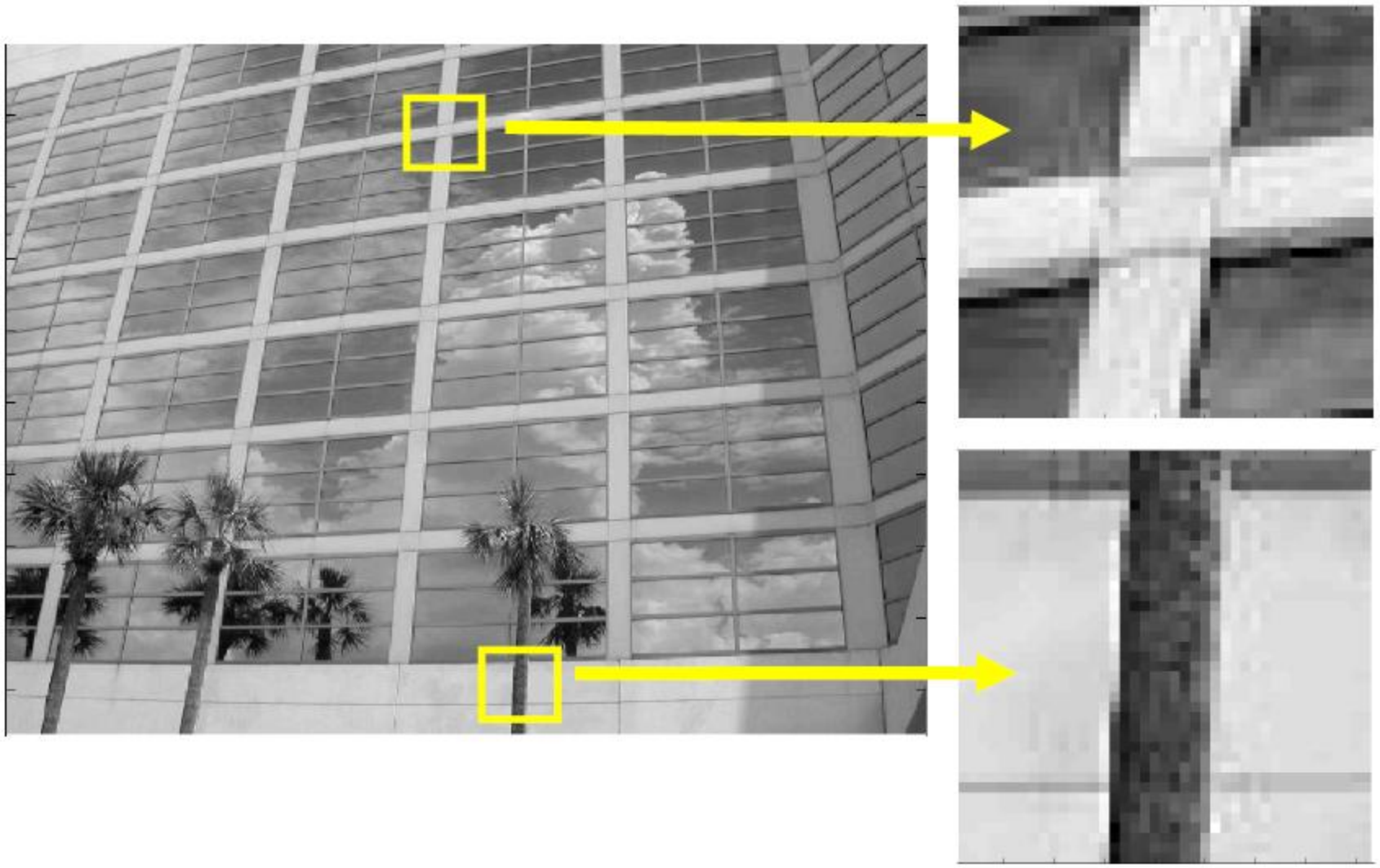
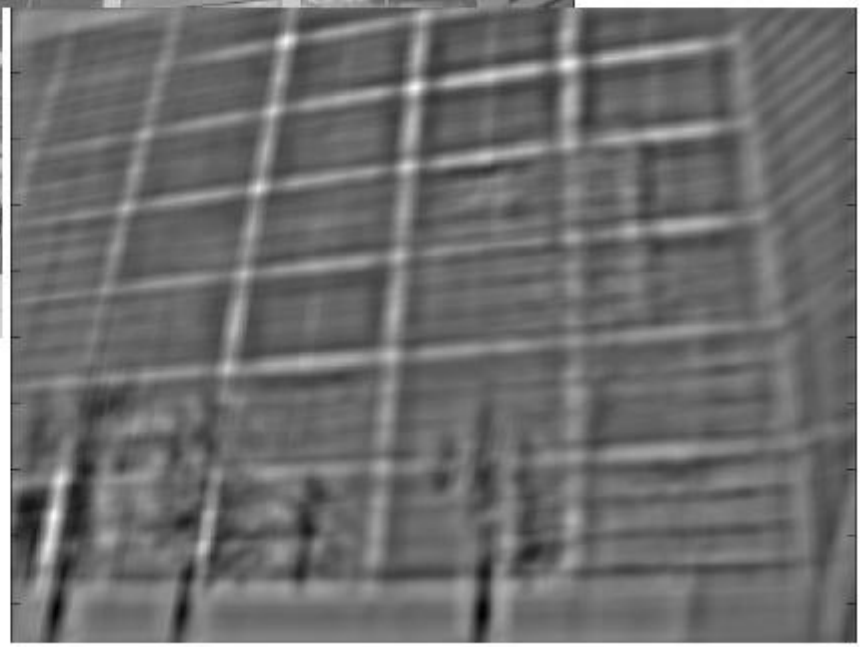
Lecture 16
Image Registration

Image Registration Methods

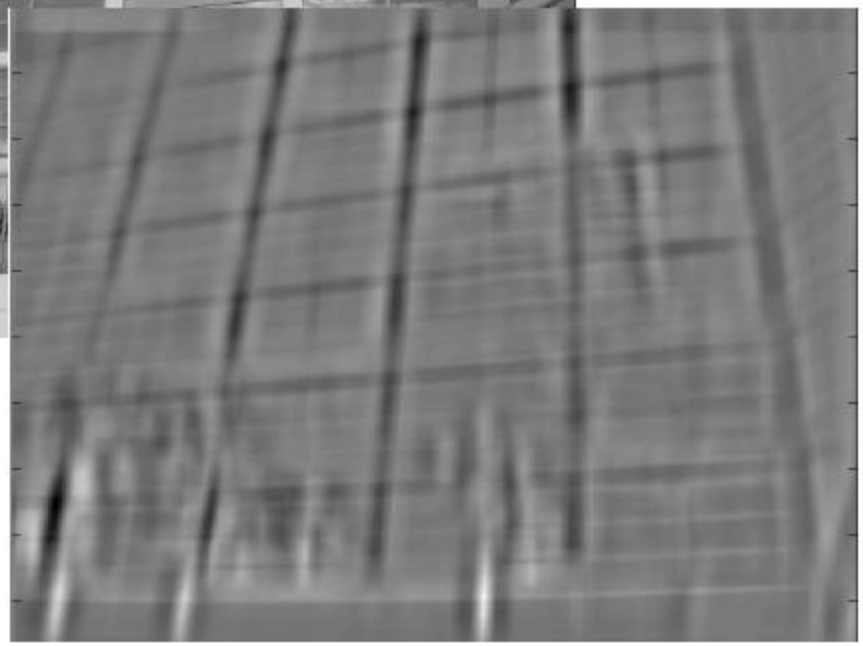
- Area based methods
  - Correlation type methods
    - Cross correlation
    - Normalized correlation
    - Binary correlation
    - Phase Correlation in Frequency domain (speed)
- Localized feature based methods
  - Edge based matching
    - Hausdorff distance based matching
  - Interest points based matching



*



*



Characteristics of Registration Methods

Every registration algorithm is a specific choice of three steps: All three should be chosen to improve registration.

