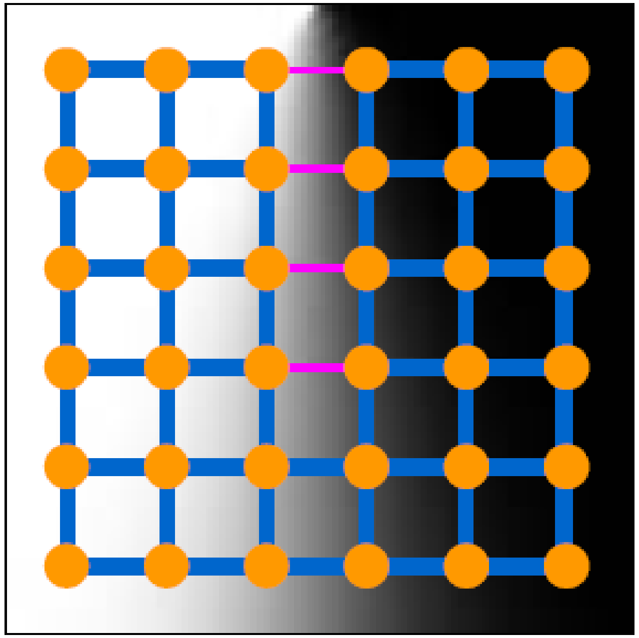
The Pixel Graph

Couplings $\{w_{ij}\}$

Reflect intensity similarity

Low contrast – strong coupling

High contrast – weak coupling



V: graph nodes: ← → Image = { pixels }

E: edges connection nodes: ← → Pixel similarity

Segmentation and Graph Cut

1) Given a source (**s**) and a sink node (**t**)

2) Define Capacity on each edge, C_ij = W_ij

3) Find the maximum flow from s->t, satisfying the capacity constraints

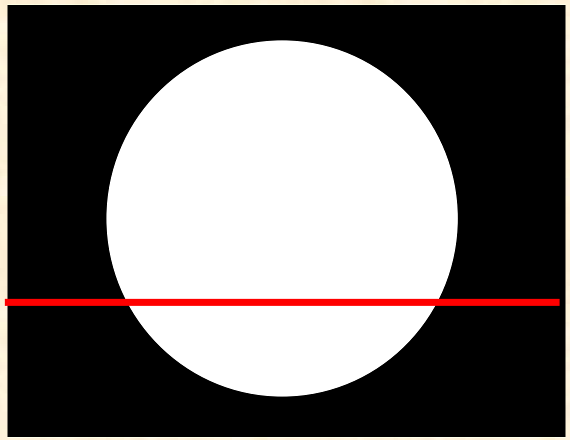
Min. Cut = Max. Flow

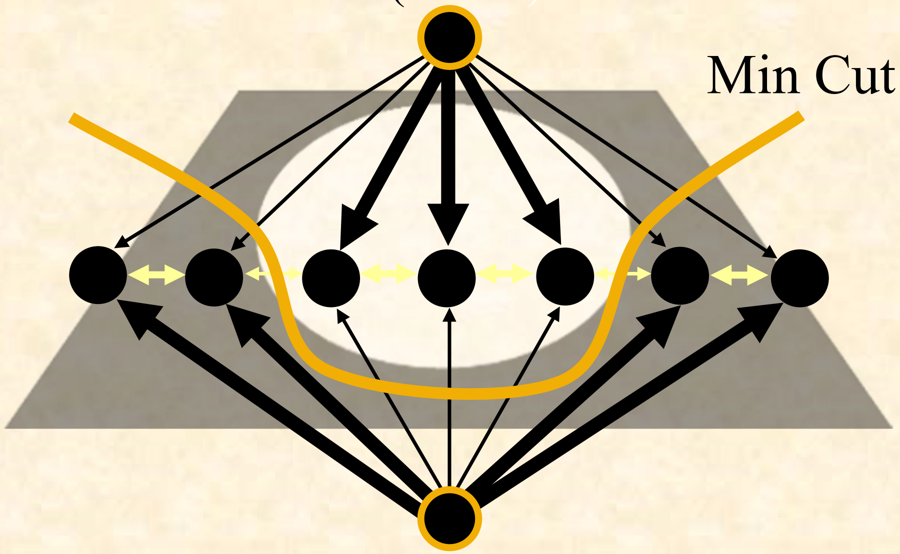
Max-flow/Min-cut theorem:

For any network having a single origin mode and destination node, the maximum flow from origin to destination equals the minimum cut value for all cuts in the network.

