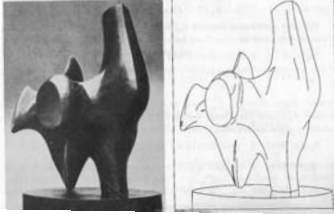


EDGE DETECTION



- What's an edge
- Edge Operators
- Edge Aggregation and Linking

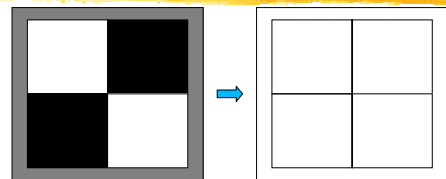
CORRIDOR



CORRIDOR



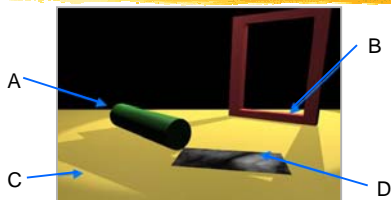
EDGES AND REGIONS



Edges: Boundary between bland image regions

Regions: Areas without edge
→ Duality Edge/Regions.

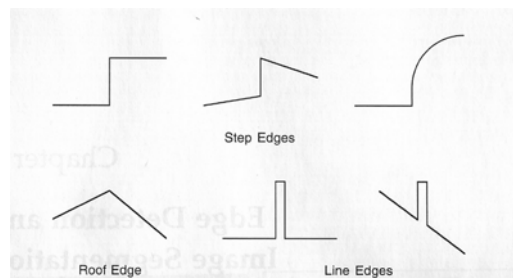
DISCONTINUITIES



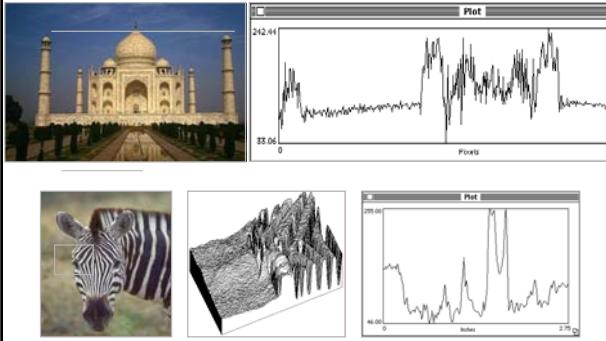
- A. Depth discontinuity: Abrupt depth change in the world
 B. Surface normal discontinuity: Change in surface orientation
 C. Illumination discontinuity: Shadows, lighting changes
 D. Reflectance discontinuity: Surface properties, markings

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

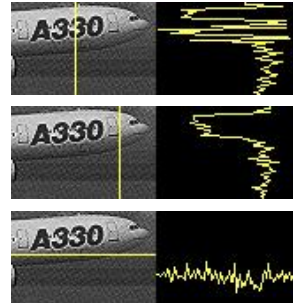
EDGE PROFILES



REALITY

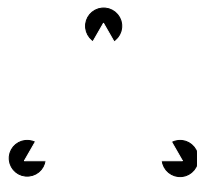


MORE REALITY



Very noisy signals
→ Much knowledge
is required!!

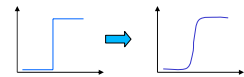
ILLUSORY CONTOURS



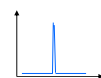
No closed contour, but

IDEAL STEP EDGE

$f(x) = \text{step edge}$

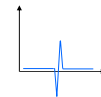


1st Derivative $f'(x)$



maximum

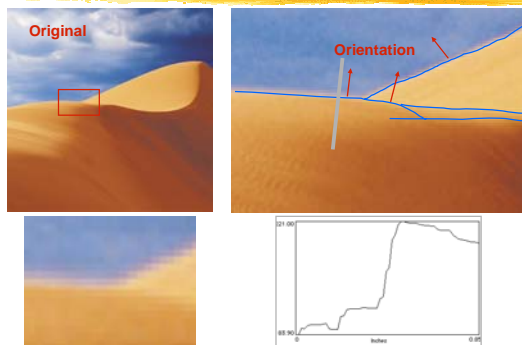
2nd Derivative $f''(x)$



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:

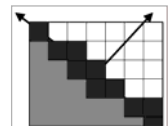
- Unit vector perpendicular to the edge normal

Edge position or center

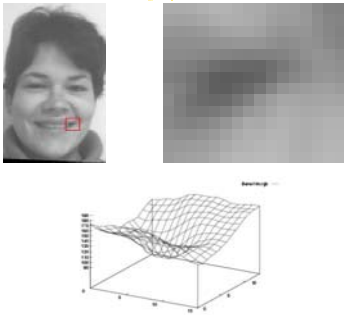
- Image position at which edge is located

Edge Strength

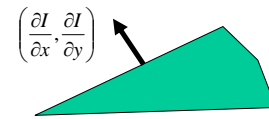
- Speed of intensity variation across the edge.



