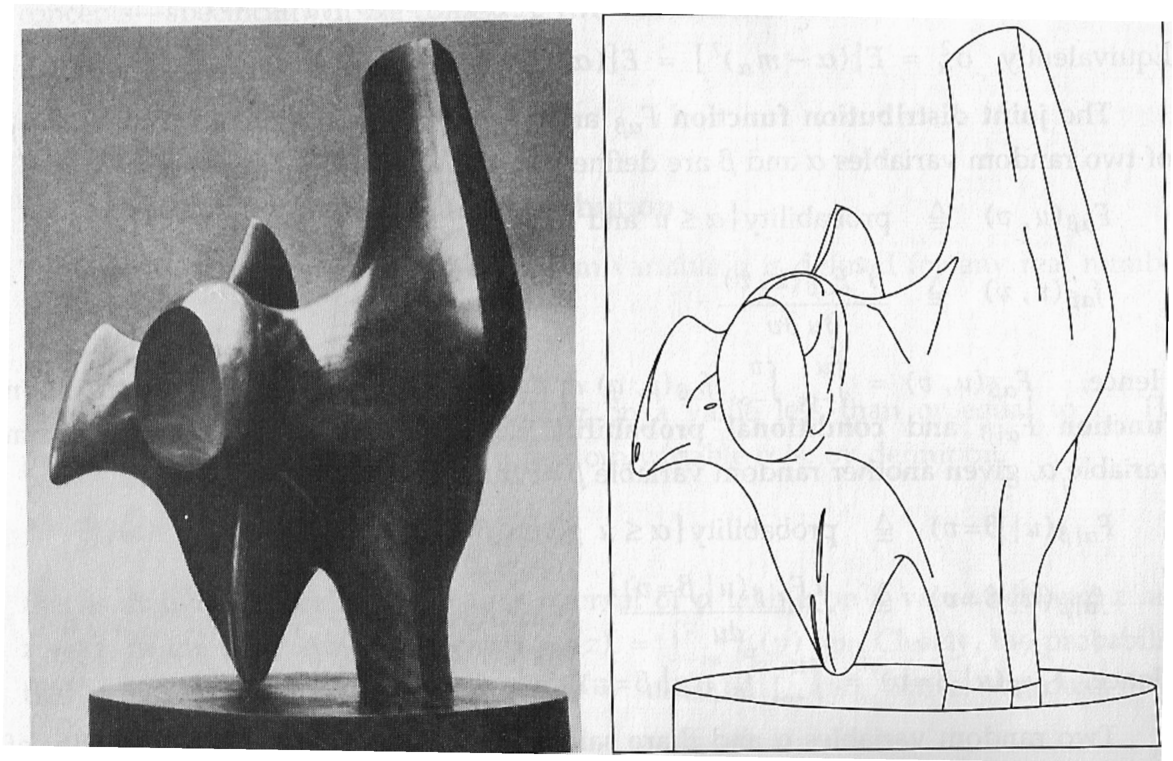
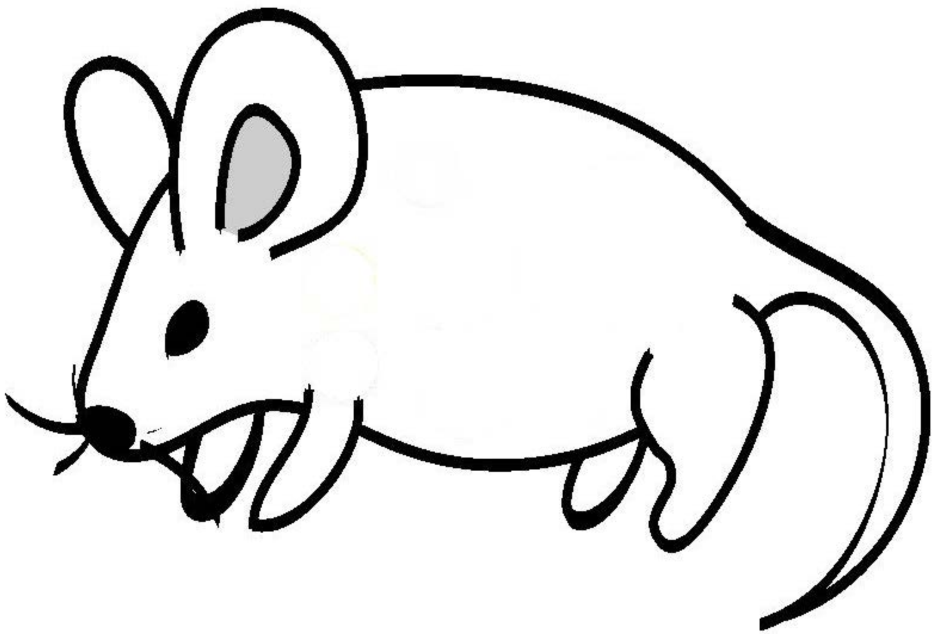
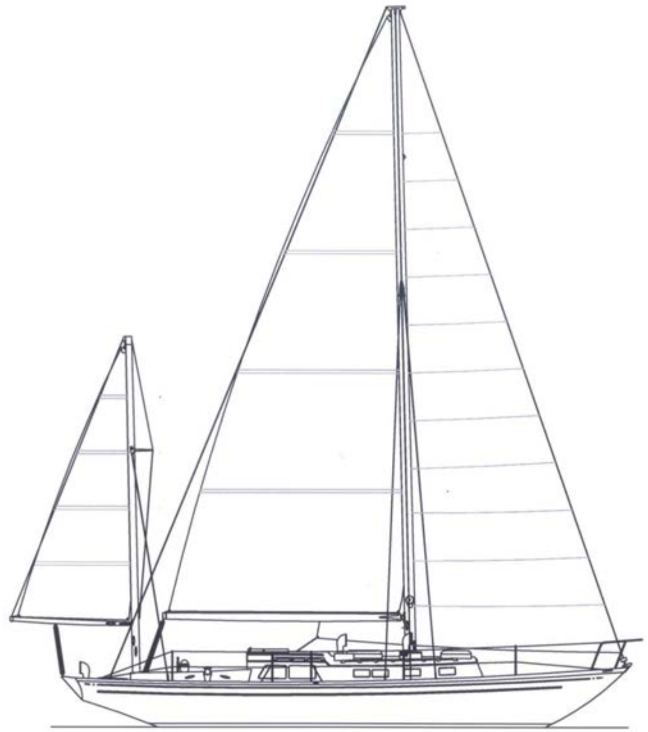
EDGE DETECTION



- What’s an edge
- Image gradients
- Edge operators

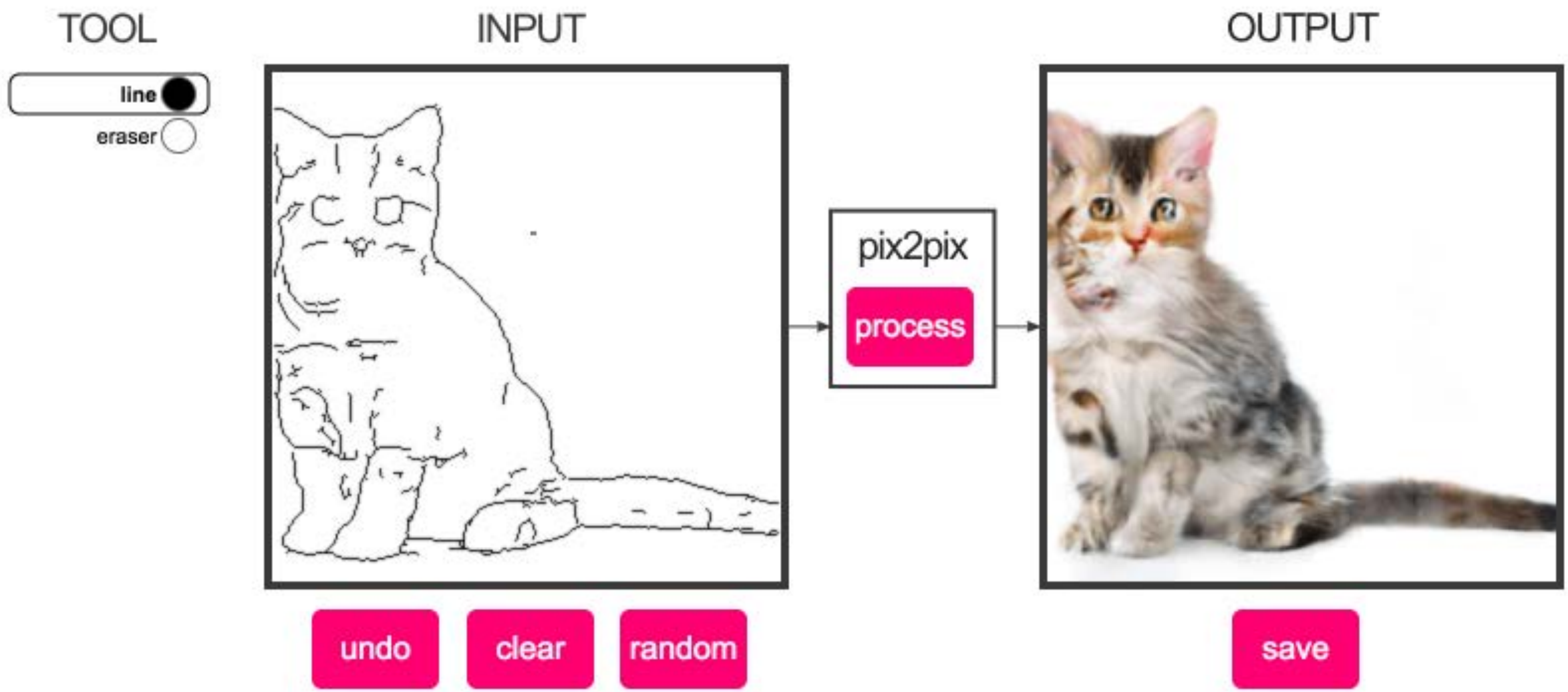
LINE DRAWINGS





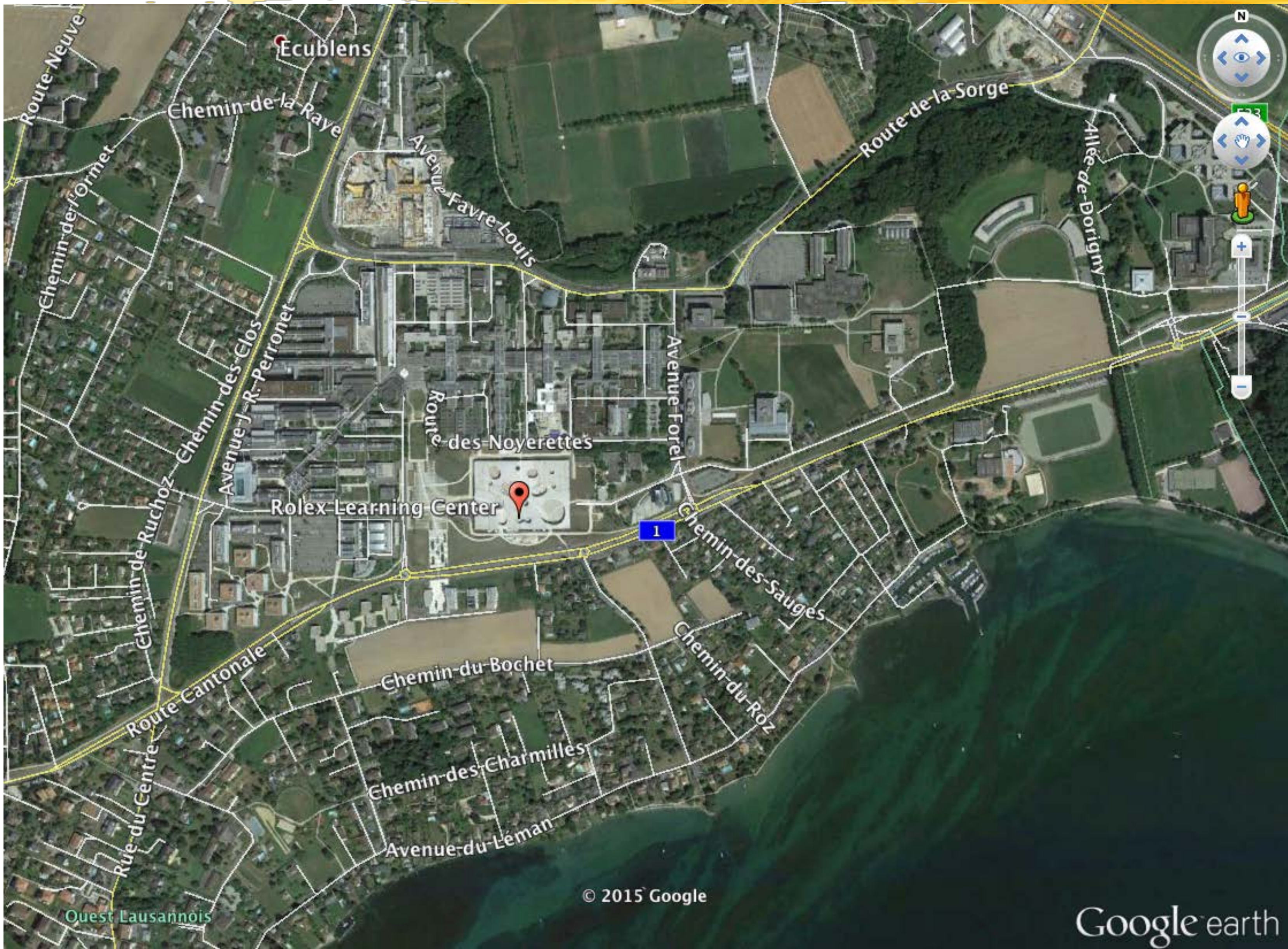
- Edges seem fundamental to human perception.
- They form a compressed version of the image.

FROM EDGES TO CATS



https://affinelayer.com/pixsrv/

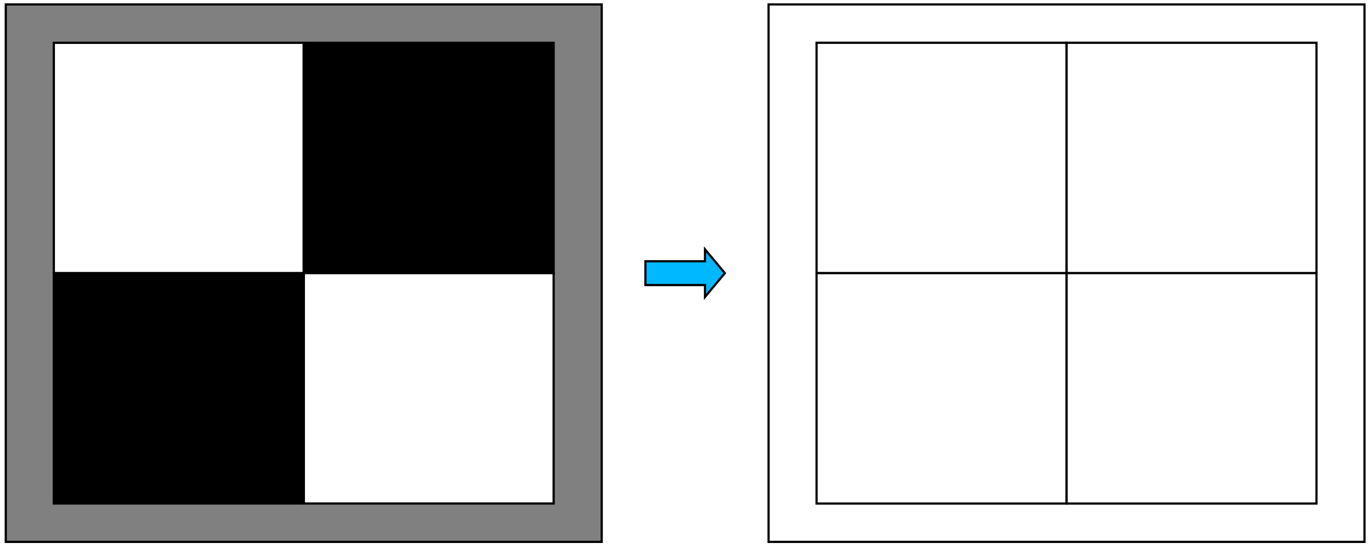
MAPS AND OVERLAYS



CORRIDOR

CORRIDOR

EDGES AND REGIONS



Edges:

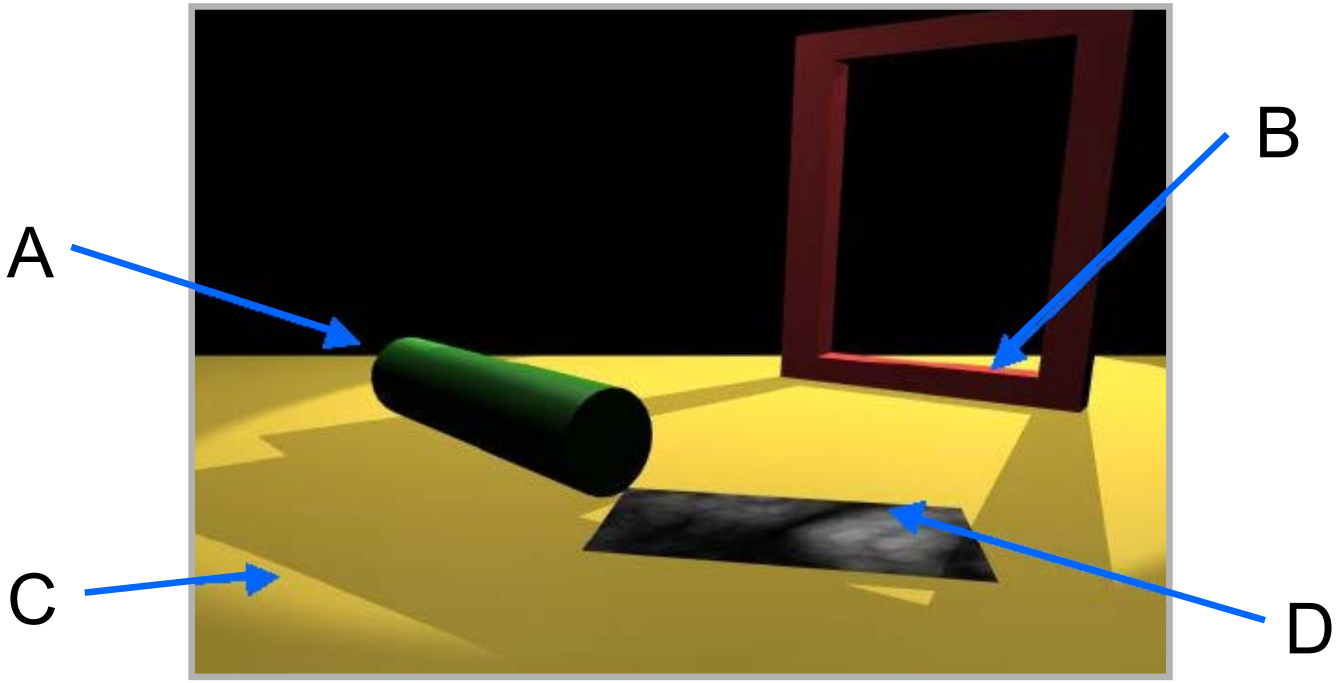
Boundary between bland image regions.

Regions:

Homogenous areas between edges.

→ Duality Edge/Regions.

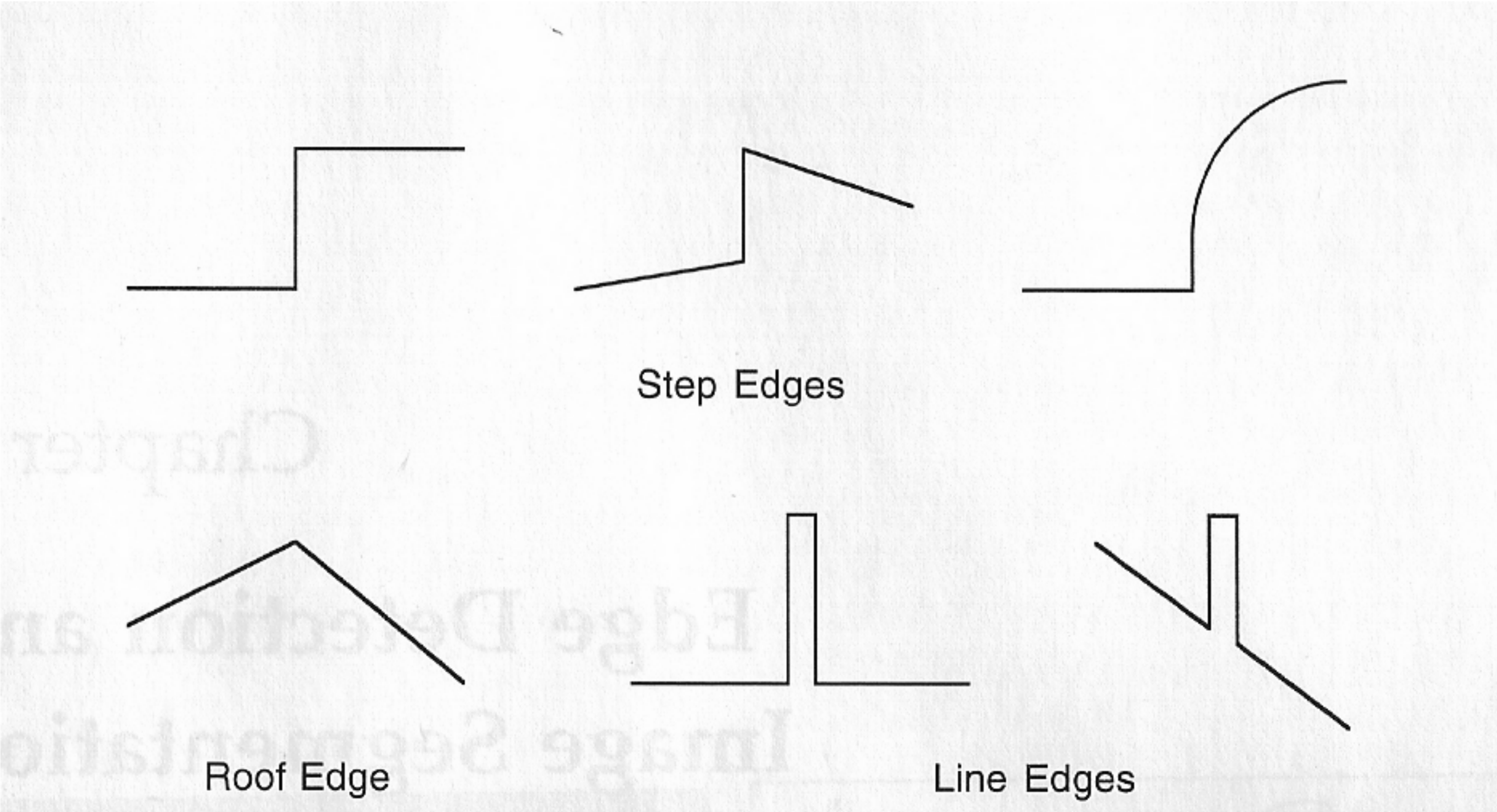
DISCONTINUITIES



A. Depth discontinuity: Abrupt depth change in the world
B. Surface normal discontinuity: Change in surface orientation
C. Illumination discontinuity: Shadows, lighting changes
D. Reflectance discontinuity: Surface properties, markings

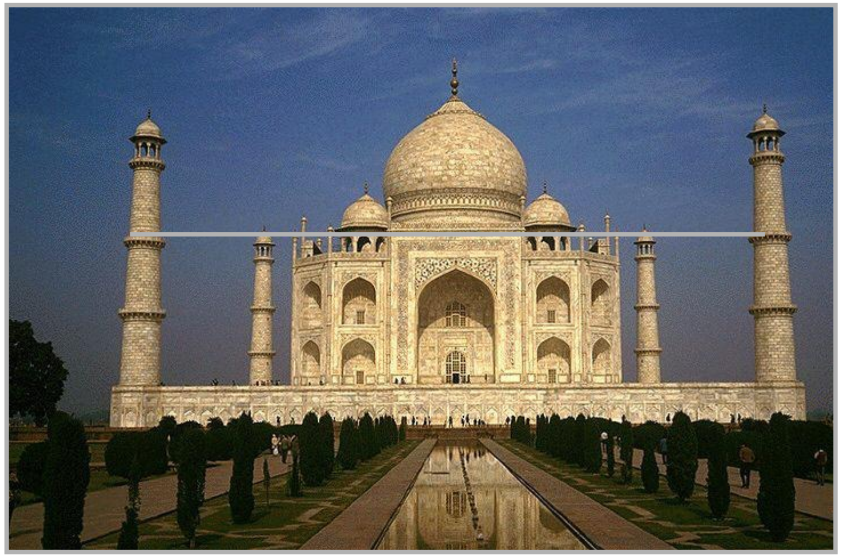
→ **Contrast:** Gray levels different on both sides

