Feature Detection

Prof. Hager
Cs461

Feature Detection

- Edges
- Corners
- SIFT features

From Pixels to Edges

- Various operators can be used to *enhance* rapid contrast changes
- Detecting these contrast changes involves *thresholding* to separate noise from signal
- *Edges* are a result of *grouping* pixels (sometimes called "edgels") into groups forming continuous curves.

Definitions:

Edge normal: Unit vector in direction of maximum intensity variation

Edge direction: Perpendicular to edge normal

Edge position: Image position of pixels of edge

Edge strength: Change in contrast along normal

From Pixels to Edges

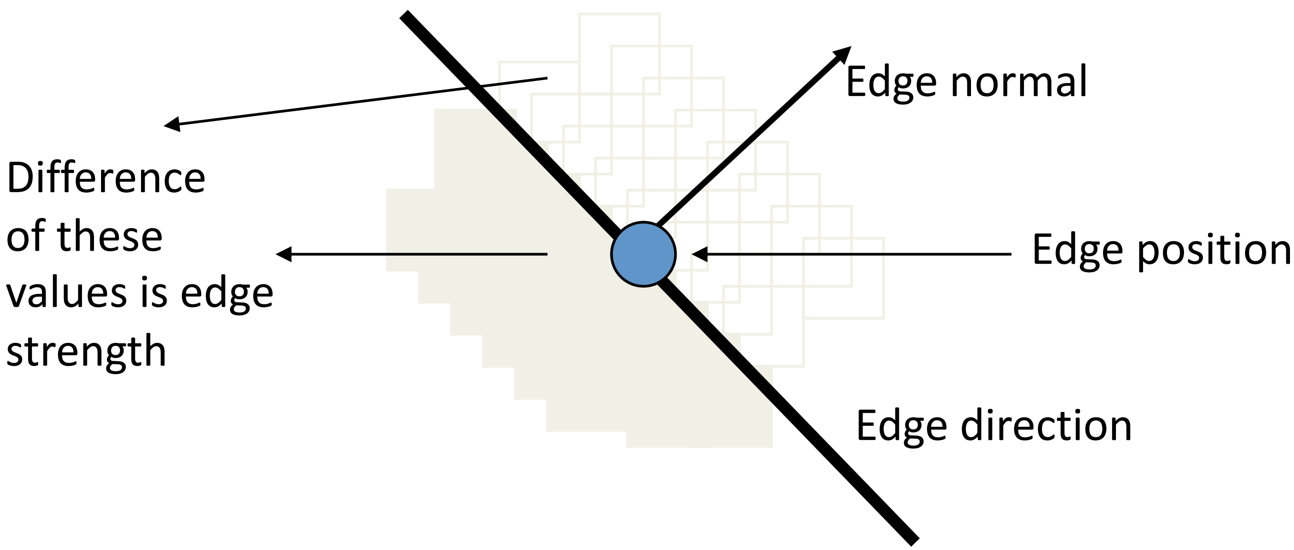
Definitions:

Edge normal: Unit vector in direction of maximum intensity variation

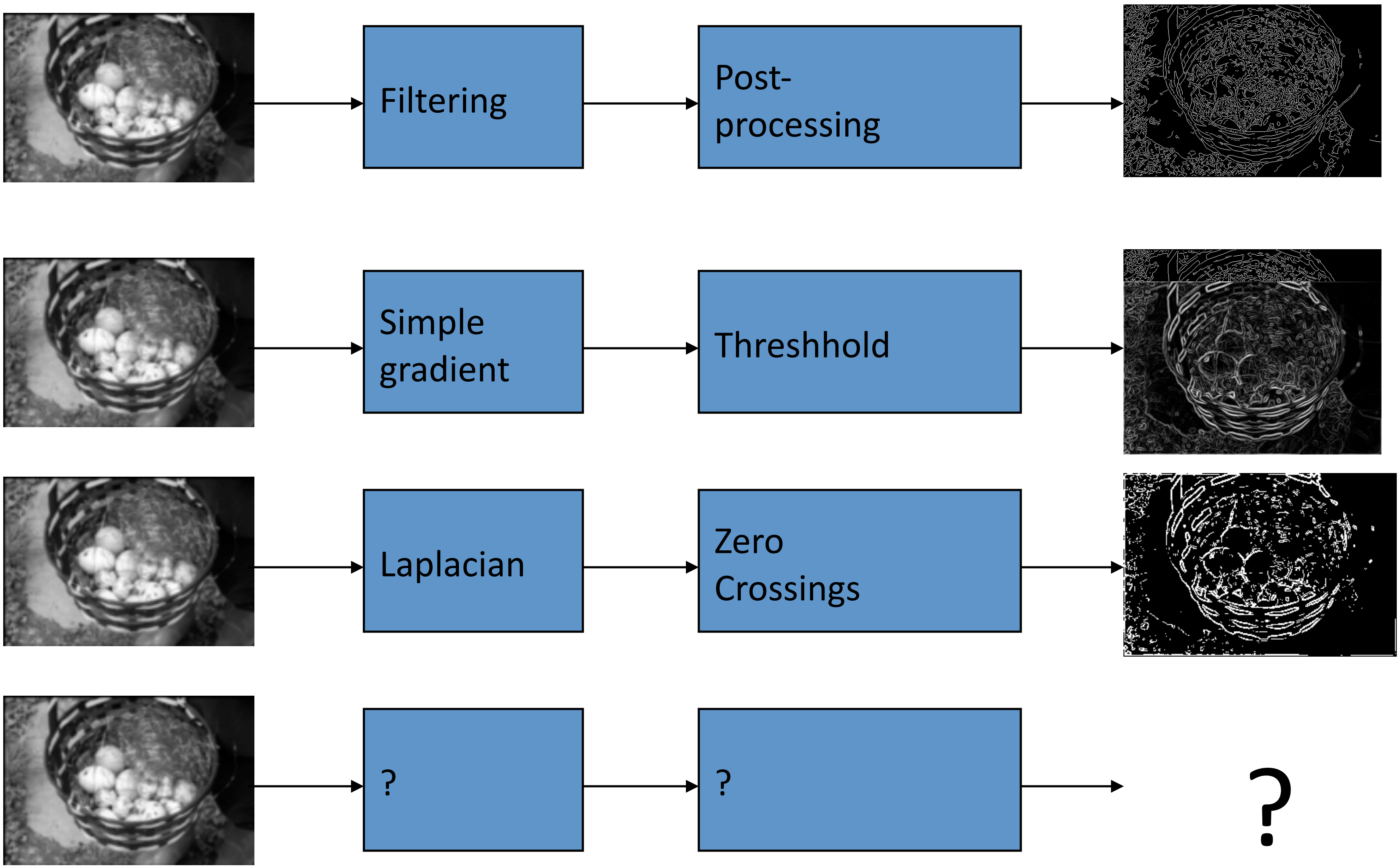
Edge direction: Perpendicular to edge normal

Edge position: Image position of pixels of edge

Edge strength: Change in contrast along normal



Edges: The Problem



What is optimal?

Canny Edge Detector

- The Plan:
  - Formulate an optimization problem for detection on 1-D signals
  - Generalize to 2D signals
  - Apply thresholding with hysteresis
  - Apply this operator at various scales
- The Assumptions:
  - Edge enhancement is linear
  - The edge model is step edges with amplitude A
  - Noise is additive, white and Gaussian

Optimization Criteria

- **Good Detection:** ***Minimize the probability of false positives and false negatives***
  - **Maximize the SNR**

$$\frac{A}{n_0}\Sigma(f) = \left| \frac{A\int_{-W}^{0} f(t)dt}{n_0\sqrt{\int_{-W}^{W} f^2(t)dt}} \right|$$

- **Good Localization:** ***Edgels detected should lie as close as possible to the true edge***
  - **Maximize 1/distance to edge center which leads to maximizing LOC**

$$\frac{A}{n_0}\Lambda(f') = \left| \frac{A\,|\,f'(0)\,|}{n_0\sqrt{\int_{-W}^{W} f'^2(t)dt}} \right|$$