IMAGES AS 3-D SURFACES



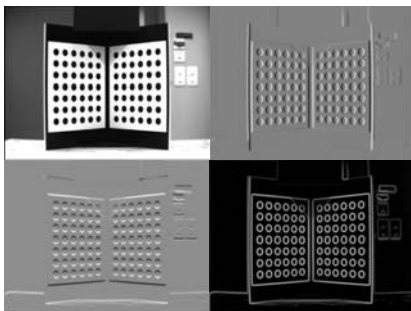
GRADIENT AND ORIENTATION



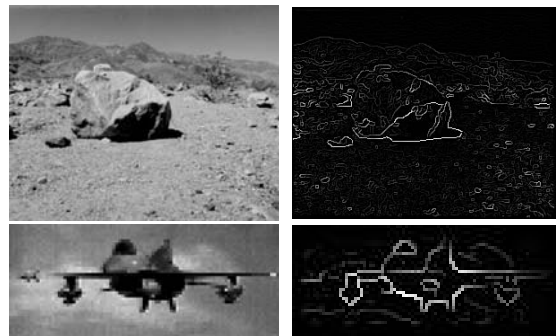
Measure of contrast : $G = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}}$

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

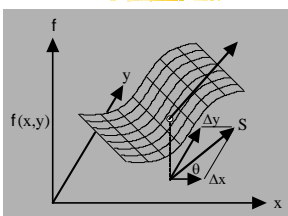
GRADIENT IMAGES



REAL IMAGES



GEOMETRIC INTERPRETATION

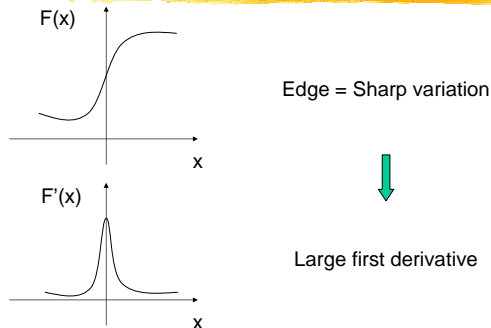


But
 $I(x,y)$ is not a continuous function.
 Therefore
 look for discrete approximations to the gradient.

EDGE OPERATORS

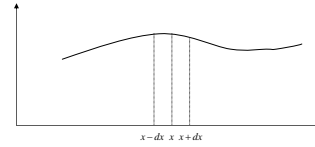
- Difference Operators
- Convolution Operators
- Parametric Matchers

GRADIENT METHODS



FINITE DIFFERENCES (1-D)

In one dimension:



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

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

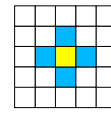
FINITE DIFFERENCES IN C (1-D)

p →

Line stored as an array:

```
{
  int i;
  for(i=0;i<n;i++){
    q[i]=p[i+1]-p[i];
  }
}
```

FINITE DIFFERENCES (2-D)



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

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

FINITE DIFFERENCES IN C (2-D)

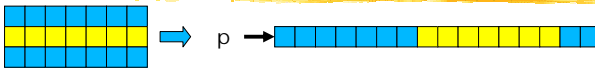
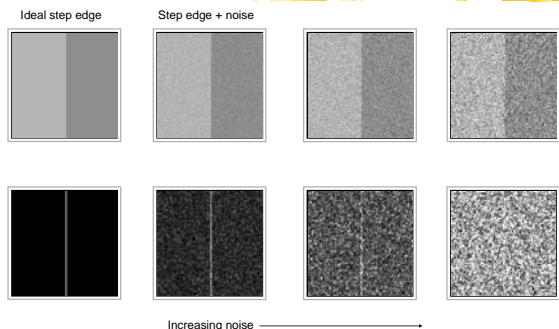


Image stored in raster format:

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

EFFECT OF NOISE



BACK TO FOURIER

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

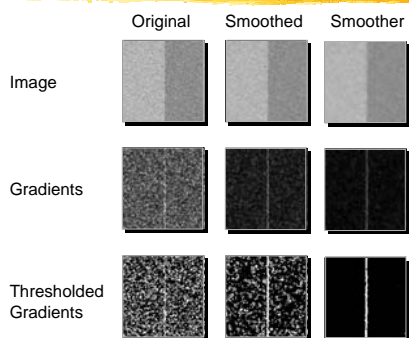
→ Differentiating emphasizes noise!