- Feature space: which properties of the sensor or scene the data is sensitive to. Features are chosen to reduce sensor noise, and other distortions like illumination and atmospheric conditions.
- Similarity measure: Cross correlation, MAD, MSD, etc., is linked to feature space
- Search space and Search strategy

Image matching using correlation

- To locate a template image T within a larger reference image R the simplest way is to find the sum of squared differences (or sum of absolute difference) given by:

$$d = \Sigma (T - R)^2$$

$$d = \Sigma T^2 - 2\Sigma T R + \Sigma R^2$$

The summation is done over the support of T

- For a good match
  - *d* should be small
  - The matching point location is given by the location of smallest *d*
- The above equation implies that a small *d* means a larger value of $2\Sigma T R$: Correlation

Cross Correlation

- A Template image *T*(*x,y*) is located within an reference image *R*(*x,y*) using the *correlation function:*

$$\gamma(x,y) = \sum_s \sum_t T(s,t) R(x+s, y+t)$$

- Compares similarity of grey levels
- Maximum γ(x,y) indicates the match and its value indicates the confidence level
- Limitations
  - Mild occlusion leads to large errors making it practically ineffective
  - If template and reference have a large number of zeros in them the correct match is not reflected by a large correlation function
  - If reference image has large light areas the correlation has large values there

Normalized Correlation

- Solution to some of the problems in cross correlation:

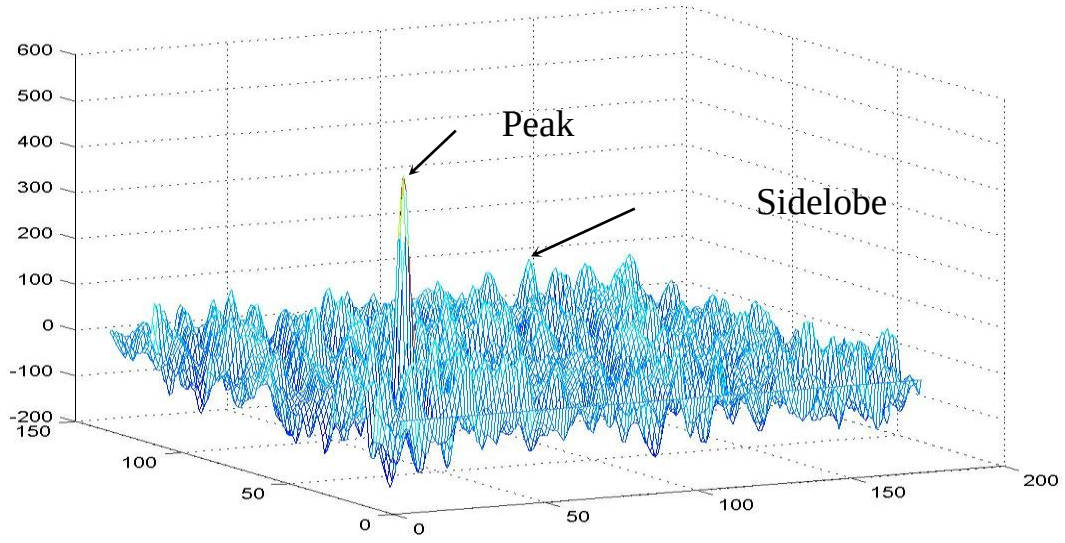
$$\hat{\gamma}(x,y)=\frac{\sum_s\sum_t[\left(T(s,t)-\overline{T(s,t)}\right)\left(R(x+s,y+t-\overline{R(x+s,y+t)}\right)]}{\left[\sum_s\sum_t[T(s,t)-\overline{T(s,t)}]^2\right]^{1/2}\left[\sum_s\sum_t[R(x+s,y+t)-\overline{R(x+s,y+t)}]^2\right]^{1/2}}$$

  - Which is between –1 to +1

Binary correlation

- The binary template image is denoted by $T_b(s,t)$
- The binary reference image is denoted by $R_b(x,y)$

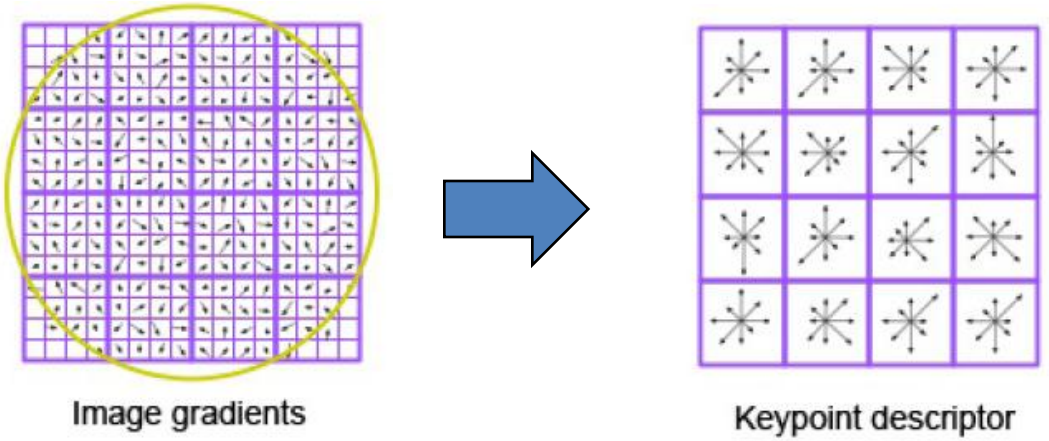
$$\chi(x,y) = \sum_s \sum_t [T_b(s,t) \otimes R_b(x+s, y+t)]$$



Scale Invariant Feature Transform (SIFT)

1. Take a 16 x16 window around interest point (i.e., at the scale detected).
2. Divide into a 4x4 grid of cells.
3. Compute histogram of image gradients in each cell (8 bins each).

16 histograms x 8 orientations
= 128 features



SIFT Computation – Steps

(1) Scale-space extrema detection

- Extract scale and rotation invariant interest points (i.e., keypoints).

(2) Keypoint localization

- Determine location and scale for each interest point.
- Eliminate "weak" keypoints

(3) Orientation assignment

- Assign one or more orientations to each keypoint.

(4) Keypoint descriptor

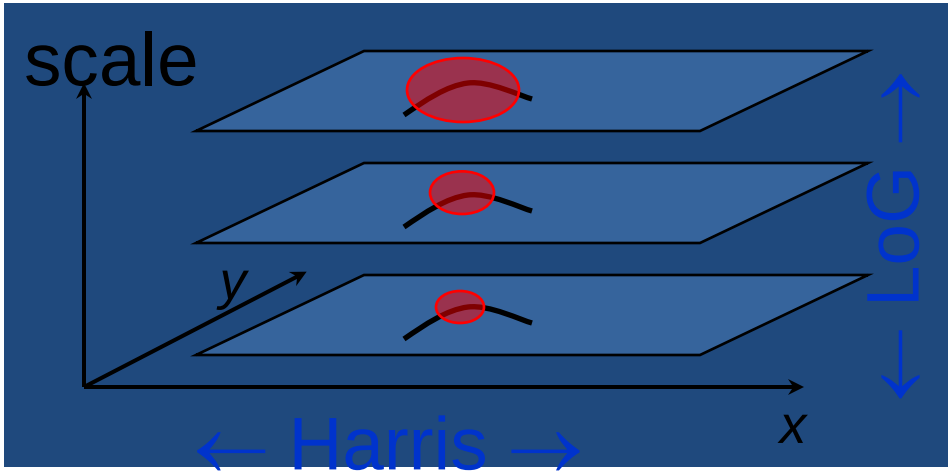
- Use local image gradients at the selected scale.