Optimization Cont'd

- Consider the maximizing the product of both criteria
  - result is itself a step filter
  - step filters are noise amplifying!
- Additional criterion: single response constraint:
  - detector should minimize the number of local maxima about an edge (recall what happens with step filter)
  - ***RESULT 1: localization vs. detection***

$$\Sigma(f_w) = \sqrt{w}\Sigma(f) \quad and \quad \Lambda(f'_w) = \frac{1}{\sqrt{w}}\Lambda(f') \quad where \;\; f_w(x) = f(x/w)$$

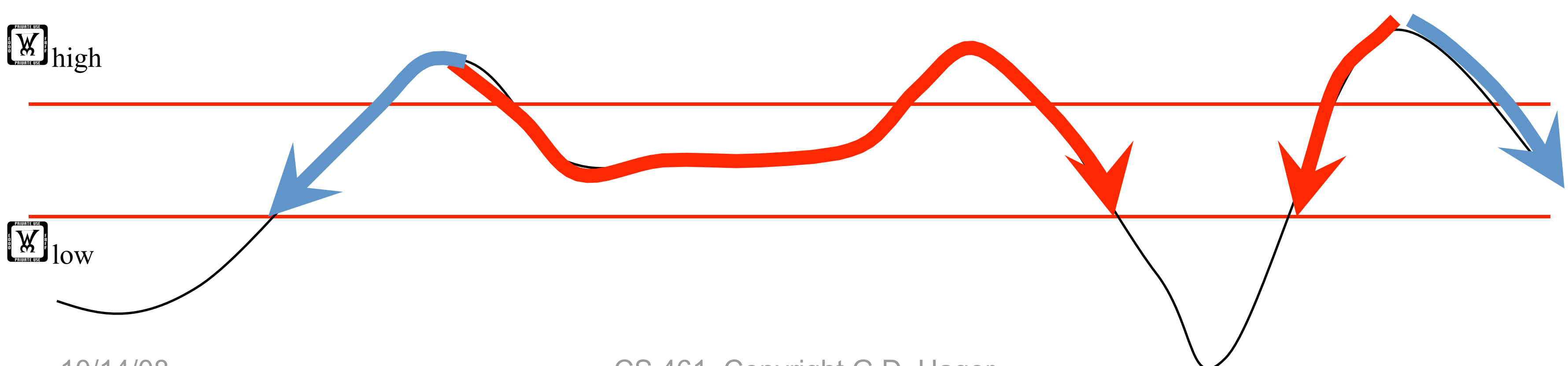
  - ***RESULT 2: optimal detector is very close to the first derivative of a Gaussian.***

The Procedure

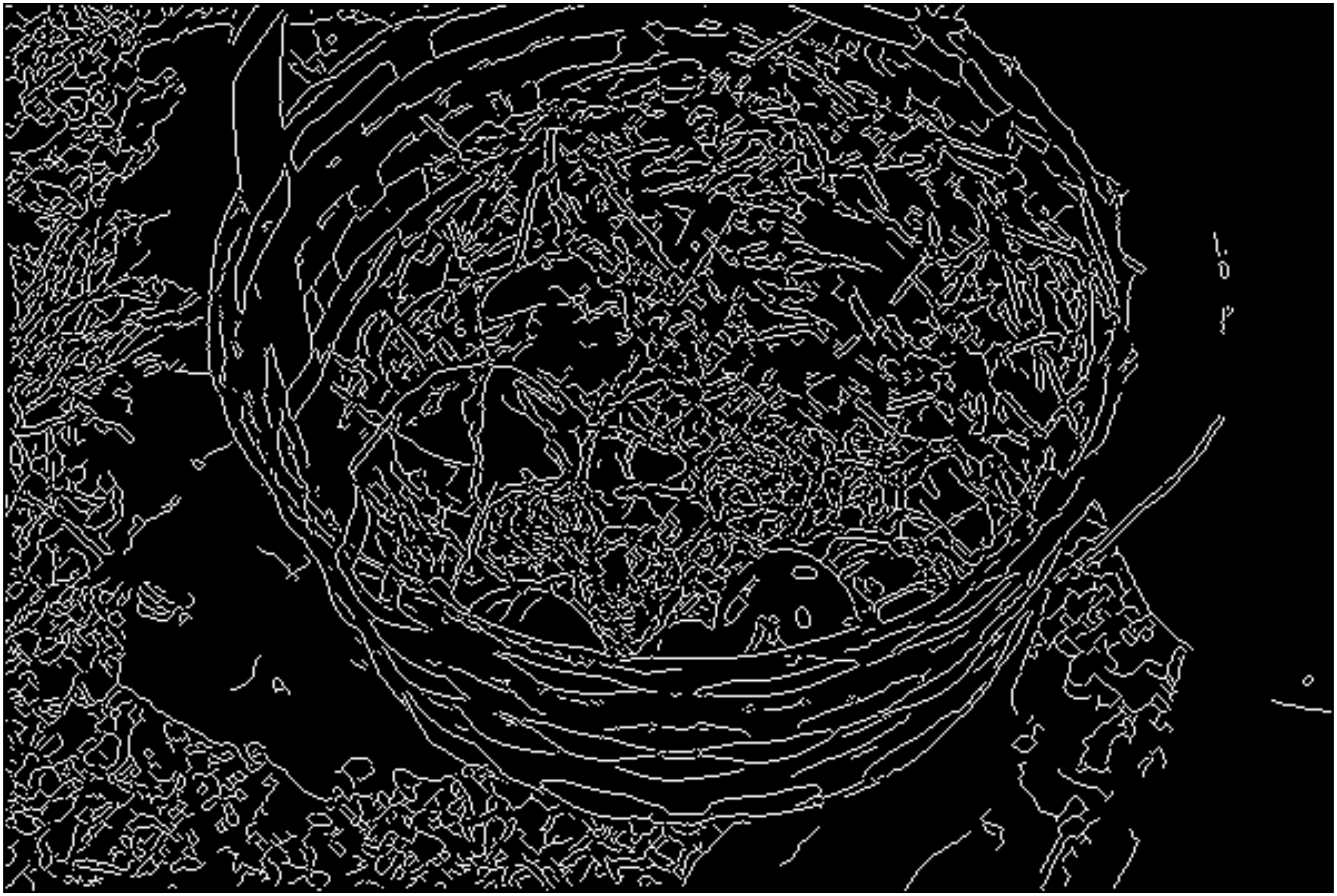
- Enhancement:
  - compute x and y derivatives using DoG's.
  - compute direction and magnitude of gradient (two images)
- Nonmaximal Suppression:
  - Sample along the gradient direction
  - If given pixel is smaller than neighbor, set it to zero
- Hysteresis Thresholding:
  - Starting from upper left, visit pixels until one exceeds t_{upper}
  - Follow chains of maxima in edge direction until value drops below t_{lower}
  - Mark and save all visited values as a connected contour

Hysteresis

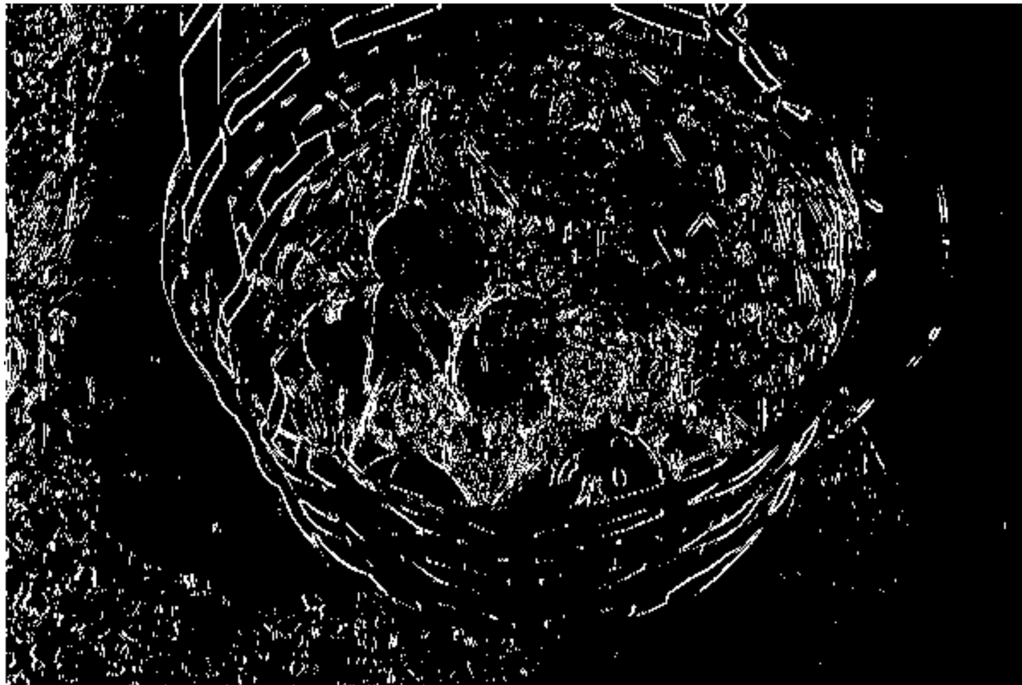
- Track edge points by starting at point where gradient magnitude > $t_{high.}$
- Follow edge in direction orthogonal to gradient.
- Stop when gradient magnitude < t_{low}.
  - i.e., use a high threshold to start edge curves and a low threshold to continue them.