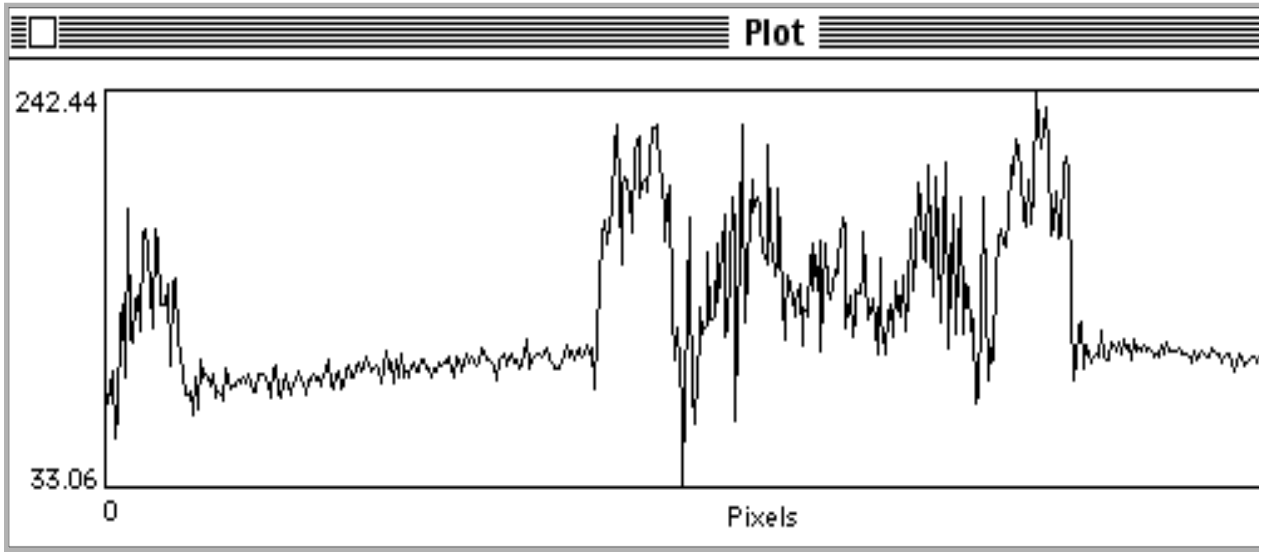
EDGE PROFILES

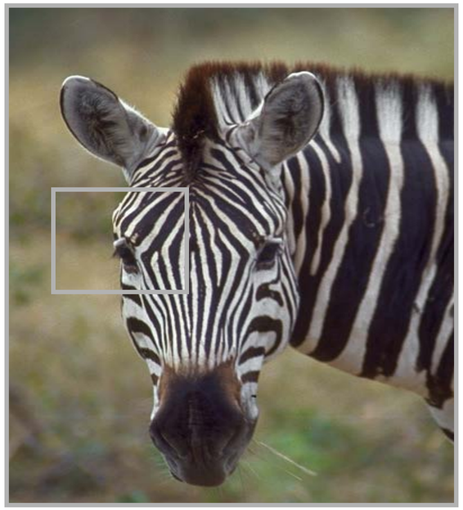


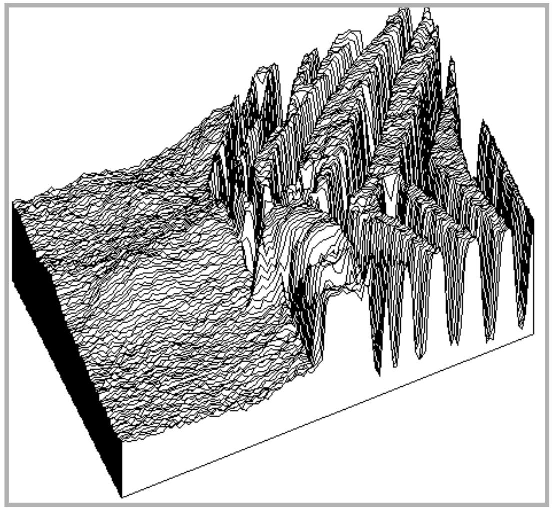
Edges are where a change occurs

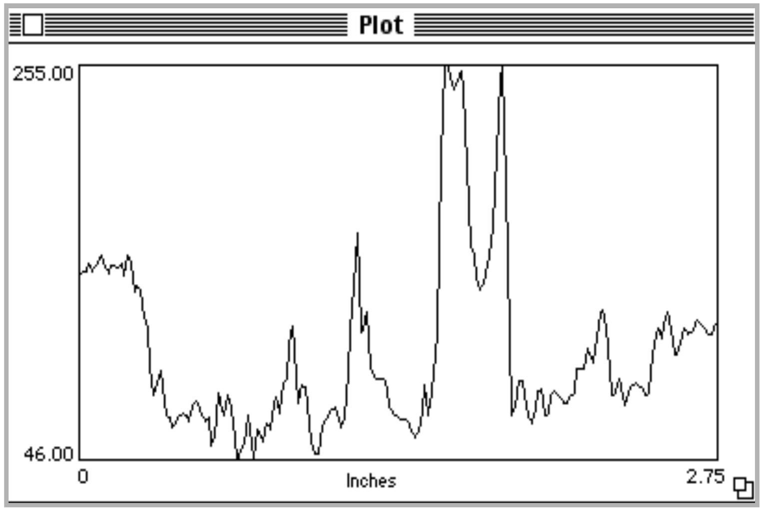
REALITY



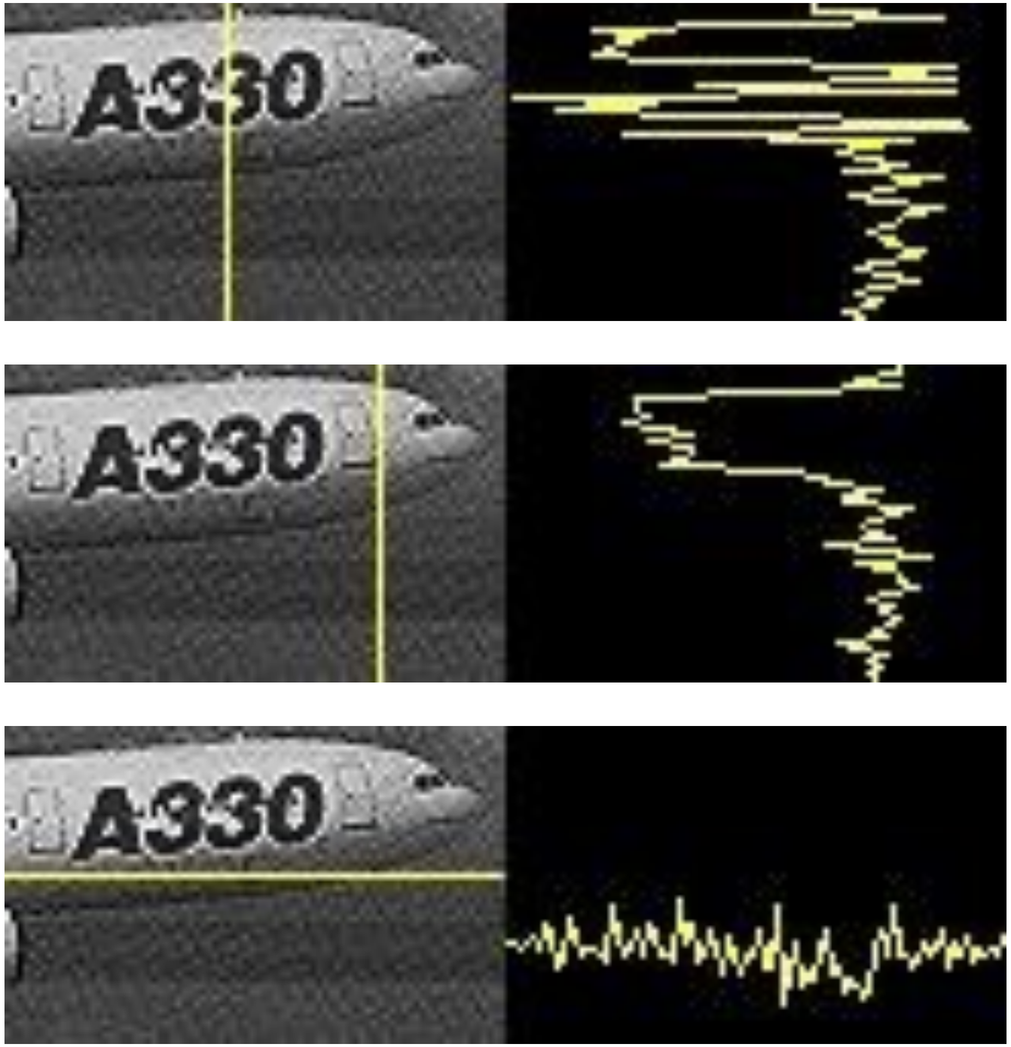








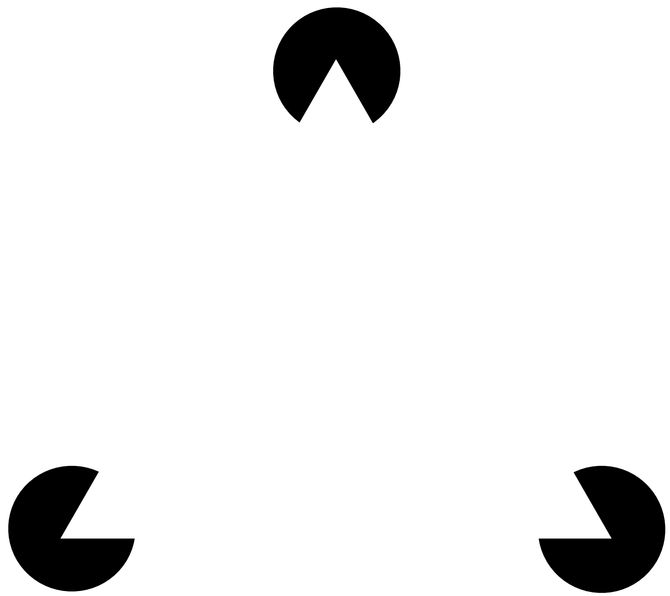
MORE REALITY



Very noisy signals

→ Much knowledge is required!!

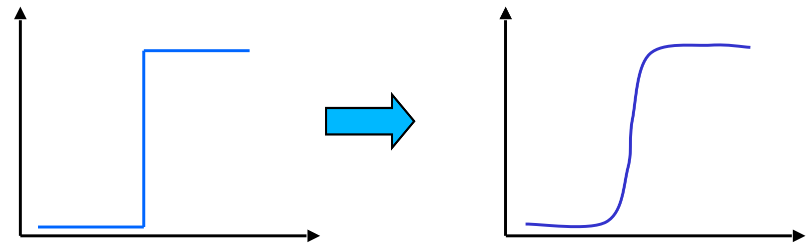
ILLUSORY CONTOURS



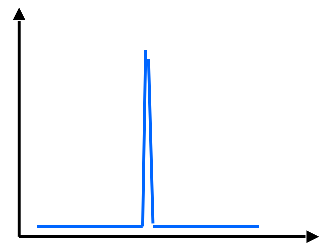
No closed contour, but ......

IDEAL STEP EDGE

f(x) = step edge

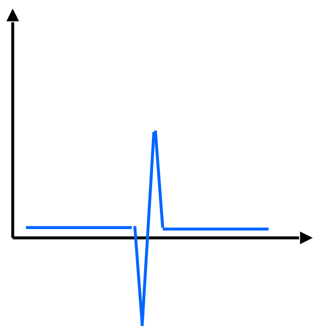


1st Derivative f’(x)



maximum

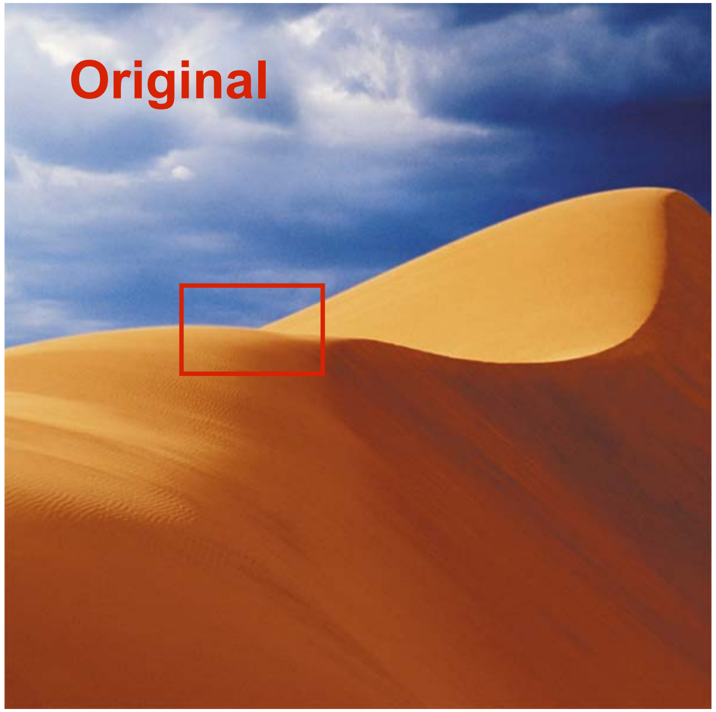
2nd Derivative f’’(x)

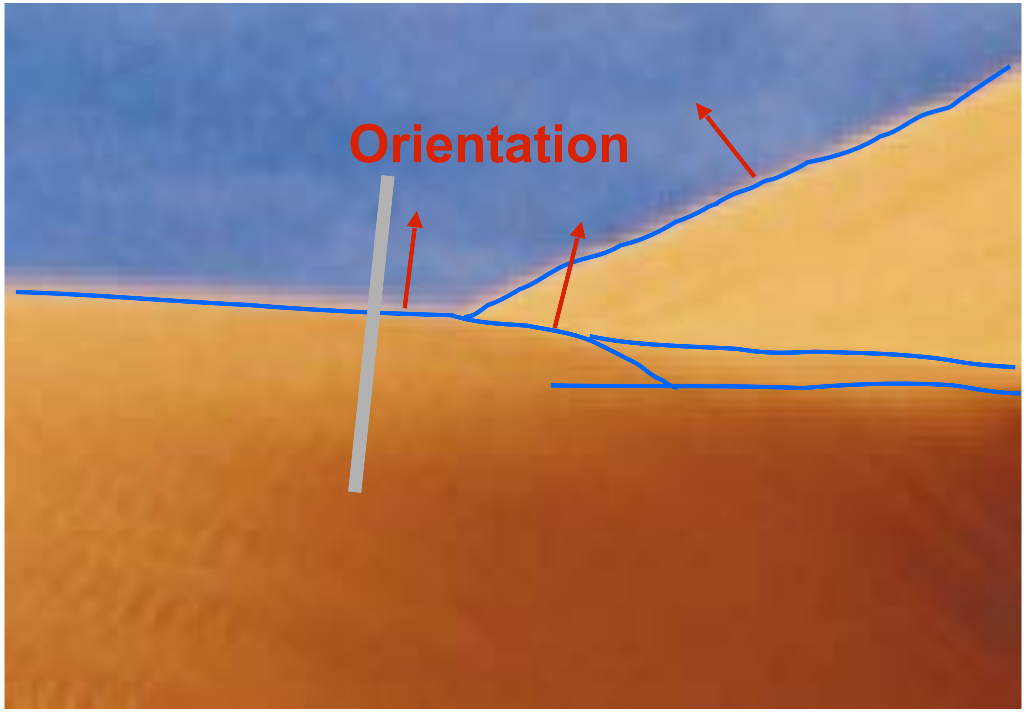


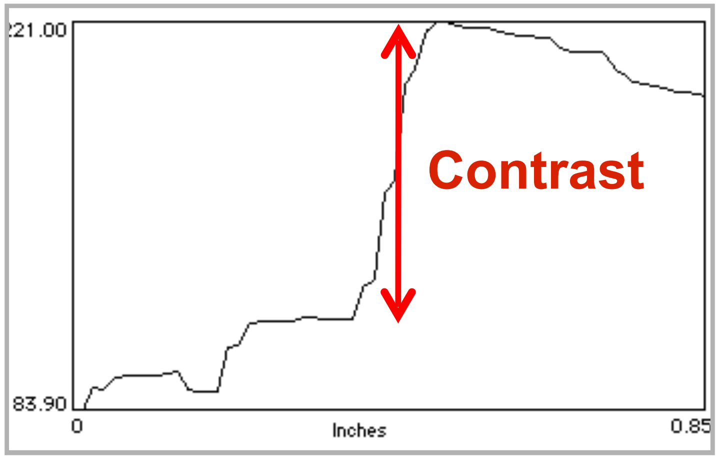
zero crossing

Rapid change in image => High local gradient

EDGE PROPERTIES







EDGE DESCRIPTORS

Edge Normal:

- Unit vector in the direction of maximum intensity change

Edge Direction:

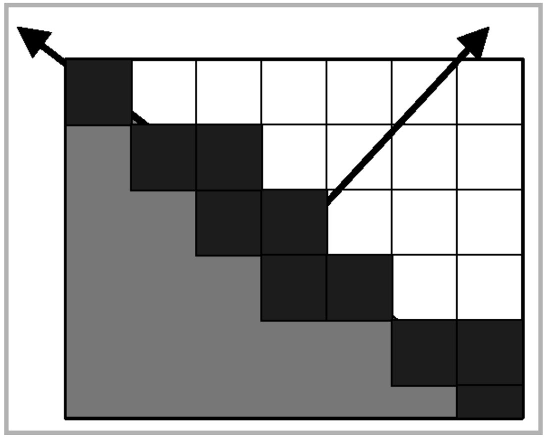
- Unit vector perpendicular to the edge normal

Edge position or center

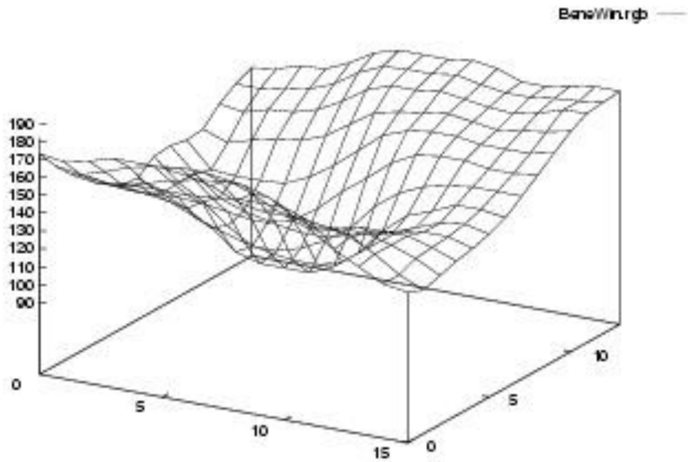
- Image location at which edge is located

Edge Strength

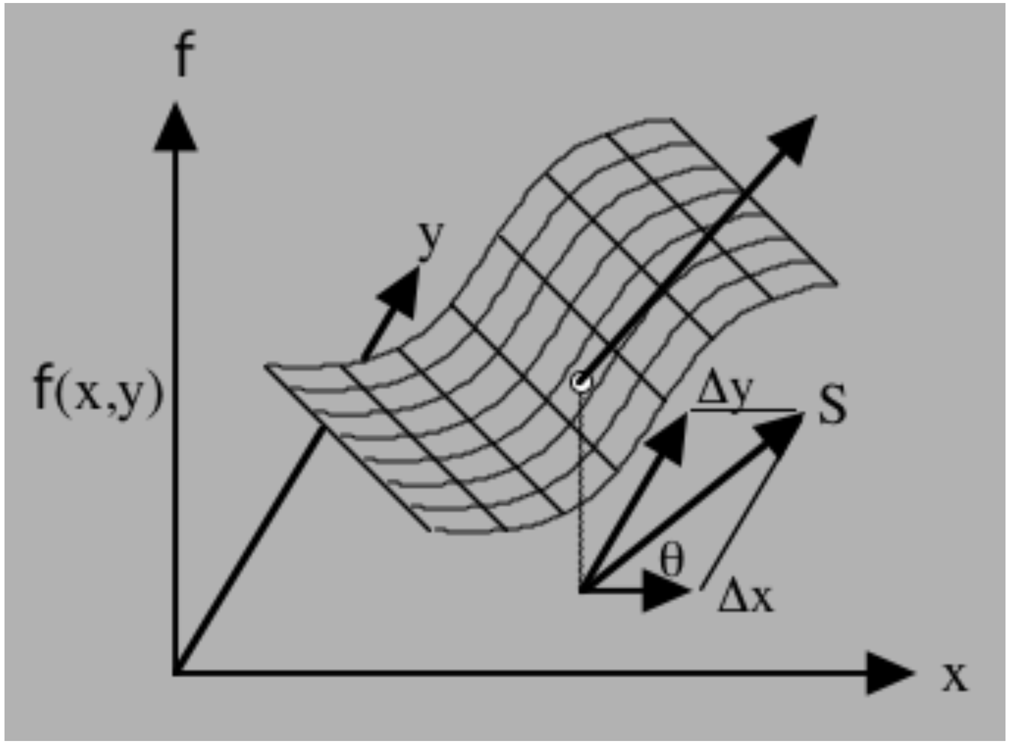
- Speed of intensity variation across the edge.