Graph cuts

Image





Cut: separating source and sink; Energy: collection of edges

Min Cut: Global minimal energy in polynomial time

Need to partition the nodes of a graph, V, into two sets A and B.

Let x be an N = |V| dimensional indicator vector, x_i = 1, if node i is in A, else -1.

Let , $$d(i) = \sum_j w(i, j)$$

be the total connection from node i to all other nodes.

Let D be an NxN diagonal matrix with d on its diagonal;

W be an NxN symmetrical **matrix with W(i, j) =** w**(i, j);**

W is also an adjacency matrix.

Spectral CUT - Partition (grouping) algorithm steps:

1. Given an image or image sequence, set up a weighted graph G = (V, E), and set the weight on the edge connecting two nodes to be a measure of the similarity between the two nodes.

2. Solve (D – W).x = λDx for eigenvectors with the smallest eigenvalues.

3. Use the eigenvector with the second smallest eigenvalue to bipartition the graph.

4. Decide if the current partition should be subdivided and recursively

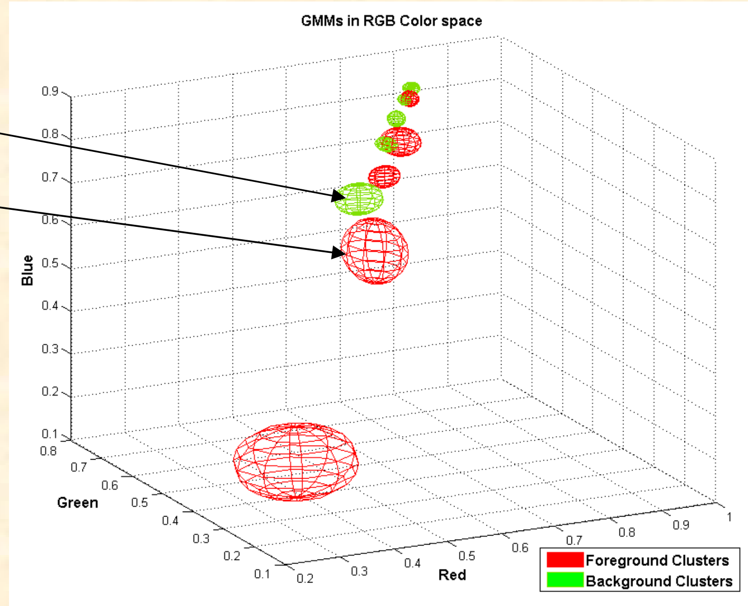
Rayliegh Quotient: $\min_x NCut(x) = \min_y \frac{y^T (D-W) y}{y^T D y}$

Object Extraction From an Image

Alpha-Matte based Foreground Extraction:

Unknown foreground

Known Background

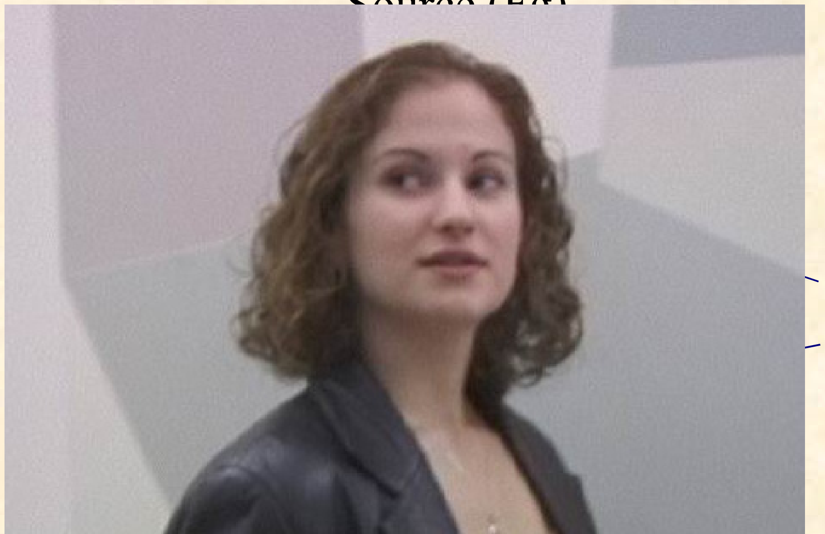


Create GMMs with K components for foreground and background separately

Learn GMMs and perform GraphCut to find tentative classification of foreground and background

Object Extraction From an Image

Source (Fg)



Sink (Bkg)

Pixel type (m)	BackGR T-link	Fore –GR T-link
Foreground	0	constant X
Background	constant X	0
Unknown	D_{Fore}	D_{Back}

$$D(m) = -\log \sum_{i=1}^{K} \left[\pi_i \frac{1}{\sqrt{\det \Sigma_i}} \exp\left(\frac{1}{2} [z_m - \mu_i]^T \Sigma_i^{-1} [z_m - \mu_i] \right) \right]$$

$$N(m,n) = \frac{\gamma}{dist(m,n)} e^{-\beta \| z_m - z_n \|^2}$$

Learn GMMs with newly classified set, and repeat the process until classification converges

GrabCut segmentation

1. Define graph

$$E(L) = \sum_{p} D_p(f_p) + \sum_{p,q \in N} V(f_p, f_q)$$

   – usually 4-connected or 8-connected
2. Define unary potentials (data/region term; t-links)
   – Color histogram or mixture of Gaussians for background and foreground

$$unary\_potential(x) = -\log\left(\frac{P(c(x);\theta_{foreground})}{P(c(x);\theta_{background})}\right)$$

3. Define pairwise potentials (smoothness / boundary term; interaction/n-links)

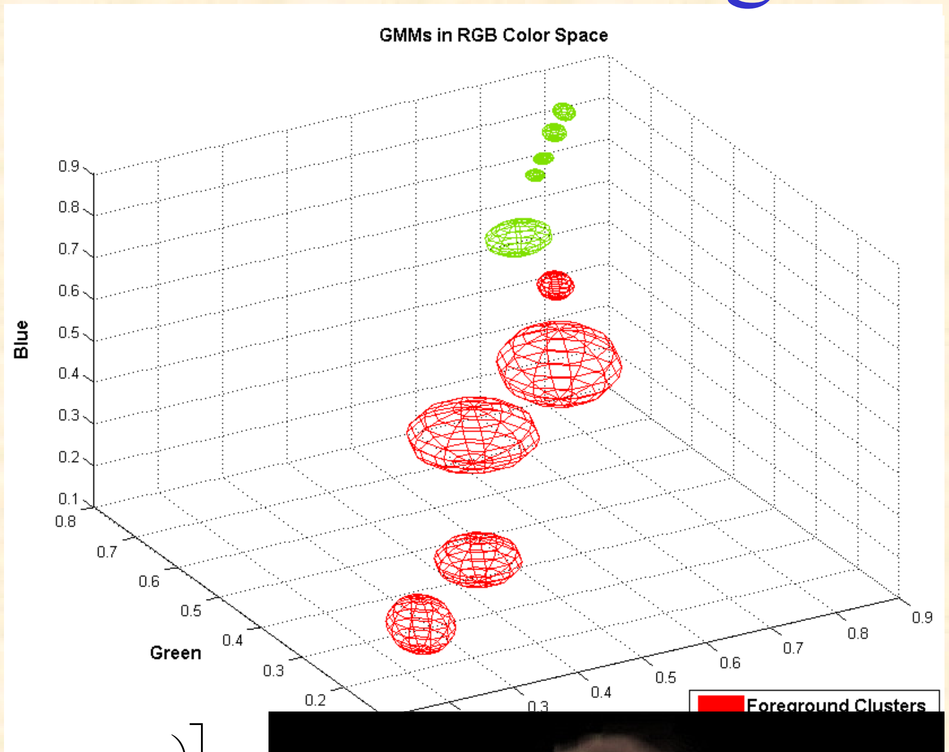
$$edge\_potential(x,y) = k_1 + k_2 \exp\left\{\frac{-\left\|c(x)-c(y)\right\|^2}{2\sigma^2}\right\}$$