D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", **International Journal of Computer Vision**, 60(2):91-110, 2004.

Cited 9589 times (as of 3/7/2011)

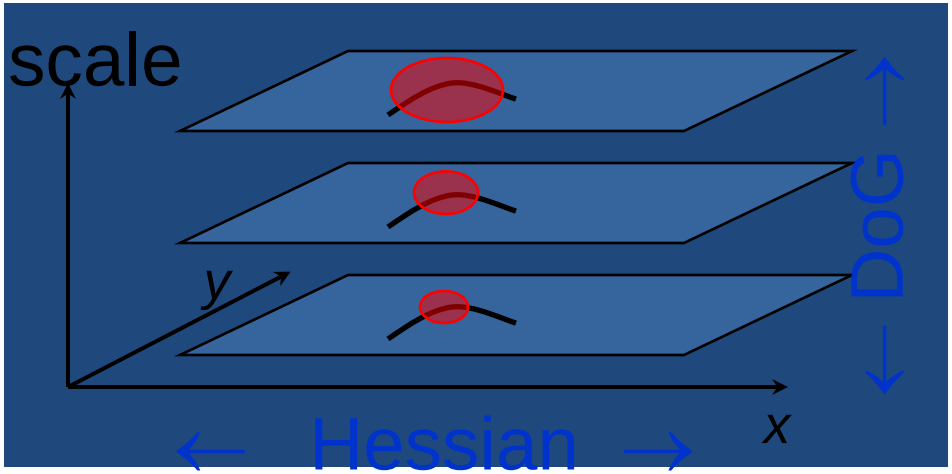
1. Scale-space Extrema Detection

- Harris-Laplace
- *Find local maxima of:*
  - Harris detector in space
  - LoG in scale



- SIFT

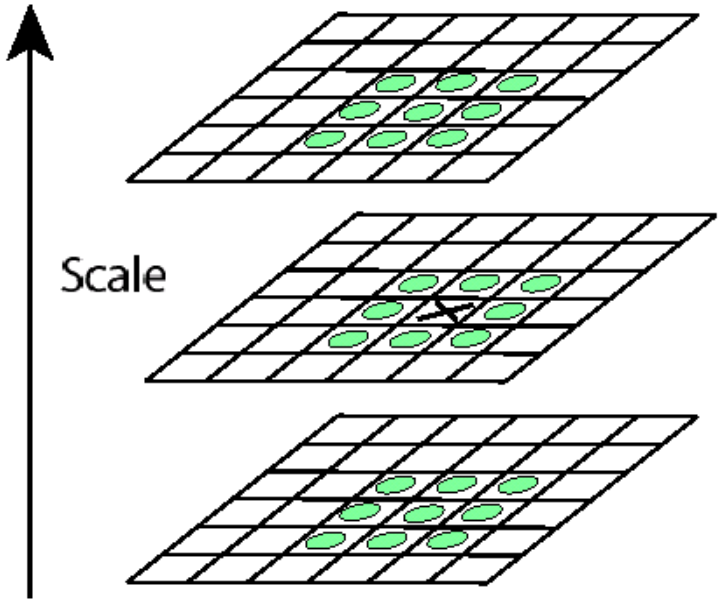
Find local maxima of:
- *Hessian in space*
- *DoG in scale*



1. Scale-space Extrema Detection (cont'd)

- Extract local extrema (i.e., minima or maxima) in DoG pyramid.

  -Compare each point to its 8 neighbors at the same level, 9 neighbors in the level above, and 9 neighbors in the level below (i.e., 26 total).



2. Keypoint Localization (cont'd)

- SIFT uses the Hessian matrix (for efficiency).
  - i.e., Hessian encodes principal curvatures

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

α: largest eigenvalue (λ_{max})
β: smallest eigenvalue (λ_{min})
(proportional to principal curvatures)

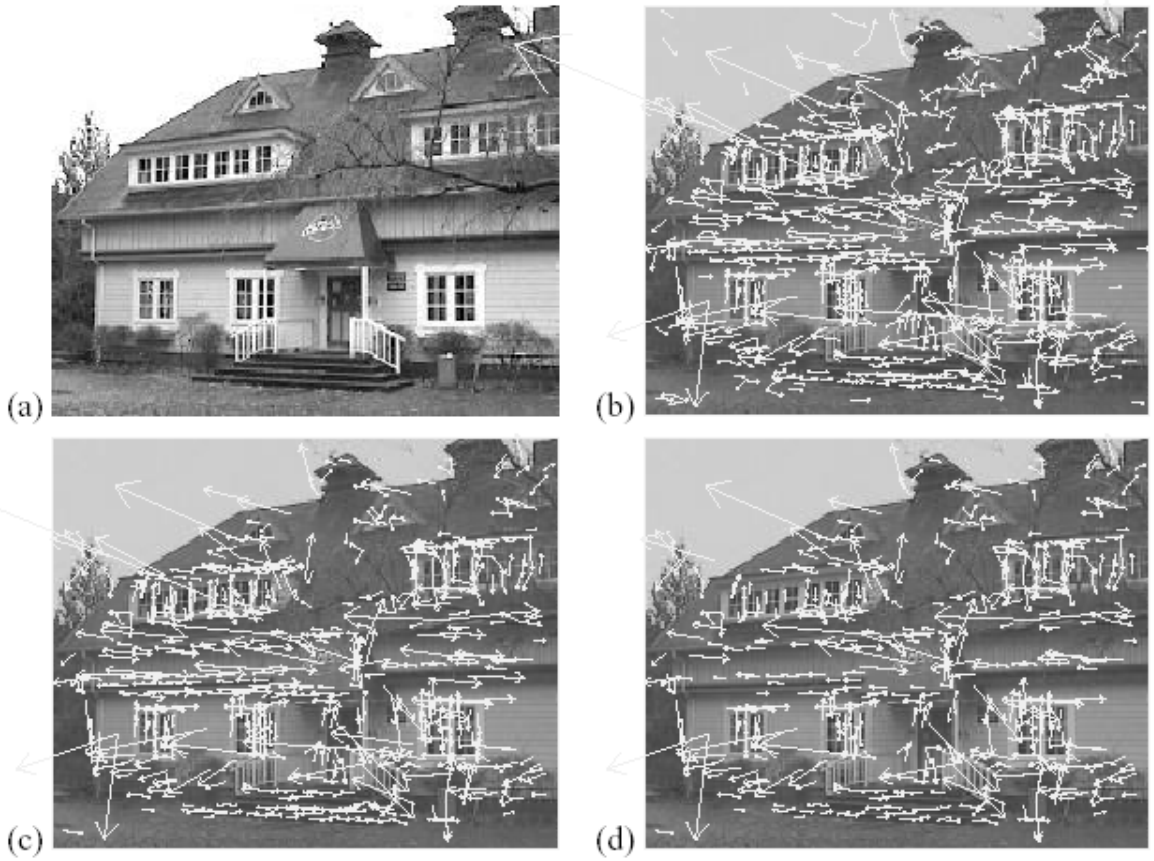
$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

$$\rightarrow \quad \frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r},$$

(r = α/β)

Reject keypoint if: $\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$ (SIFT uses r = 10)