IMAGES AS 3-D SURFACES



GEOMETRIC INTERPRETATION



Since I(x,y) is not a continuous function:

1. Locally fit a smooth surface.
2. Compute its derivatives.

IMAGE GRADIENT

The gradient of an image

$$\nabla I = \left[\frac{\delta I}{\delta x}, \frac{\delta I}{\delta y}\right]$$

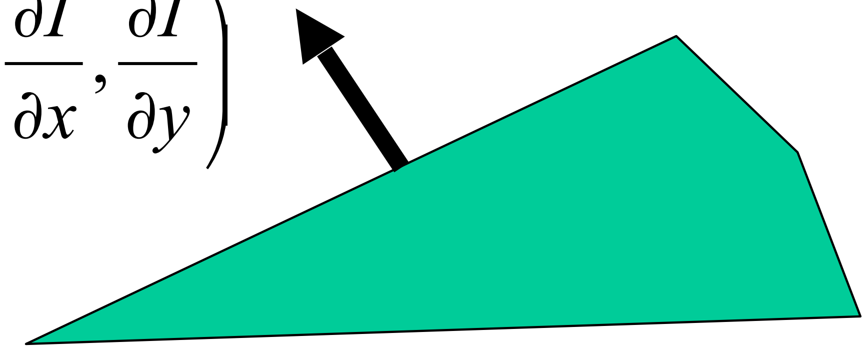
points in the direction of most rapid change in intensity.

$$\nabla I = \left[\frac{\delta I}{\delta x}, 0\right]$$

$$\nabla I = \left[0, \frac{\delta I}{\delta y}\right]$$

$$\nabla I = \left[\frac{\delta I}{\delta x}, \frac{\delta I}{\delta y}\right]$$

MAGNITUDE AND ORIENTATION

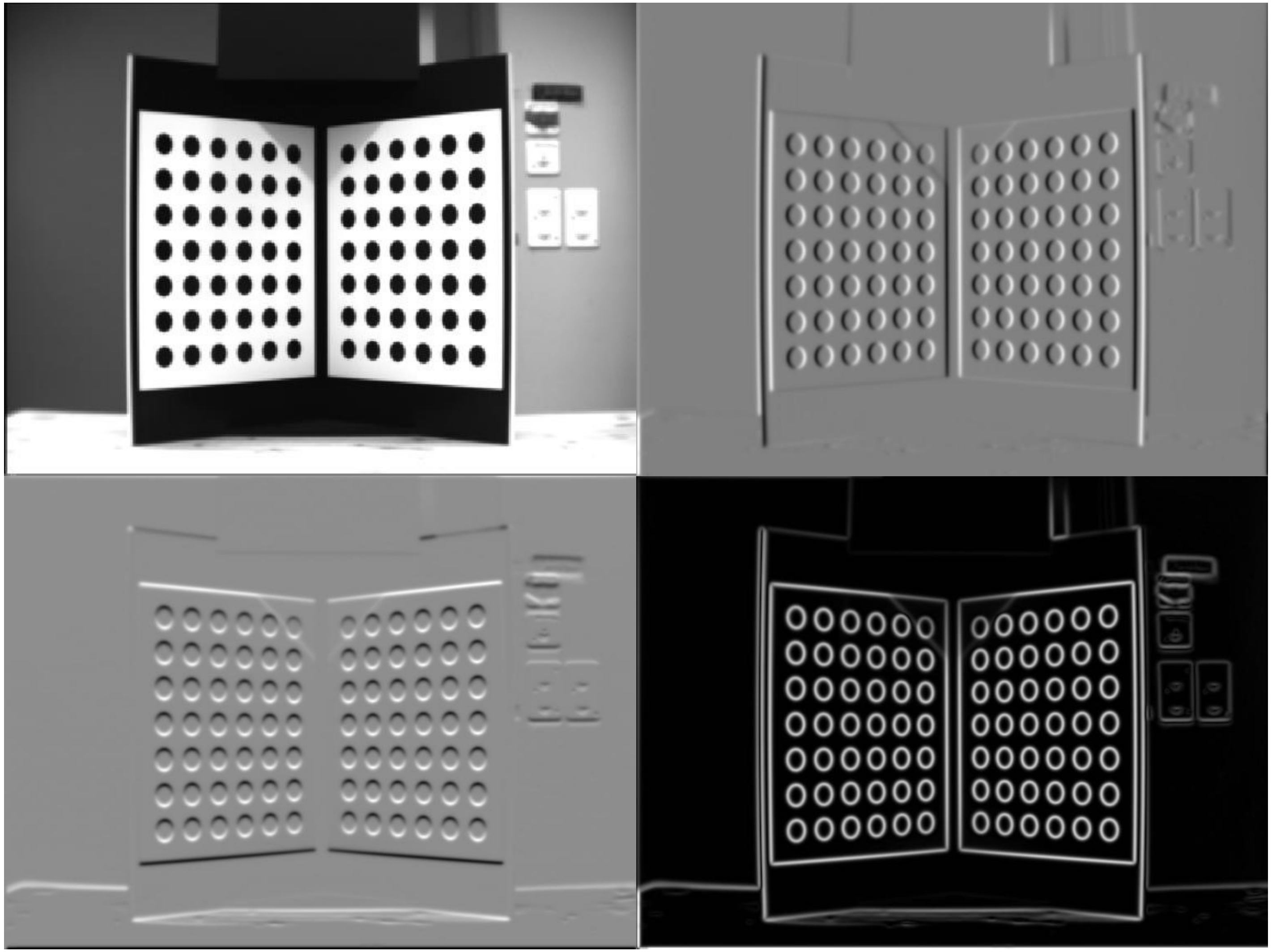


$$\left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$$

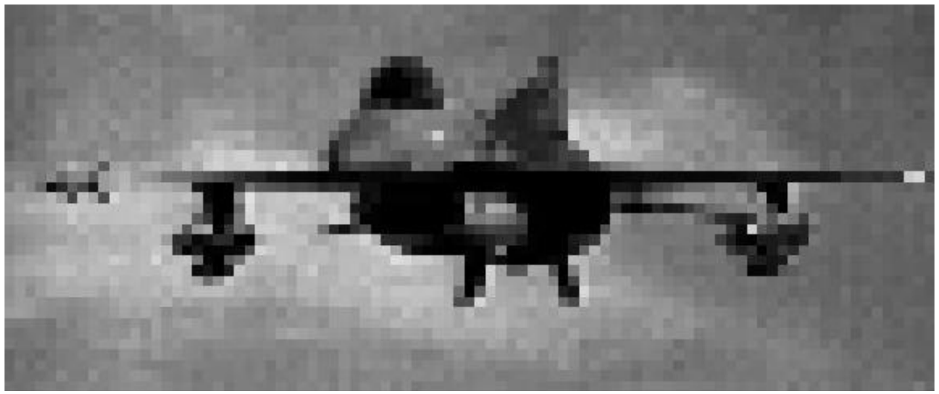
$$\text{Measure of contrast}: \; G = \sqrt{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2}$$

$$\text{Edge orientation}: \; \theta = \arctan\left(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x}\right)$$

GRADIENT IMAGES



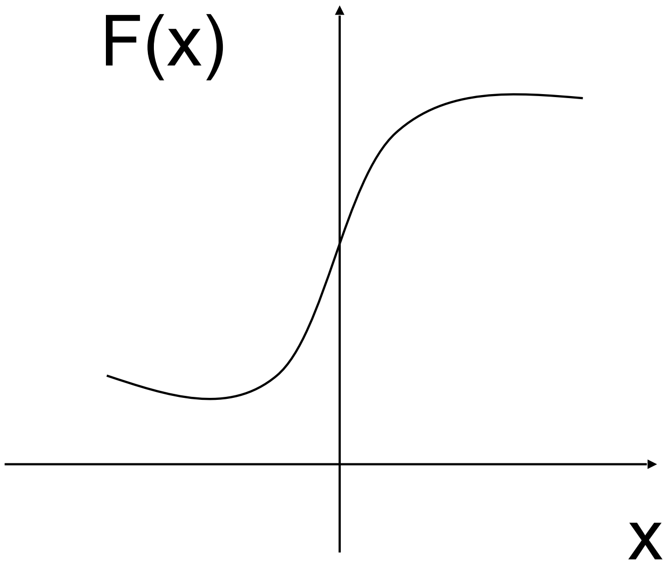
REAL IMAGES

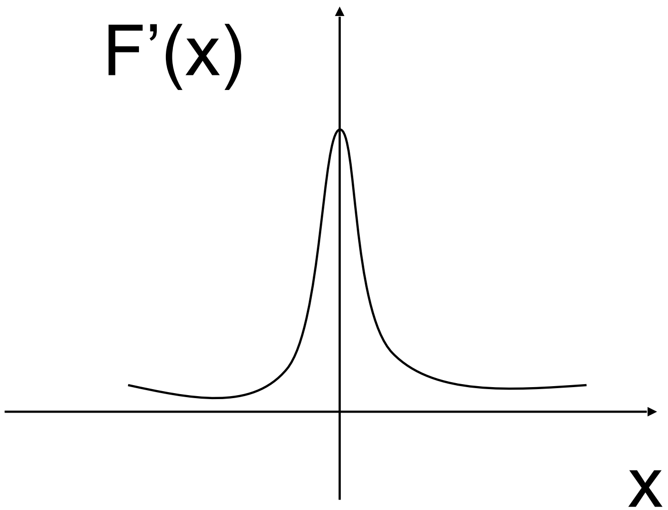


EDGE OPERATORS

- Difference Operators
- Convolution Operators
- Parametric Matchers
- Trained Detectors
- Deep Nets

GRADIENT METHODS





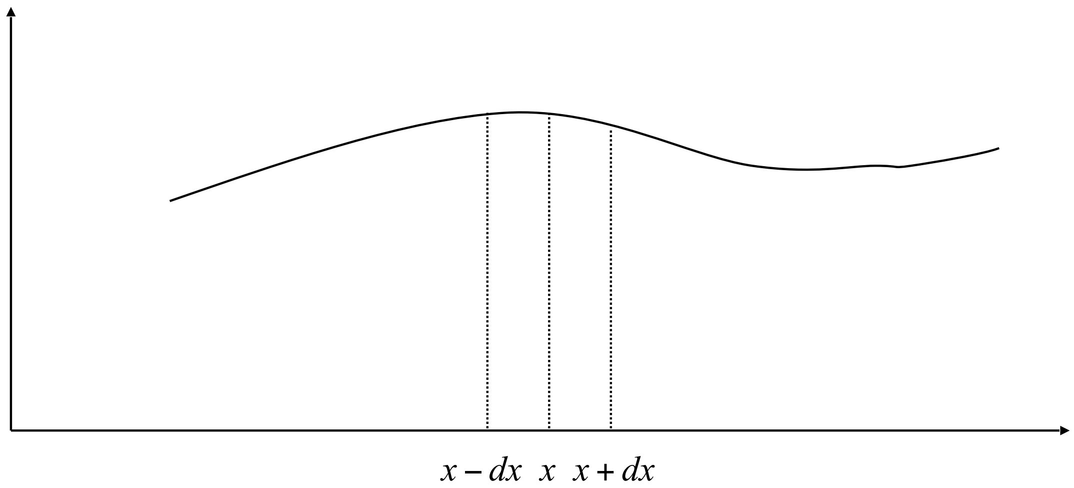
Edge = Sharp variation

↓

Large first derivative

1D FINITE DIFFERENCES

In one dimension:



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

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

CODING 1D FINITE DIFFERENCES

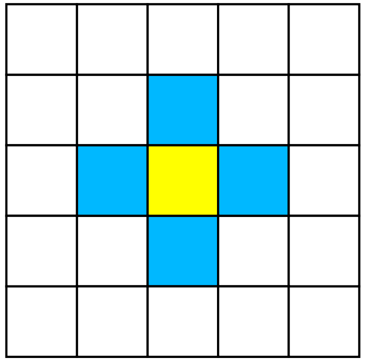
Line stored as an array:

- for i in range(n-1):
  q[i]=(p[i+1]-p[i])

- for i in range(1,n-1):
  q[i]=(p[i+1]-p[i-1])/2

- q=(p[2:]-p[:-2])/2

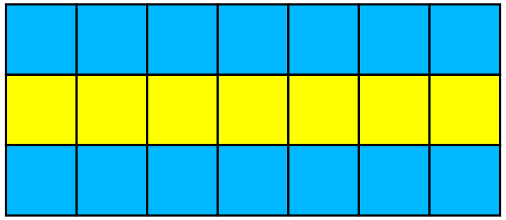
2D FINITE DIFFERENCES



$$\frac{\partial f}{\partial x} \approx \frac{f(x+dx, y) - f(x, y)}{dx} \approx \frac{f(x+dx, y) - f(x-dx, y)}{2dx}$$

$$\frac{\partial f}{\partial y} \approx \frac{f(x, y+dy) - f(x, y)}{dy} \approx \frac{f(x, y+dy) - f(x, y-dy)}{2dy}$$

CODING 2D FINITE DIFFERENCES



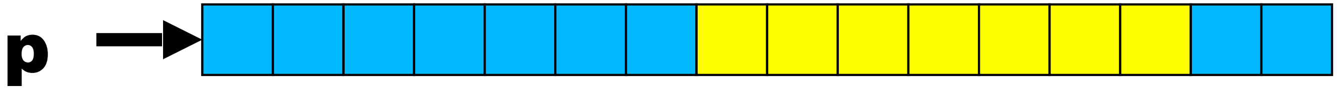


Image stored as a 2D array:

- dx = p[1:,:]-p[:-1,:]
  dy = p[:,1:]-p[:,:-1]

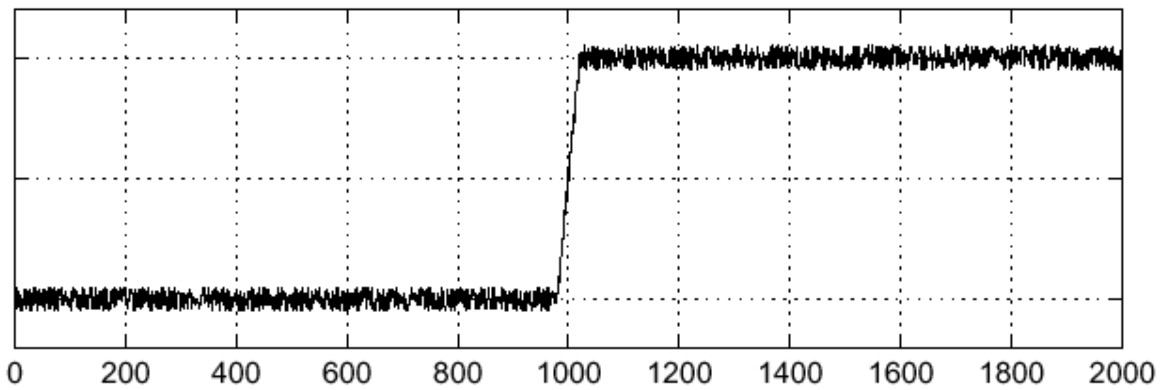
- dx = (p[2:,:]-p[:-2,:])/2
  dy = (p[:,2:]-p[:,:-2])/2

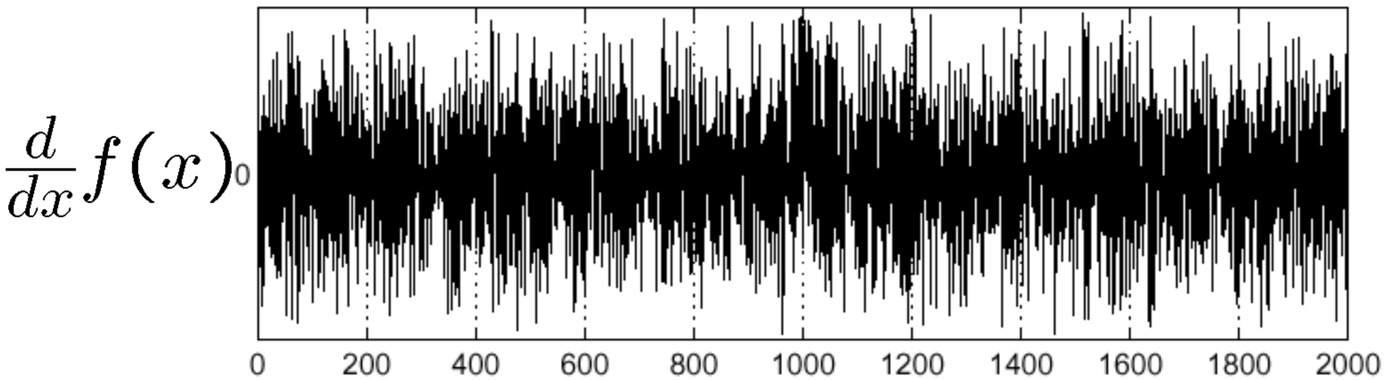
Image stored in raster format:

```
{
  int i;
  for(i=0;i<xdim;i++){
    dx[i] = p[i+1]     -p[i];
    dy[i] = p[i+xdim]-p[i];
  }
}
```

NOISE IN 1D

Consider a single row or column of the image:



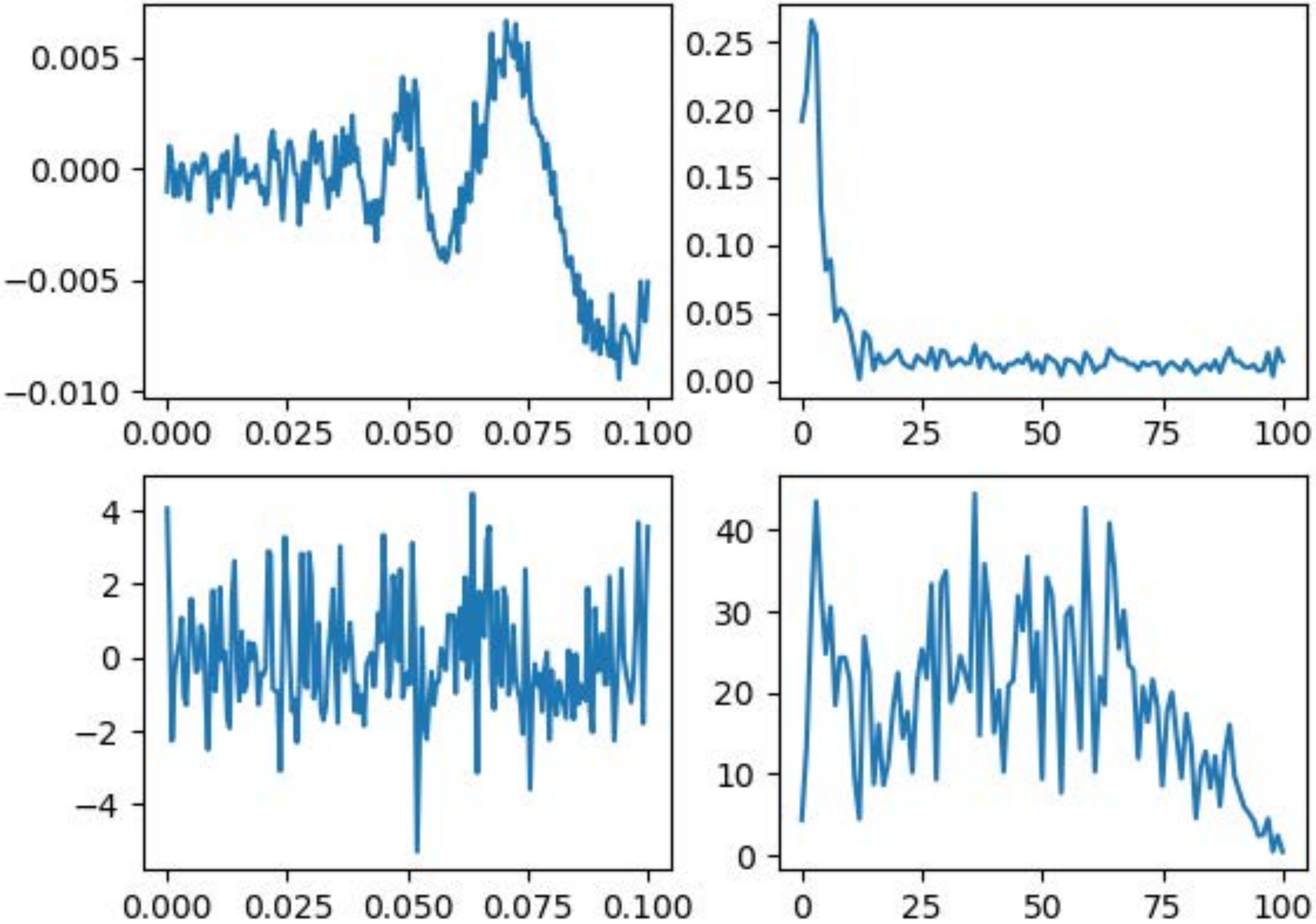


FOURIER INTERPRETATION

Function	Fourier Transform
$\frac{df}{dx}(x)$	$uF(u)$
$\frac{\delta f}{\delta x}(x,y)$	$uF(u,v)$
$\frac{\delta f}{\delta y}(x,y)$	$vF(u,v)$

→ Differentiating emphasizes high frequencies and therefore noise!

$$f(x) = x^2 sin(1/x)$$



f F

$\frac{df}{dx}$ uF

REMOVING NOISE

Problem:

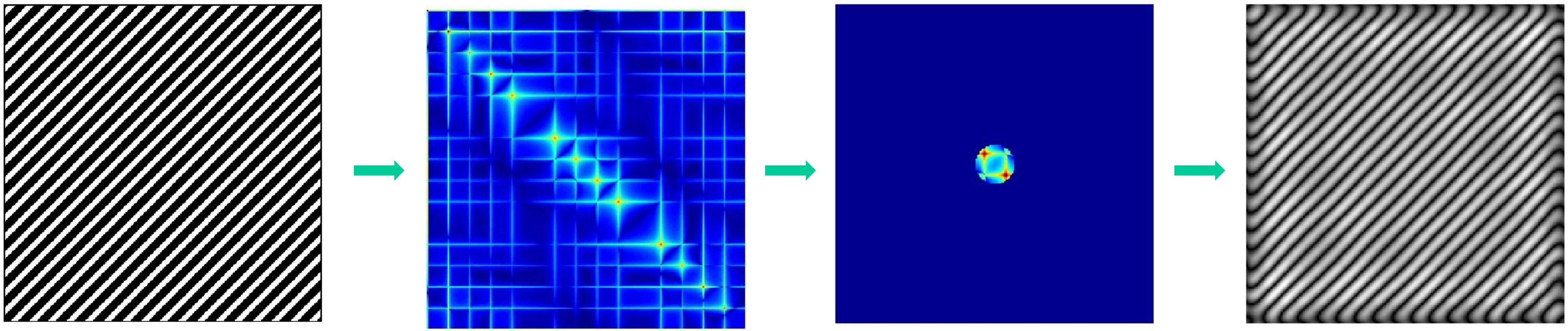
High frequencies lead to trouble with derivation.

Solution:

Suppress high frequencies by

- multiplying DFT of the signal with something that suppresses high frequencies.

DIAGONAL STRUCTURES



Rotated stripes:

- Dominant diagonal structures
- Discretization produces additional harmonics

Removing higher frequencies and reconstructing:

- Smoothed image

REMOVING NOISE

Problem:

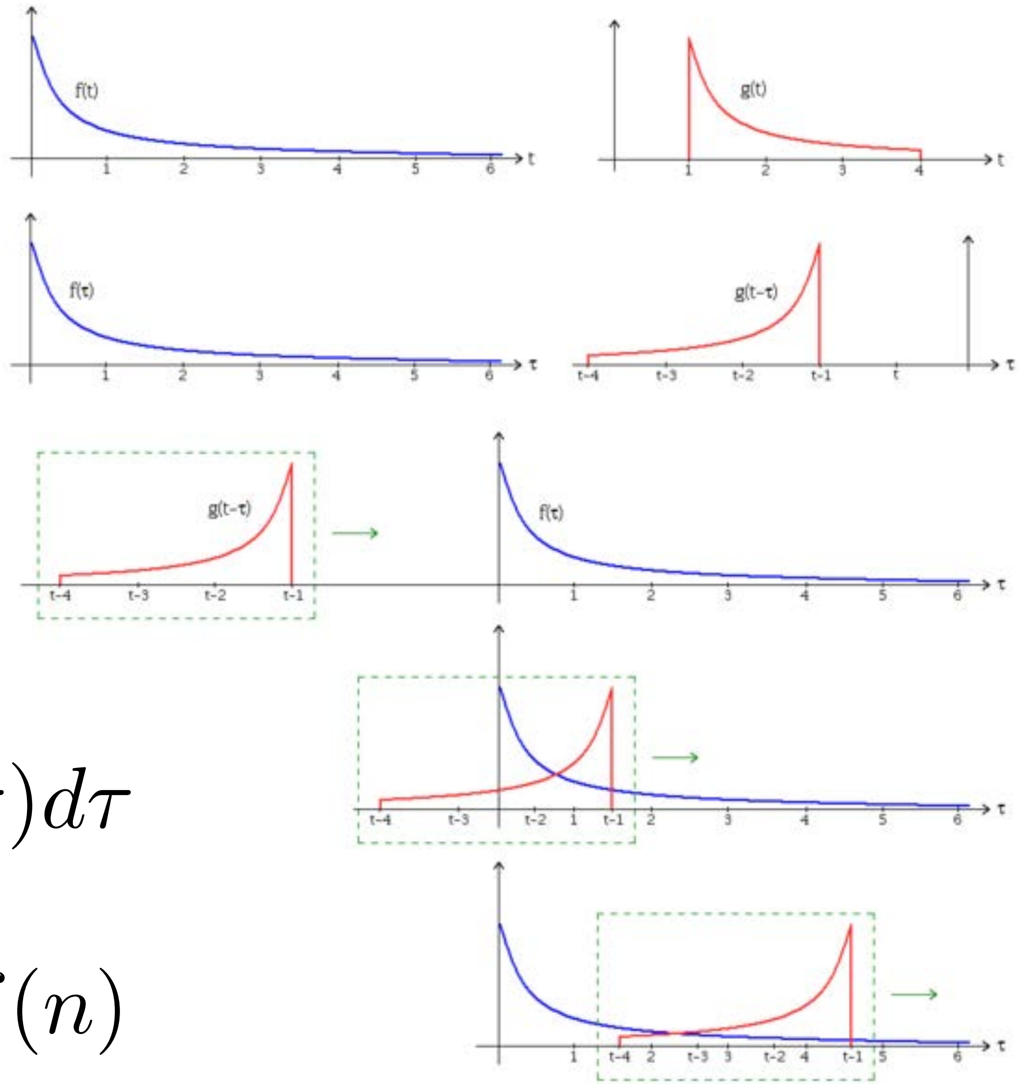
High frequencies lead to trouble with derivation.

Solution:

Suppress high frequencies by

- multiplying DFT of the signal with something that suppresses high frequencies;
- convolving with a low-pass filter.