4. Apply graph cuts
5. Terminate iteration when potential ceases to decrease significantly
6. Else return to 2, using current labels to compute foreground, background models

Object Extraction From an Image

Initial State

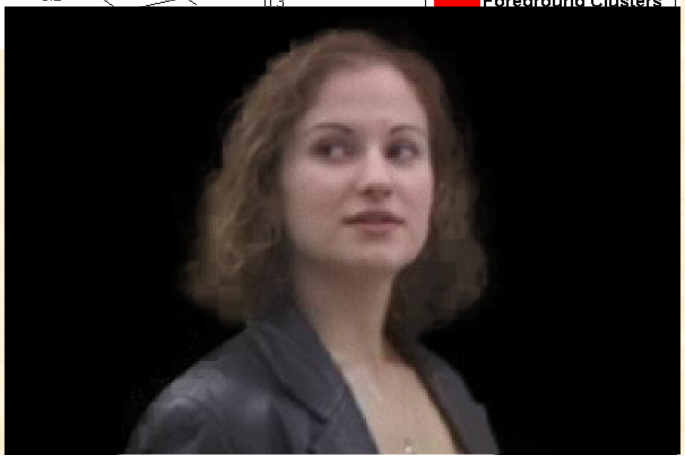
Final State



$$P(m) = \log \sum_{i=1}^{K} \left[w_i \frac{1}{\sqrt{\det \Sigma_i}} \times \exp\left(\frac{1}{2} [I_m - \mu_i]^T \Sigma_i^{-1} [I_m - \mu_i] \right) \right]$$

$$\alpha_m = \begin{cases} 1 \;\; if \;\; (P_{fore}(m) - P_{back}(m)) > \tau \\ 0 \;\; if \;\; (P_{back}(m) - P_{fore}(m)) > \tau \\ unknown \;\; if \;\; \left| P_{fore}(m) - P_{back}(m) \right| < \tau \end{cases}$$

$$\min \;\; J(\alpha) = \alpha^T L \alpha;$$



$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i$$

$$\alpha_i \approx a I_i + b, \quad \forall i \in w,$$

where $a =$ and w is a small image window.

goal in this paper will be to find α, a and b minimizing the cost function

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} (\alpha_i - a_j I_i - b_j)^2 + \varepsilon a_j^2 \right), \qquad (3)$$

where w_j is a small window around pixel j.

A Closed Form Solution to Natural Image Matting

Anat Levin, Dani Lischinski, Yair Weiss; CVPR-2006.

Object Extraction From an Image

Motion Detection and Tracking

Indian Institute of Technology, Madras

Visualization and Perception Lab

Definition of Motion Detection

- Action of sensing physical movement in a given area

- Motion can be detected by measuring change in speed or vector of an object

Background Subtraction

- **Motivation:** Simple difference (frame differencing) of two images shows moving objects
- Uses a reference background image for comparison purposes
- Current image (containing target object) is compared to reference image pixel by pixel
- Places where there are differences are detected and classified as moving objects

Overview of Various BGS Algorithms

BGS Algorithm	Reference Paper	Salient Features
Adaptive Median Filtering (AMF) (Running Average)	*N. McFarlane and C. Schofield, "Segmentation and Tracking of Piglets in Images", Machine Vision and Applications, Vol. 8, No. 3. (1 May 1995*), pp. 187-193	• Background pixel is modeled as weighted average where recent frames have higher weight • Parametric thus less memory intensive
Running Gaussian Average	"Pfinder: real-time tracking of the human body" by C. Wren et al *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on* , vol., no., pp.51-56, 14-16 Oct 1996	• Pfinder adopts a Maximum A Posteriori Probability (MAP) approach. • It first models the person, then the scene and then does analysis
Mixture of Gaussians (MoG) (Stauffer and Grimson method)	Stauffer, C.; Grimson, W.E.L. , "Learning patterns of activity using real-time tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.22, no.8, pp.747-757, Aug 2000	• Each pixel is a mixture of Gaussians. • Gaussians modify and adapt with each new incoming frame

Overview of Various BGS Algorithms (contd..)