2. Keypoint Localization (cont'd)



(a) 233x189 image

(b) 832 DoG extrema

(c) 729 left after low contrast threshold

(d) 536 left after testing ratio based on Hessian

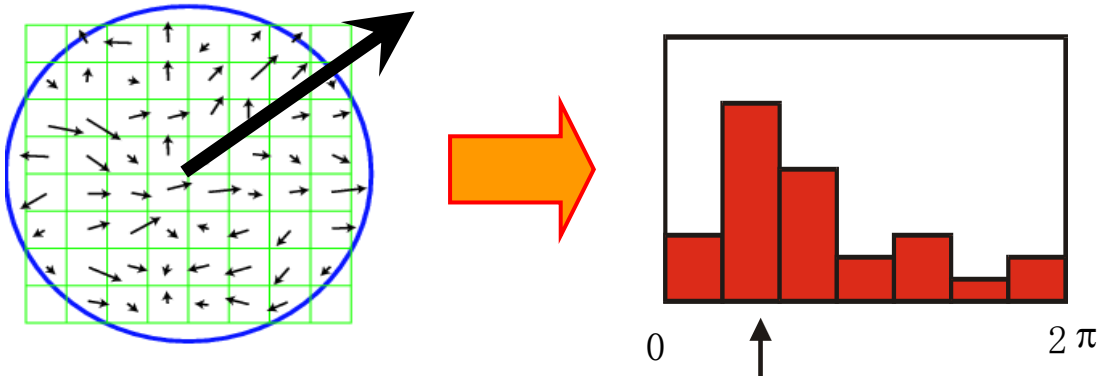
3. Orientation Assignment

- Create histogram of gradient directions, within a region around the keypoint, at selected scale:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

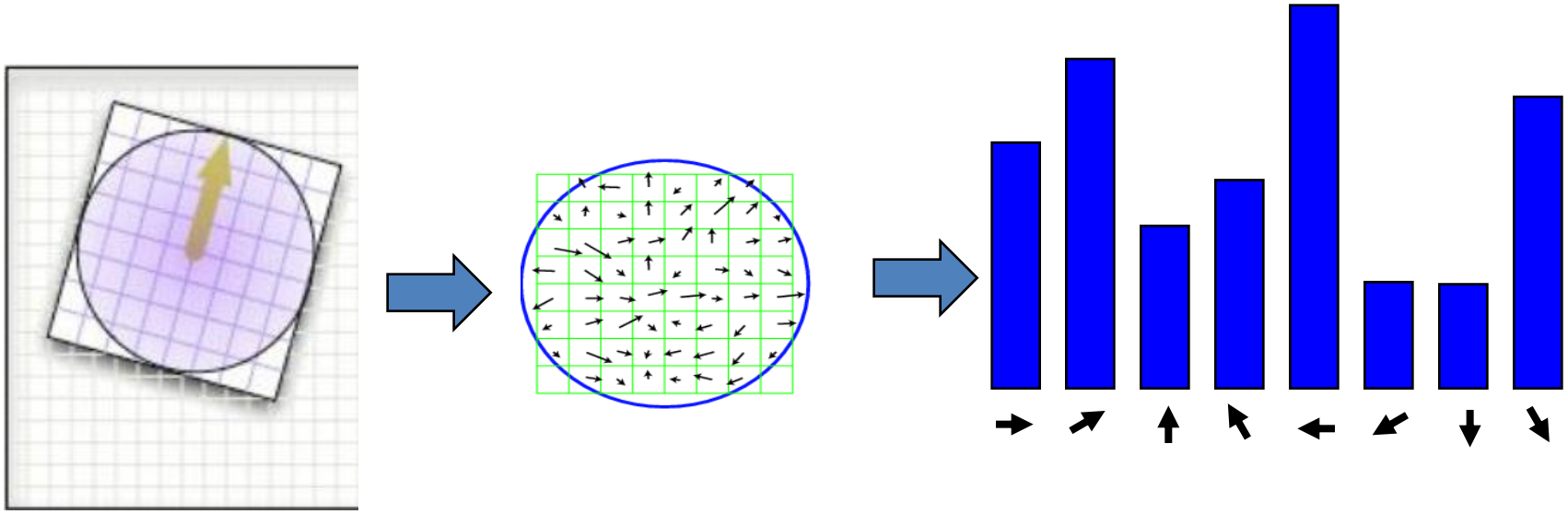
$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = a\tan 2((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$



36 bins (i.e., 10^o per bin)

4. Keypoint Descriptor



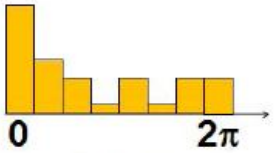
8 bins

4. Keypoint Descriptor (cont'd)

1. Take a 16 x16 window around detected interest point.

2. Divide into a 4x4 grid of cells.

3. Compute histogram in each cell.



(8 bins)

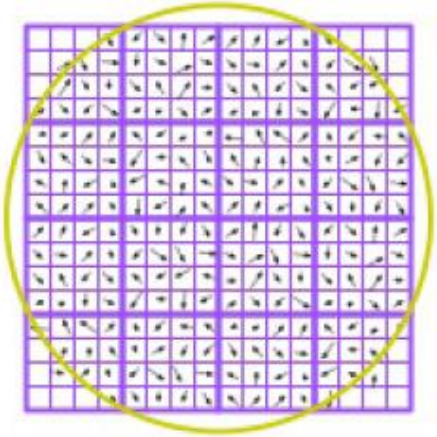
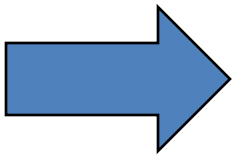
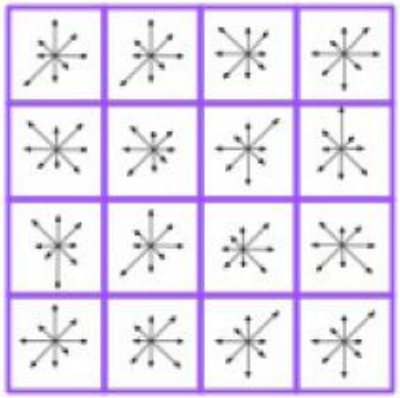