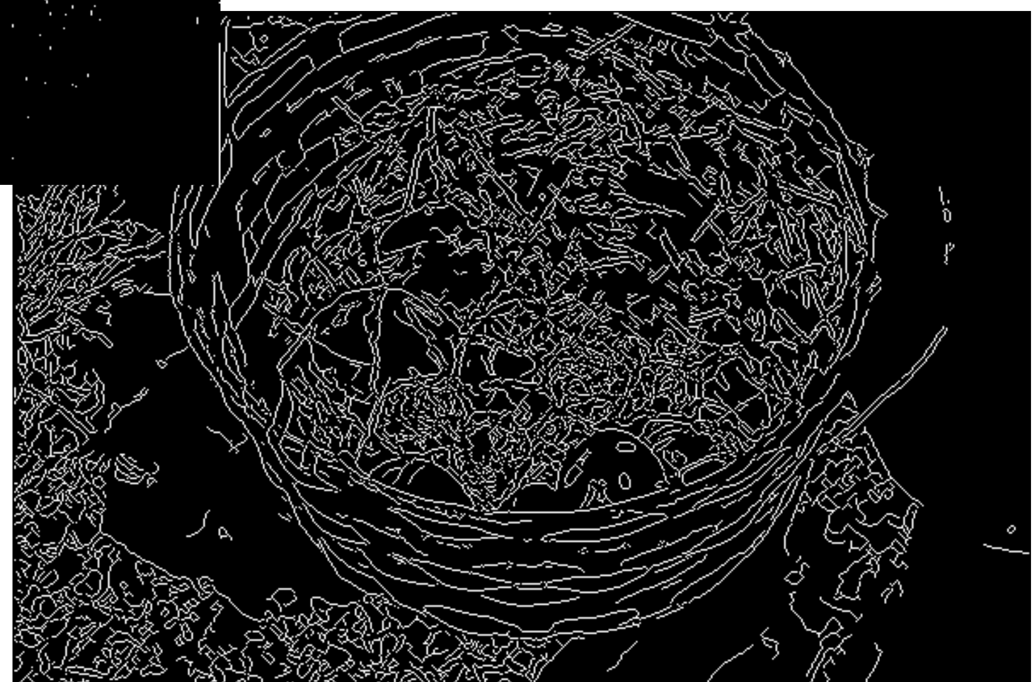


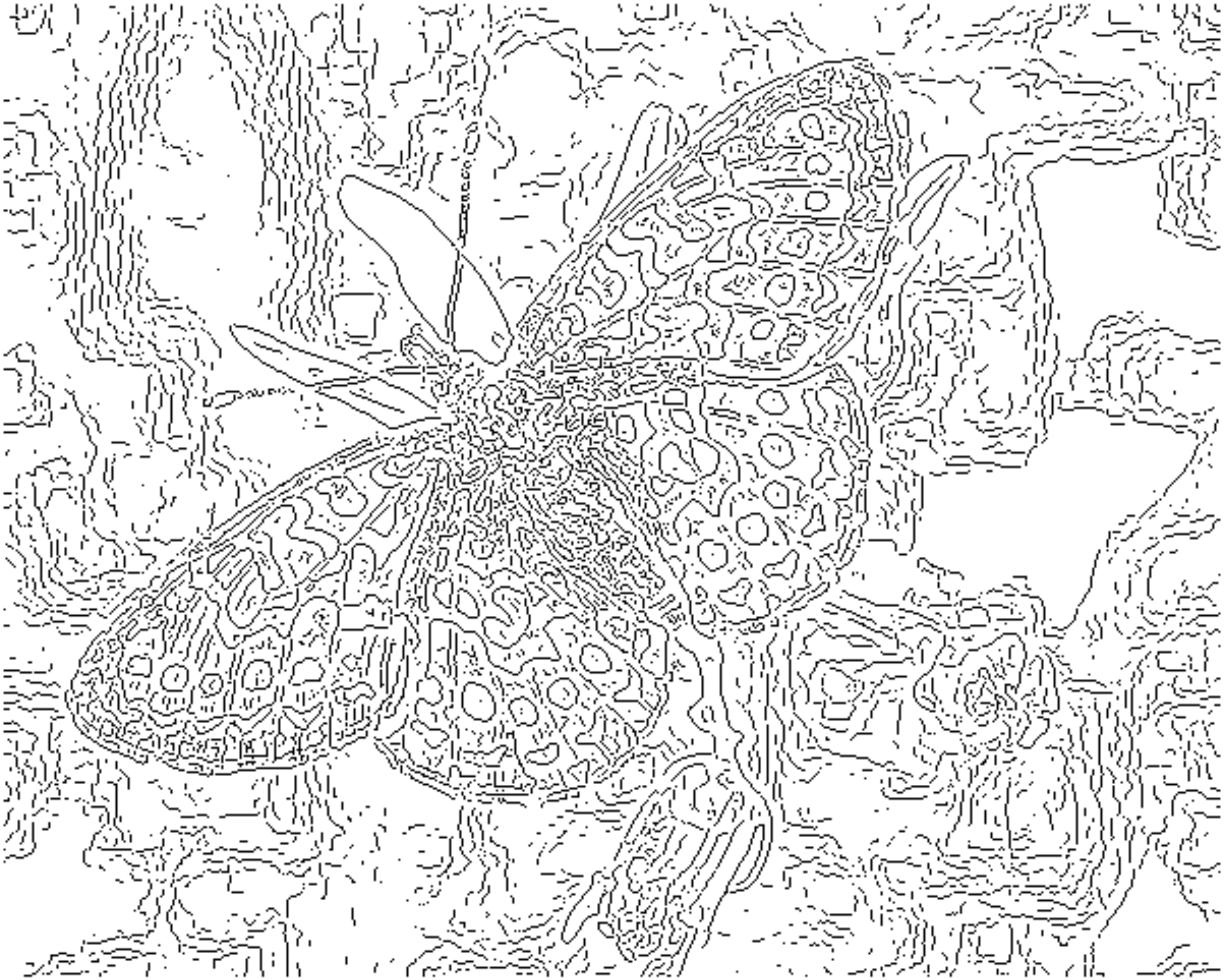
Canny Output



Canny Comparison







fine scale
high
threshold

coarse
scale,
high high
threshold

coarse
scale
Low high
threshold

Why is Canny so Dominant

Still widely used after 20 years.

1. Theory is nice (but end result same,).
2. Details good (magnitude of gradient, non-max supression).
3. Hysteresis an important heuristic.
4. Code was distributed.

SubPixel Precision

- It is often hard to exactly localize the maximum of a function
- Many algorithms need sub-pixel precision
- Thus, it is common to apply a *second derivative* operator locally to locate the edge
  - note we know the edge direction, so we can compute second directional derivatives!