BGS Algorithm	Reference Paper	Salient Features
Zivkovic AGMM (adaptive Gaussian mixture models)	Zivkovic, Z.; “Improved adaptive Gaussian mixture model for background subtraction”, Pattern Recognition, 2004, Proceedings of the 17th International Conference on ICPR 2004, vol.2, no., pp. 28-31 Vol.2, 23-26 Aug. 2004	• Uses Gaussian mixture probability density • The Gaussian mixture parameters and components of each pixel is updated online
Eigenbackgrounds	Oliver, N.M.; Rosario, B.; Pentland, A.P.; "A Bayesian computer vision system for modeling human interactions", IEEE Transactions on Pattern Analysis and Machine Intelligence , vol.22, no.8, pp.831-843, Aug 2000	• PCA by way of eigenvector decomposiion is a way to reduce the dimensionality of a space • PCA can be applied to a sequence of n frames to compute the eigenbackgrounds • Faster than MoG approach
Prati Mediod (mediod filtering)	Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A.; "Detecting moving objects, ghosts, and shadows in video streams," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.25, no.10, pp. 1337- 1342, Oct. 2003	• Pixels of moving objects, shadows etc., are processed differently • Uses Median function

Basic BGS Algorithms

- Background as the **average** or the **median** (Velastin, 2000; Cucchiara, 2003) of the previous *n* frames:
  - rather fast, but very memory consuming: the memory requirement is *n* * size(frame)
- Background as the Approximate Median Filtering (AMF) (**running average**)

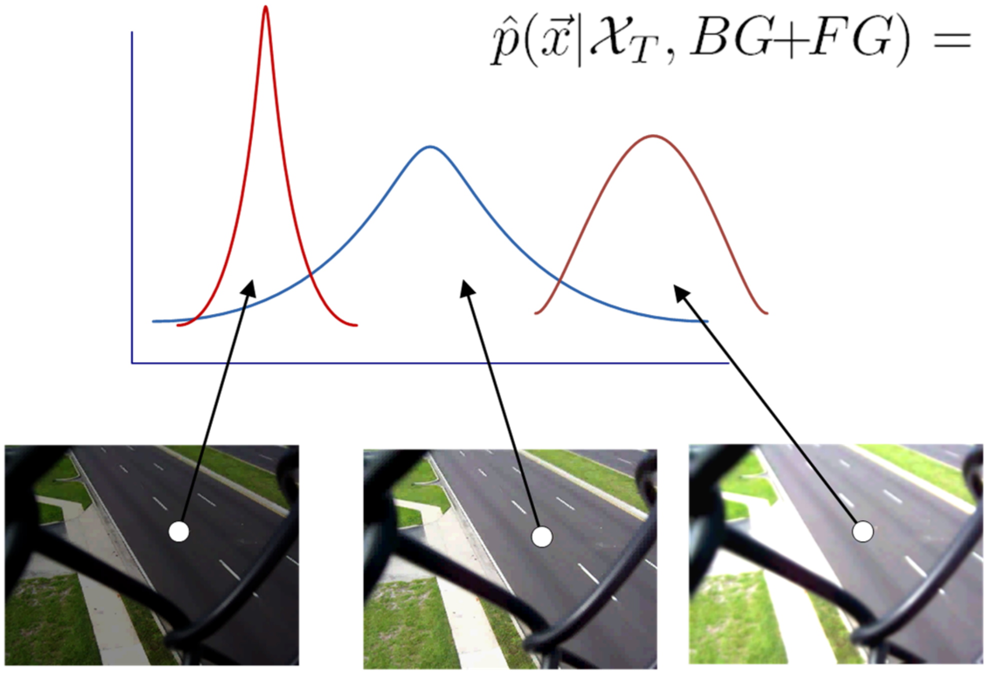
$$B_{i+1} = \alpha * I_i + (1 - \alpha) * B_i$$