Image gradients





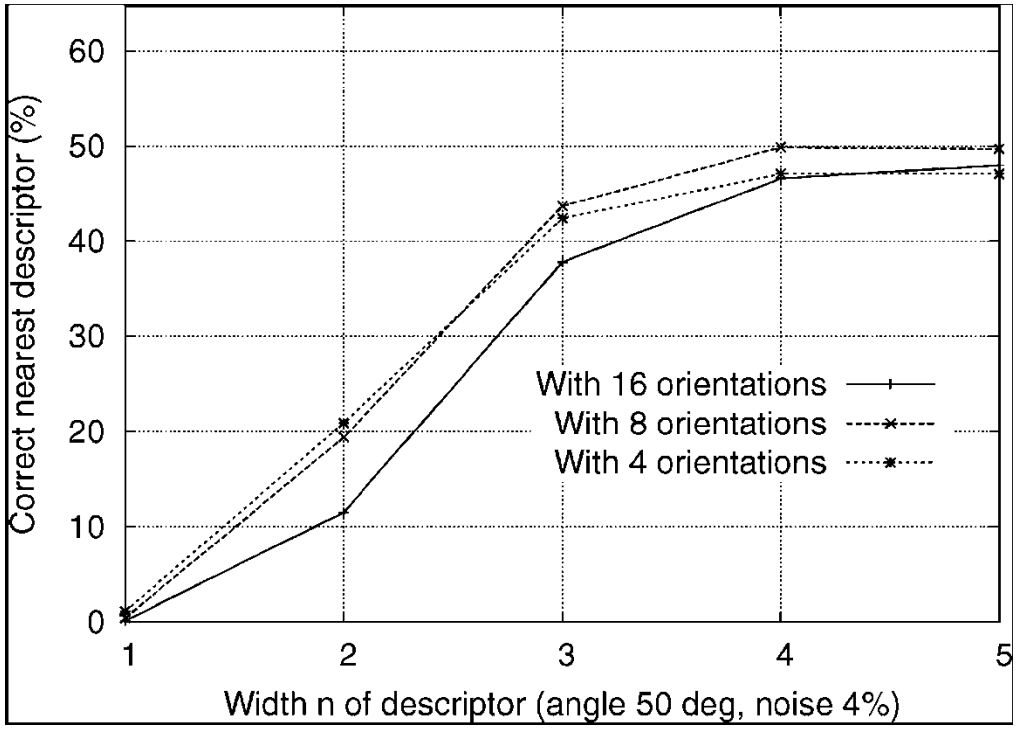
Keypoint descriptor

16 histograms x 8 orientations
= 128 features

4. Keypoint Descriptor (cont'd)

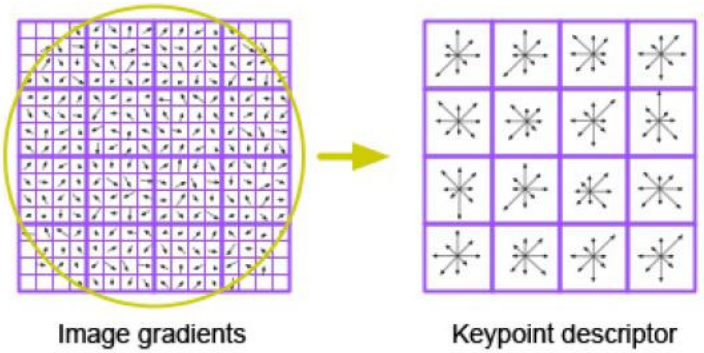
- Descriptor depends on two main parameters:
  (1) number of orientations r
  (2) n x n array of orientation histograms
  •

rn^2 features



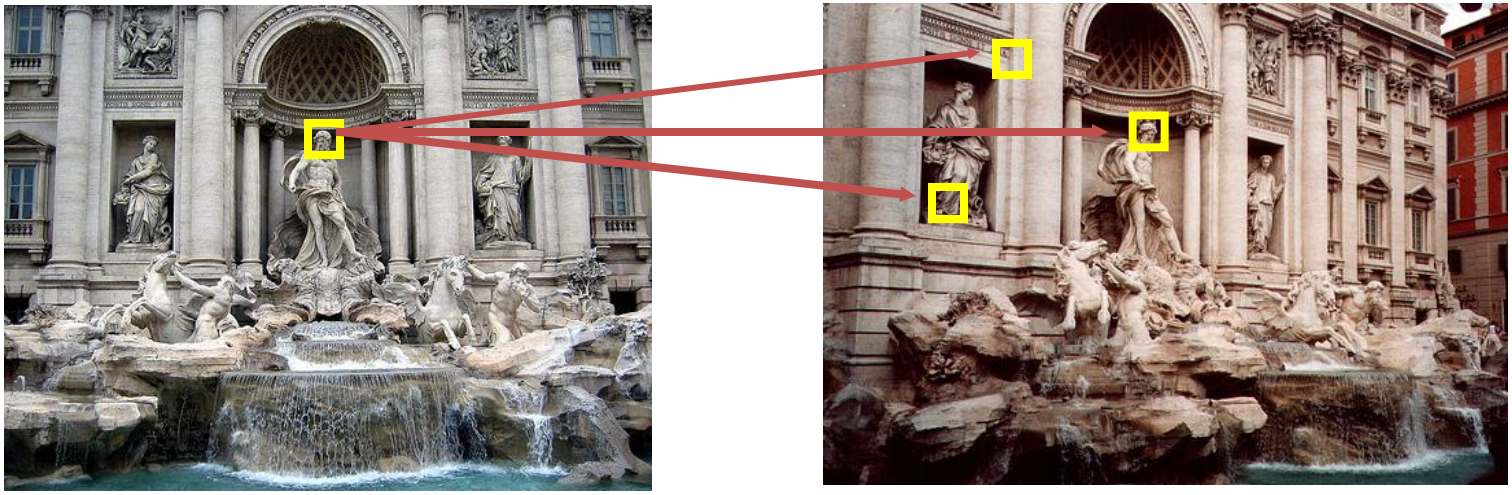
SIFT: r=8, n=4

128 features



Matching SIFT features

- Given a feature in I_1, how to find the best match in I_2?
  1. Define distance function that compares two descriptors.
  2. Test all the features in I_2, find the one with min distance.



I_1 I_2

SURF: Speeded Up Robust Features

- Speed-up computations by fast approximation of (i) Hessian matrix and (ii) descriptor using “integral images”.

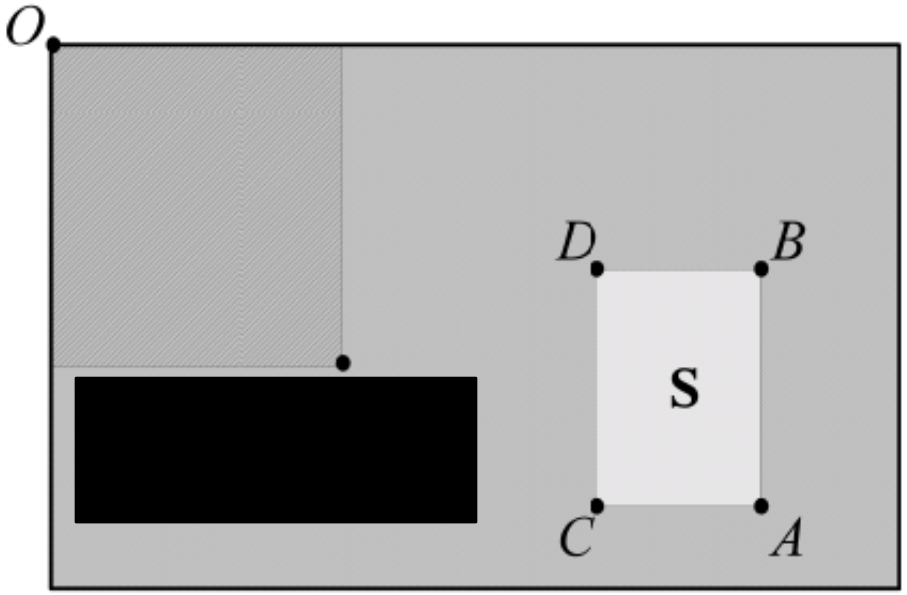
$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix},$$

- What is an “integral image”?

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, “SURF: Speeded Up Robust Features”, **European Computer Vision Conference (ECCV),** 2006.

Integral Image

- The integral image $I_{\Sigma}(x,y)$ of an image I(x, y) represents the sum of all pixels in I(x,y) of a rectangular region formed by (0,0) and (x,y).



Using integral images, it takes only four array references to calculate the sum of pixels over a rectangular region of any size.

$$\mathbf{S}=A-B-C+D$$

Hausdorff distance based matching

- Hausdorff Distance
  - Given two finite point sets $A=\{a_1,a_2,\ldots,a_n\}$ and $B=\{b_1,b_2,\ldots,b_n\}$, the Hausdorff distance is given by:

$$H(A,B)=\max\{h(A,B),h(B,A)\} \qquad where,$$

$$h(A,B)=\max_{a\in A}\min_{b\in B}\left(\|a-b\|\right)$$

Measures the mismatch between the sets A and B by finding the point of A that is farthest from any point of B.

- Works on edge images rather than intensity images
- Robust against mild rotation and scale transformations

Edge based Hausdorff matching

- Conventional Hausdorff distance is very sensitive to 'outliers'
- Sensitivity is reduced by using ranked distance instead of maximum distance

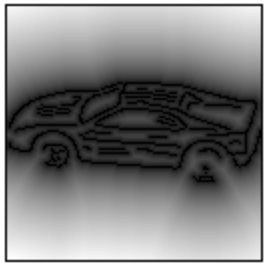
$$h_k(A,B) = K^{th}_{a\in A} \min_{b\in B} \|a-b\|$$

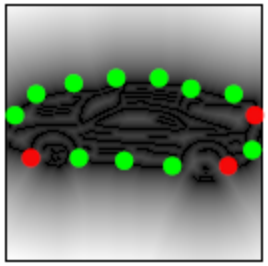
- K 'best' matching points are 'automatically' selected without pre-specifying the part of the image to be matched
- Works well for images where conventional Hausdorff distance based matching fails.