Summary: Types of Edge Detection Operators

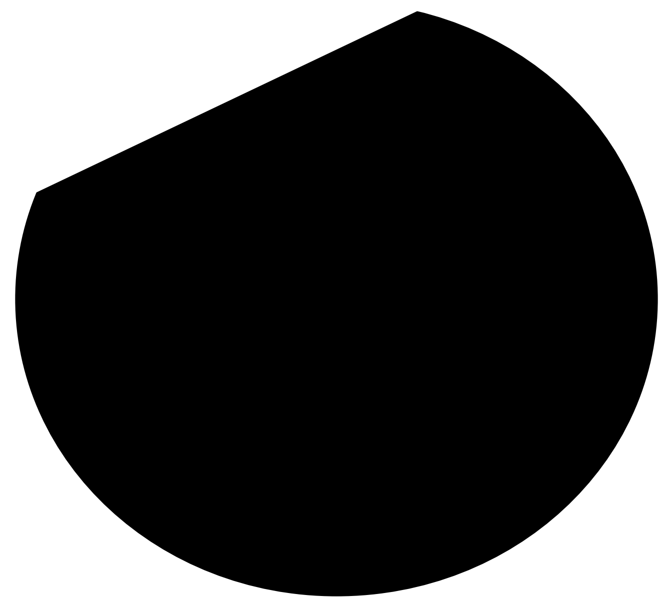
1. Operators approximating derivatives using differences.
    - directional: Roberts, Prewitt, DoG, etc.
    - Rotationally invariant: Laplacian (sum of second derivatives)

2. Operators based on the zero crossing of the second derivative (e.g. Canny).

3. Operators that attempt to match a specific image profile.

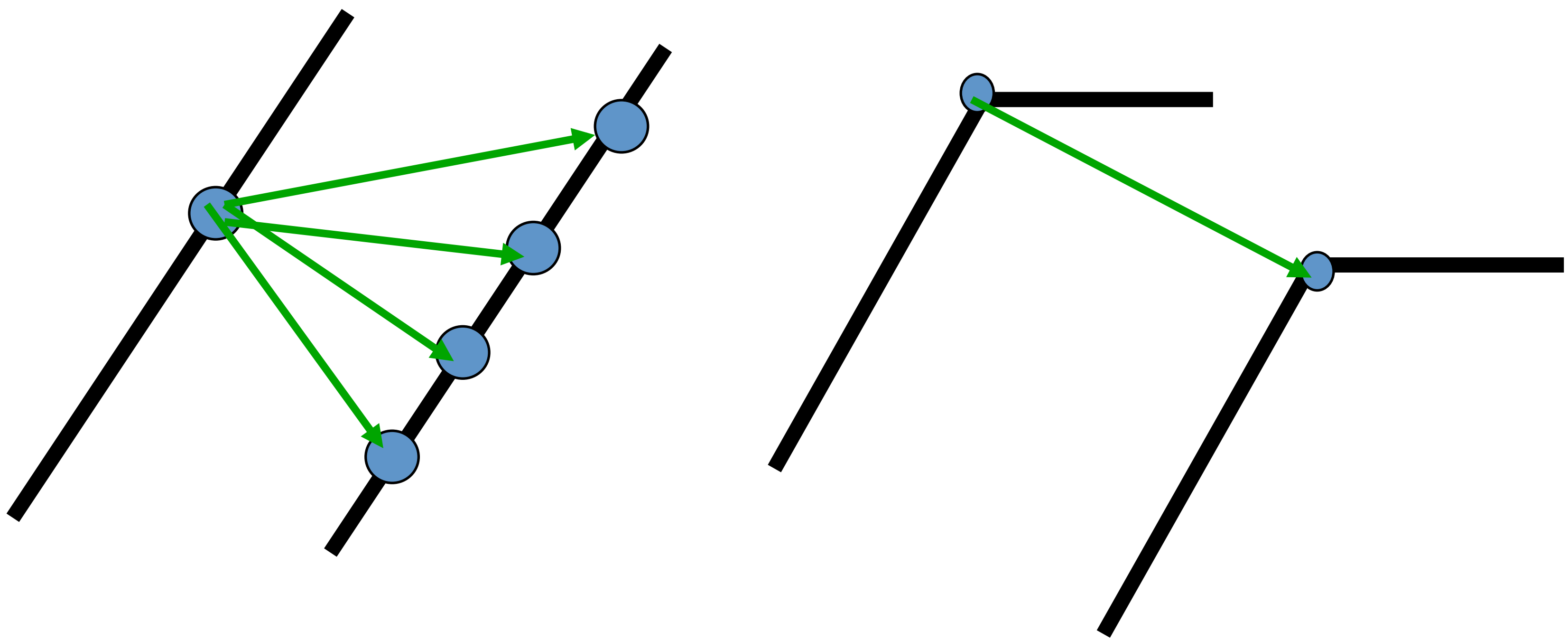
Corners

- Why are they important?

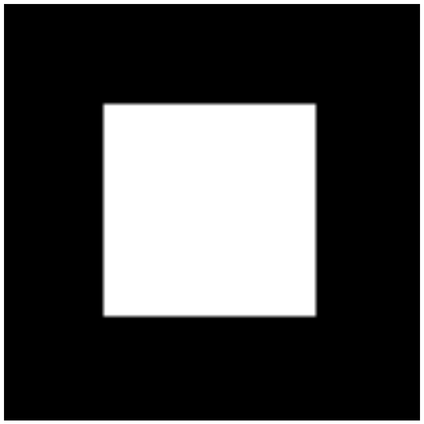


Corners contain more info than lines.

- A point on a line is hard to match; corners are "easy"

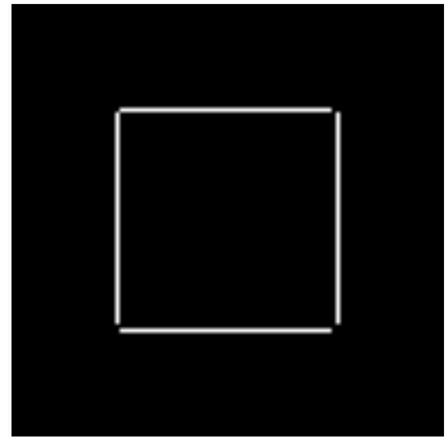


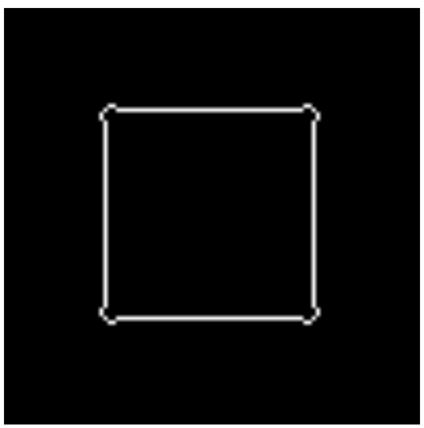
Edge Detectors Tend to Fail at Corners



Original

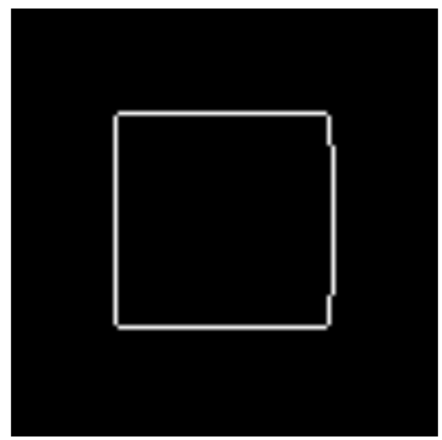
Prewitt





LoG

Canny



Finding Corners

Intuition:

- Right at corner, gradient is ill defined.
- Near corner, gradient has two different values.

Formula for Finding Corners

We look at matrix:

Sum over a small region, the hypothetical corner

Gradient with respect to x, times gradient with respect to y

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

WHY THIS?

Matrix is symmetric

First, consider case where:

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

This means all gradients in neighborhood are:

(k,0) or (0, c) or (0, 0) (or off-diagonals cancel).

What is region like if:

1. lambda$1 = 0$?
2. lambda$2 = 0$?
3. lambda$1 = 0$ and lambda$2 = 0$?
4. lambda$1 > 0$ and lambda$2 > 0$?

General Case:

From Linear Algebra we haven't talked about it follows that since C is symmetric:

$$C = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

So every case is like one on last slide.