CONVOLUTION OPERATORS

Images are far too noisy for simple finite difference computation:

- Fit a surface to local image patch.
 - Convolve with the derivative of a Gaussian.
- Convolve the image with a mask

BENEFITS OF BLURRING



CONVOLUTION

Continuous case:

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

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

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 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 ...

DIFFERENTIATION AS CONVOLUTION

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

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

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

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

→ Use wider masks to add an element of smoothing

2x2 Masks: Roberts Operator

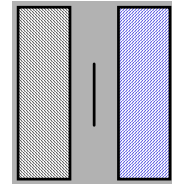
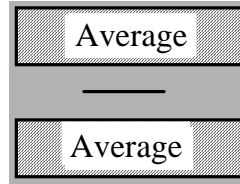


$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$G = \sqrt{[I(x+1,y+1) - I(x,y)]^2 + [I(x+1,y-1) - I(x-1,y+1)]^2}$$

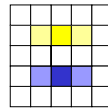
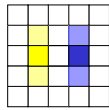
→ Equivalent to fitting plane to patch.

SMOOTHING AND DIFFERENTIATING



Taking the difference of averages on either side of the central pixel.

3X3 MASKS: PREWITT AND SOBEL



$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

→ Equivalent to fitting a quadric to patch.

→ Roberts followed by smoothing.

PREWITT EXAMPLE



Santa Fe Mission



Prewitt Horizontal and Vertical Edges Combined

SOBEL EXAMPLE



SEPARABILITY

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$= 1/4 * [-1 \ 0 \ -1] \otimes \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

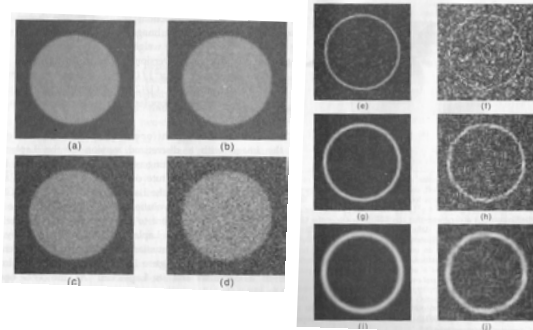
$$= 1/4 * [1 \ 2 \ 1] \otimes \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Sobel kernel is separable -> Faster implementation

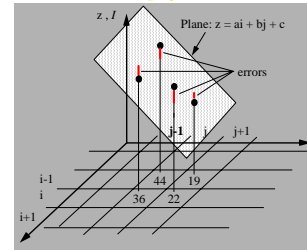
Non separable: N^2W^2 multiplications for a $N \times N$ image and a $W \times W$ mask.

Separable: $2N^2W$ multiplications for a $N \times N$ image and a $W \times W$ mask.

MASK SIZE



LARGER WINDOWS



Minimize the squared error between surface model and image data.

REMOVING NOISE

Problem:

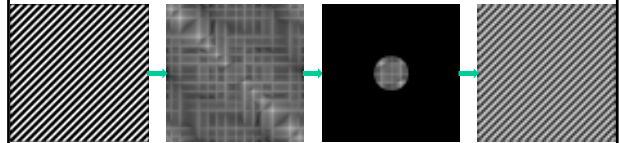
High frequencies lead to trouble with derivation.

Solution:

Suppress high frequencies by

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

DIAGONAL STRUCTURES



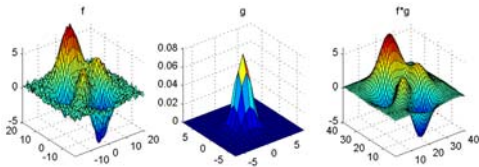
Rotated stripes:

- Dominant diagonal structures
- Discretization produces additional harmonics

Removing higher frequencies and reconstructing:

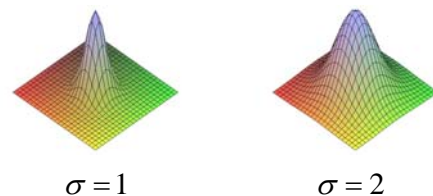
- Smoothed image

GAUSSIAN SMOOTHING



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

GAUSSIAN MASKS

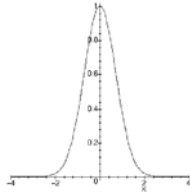


$\sigma = 1$

$\sigma = 2$

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

SEPARABILITY



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

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

$$\iint_{u,v} g_2(u, v) f(x-u, y-v) dudv = \int_u g_1(u) \left(\int_v g_1(v) f(x-u, y-v) dv \right) du$$