1D CONVOLUTION



$$g * f(t) = \int_{\tau} g(t - \tau) f(\tau) d\tau$$

$$g * f(m) = \sum_{n} g(m - n) f(n)$$

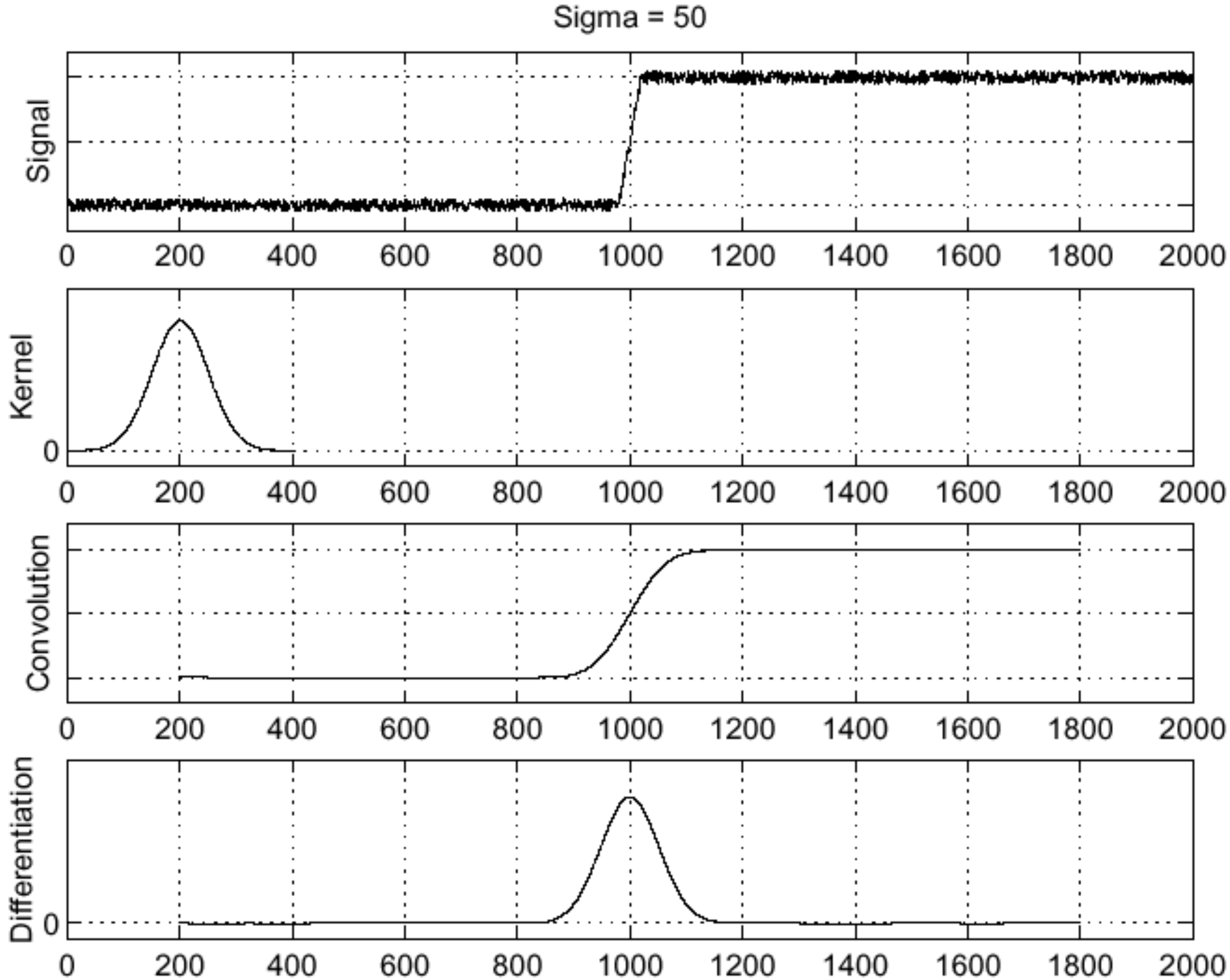
SOLUTION: SMOOTH FIRST

f

g

$g * f$

$$\frac{\partial}{\partial x}(g * f)$$

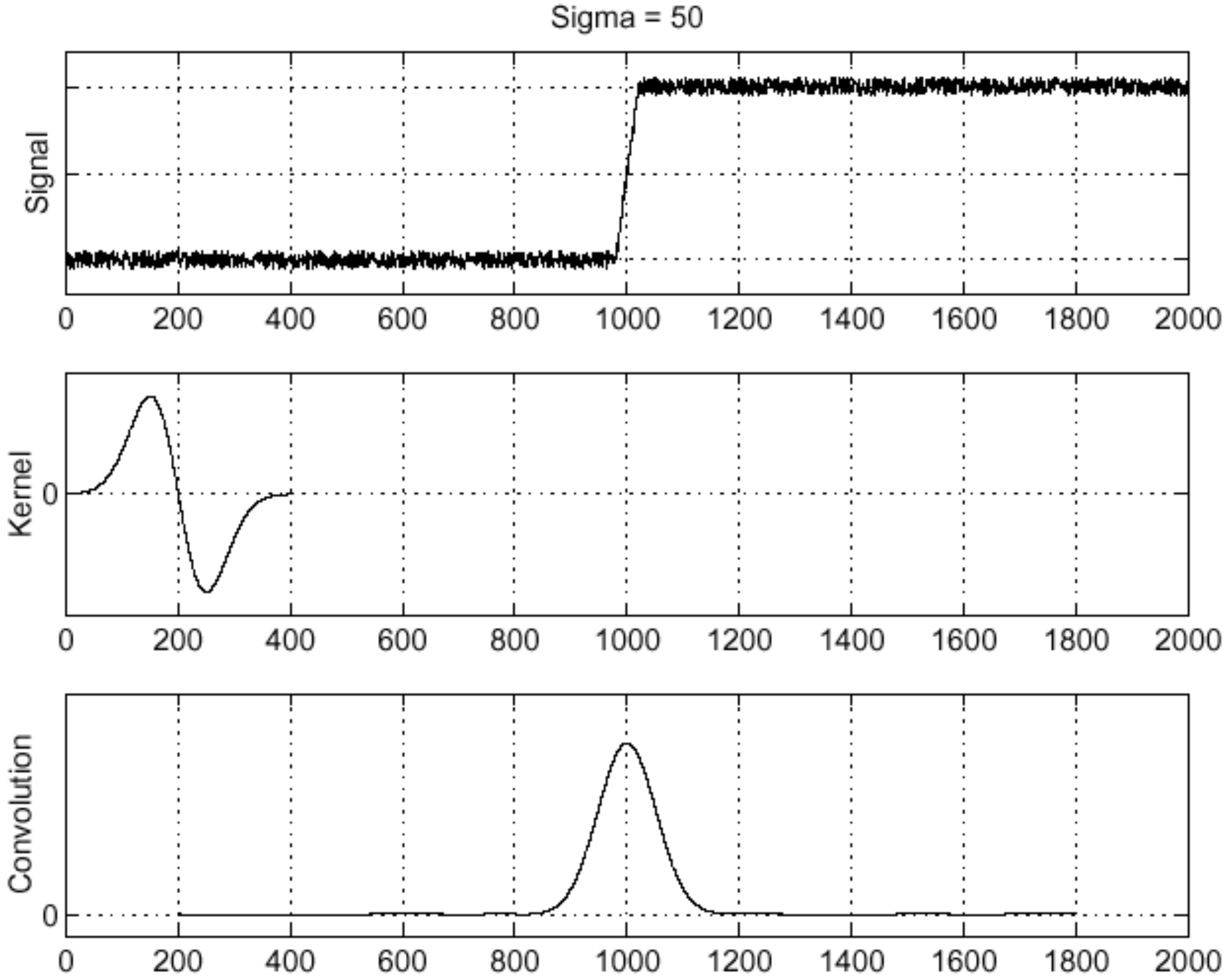


DERIVATIVE THEOREM OF CONVOLUTION

$$f$$

$$\frac{\partial g}{\partial x}$$

$$\frac{\partial}{\partial x}(g * f) = \frac{\partial g}{\partial x} * f$$



--> Faster because dg/dx can be precomputed.

1D AND 2D CONVOLUTION

Continuous case:

$$m \bullet f(x) = \int_u m(u) f(x-u) du$$

$$m \bullet f(x,y) = \int_u \int_v m(u,v) f(x-u, y-v) du dv$$

Discrete case:

$$m \bullet f(x) = \sum_{i=-w}^{w} m(i) f(x-i)$$

$$m \bullet f(x,y) = \sum_{i=-w}^{w} \sum_{j=-h}^{h} m(i,j) f(x-i, y-j)$$

CONVOLUTION IN C

Naive C implementation:

```
static double g[][]={{-1.0,-2.0,-1.0},{0.0,0.0,0.0},{1.0,2.0,1.0}};
{
        for(i=i0;i<N;i++)
            for(j=j0;j<N;j++){
              q[i][j]=0;
              for(a=a0;a<W;a++)
                for(b=b0;b<W,b++)
                  q[i][j]+=g[a][b]*p[i-a][j-b];
    }
}
```

Computational complexity:

- N^2W^2 multiplications for a $N \times N$ image and a $W \times W$ mask.
- Lots of memory access

→ Slow, but can be sped up when the filters are separable.

DIFFERENTIATION AS CONVOLUTION

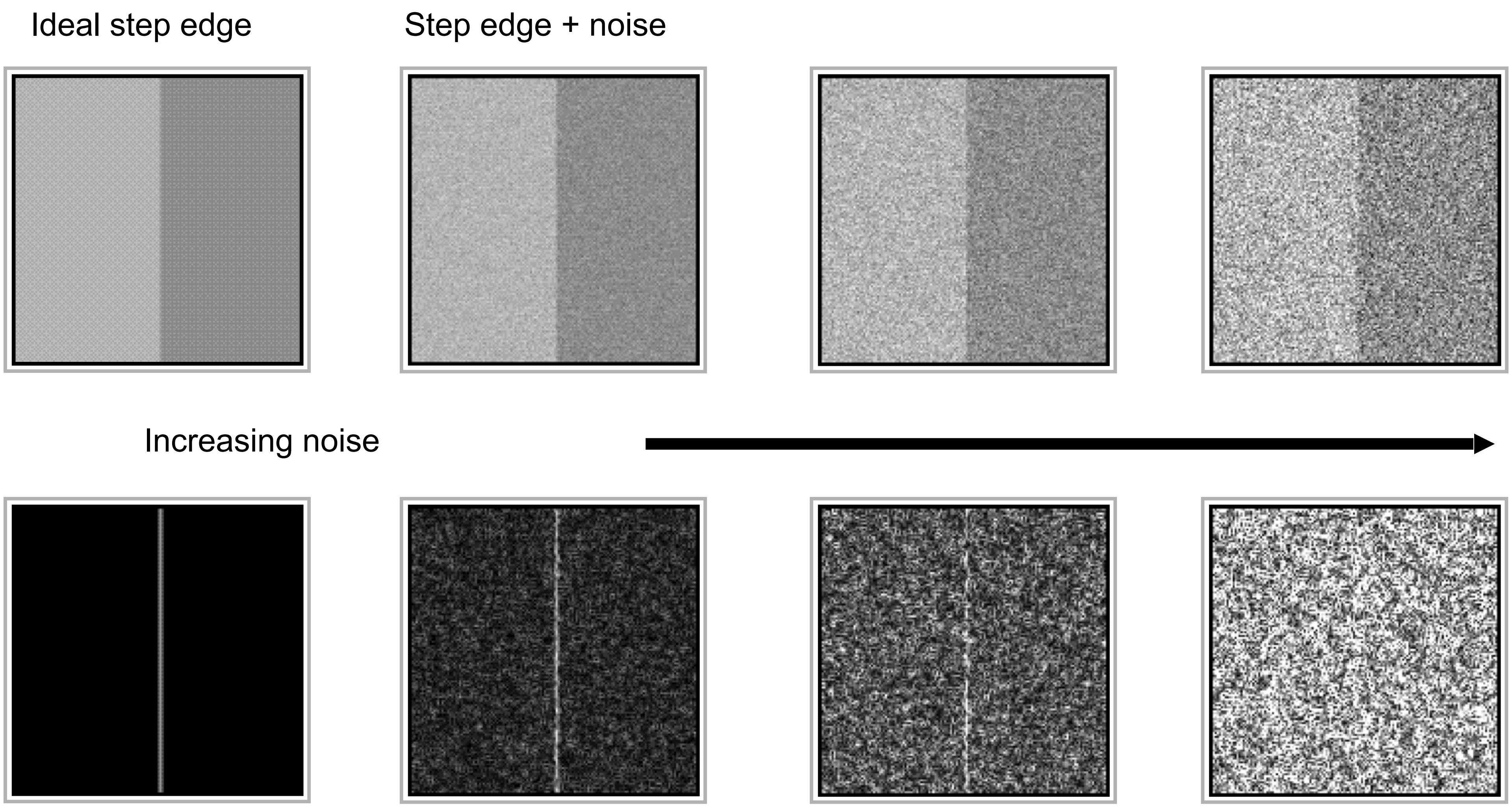
$$\left[-1,1\right] \rightarrow \frac{\partial f}{\partial x} \approx \frac{f(x+dx,y)-f(x,y)}{dx}$$

$$\left[-0.5,0,0.5\right] \rightarrow \frac{\partial f}{\partial x} \approx \frac{f(x+dx,y)-f(x-dx,y)}{2dx}$$

$$\begin{bmatrix}-1\\1\end{bmatrix} \rightarrow \frac{\partial f}{\partial y} \approx \frac{f(x,y+dy)-f(x,y)}{dy}$$

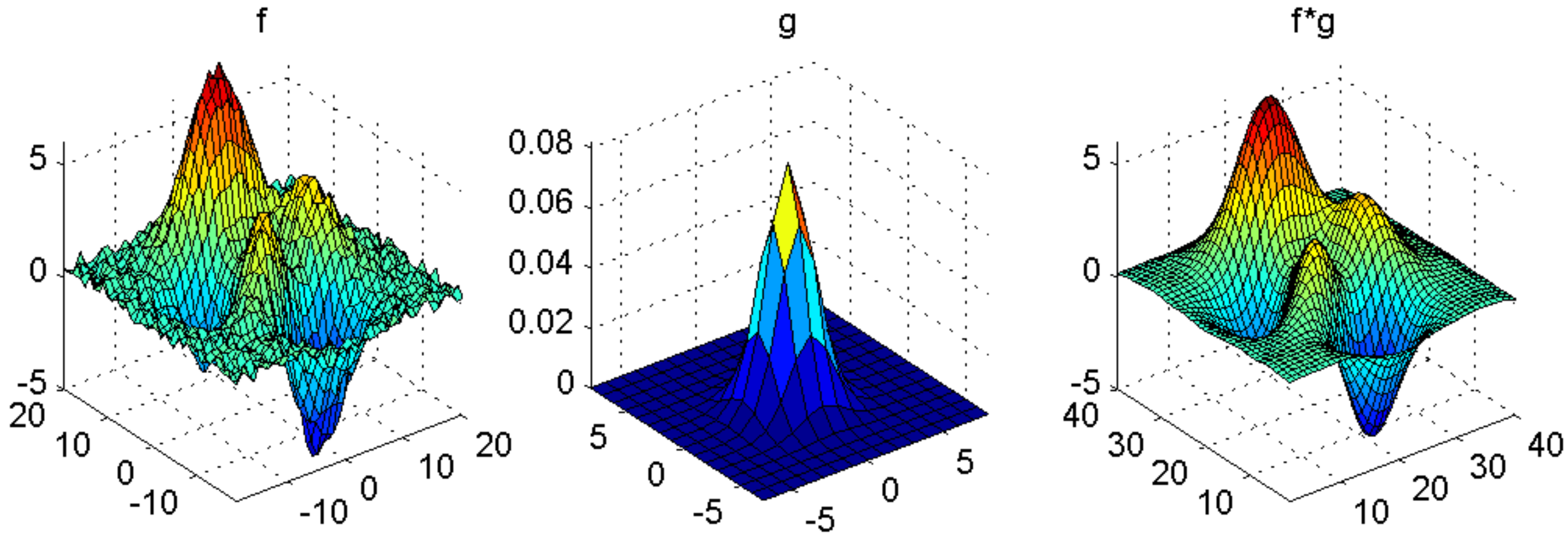
$$\begin{bmatrix}-0.5\\0\\0.5\end{bmatrix} \rightarrow \frac{\partial f}{\partial y} \approx \frac{f(x,y+dy)-f(x,y-dy)}{2dy}$$

NOISE IN 2D



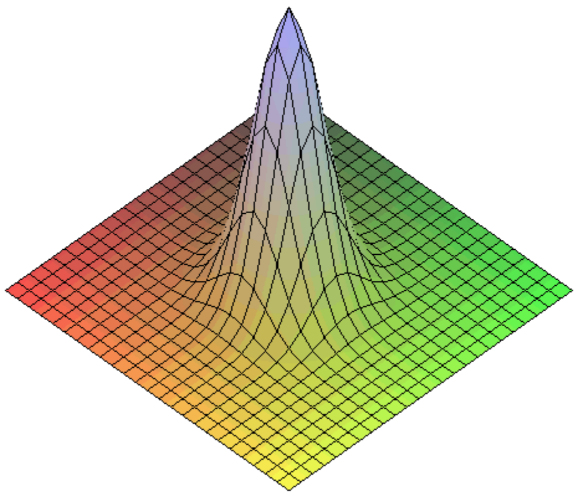
→ Use wider masks to add an element of smoothing

GAUSSIAN SMOOTHING

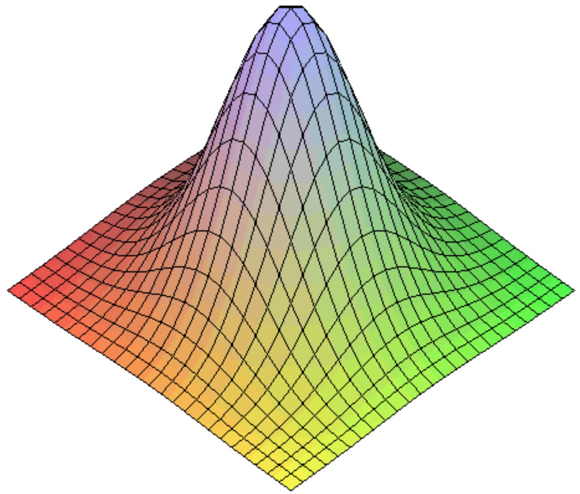


- Eliminates high frequency noise.
- Is fast because the kernel is
  - small,
  - separable.

GAUSSIAN MASKS



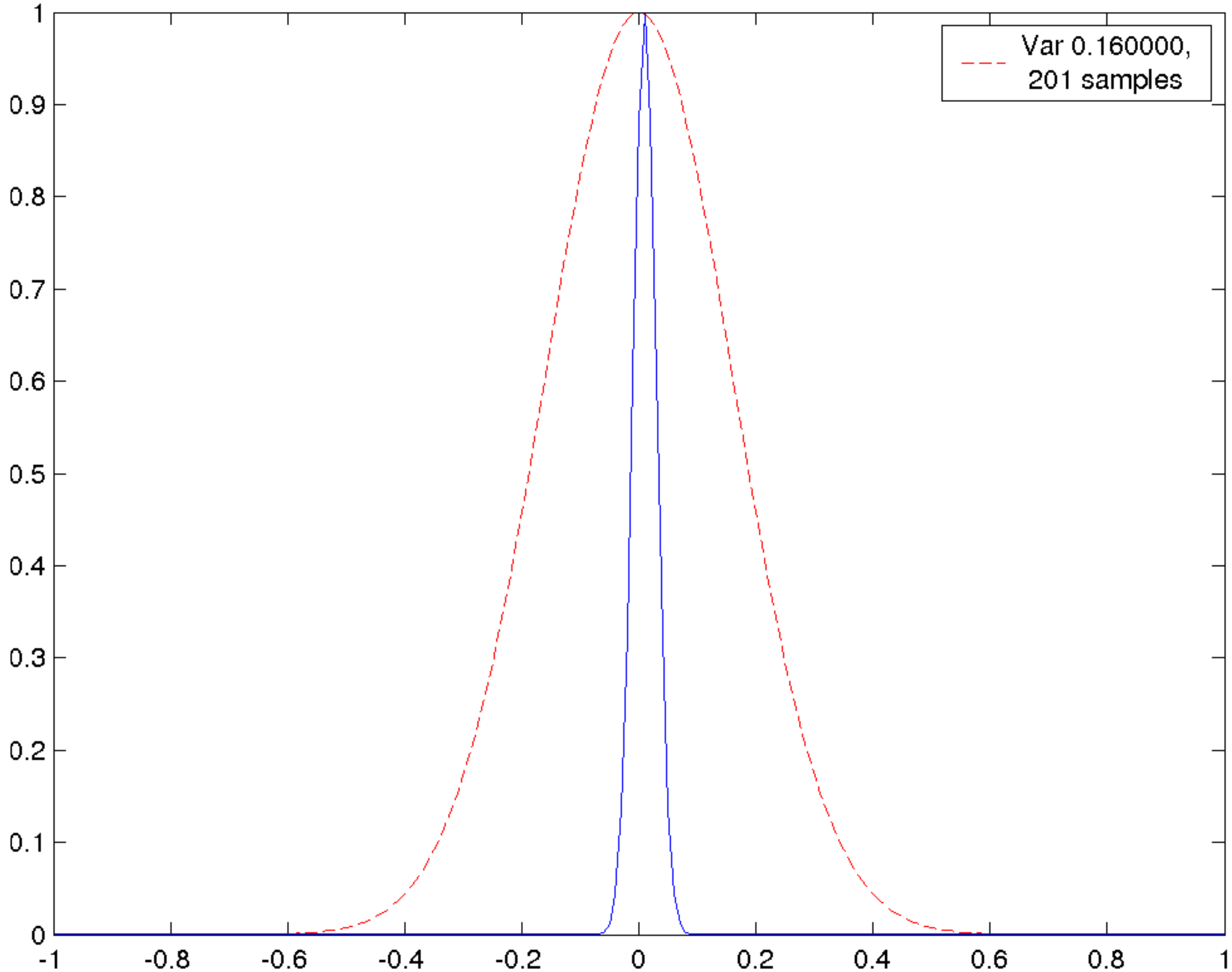
$\sigma = 1$



$\sigma = 2$

$$g_2(x, y) = \frac{1}{2\pi\sigma^2} \exp(-(x^2 + y^2) / 2\sigma^2)$$

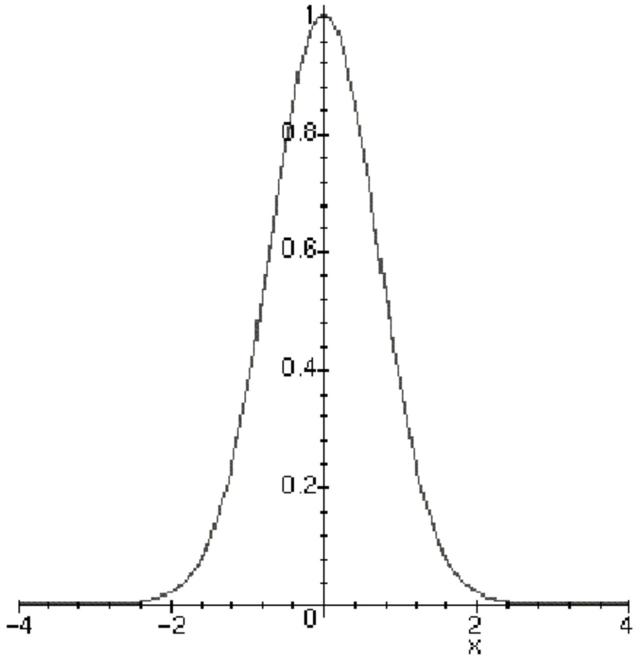
FOURIER TRANSFORM



- The DFT of a Gaussian is a Gaussian.
- It has finite support.
- Its width is inversely proportional to that of the original Gaussian.

—> Convolving with a Gaussian suppresses the high frequencies.