Corners Algorithm

- For every point (u,v)’, compute C for a neighborhood about (u,v)
- Sort by the minimum singular value of C
- Perform non-maximal suppression
  - If the neighbor of a pixel has a larger value than the pixel itself, set the value to zero.
- Read off locations starting with highest values and working down until enough locations are found or we run out of locations
  - optional: as we read down, discard corners that are within a small distance of corners that have appeared higher in the list

Problems With Corners

- In clutter, there are many corners
  - Features should be numerous but distinctive
- The choice of corners depends on scale
  - Features should be comparable under similarity transforms
- Local illumination changes can affect which areas are chosen as corners
  - Features should be easily detected under changes in lighting
- The choice of corners is sensitive to out of plane rotation
  - Features should be easily detected under changes in pose

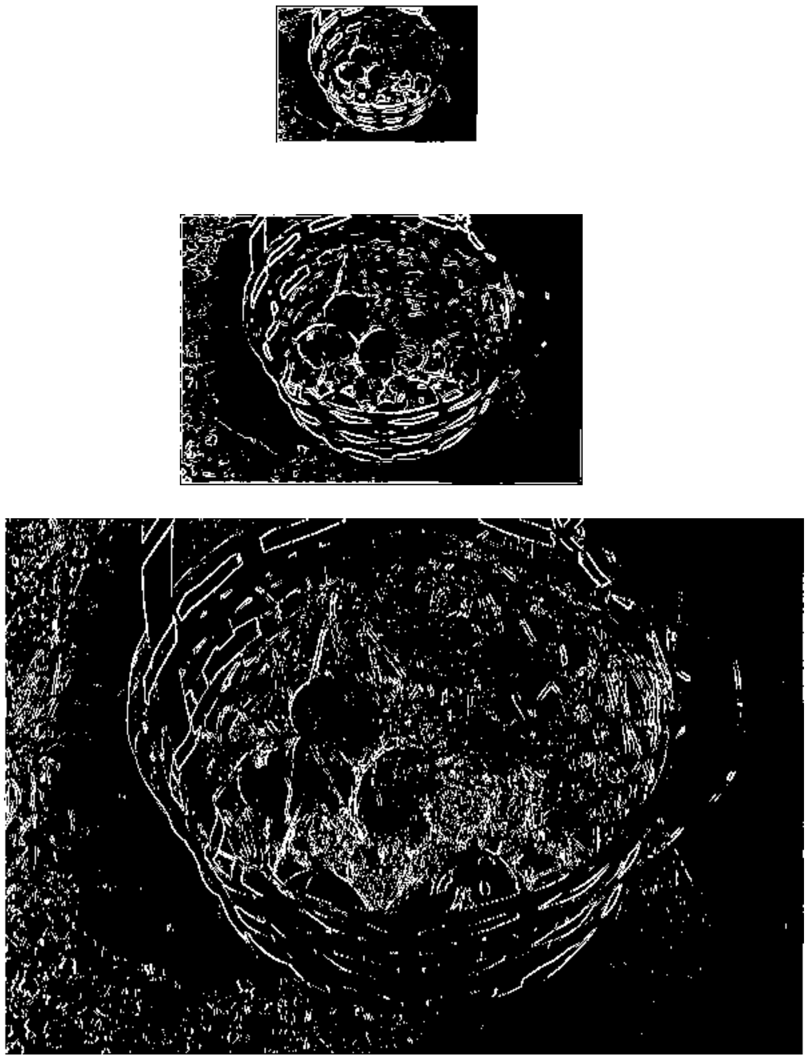
A “Recent” Addition

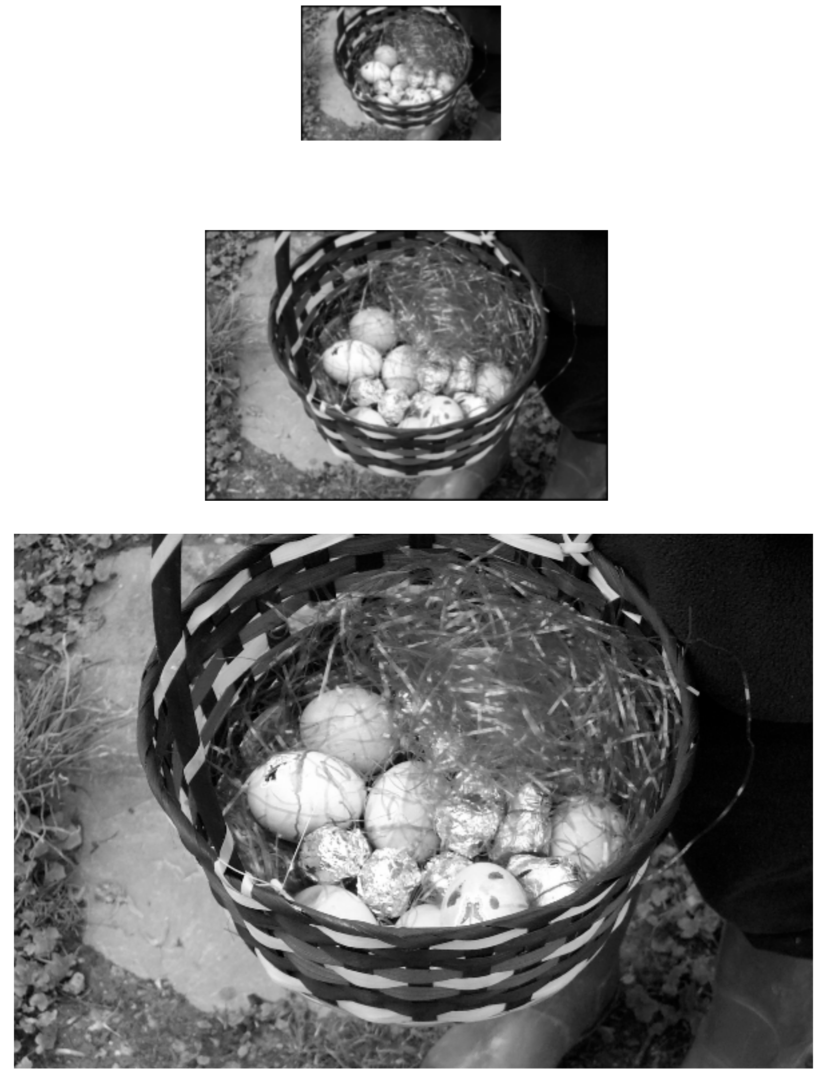
- Sift features were introduced by David Lowe in 1999
  - Lowe, David G. (1999). "Object recognition from local scale-invariant features". *Proceedings of the International Conference on Computer Vision* ***2: 1150–1157.***
- One of the most significant innovations in the last decade
  - Used for object recognition, image mosaicking, robot navigation, wide baseline stereo …
  - Is patented
- A more recent version, SURF, is much faster and works roughly as well
  - Herbert Bay, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Proceedings of the ninth European Conference on Computer Vision, May 2006
  - And is not patented …