  - α, the learning rate, is typically 0.05
  - no more memory requirements

Gaussian Mixture Models

- Each pixel modeled with a mixture of Gaussians
- Flexible to handle variations in the background

$$\hat{p}(\vec{x}|\mathcal{X}_T, BG{+}FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}; \widehat{\vec{\mu}}_m, \widehat{\sigma}_m^2 I)$$



The GMM Model

- Choose a reasonable time period T and at time t we have
  $\mathcal{X}_T = \{x^{(t)}, ..., x^{(t-T)}\}$
- For each new sample update the training data set $\mathcal{X}_T$
- Re-estimate $\hat{p}(\vec{x}|\mathcal{X}_T, BG)$
- Full scene model (BG + FG)

$$\hat{p}(\vec{x}|\mathcal{X}_T, BG{+}FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}; \widehat{\vec{\mu}}_m, \widehat{\sigma}_m^2 I)$$

GMM with M Gaussians where

- $\widehat{\vec{\mu}}_1, ..., \widehat{\vec{\mu}}_M$ - estimates of the means
- $\widehat{\sigma}_1, ..., \widehat{\sigma}_M$ - estimates of the variances
- $\hat{\pi}_m$ - mixing weights non-negative and add up to one.

Results - Simple frame differencing

INPUT Video

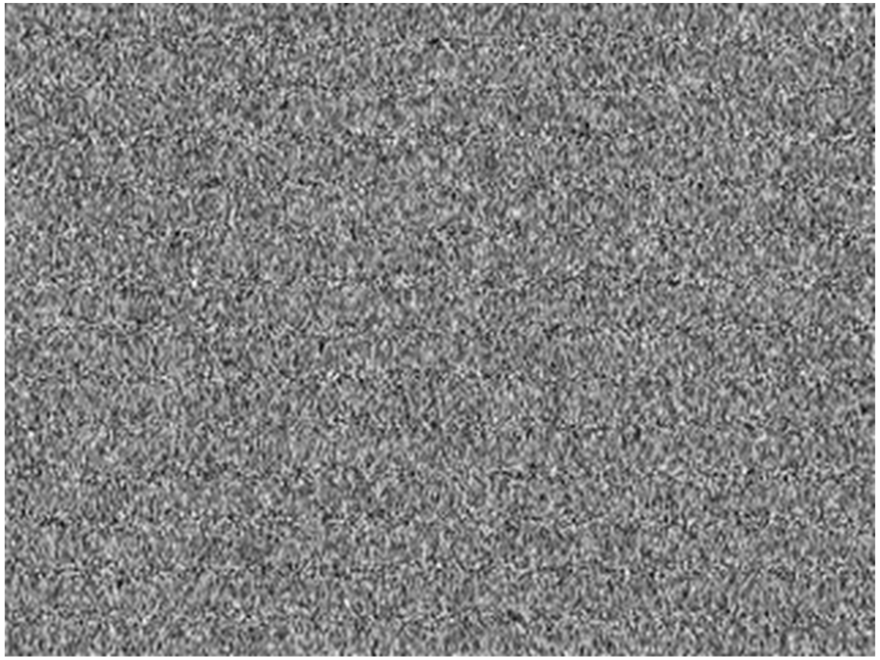
Foreground Mask

Background
Model

Results - Approximate Median Filtering

Foreground
Mask

Results - Mixture of Gaussians (MoG)