SEPARABILITY



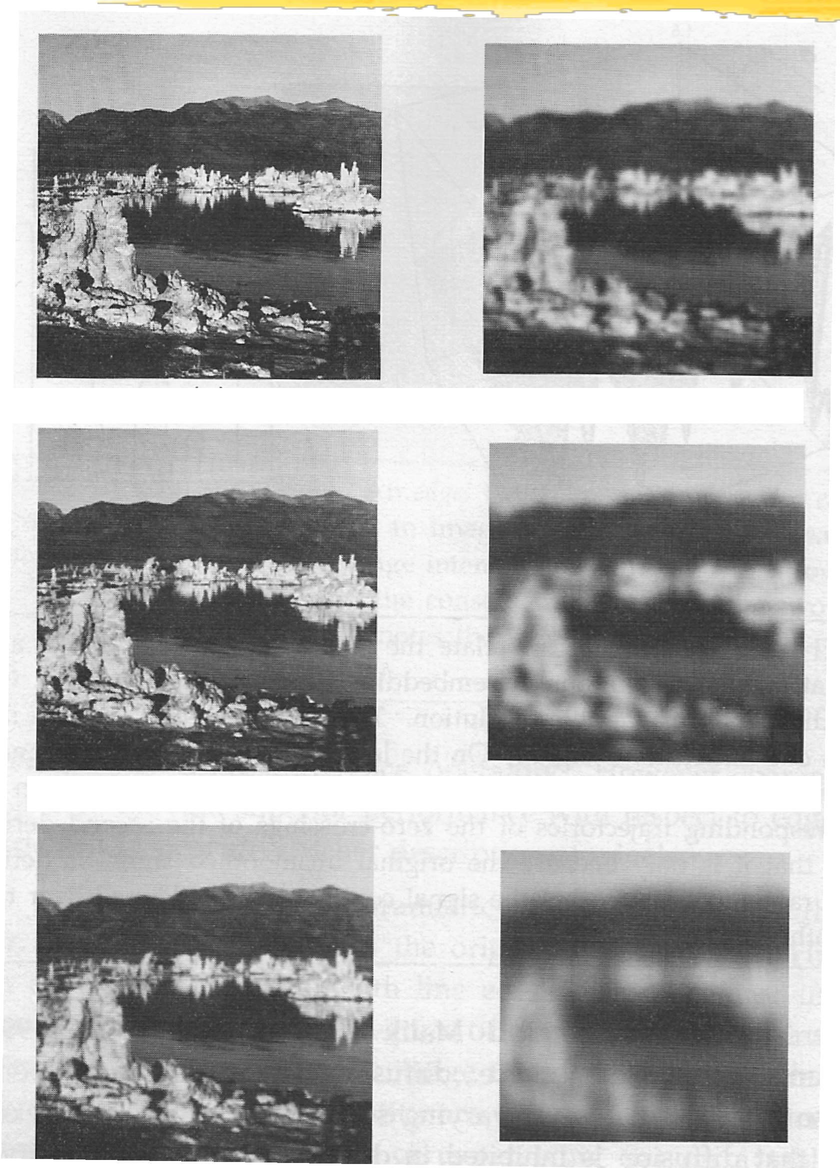
$$g_1(x) = \frac{1}{\sqrt{\pi}\sigma} \exp(-x^2/\sigma^2)$$

$$g_2(x,y) = g_1(x)g_1(y)$$

$$\int_u\int_v g_2(u,v)f(x-u,y-v)dudv = \int_u g_1(u)(\int_v g_1(v)f(x-u,y-v)dv)du$$

$$= \int_v g_1(v)(\int_u g_1(u)f(x-u,y-v)du)dv$$

SMOOTHED IMAGES



GAUSSIAN DERIVATIVES

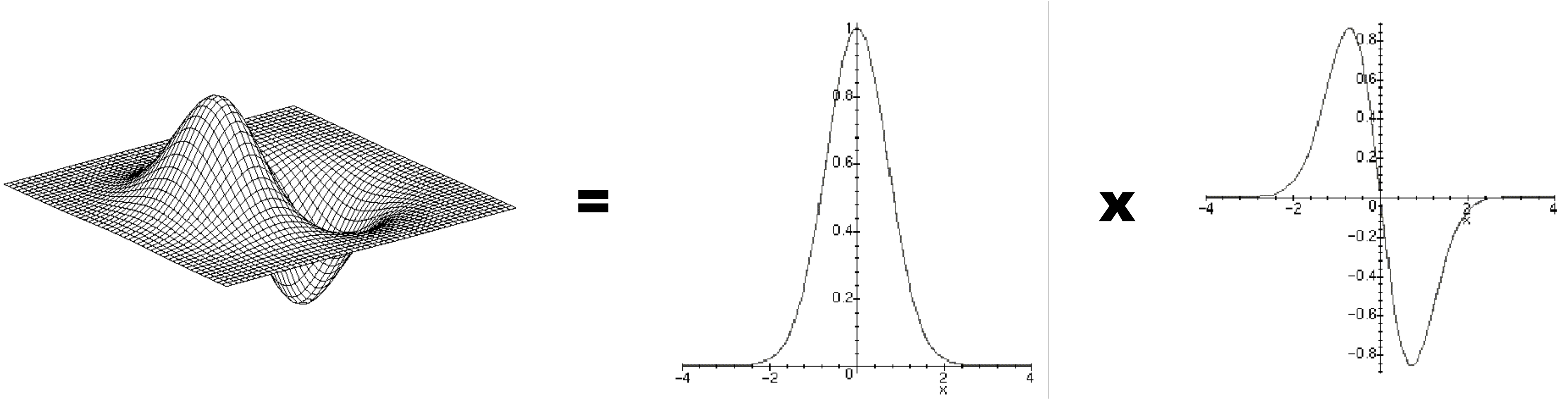
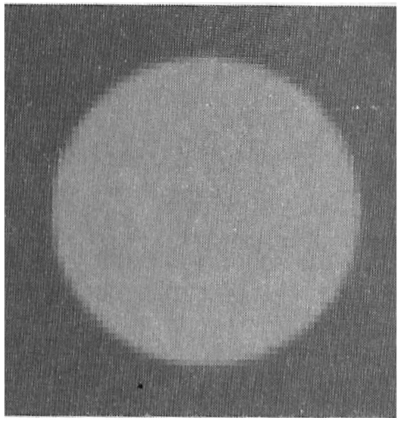


Image derivatives computed by convolving with the derivative of a Gaussian:

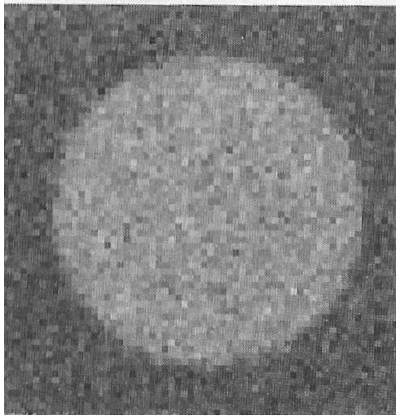
$$\frac{\partial}{\partial x}\int_u\int_v g_2(u,v)\,f(x-u,y-v)\,du\,dv = \int_u g_1^{'}(u)\left(\int_v g_1(v)\,f(x-u,y-v)\,dv\right)du$$

$$\frac{\partial}{\partial y}\int\int g_2(u,v)\,f(x-u,y-v)\,du\,dv = \int_v g_1^{'}(v)\left(\int_u g_1(u)\,f(x-u,y-v)\,du\right)dv$$

INCREASING SIGMA



No Noise

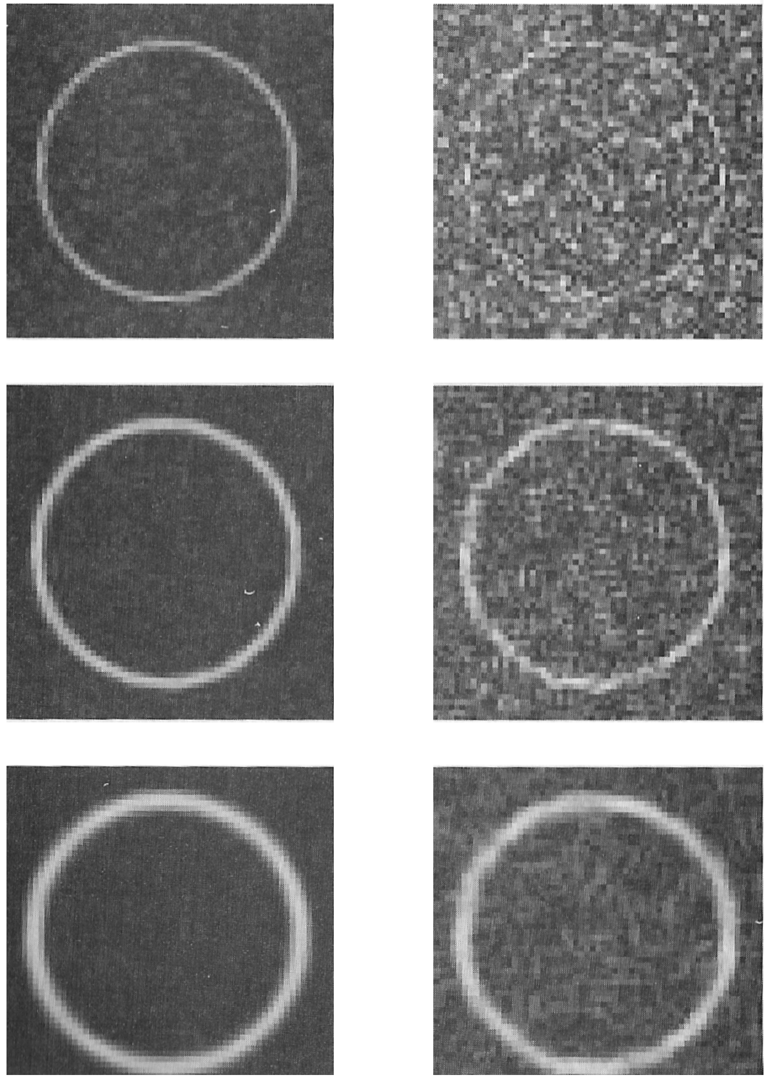


Noise Added

$\sigma=1$

$\sigma=2$

$\sigma=4$



No Noise | Noise Added

GAUSSIAN MASKS

Sigma=1:

g : 0.000070 0.010332 0.207532 0.564131 0.207532 0.010332 0.000070

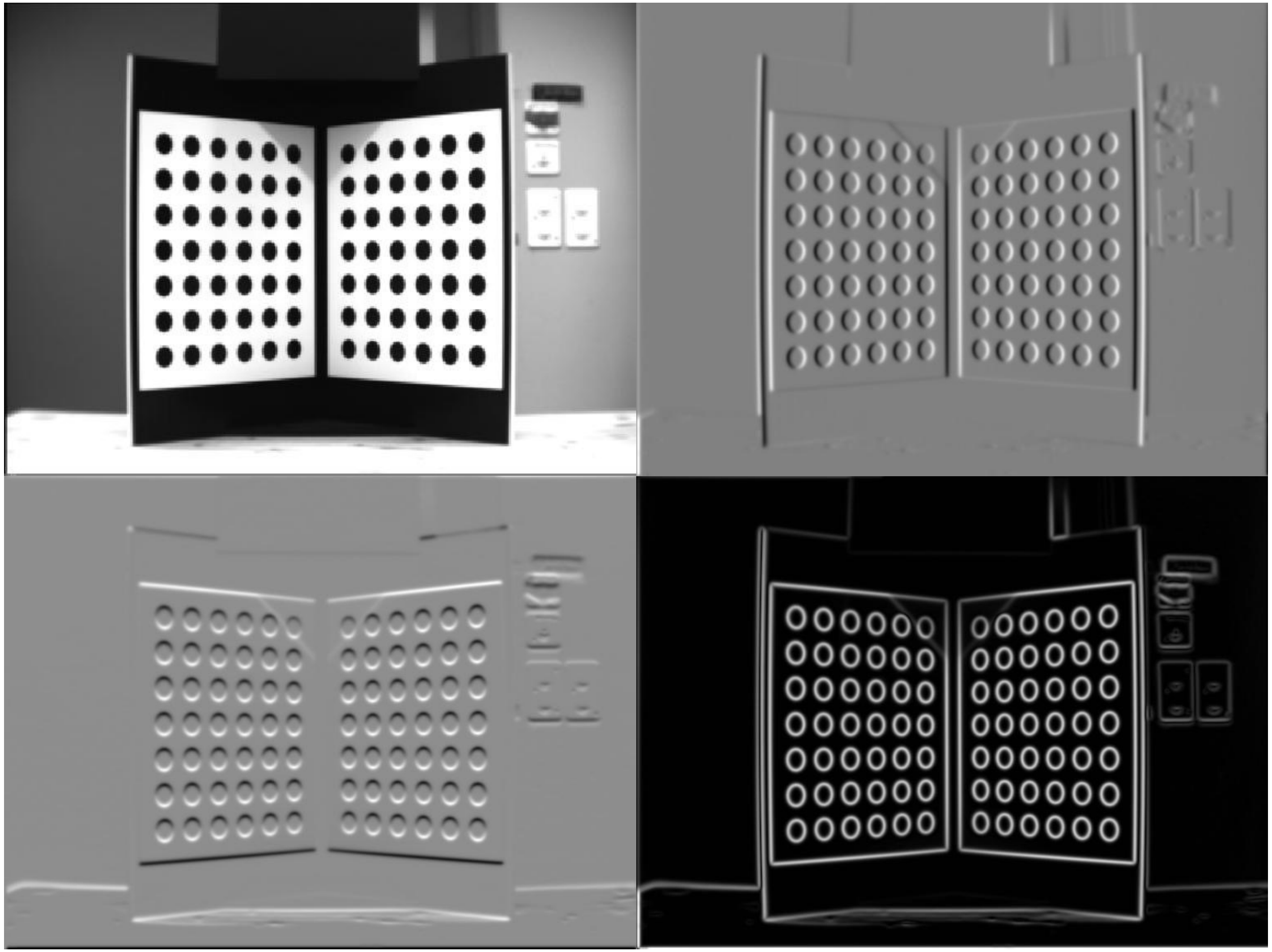
g’: 0.000418 0.041330 0.415065 0.000000 -0.415065 -0.041330 -0.000418

Sigma=2:

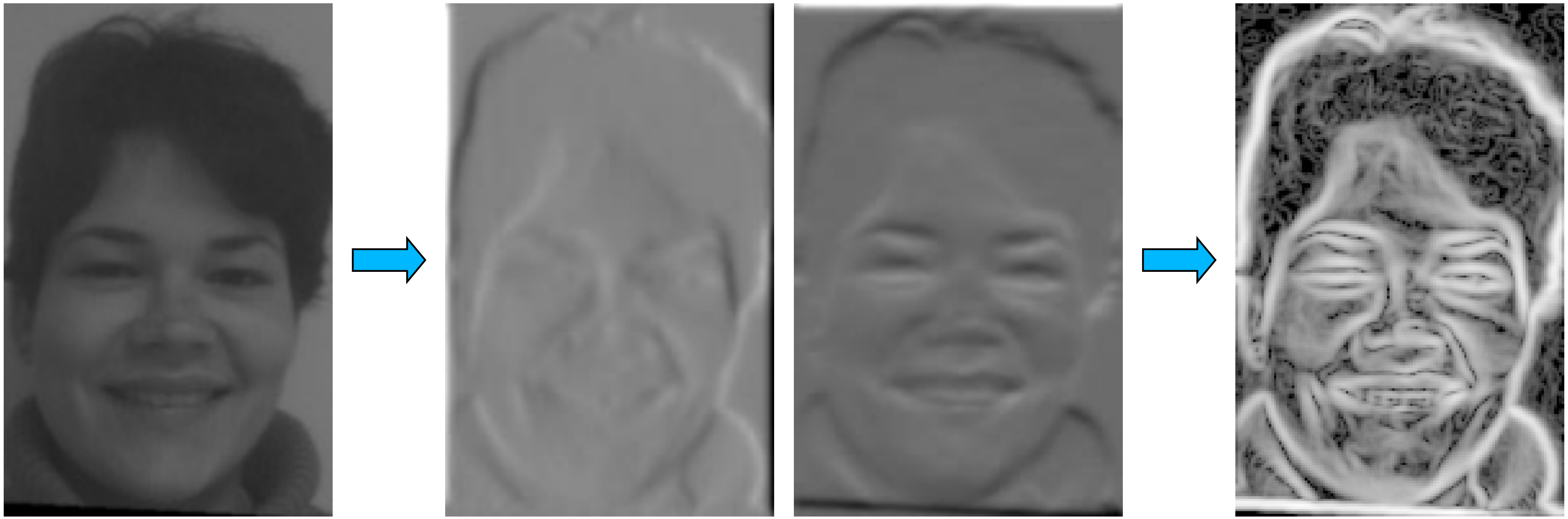
g : 0.005167 0.029735 0.103784 0.219712 0.282115 0.219712 0.103784 0.029735 0.005167

g’: 0.010334 0.044602 0.103784 0.109856 0.000000 -0.109856 -0.103784 -0.044602 -0.010334

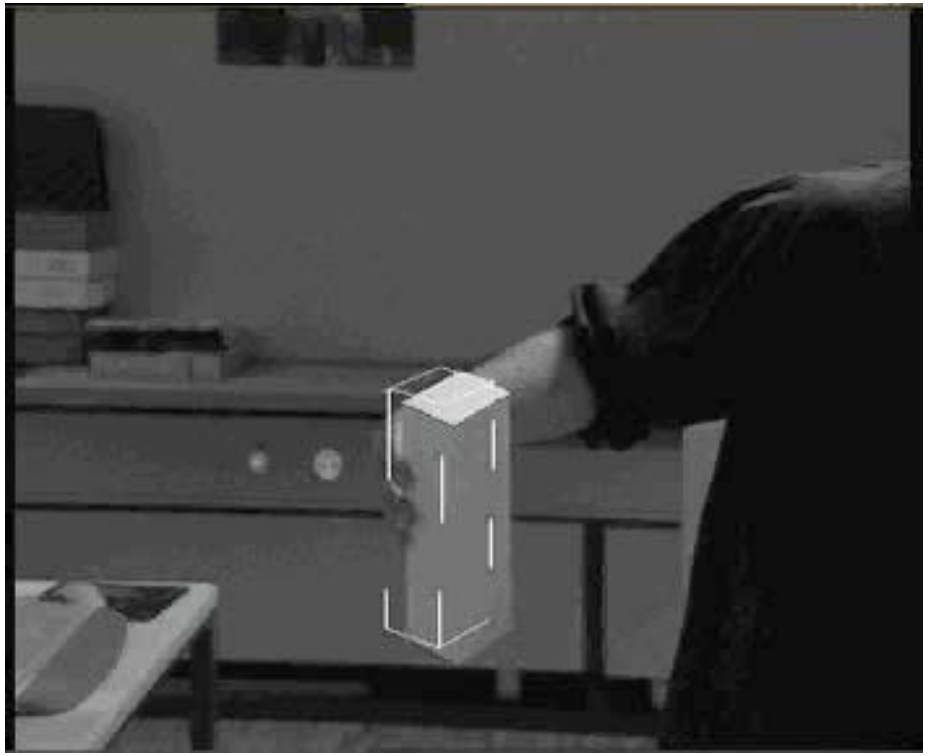
DERIVATIVE IMAGES



DERIVATIVE IMAGES

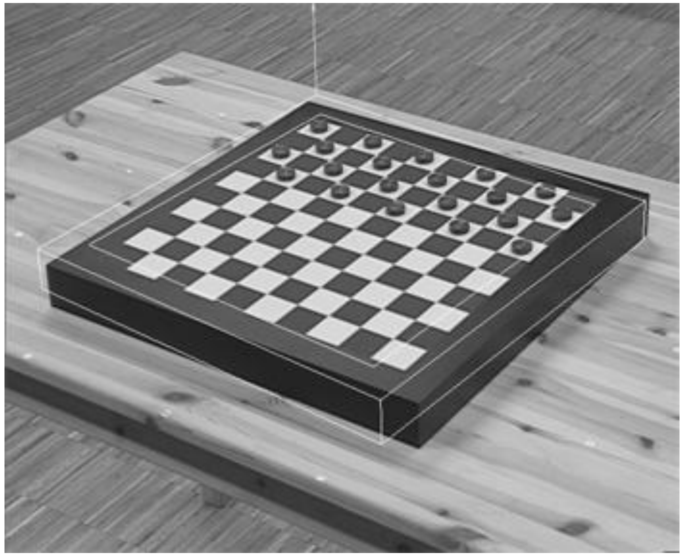


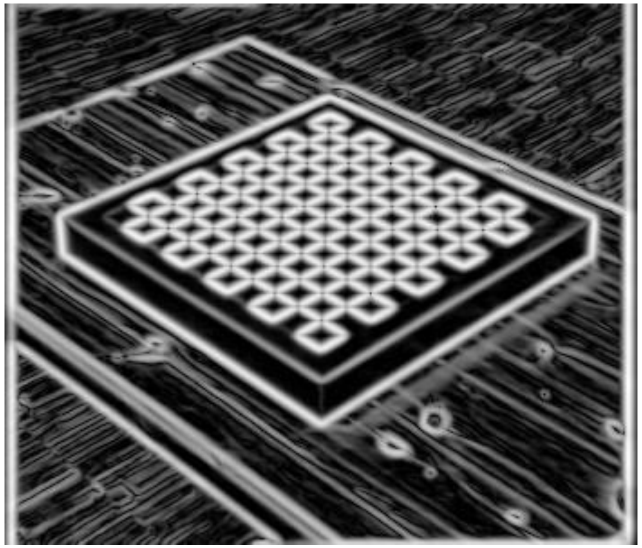
GRADIENT-BASED TRACKING

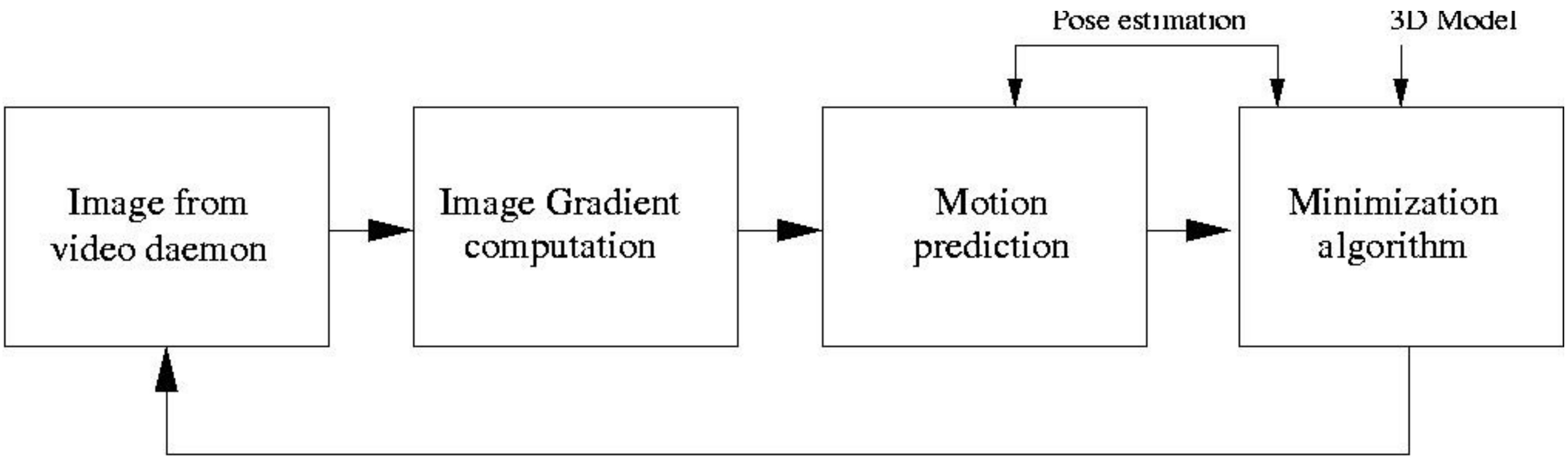


Maximize edge-strength along projection of the 3—D wireframe.