Steps in SIFT Feature Selection

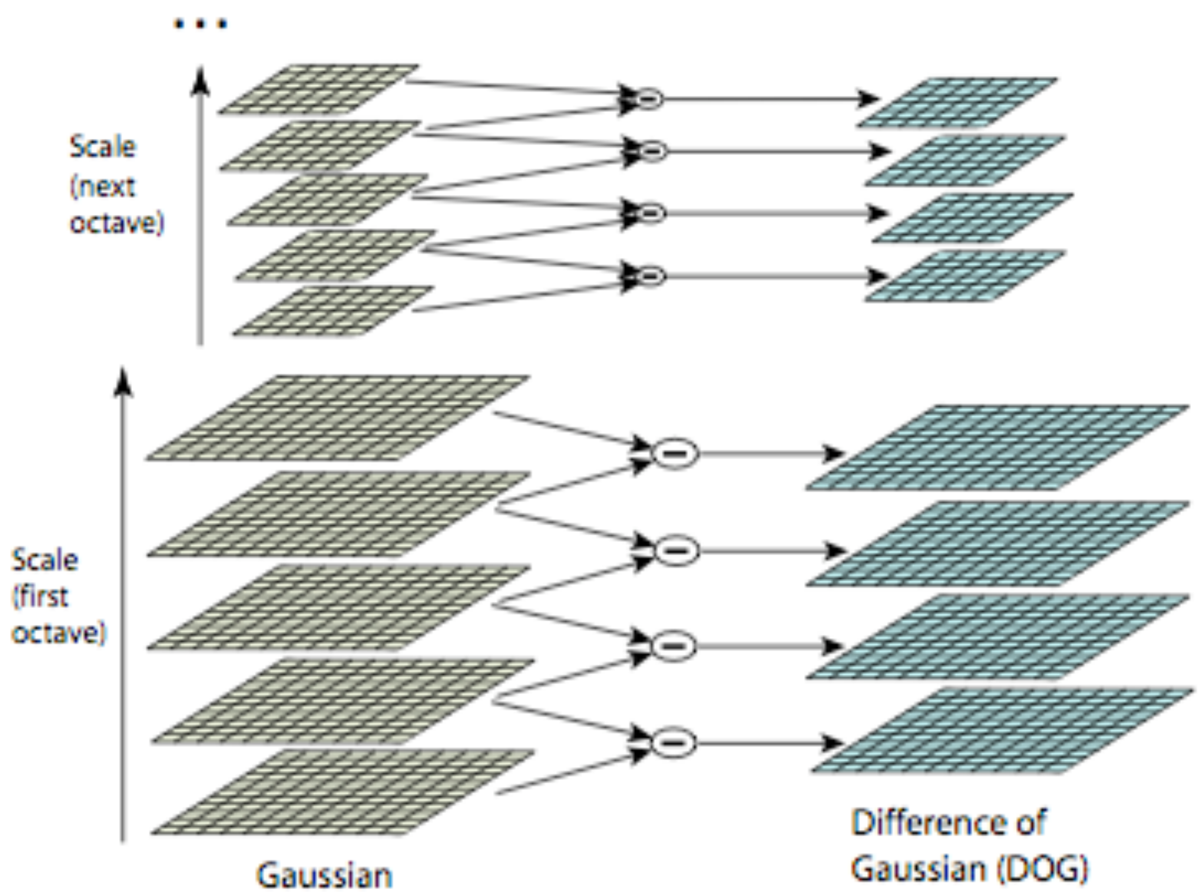
- Scale-space peak selection
  - Uses scale space pyramid
- Keypoint localization
  - includes rejection due to poor localization and low contrast
  - also perform cornerness check using eigenvalues; reject those with eigenvalue ratio greater than 10
- Orientation Assignment
  - dominant orientation plus any within 80% of dominant
- Keypoint descriptor
  - Gradient histograms
- Normal images yield approx. 2000 stable features
  - small objects in cluttered backgrounds require 3-6 features

Laplacian of Gaussian Pyramid



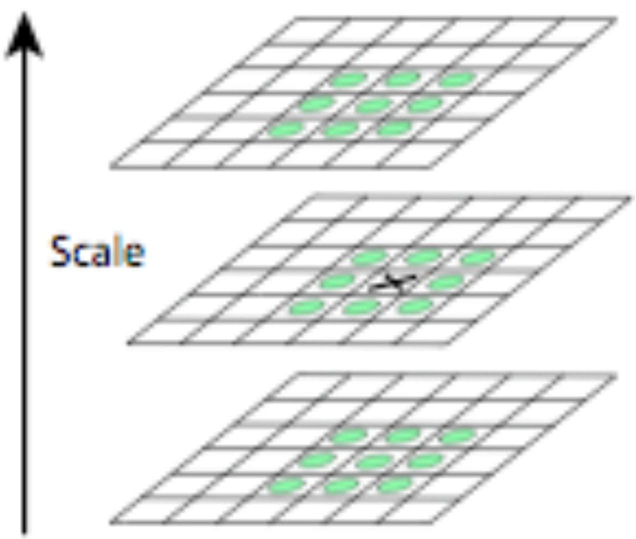


Approximating LoG with DoG



Peak Detection

- Find all max and min is LoG images in both space and scale
  - N samples per octave (factor 2 in sigma)
  - 8 spatial neighbors; 9 scale neighbors
  - orientation based on maximum of weighted histogram



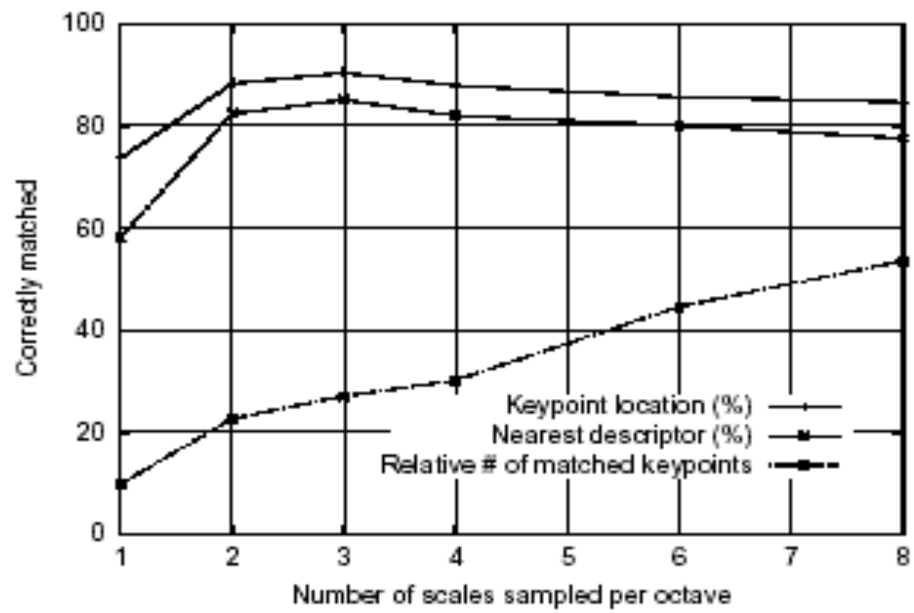


Figure 3: The top line in the graph shows the percent of keypoint locations that are repeatably detected in a transformed image as a function of the number of scales sampled per octave. The other lines show the percent of descriptors correctly matched to a large database and the total number of correctly matched keypoints (scaled arbitrarily to fi t on the graph).

Keypoint Descriptor

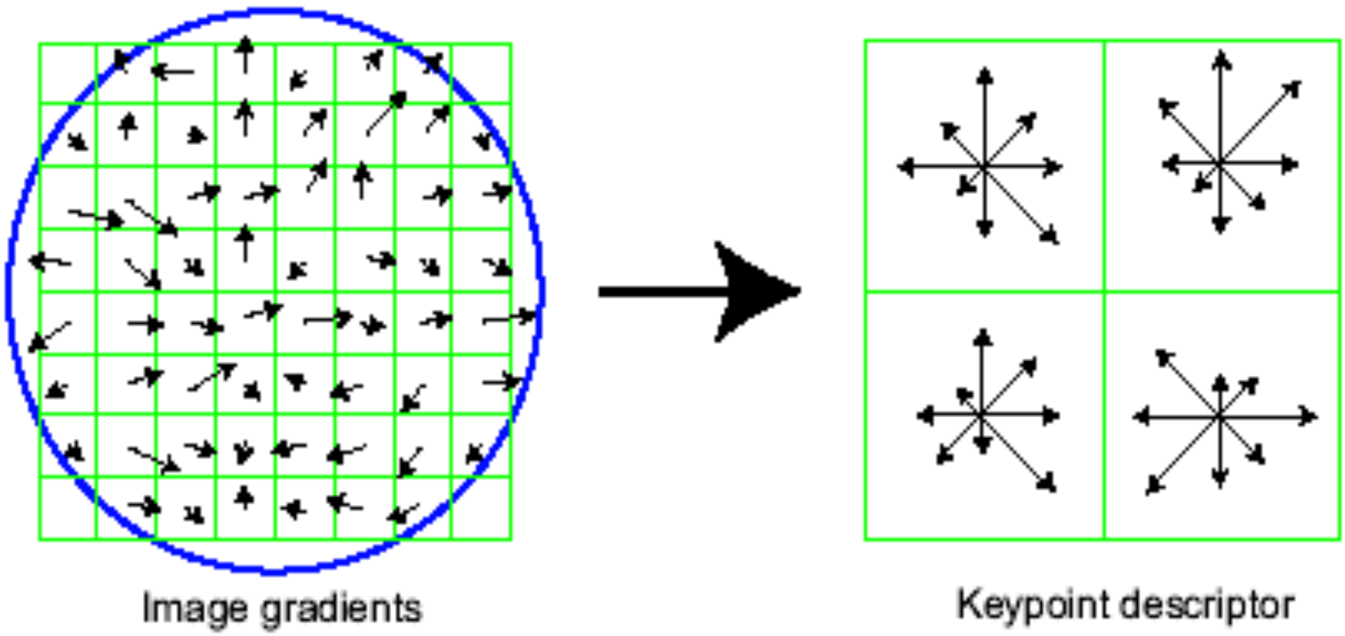


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point, as shown on the left. These are weighted by a Gaussian window, indicated by the overlayed circle. These samples are then accumulated into orientation histograms summarizing the contents over larger regions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. To reduce clutter, this figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas most experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