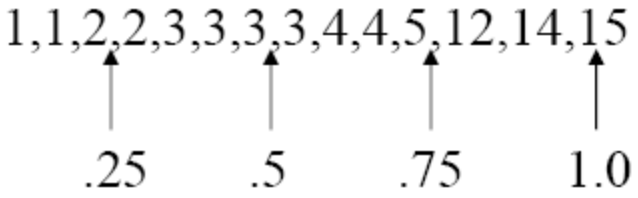
$$H_{KL}(A,B) = \max(h_K(A,B), h_L(B,A))$$

Hausdorff Matching

- Partial (or fractional) Hausdorff distance to address robustness to outliers
  - Rank rather than maximum
    - $h_k(A,B) = \text{kth}_{a \in A} \min_{b \in B} \| a-b \| = \text{kth}_{a \in A} D_B(a)$
  - K-th largest value of D_B at locations given by A
  - Often specify as fraction f rather than rank
    - 0.5, median of distances; 0.75, 75th percentile







Edge based Hausdorff matching (Contd.)

- Partial Hausdorff distance is computed through a Distance Transformation (DT) using iterated local operations

- Distances are approximated by propagating distances between the neighboring pixels over the entire image

Algorithm Flow

- Preprocess the template image to extract the low-level features (edges)
- Compute the DT of the template edge image
- For all possible translations of the template image, do
  1. Extract the corresponding portion from the reference edge image
  2. Extract the corresponding portion from the DT of reference image
  3. Multiply the template edge image to reference image DT
  4. Sort the non-zero values and pick the Kth Value
  5. Multiply the reference edge image to template image DT
  6. Sort the non-zero values and pick the Lth Value
  7. Partial Hausdorff distance is the maximum of Kth and Lth Values
- Select the Translation value which minimizes the partial Hausdorff distance

DT : Sequential Algorithm

- Start with zero-infinity image : set each edge pixel to 0 and each non-edge pixel to infinity.
- Make 2 passes over the image with a mask:
  - 1. Forward, from left to right and top to bottom
  - 2. Backward, from right to left and from bottom to top.

d2	d1	d2
d1	0	

Forward Mask

	0	d1
d2	d1	d2

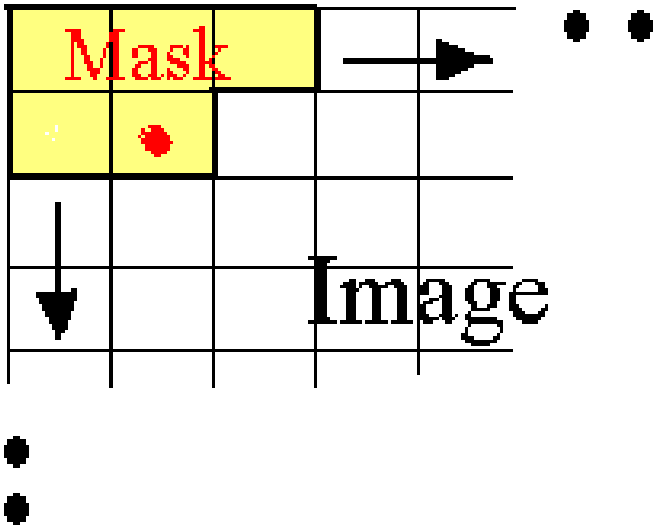
Backward Mask

*d*1 and *d*2, are added to the pixel values in the distance map and the new value of the zero pixel is the minimum of the five sums.

DT Parallel Algorithm

- For each position of mask on image,
- $V_{i,j}$ = minimum($v_{i-1,j-1}$+d2,$v_{i-1,j}$ +d1,$v_{i-,j+1}$+d2, $v_{i,j-1}$+d1,$v_{i,j}$)

d2	d1	d2
d1	0	



Distance Transformation

- Edge image is converted into zero-infinity image by replacing edges with zeros and non-edges with infinity
- Two passes are made over the image
  - DT = zero-infinity image

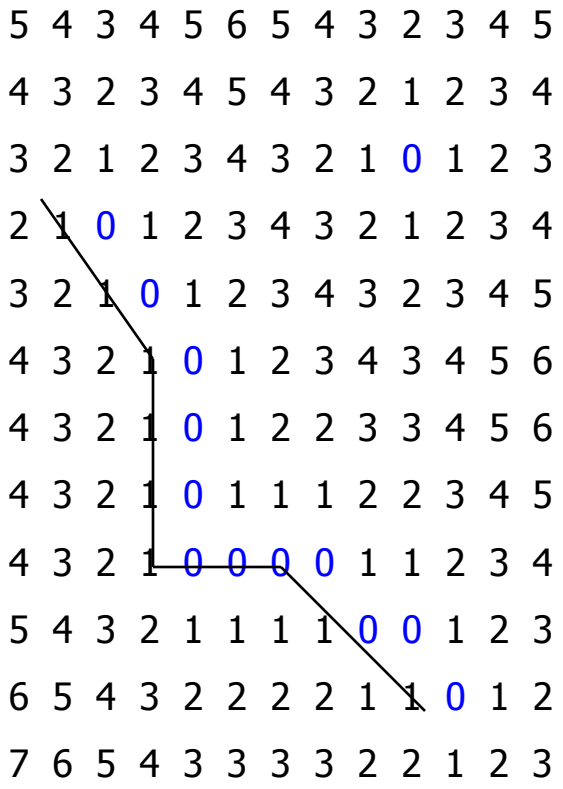
Forward Pass:

```
for i=2, …. last row, do
   for j=2, …..last column, do
      DT[i,j] = min ( DT[i-1,j-1]+4, DT[i-1,j]+3, DT[i-1,j+1]+4,
                                          DT[i,j-1]+3, DT[i,j] )
```

Backward Pass:

```
for i=last row-1, ….1, do
   for j=last column-1, …. 1, do
      DT[i,j] = min ( DT[i,j], DT[i,j+1]+3, DT[i+1,j-1]+4,
                                          DT[i+1,j]+3, DT[i+1,j+1]+4 )
```

Example distance transform



- Set image feature points to 0
- Set all neighbors of 0 values to 1
- Set all unset neighbors of 1 values to 2
- Set all unset neighbors of 2 to 3, etc.

Parallel computation:

- at each stage, every unassigned pixel P checks its neighbors;
- if any neighbor has a distance label of d, then pixel P becomes d+1

- Manhatten distance used here.
- Can use scaled Euclidean distance, where 4-neighbors are distance 3 and diagonal neighbors are distance 4.

Comparing Portions of Shapes

- Target is to find the K points of the model set which are closest to the points of the image set.

- *'Automatically' select the K 'best matching' points of B*
  - It identifies subsets of the model of size K that minimizes the Hausdorff distance
  - Don't need to pre-specify which part of the model is being compared

Problems

- Gives false matches where images are contaminated with spatial noise

- Distance values are reduced because of erroneous edges reducing overall ranked distance

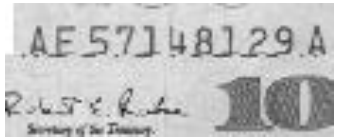
Enhanced Hausdorff distance based algorithm

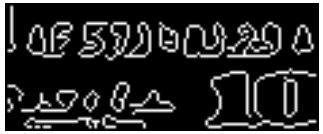
- Uses orientation information

- More Robust than Partial Hausdorff Distance based algorithm
- Composed of
  - Gradient Computation
  - Edge Extraction
  - DT map Construction
  - Oriented Hausdorff Similarity (OHS) Calculation