GRADIENT MAXIMIZATION







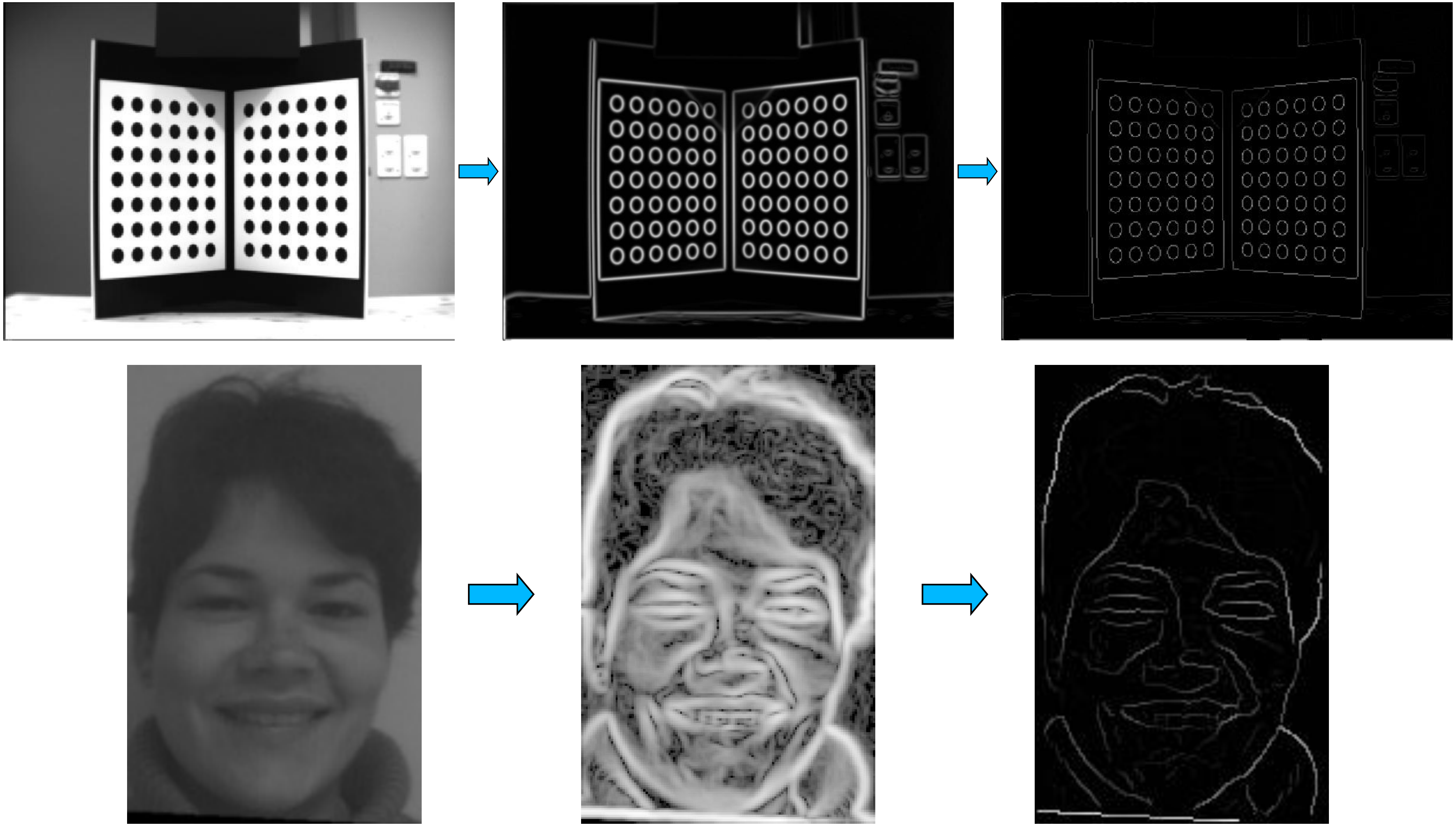
WING DEFORMATION

- Measure true behavior in flight.
- Validate computational models.

REAL-TIME TRACKING

CANNY EDGE DETECTOR



CANNY EDGE DETECTOR

Convolution

- Gradient strength
- Gradient direction

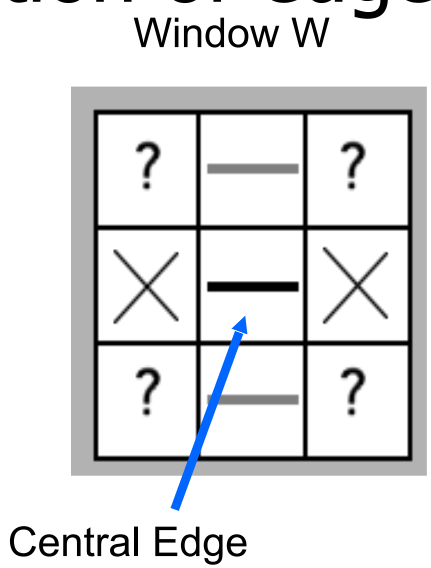
Thresholding

Non Maxima Suppression

Hysteresis Thresholding

NON-MAXIMA SUPPRESSION

In parallel, at each pixel in edge image, use window to select as a function of edge orientation:

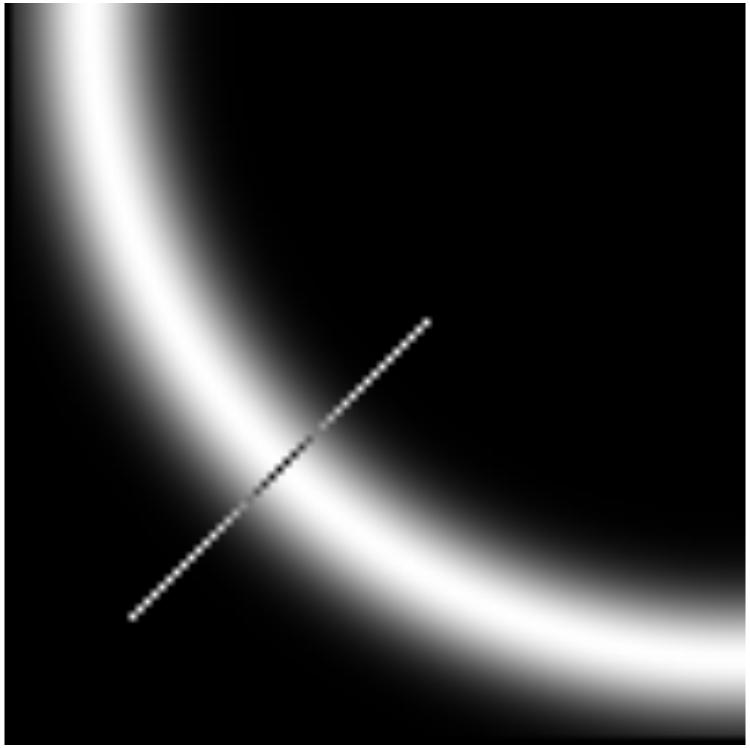


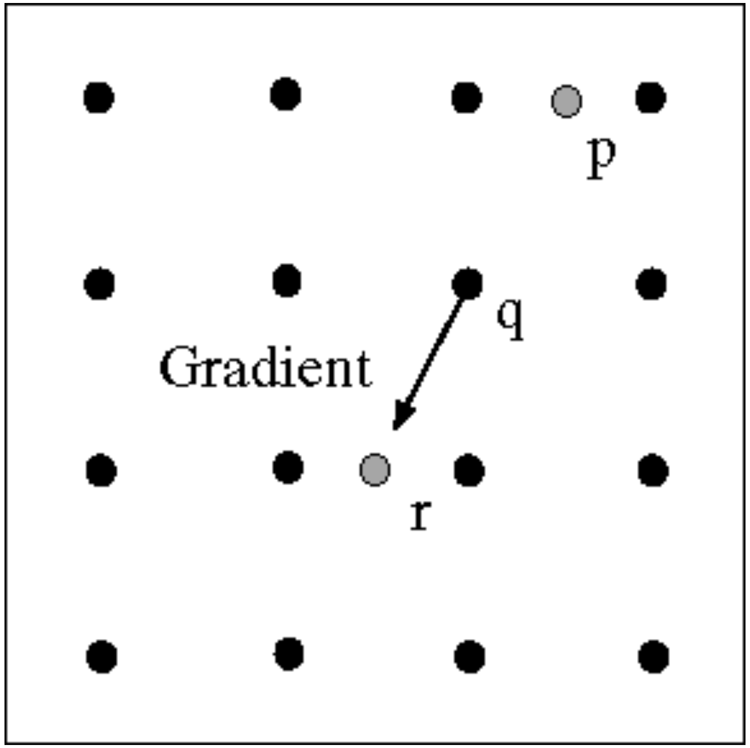
-- Definitely consider these edges

X Do not consider these edges

? Maybe consider them, depending on algorithm

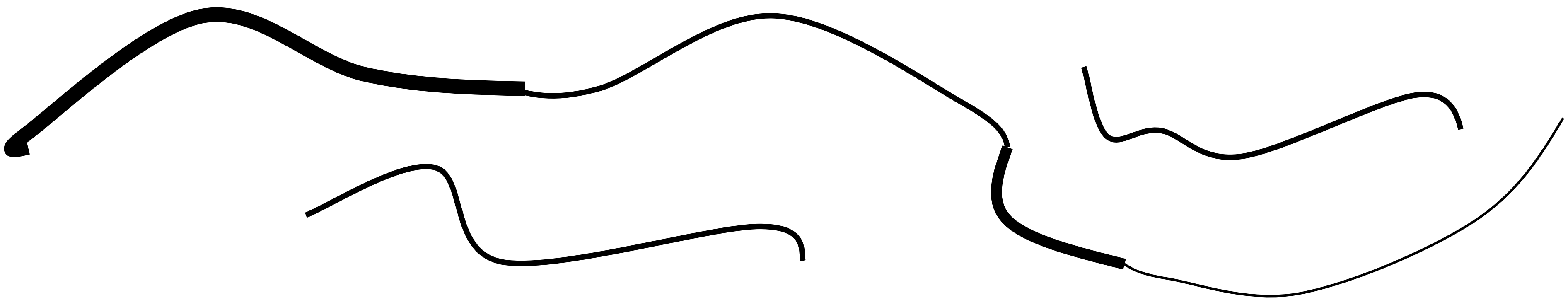
NON-MAXIMA SUPPRESSION





Check if pixel is local maximum along gradient direction, which requires checking interpolated pixels p and r.

HYSTERESIS THRESHOLDING



Algorithm takes two thresholds: high & low

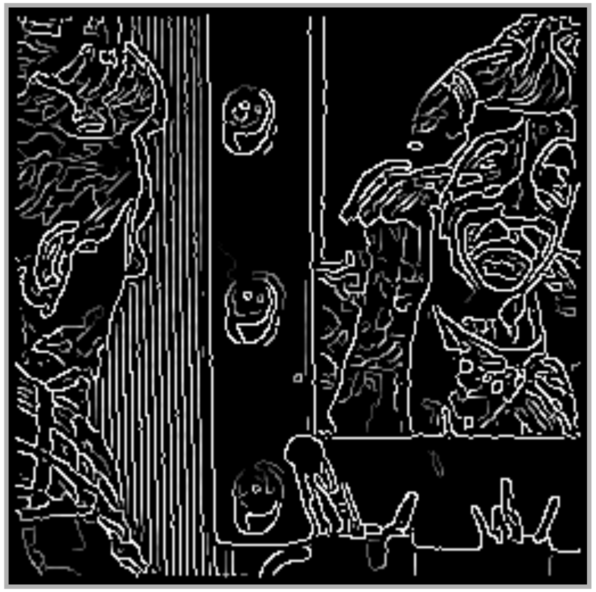
- A pixel with edge strength above high threshold is an edge.
- Any pixel with edge strength below low threshold is not.
- Any pixel above the low threshold and next to an edge is an edge.

Iteratively label edges

- Edges grow out from 'strong edges'
- Iterate until no change in image.

CANNY RESULTS

σ=1, T2=255, T1=220



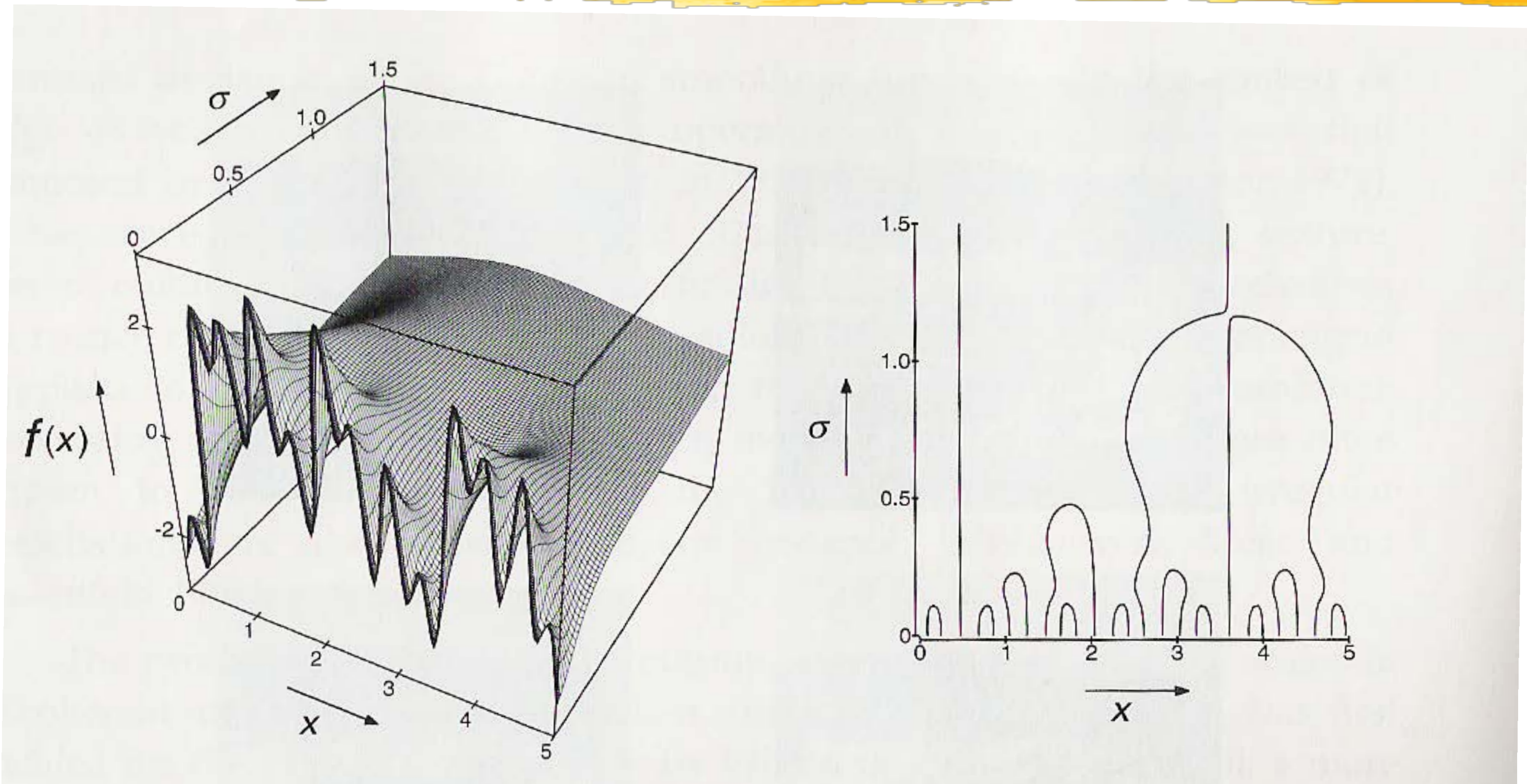
σ=1, T2=128, T1=1

σ=2, T2=128, T1=1

M. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 12, December 1997, pp. 1338-1359.

http://marathon.csee.usf.edu/edge/edge_detection.html

SCALE SPACE



Increasing scale (σ) removes details but never adds new ones:

- Edge position may shift.
- Two edges may merge.
- An edge may **not** split into two.

MULTIPLE SCALES

$\sigma = 1$

$\sigma = 2$

$\sigma = 4$

→Choosing the right scale is a difficult semantic problem.

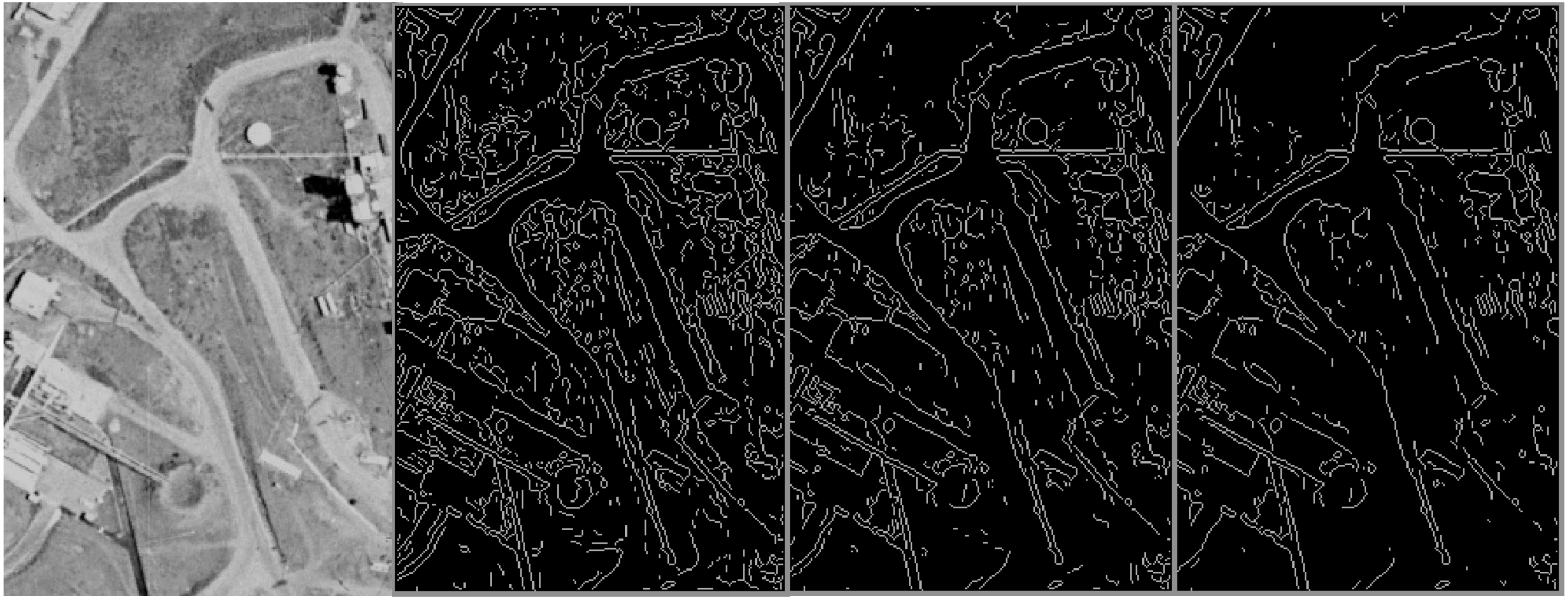
SCALE vs THRESHOLD

Fine scale
High threshold

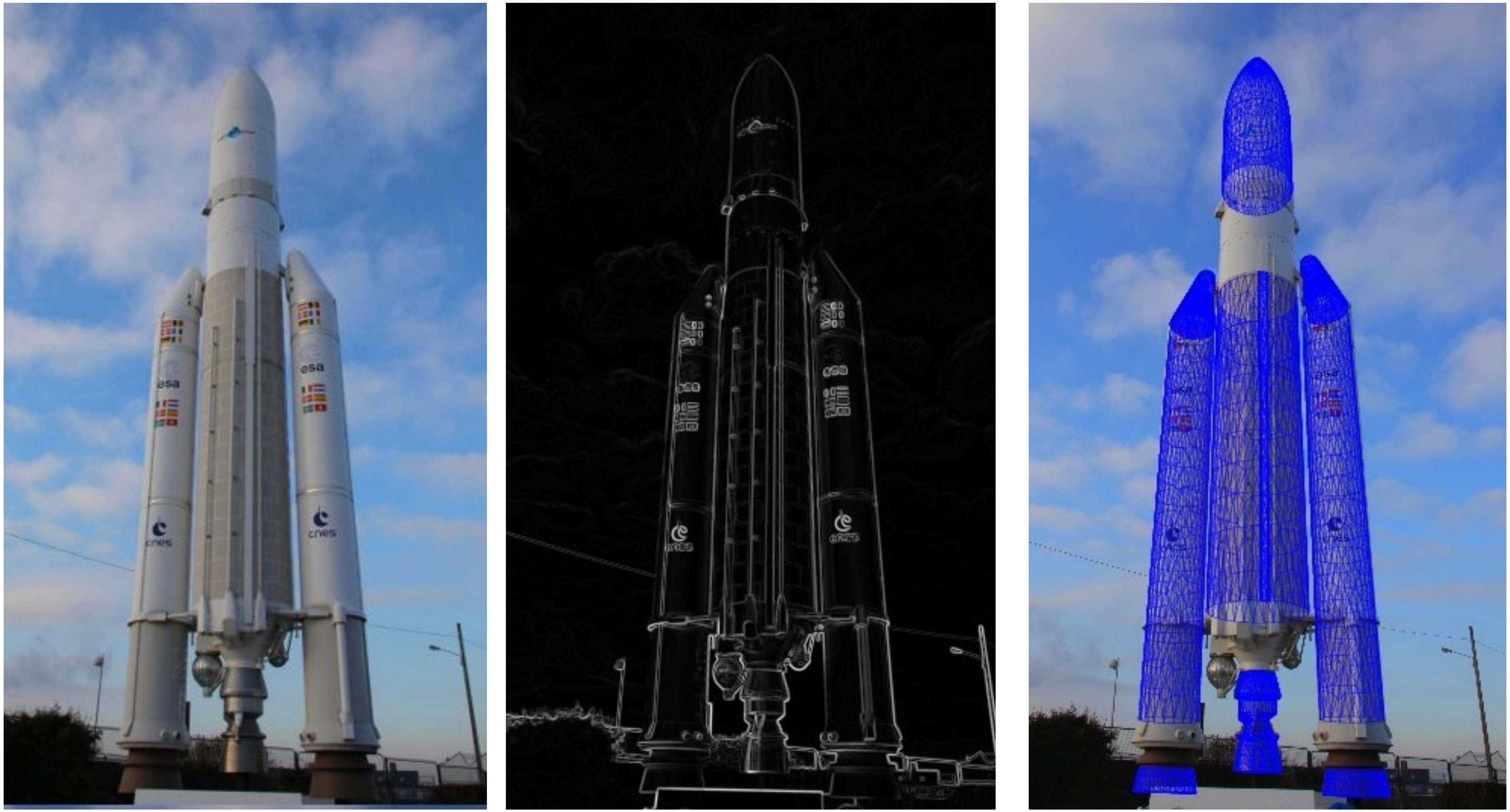
Coarse scale
High threshold

Coarse scale
Low threshold

ROAD IMAGE



TRACKING ARIANE

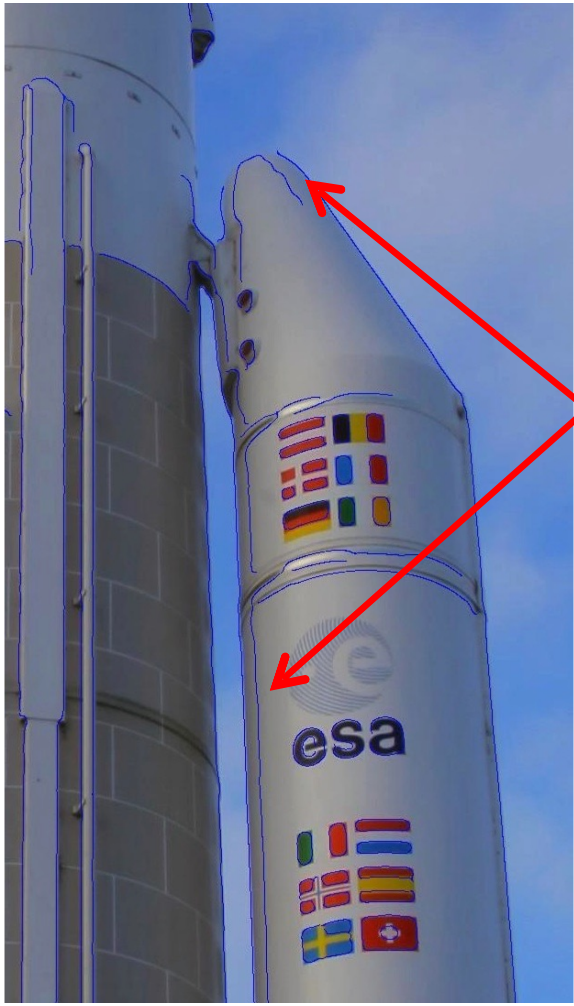


RAPID TRACKER

Given an initial pose estimate:

- Estimate the occluding contours.
- Find closest edge points in the normal direction.
- Re-estimate pose to minimize distances in the least squares sense.
- Iterate until convergence.