Example

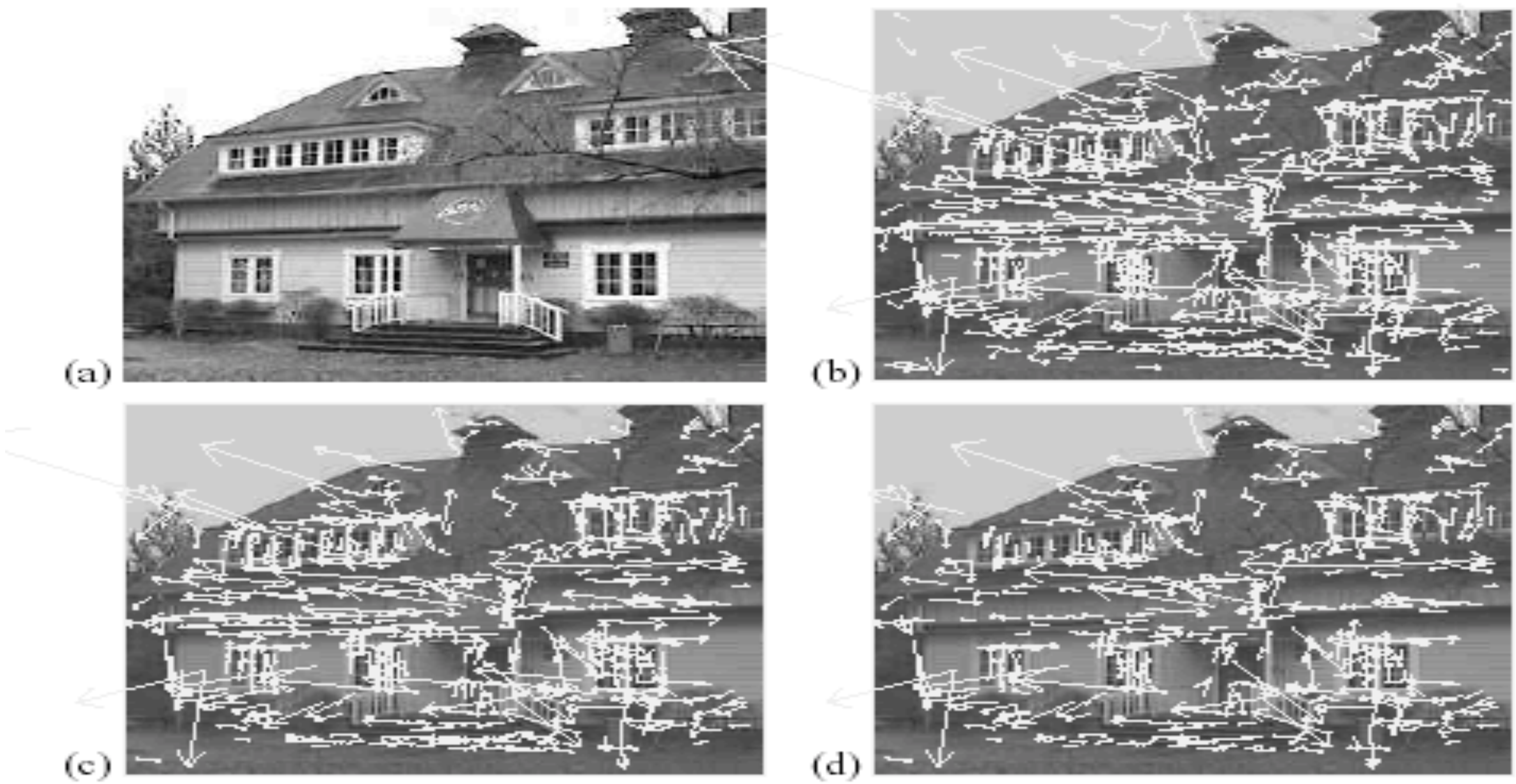


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principle curvatures.

Evalution Methodologies

- SIFT is designed to be invariant to translation, -in-plane rotation, and scale
  - Remaining degrees of freedom are two out-of-plane rotations
  - Effects include anisotropic scaling (aspect ratio), skew, and parallax
- Goal of evaluation is to tune available parameters to provide the best performance over a range of conditions
  - noise
  - out-of-plane rotations
- Data: 32 “diverse” images from which keypoints were extracted
  - artificially scale, rotation, add noise
  - ground truth is known in this case, allowing quantitative performance assessment

Matching Results

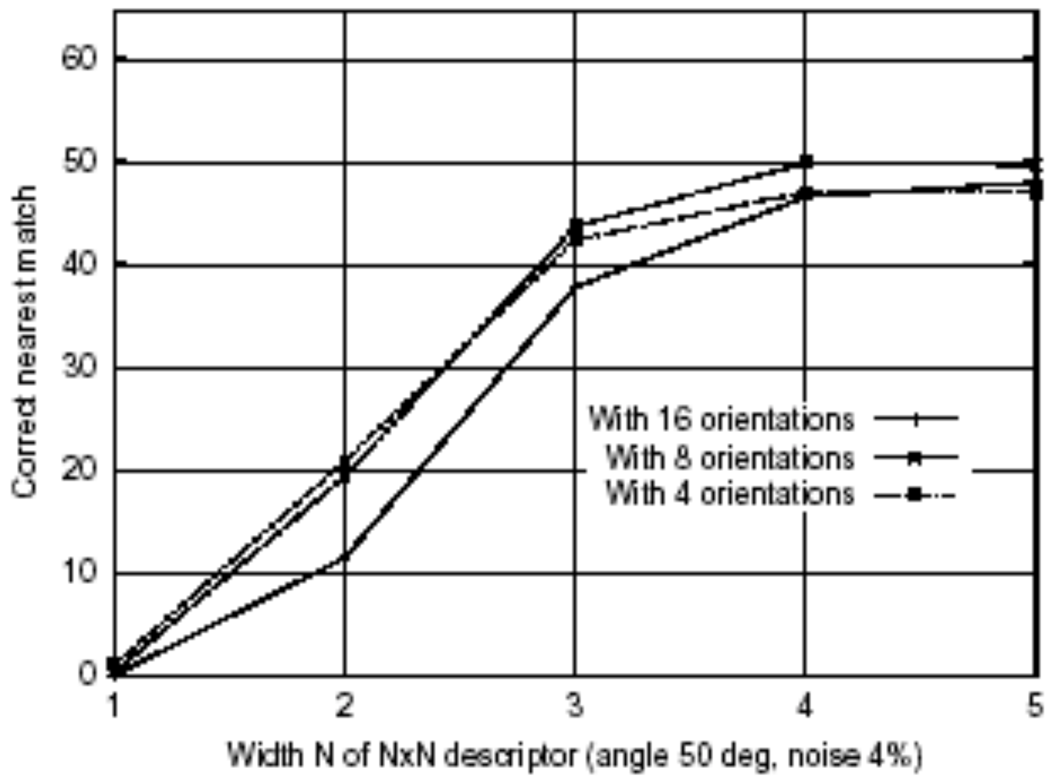


Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of size of the $n \times n$ keypoint descriptor and the number of orientations in each histogram. The graph is computed for an image with an affi ne viewpoint change of 50 degrees and addition of 4% image noise.