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

SMOOTHED IMAGES



GAUSSIAN DERIVATIVES



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

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

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

GAUSSIAN MASKS

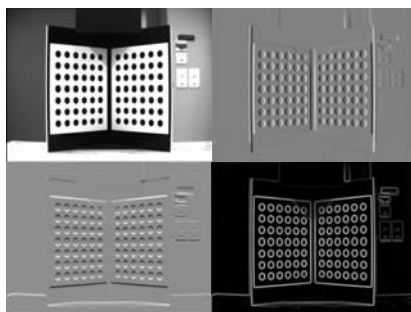
Sigma=1:

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

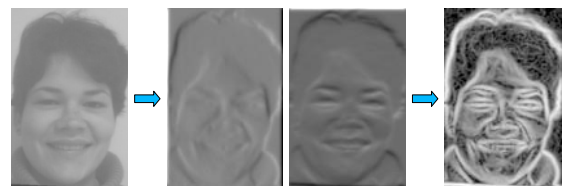
Sigma=2:

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

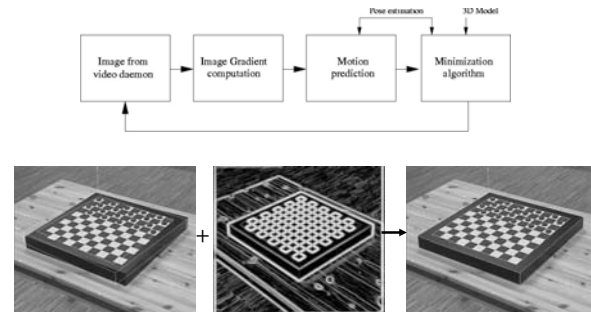


GRADIENT-BASED TRACKING



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

TRACKING WORKFLOW

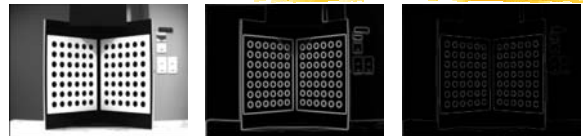


VIRTUAL CHECKERS



- Real-time tracking of the Board
- Detecting motions of the pieces
- Animating the Virtual Human

CANNY EDGE DETECTOR



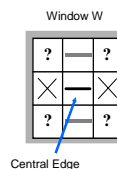
1. Convolution
 - Gradient strength
 - Gradient direction
2. Non Maxima Suppression
3. Hysteresis Thresholding

CANNY EDGES



NON-MAXIMAL SUPPRESSION

In parallel, at each pixel in edge image, apply selection window W as a function of edge orientation:



- Definitely consider these edges
- X Do not consider these edges
- ? Maybe consider them, depending on algorithm

HYSTERESIS THRESHOLDING

Algorithm takes two thresholds: high & low

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



$\sigma=1, T_2=255, T_1=1$

'Y' or 'T' junction
problem with
Canny operator

CANNY RESULTS



$\sigma=1, T_2=255, T_1=220$

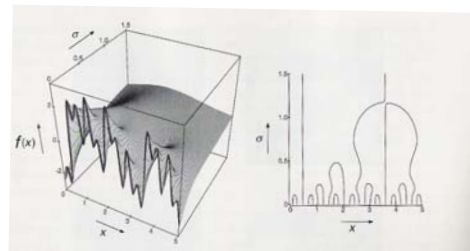
$\sigma=1, T_2=128, T_1=1$

$\sigma=2, T_2=128, T_1=1$

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



Gaussian smoothing removes details and never adds new ones.

MULTIPLE SCALES



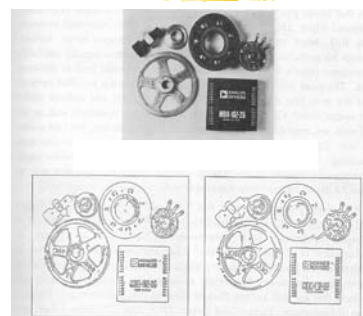
$\sigma=1$

$\sigma=2$

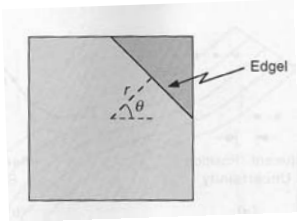
$\sigma=4$

→ Choosing the right scale is a difficult semantic problem.

REMOVING DETAILS

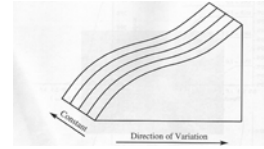


PARAMETRIC MODEL MATCHERS



→ 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.

OPEN ISSUES

- Convolution operators respond to steep smooth shading.
- Parametric matchers tend to reject non ideal edges.
- Arbitrary thresholds and scale sizes are required.

→ **Edge detection remains an open problem.**