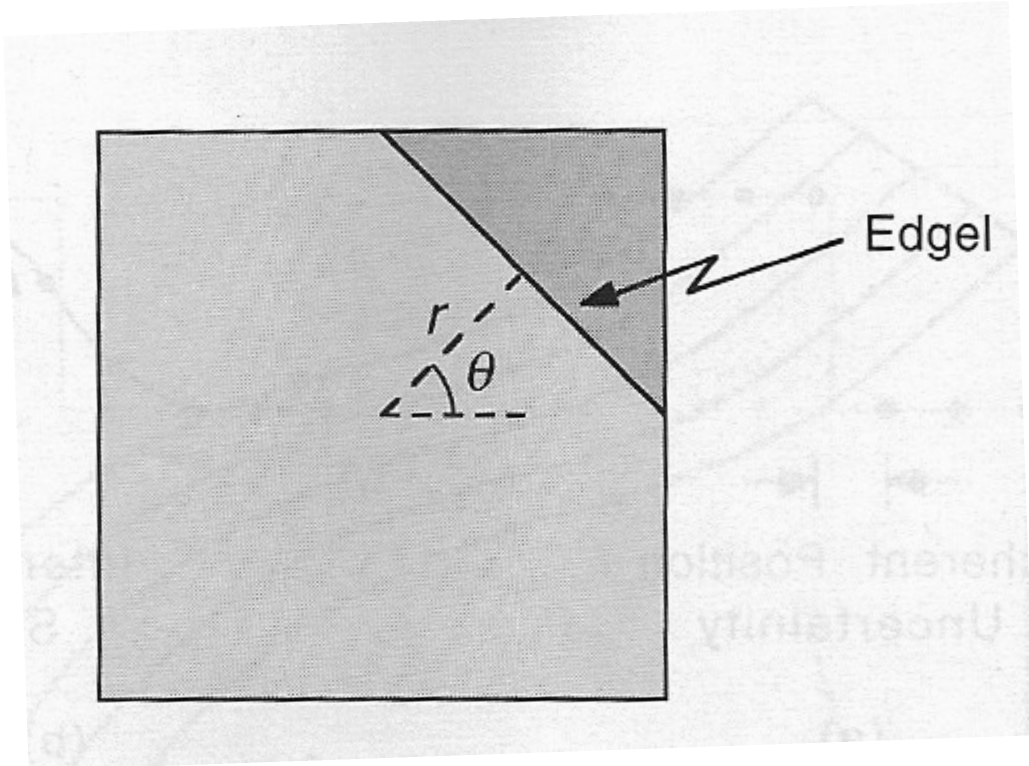
STEEP SMOOTH SHADING



- Rapidly varying gray levels.
- Large gradients.

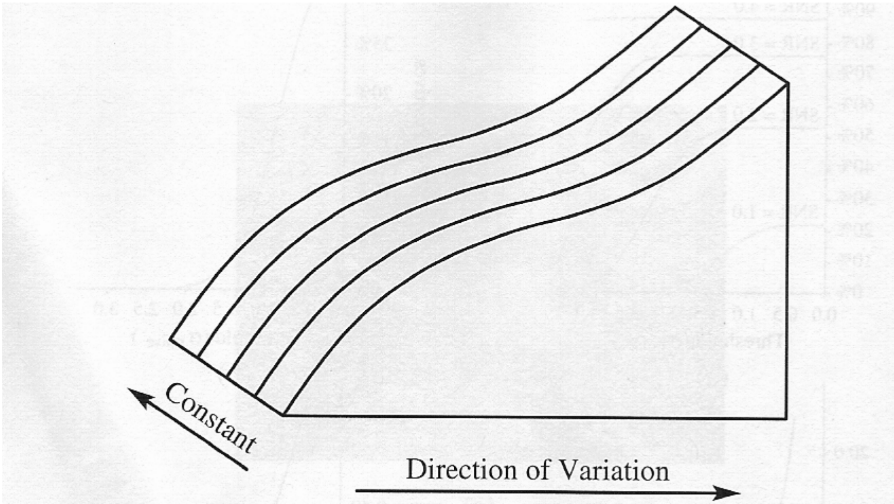
→ Shading can produce spurious edges.

PARAMETRIC MODEL MATCHERS



→ 4 parameters model to be fit in the least squares sense.

SURFACE FITTING

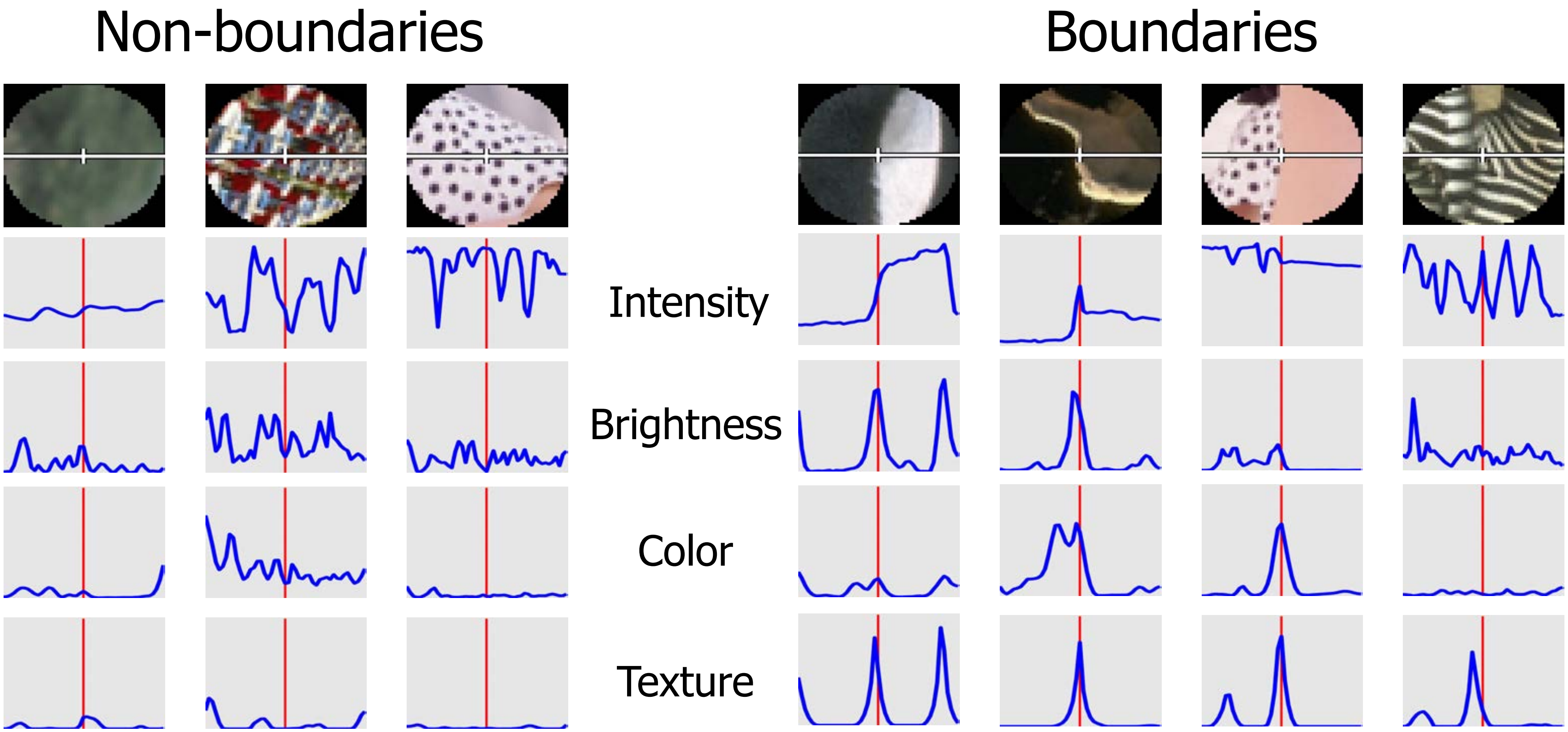


1. Estimate the edgel direction by fitting a cubic surface.
2. Fit a 1-D surface in the direction of the edgel
   - Step shaped surface,
   - Quadratic polynomial.
3. Declare an edge if step shape better than quadratic.

TEXTURE BOUNDARIES

- Not all image contours are characterized by strong contrast.
- Sometimes, textural changes are just as significant.

DIFFERENT BOUNDARY TYPES

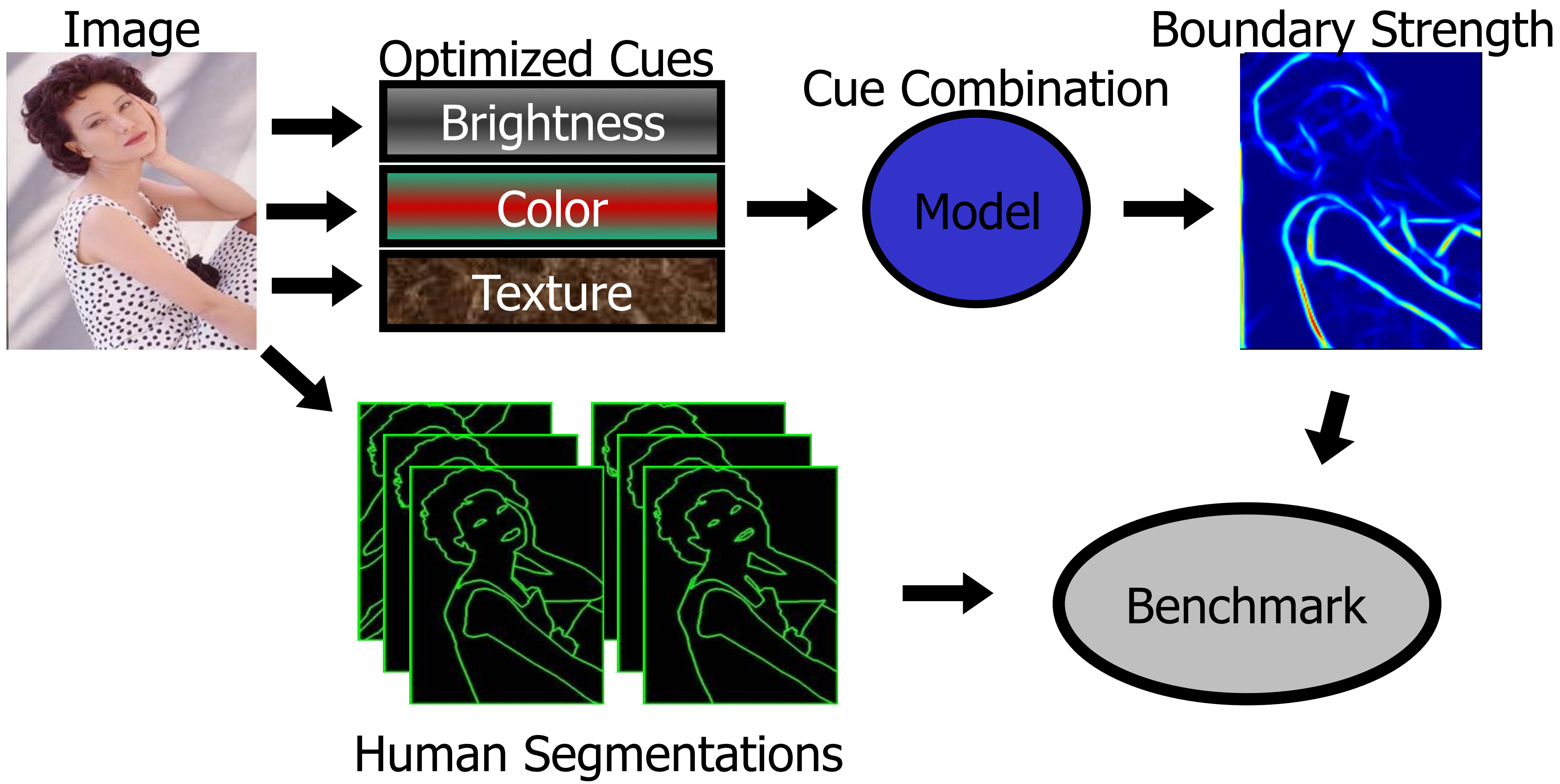


Martin et al. , PAMI'04

TRAINING DATABASE

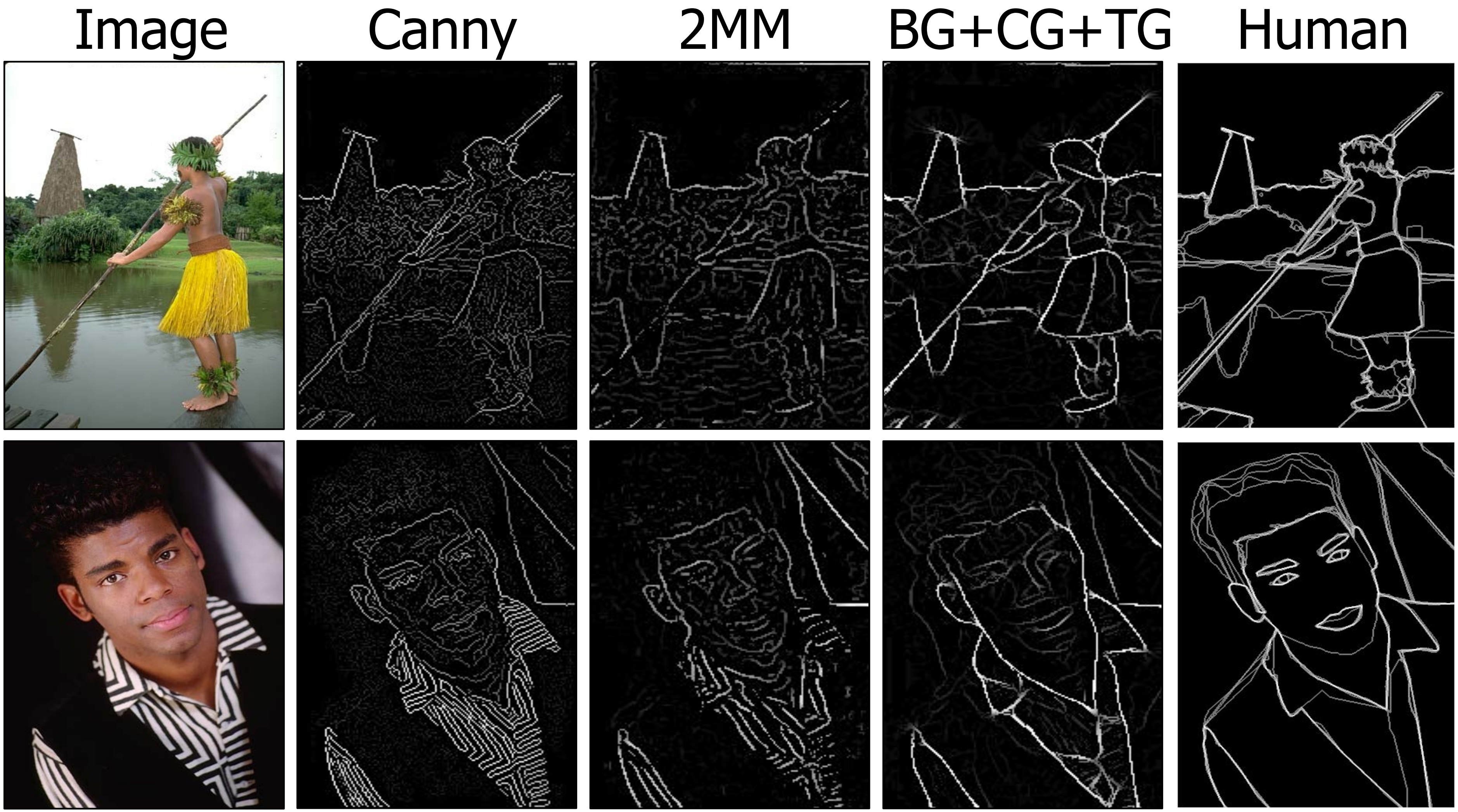
1000 images with 5 to 10 segmentations each.

MACHINE LEARNING



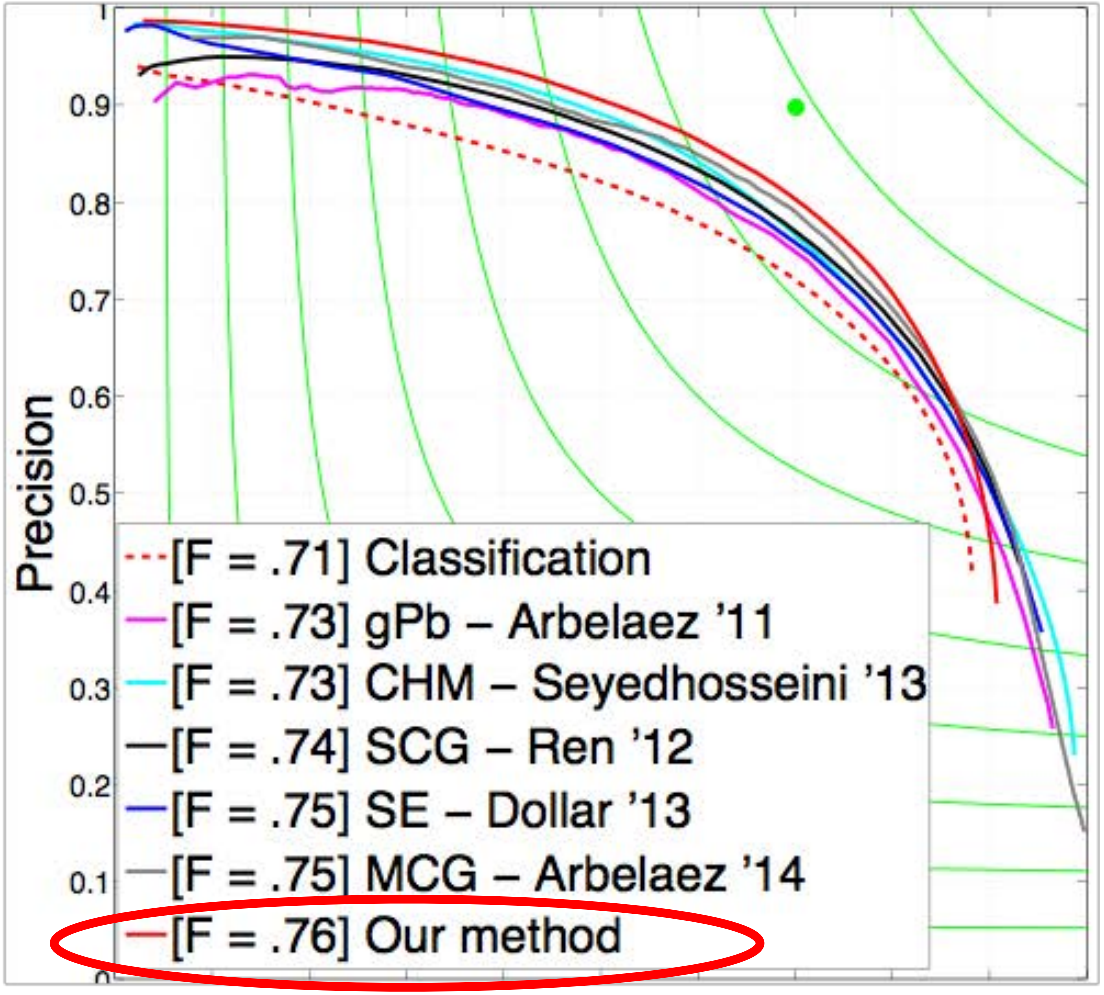
Learn the probability of being a boundary pixel on the basis of a set of features.

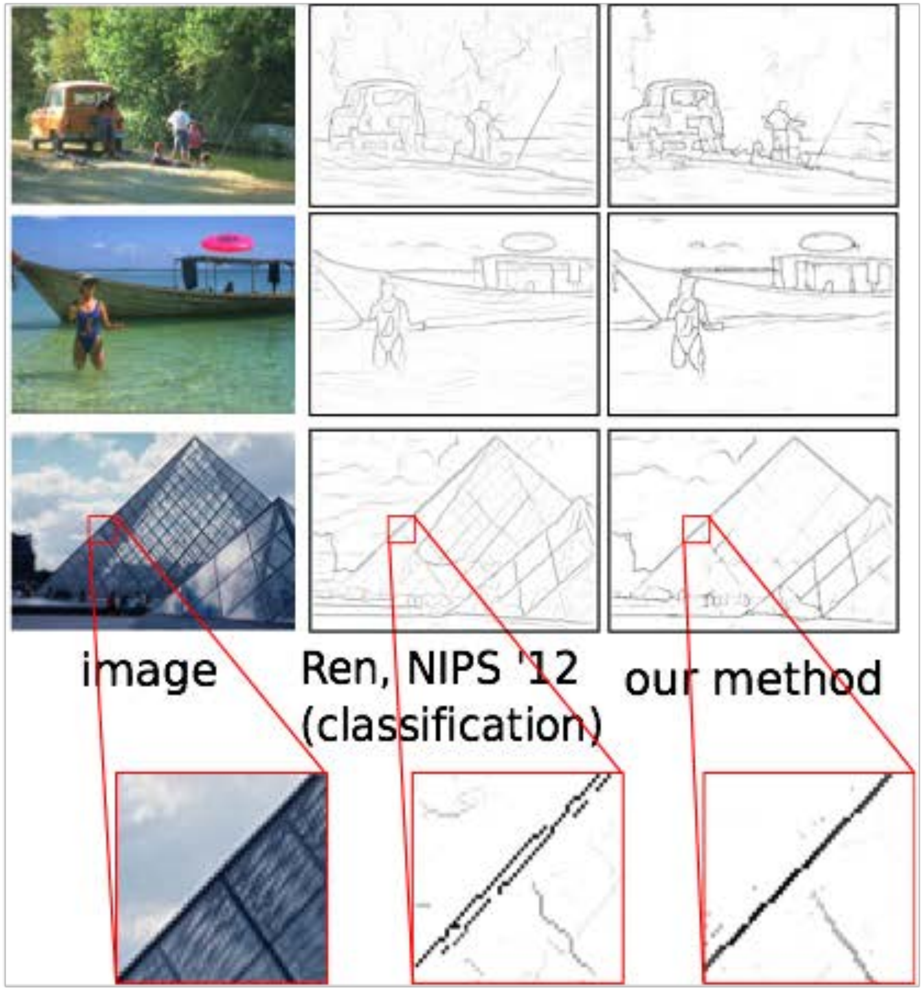
RESULTS



Martin et al. , PAMI'04

CLASSIFICATION vs REGRESSION

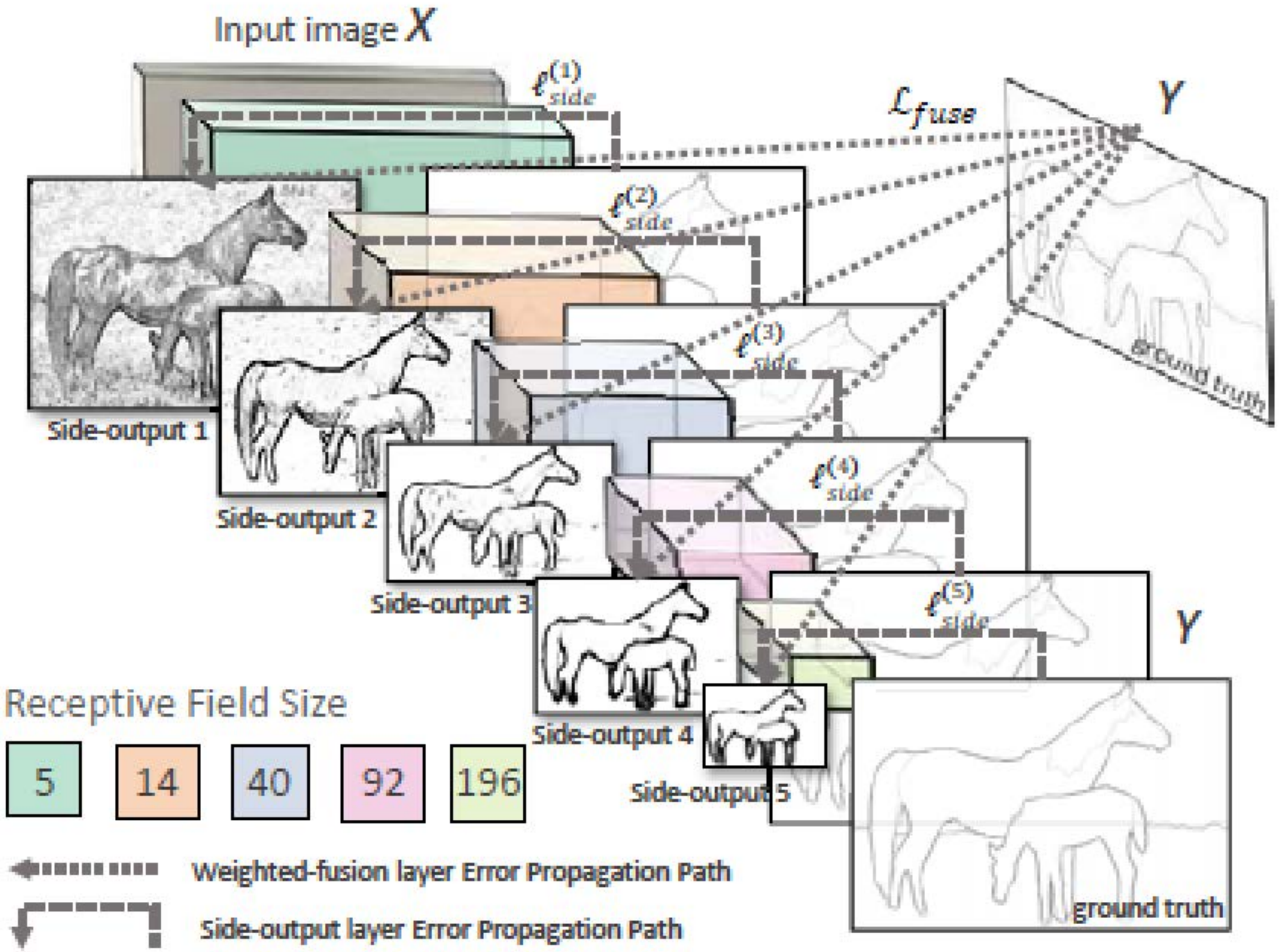




Yes!