Detection Stability

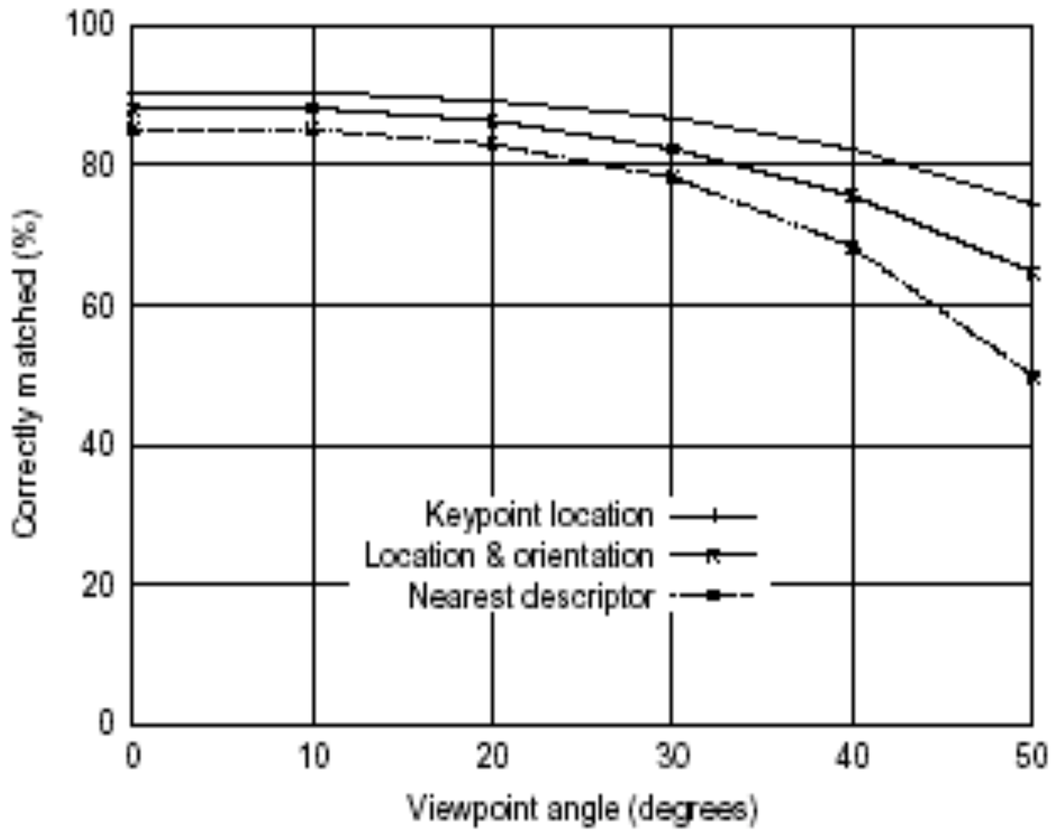


Figure 9: This graph shows the stability of detection for keypoint location, orientation, and final matching to a database as a function of affine distortion. The degree of affine distortion is expressed in terms of the equivalent viewpoint rotation in depth for a planar surface.

Smoothing Issues

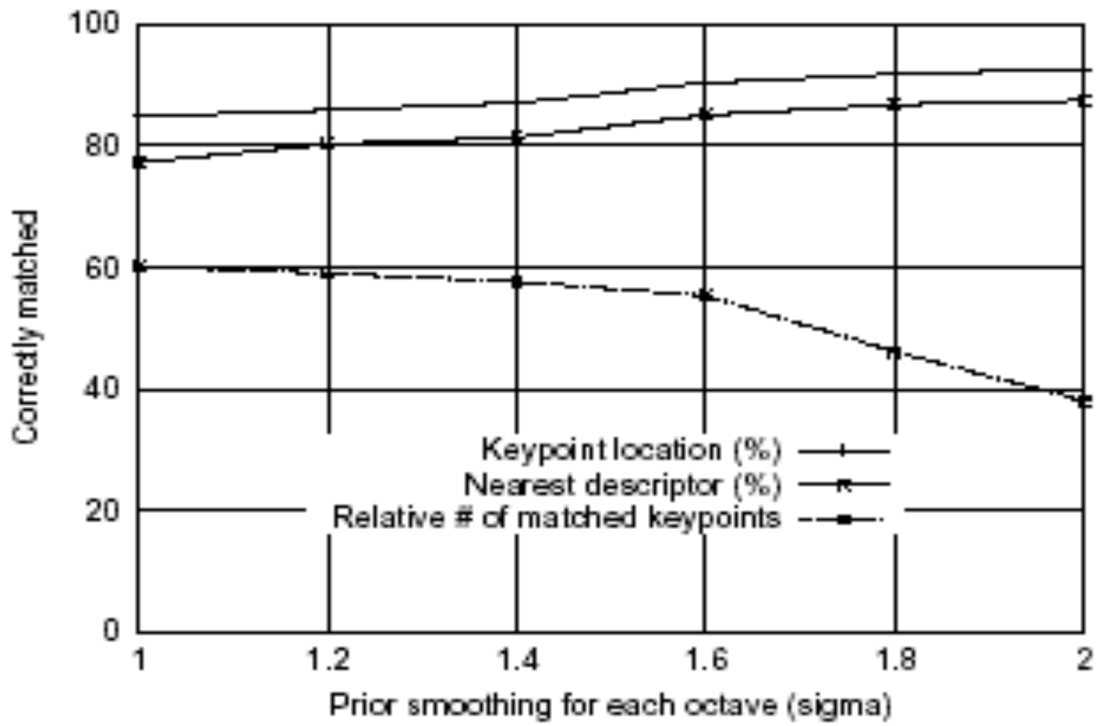


Figure 4: The top line in the graph shows the percent of keypoint locations that are repeatably detected as a function of the prior image smoothing before resampling each new octave. The other lines show the percent of descriptors correctly matched to a large database and the relative number of matched keypoints.

Match Scaling

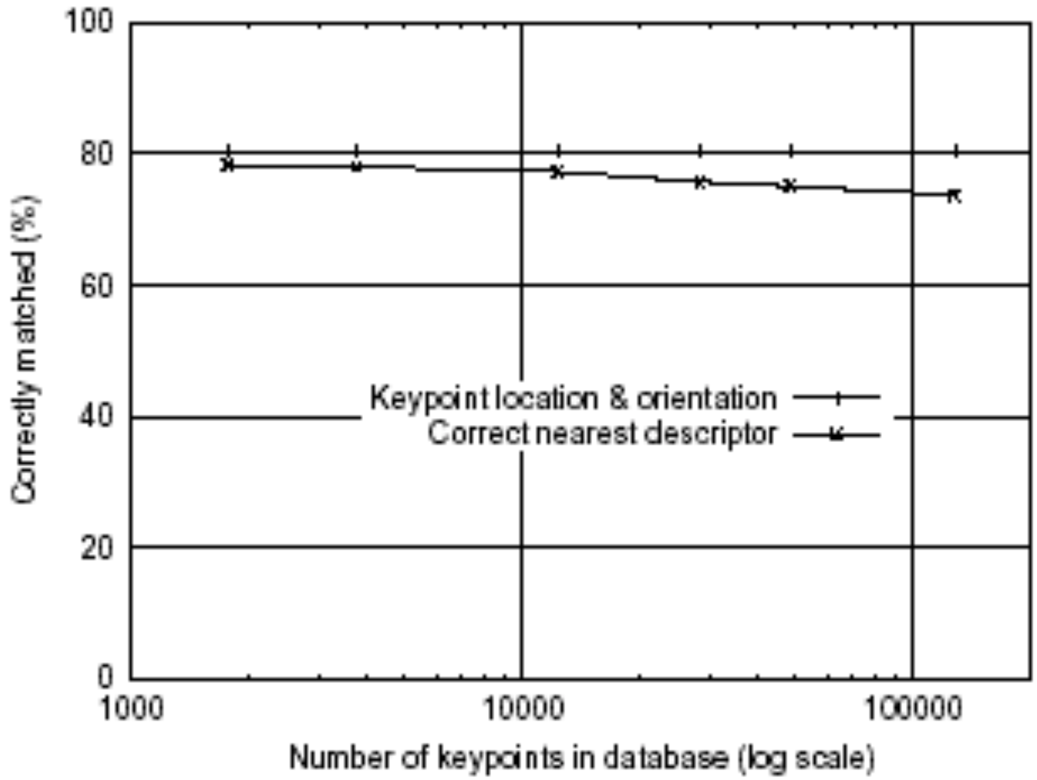


Figure 10: The dashed line shows the percent of keypoints correctly matched to a database as a function of database size (using a logarithmic scale). The solid line shows the percent of keypoints assigned the correct location and orientation.

Image Feature Detection Summary

- Filtering is a way of removing noise or suppressing/enhancing frequency content
- Typically, we combine some type of image derivative with smoothing
- Image gradients are the basic tool in 2D images
- Derivative of Gaussian is generally the gradient operator of choice
- Canny detector is probably the most widely used algorithm for performing edge detection
- Corner detectors can be formulated also in terms of image gradient structures