Example 1

- Sample Template Image





- Reference image

Example 1 Results

- Cross Correlation Results

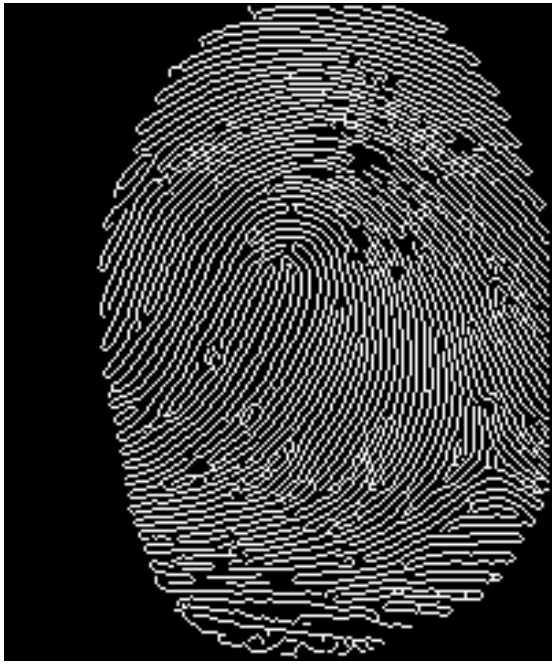
- Conventional Hausdorff Results

Enhanced Hausdorff results

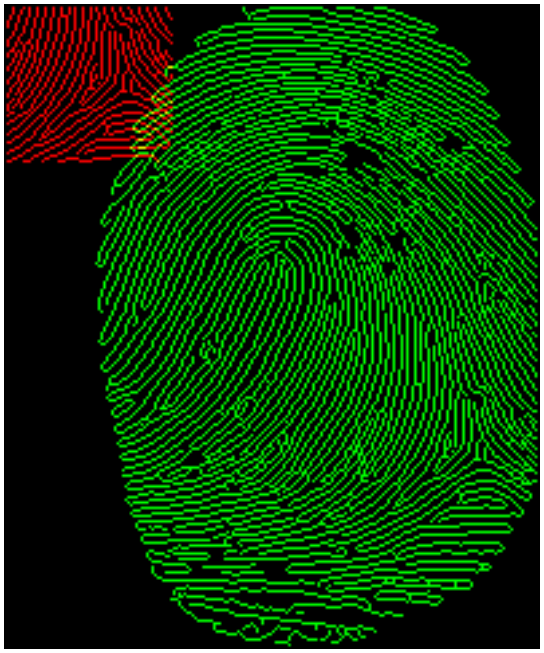
Example 2

- Sample Template Image

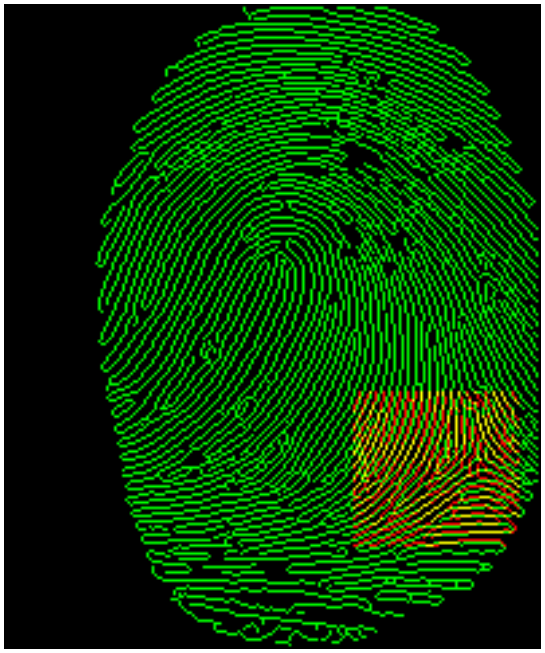
- Reference image



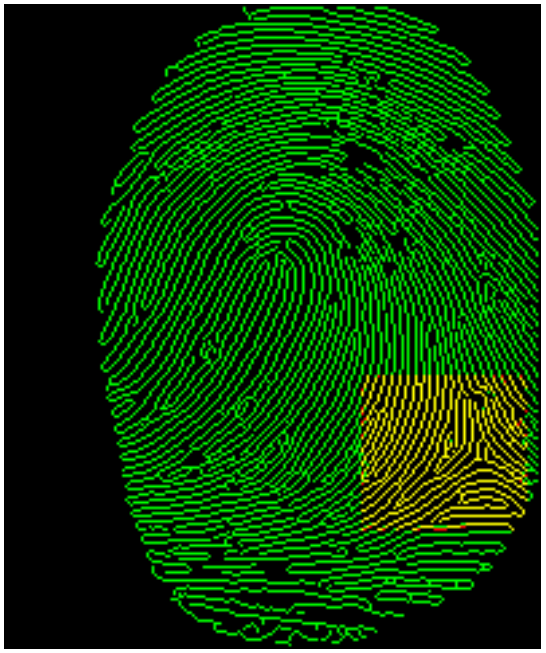
Example 2 Results



Cross Correlation Results



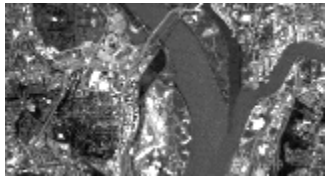