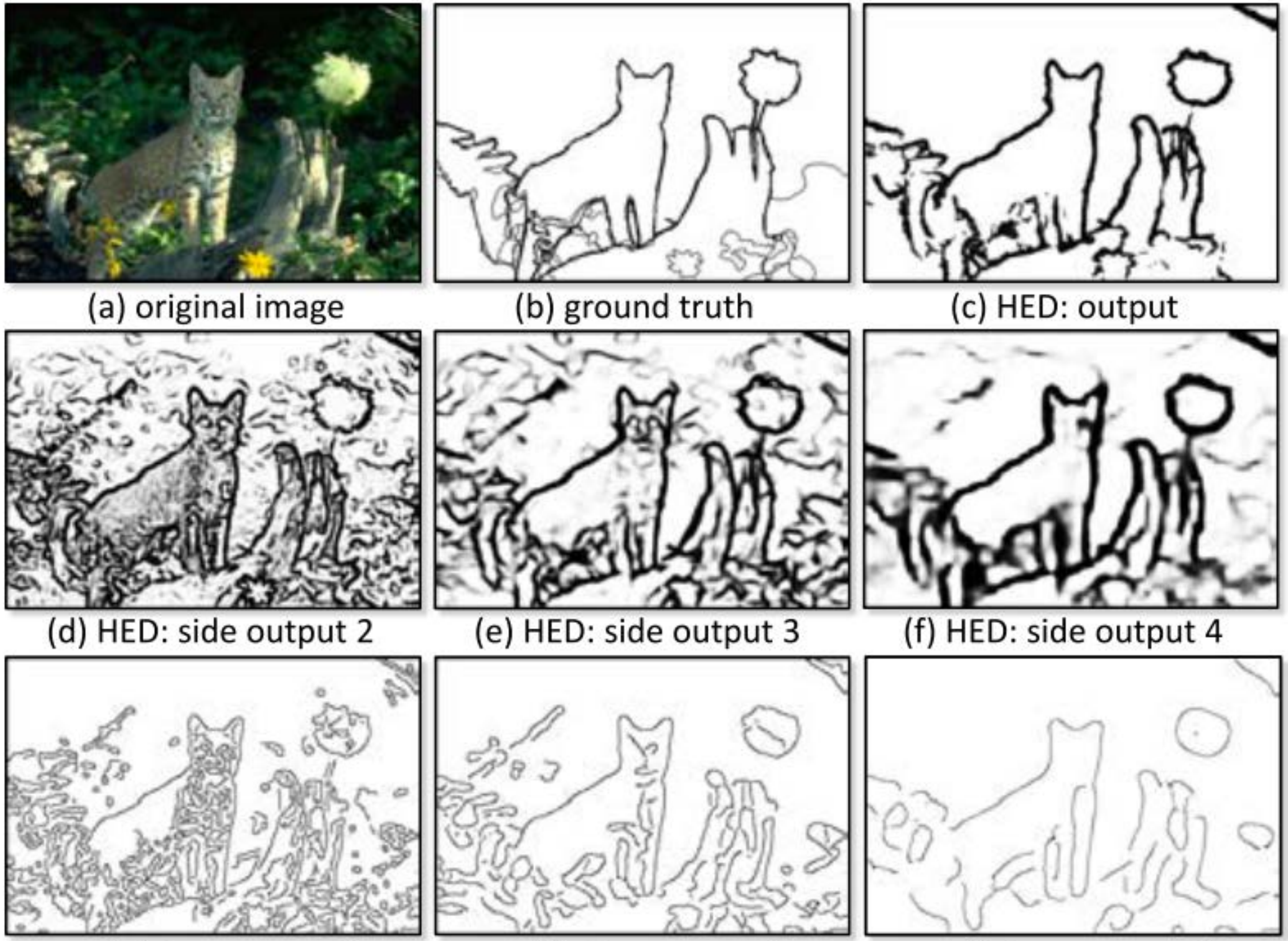
Sironi, et al. PAMI'15

DEEP LEARNING



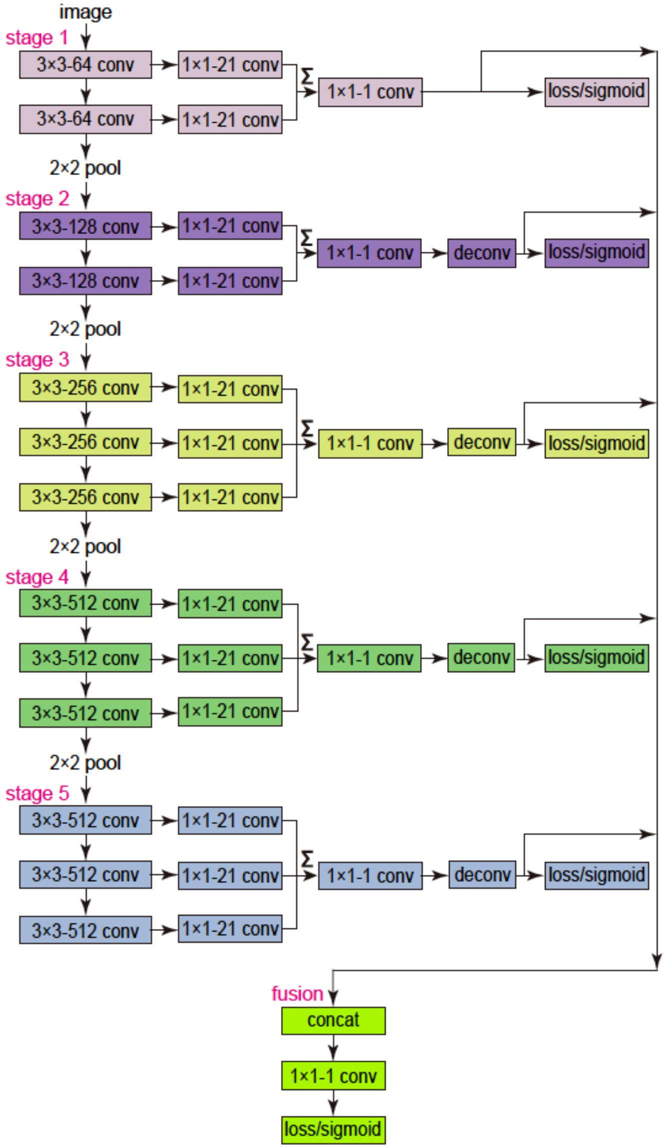
Xie and Tu, ICCV'15

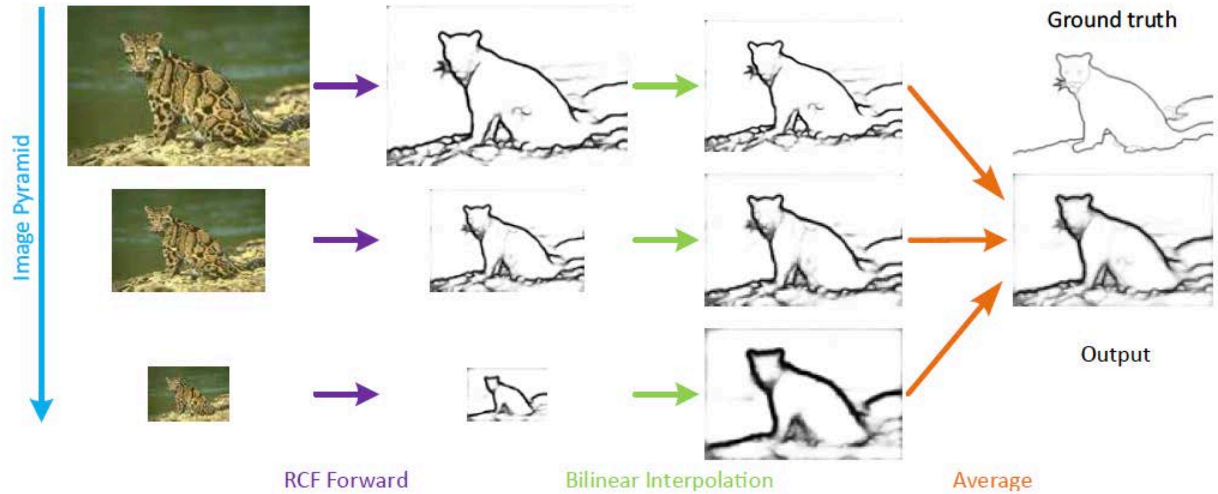
DEEP LEARNING VS CANNY



(a) original image (b) ground truth (c) HED: output
(d) HED: side output 2 (e) HED: side output 3 (f) HED: side output 4
(g) Canny: $\sigma = 2$ (h) Canny: $\sigma = 4$ (i) Canny: $\sigma = 8$

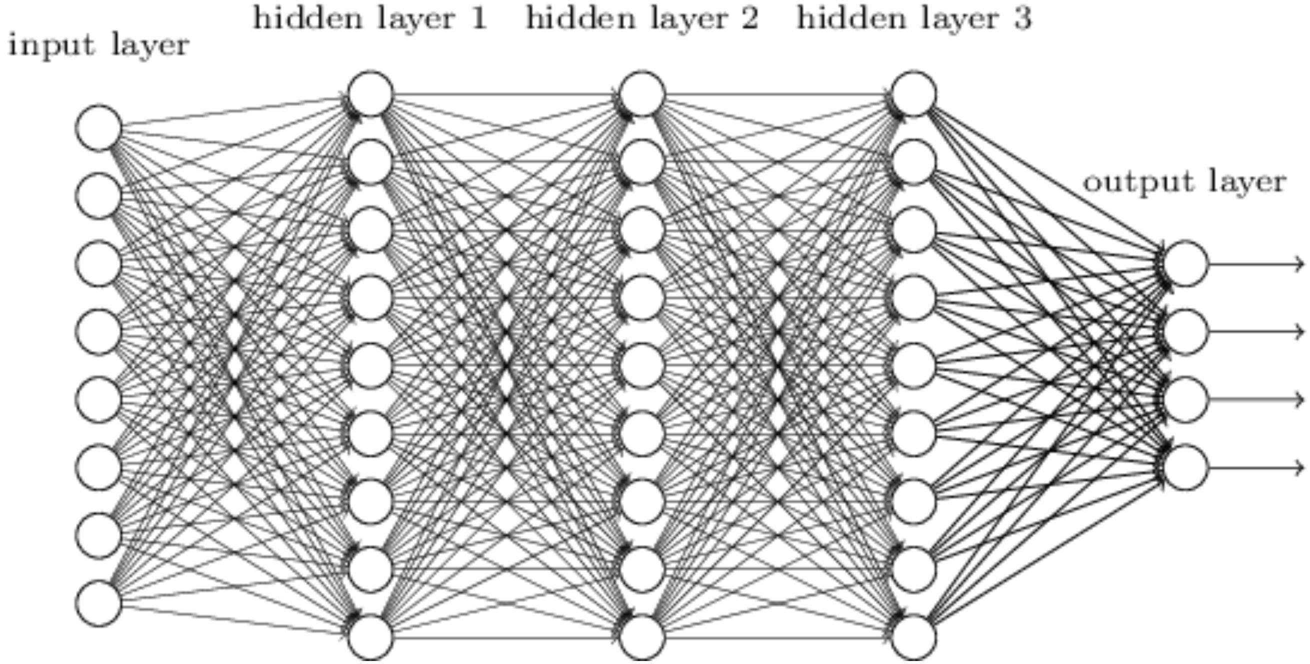
DEEPER LEARNING





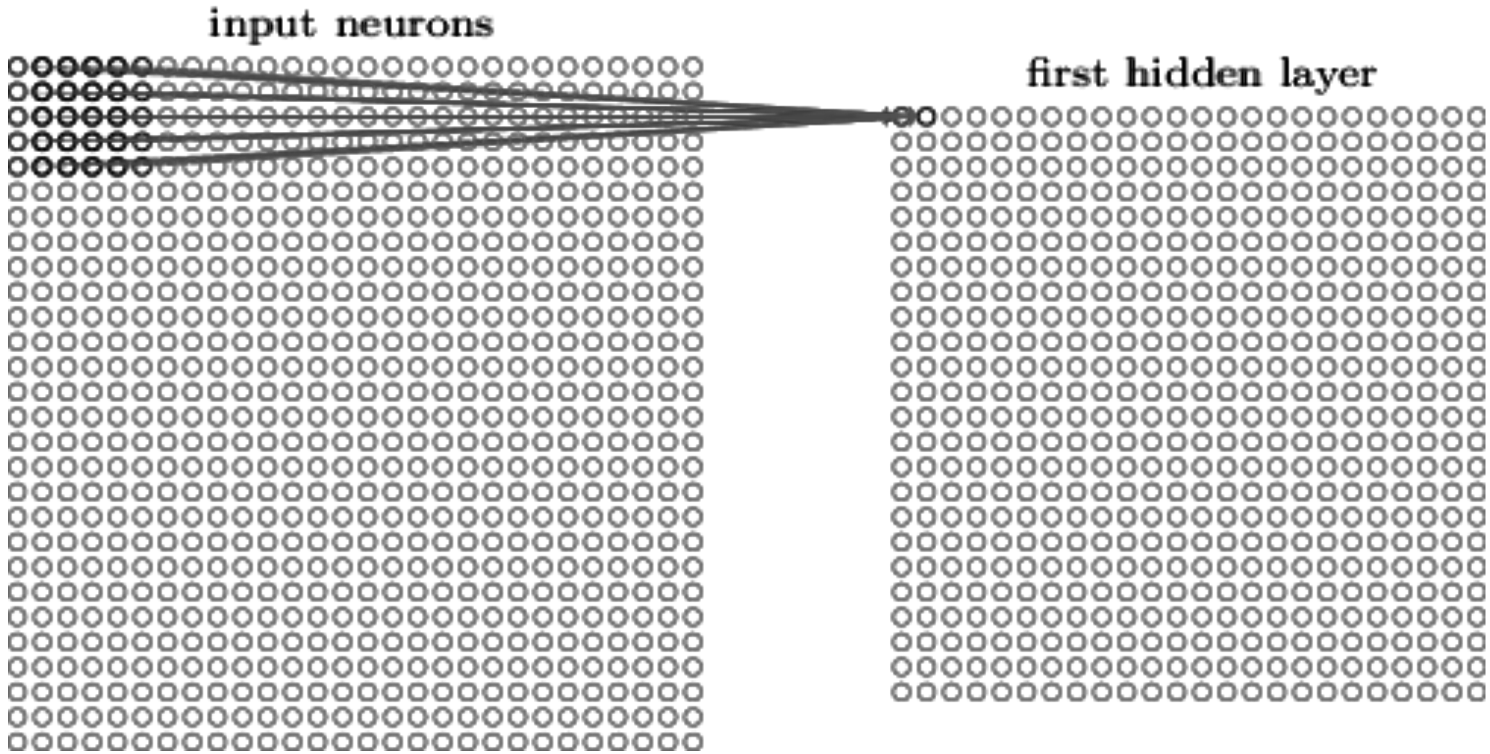
Liu et al. , CVPR'17

DEEP LEARNING



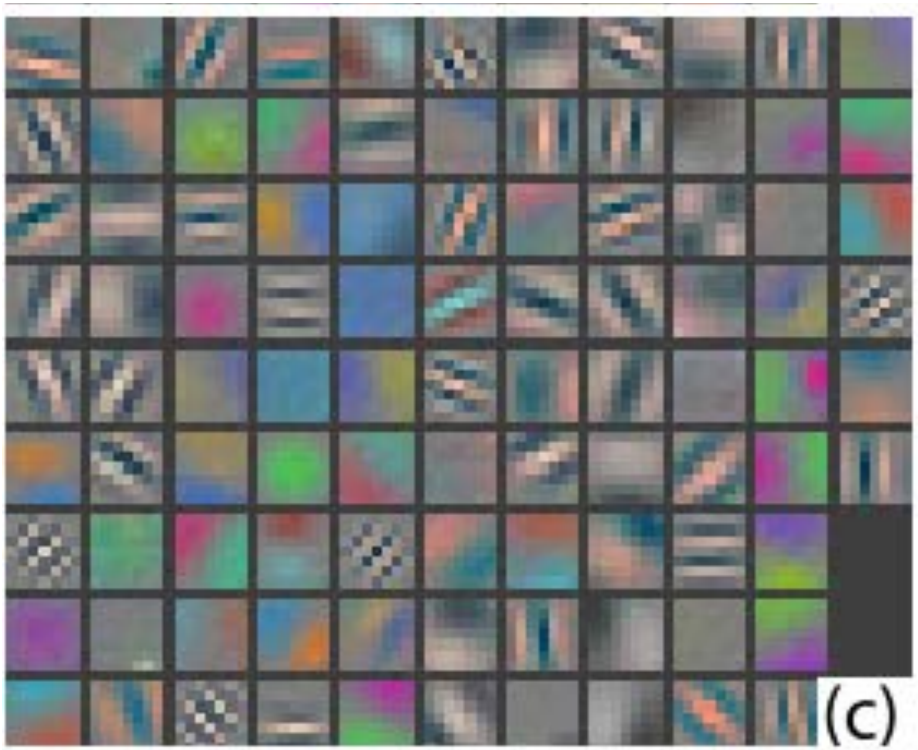
- Store in each node a function of the linear combination of the activations of all nodes in the previous layers.
- The network can be trained to produce a desired output given a specific input by learning the linear combination weights.

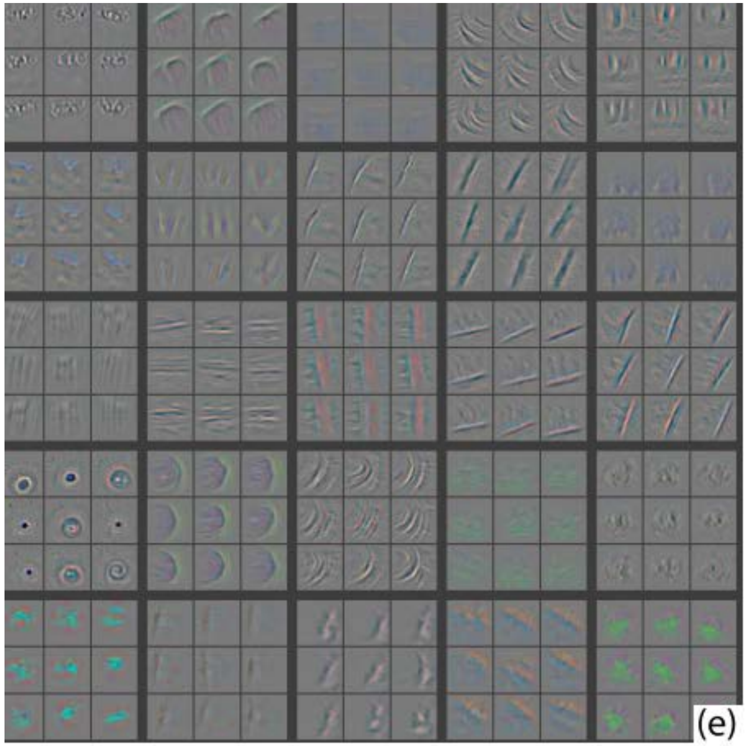
CONVOLUTIONAL LAYER



$$\sigma\left(b + \sum_{x=0}^{n_x} \sum_{y=0}^{n_y} w_{i,j} a_{i+x,j+y}\right)$$

A PARTIAL EXPLANATION?

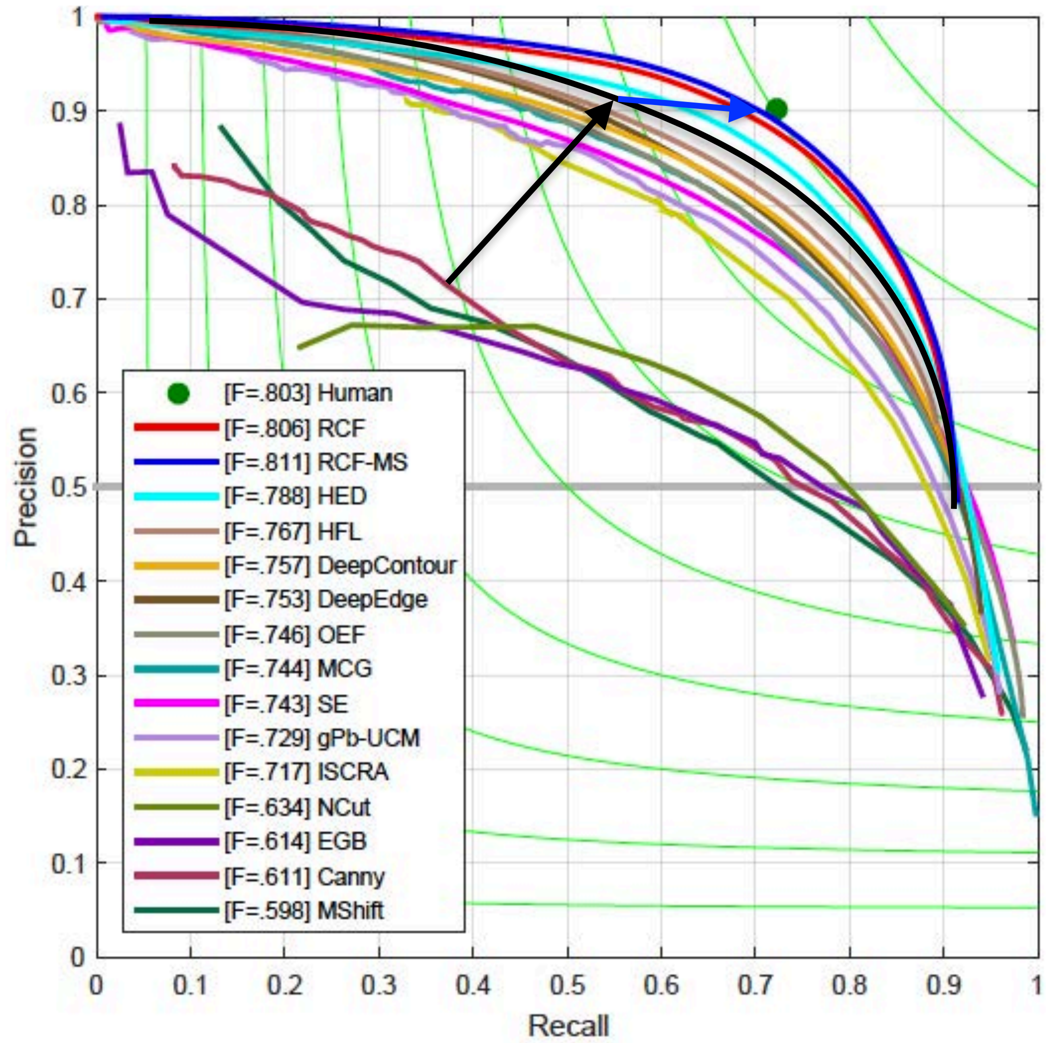




First and second layer features of a Convolutional Neural Net:

- They can be understood as performing multiscale filtering.
- The weights and thresholds are chosen by the optimization procedure.

50 YEARS OF EDGE DETECTION



- Convolution operators respond to steep smooth shading.
- Parametric matchers tend to reject non ideal edges.
- Arbitrary thresholds and scale sizes are required.
- Learning-based methods need exhaustive databases.

- There still is work to go from contours to objects.

Canny, PAMI'86 —> Sironi et al. PAMI'15 Sironi et al. PAMI'15 —> Liu et al. , CVPR'17