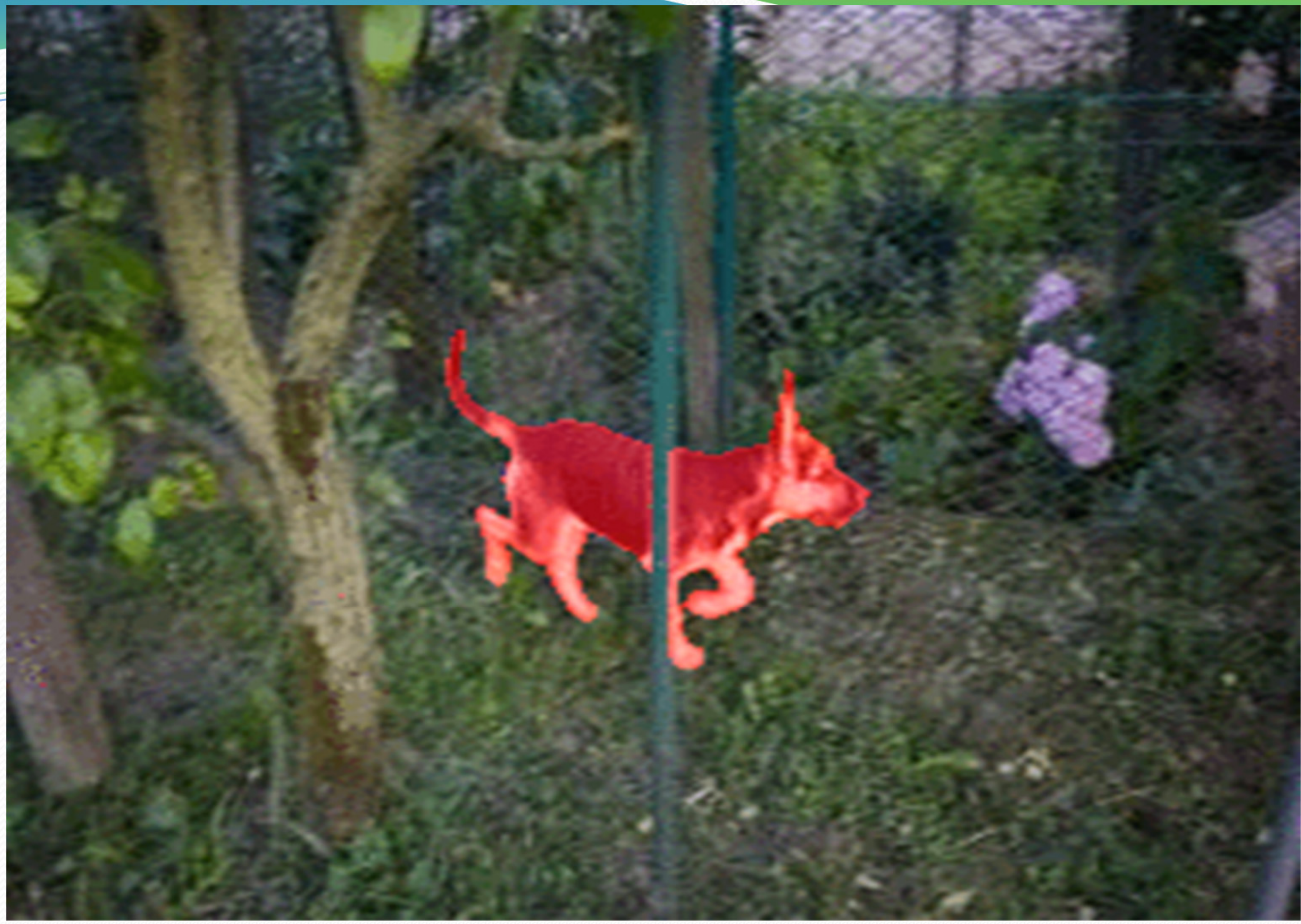


Background Model

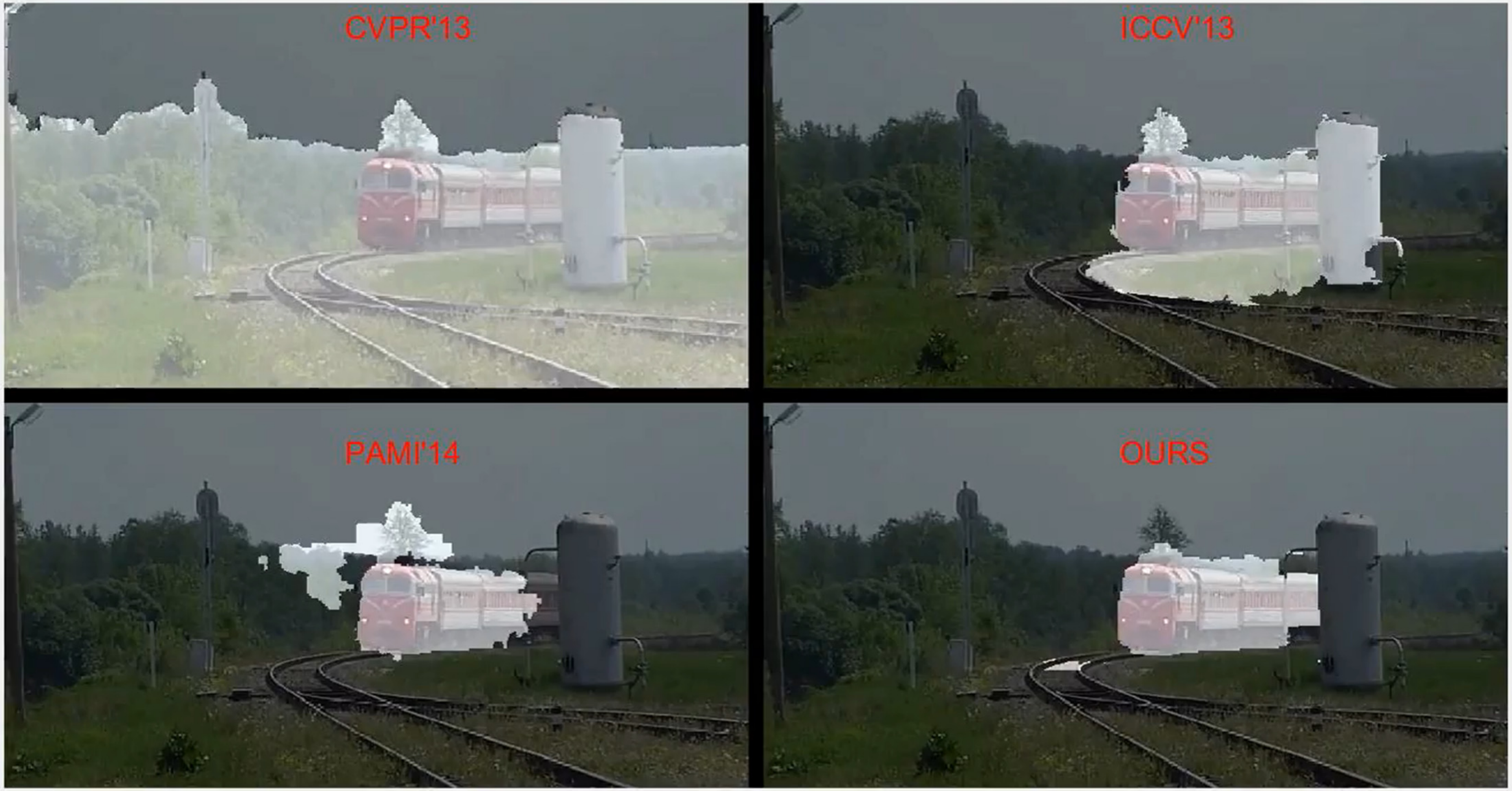
Foreground Mask

(AMF)

(MoG)



"Motion-based Occlusion-aware Pixel Graph Network for Video Object Segmentation", Saptakatha Adak and Sukhendu Das; in 26th International Conference on Neural Information Processing (ICONIP), Sydney, Australia, December 12-15, 2019; [*Rank – A; Best student paper award*].



IMAVIS ‘17

References

1. "Digital Image Processing"; R. C. Gonzalez and R. E. Woods; Addison Wesley; 1992+.

2. "Computer Vision: Algorithms and Applications"; by Richard Szeliski; Springer-Verlag London Limited 2011.

3. Jianbo Shi and Jitendra Malik; Normalized Cuts and Image Segmentation;, Member, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 22, NO. 8, AUGUST 2000, pp 888-905.

4. Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: Interactive foreground extraction using iterated graph-cuts. ACM Transactions on Graphics, 23(3):309–34, 2004.

5. J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In Proc. IEEE Intl. Conf. on Computer Vision, 2005.