Conventional Hausdorff Results



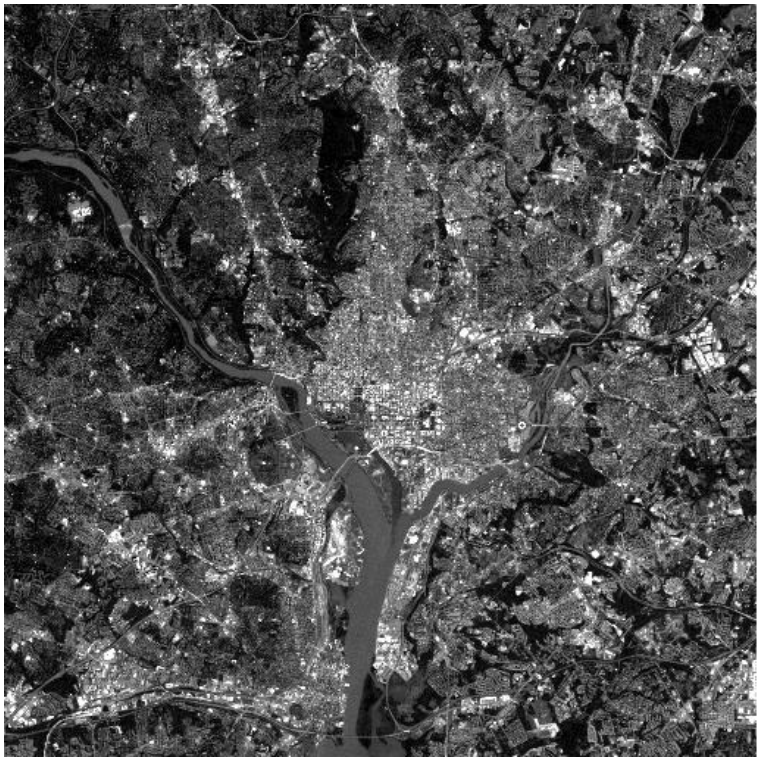
Enhanced Hausdorff results

Example 3

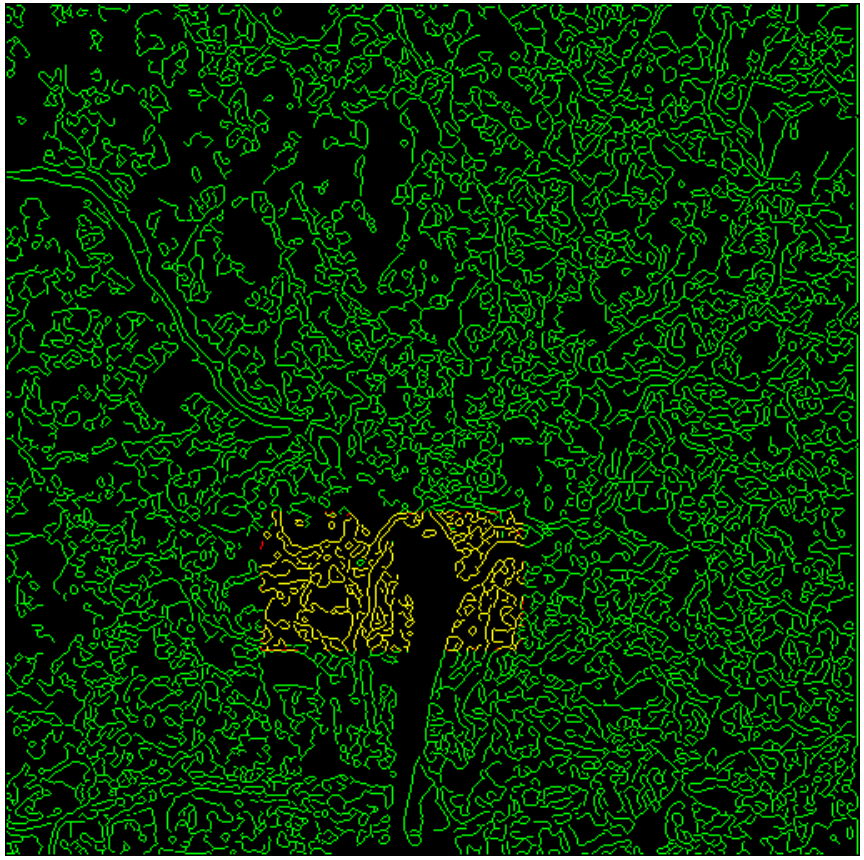
- Sample Template Image



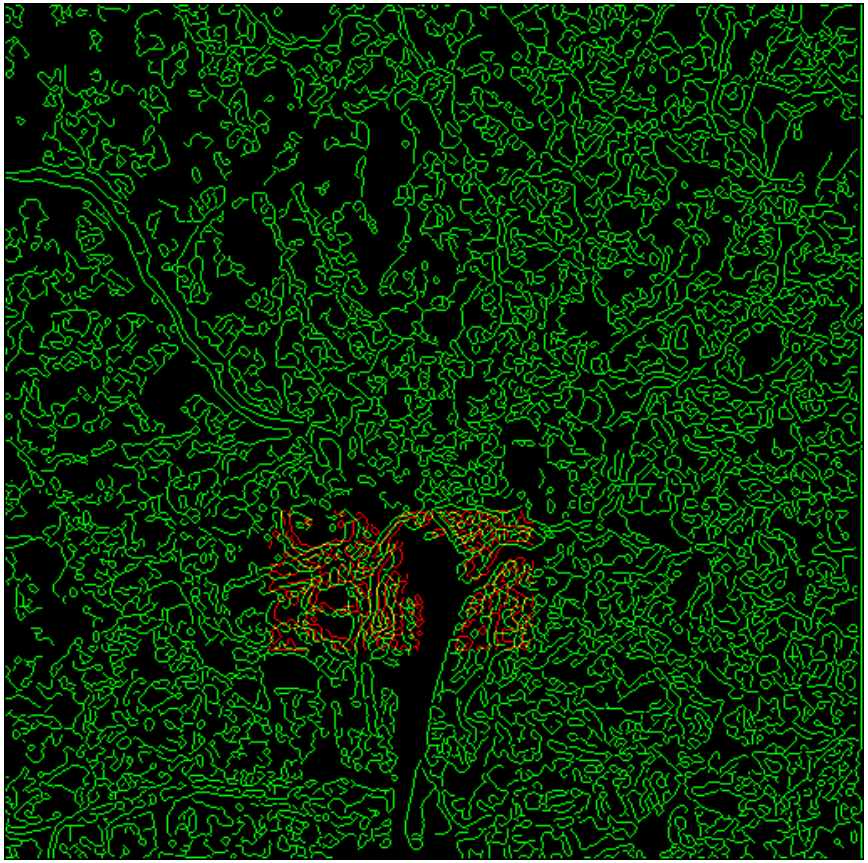
- Reference image



Example 3 Results

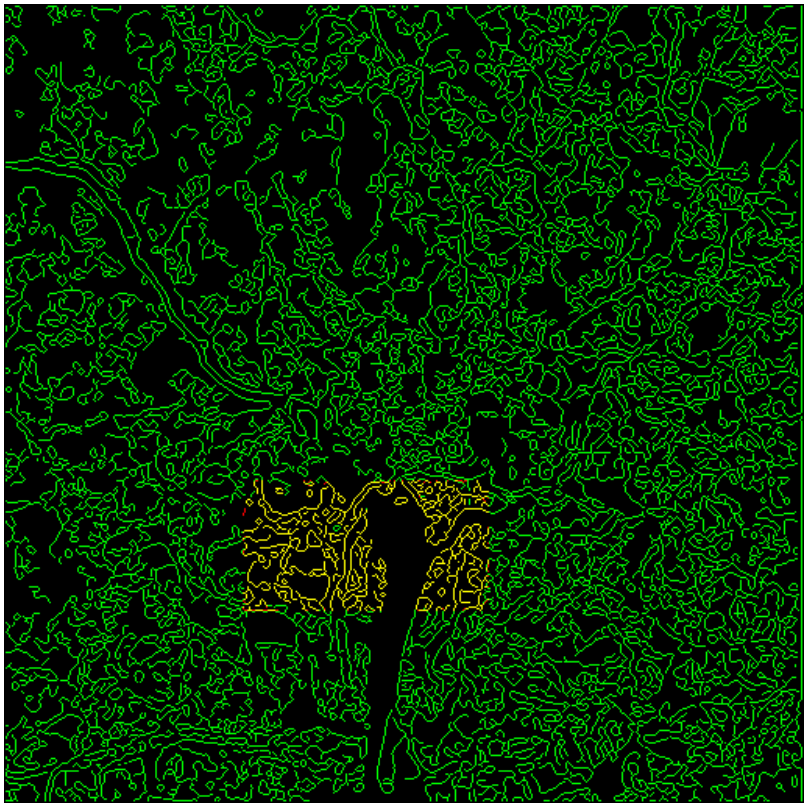


Cross Correlation Results



Conventional Hausdorff Results

Example 3 Results

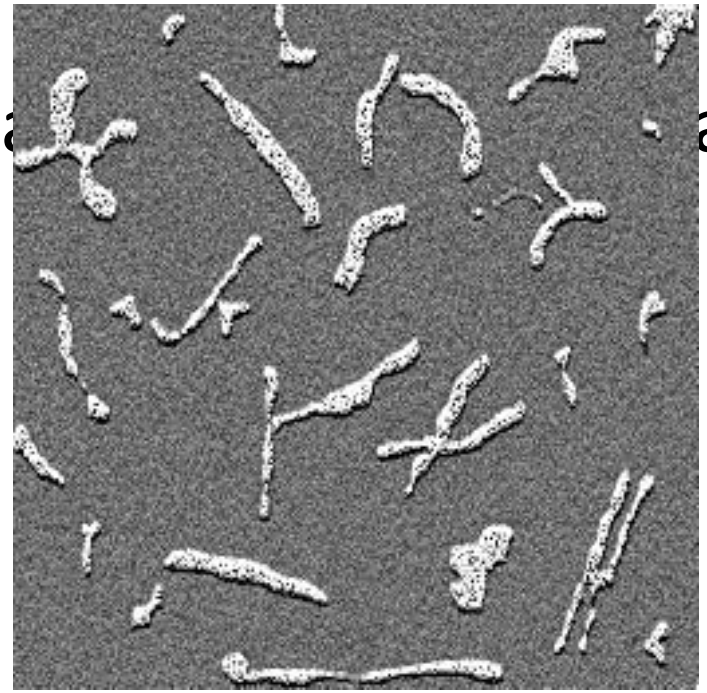


Enhanced Hausdorff results

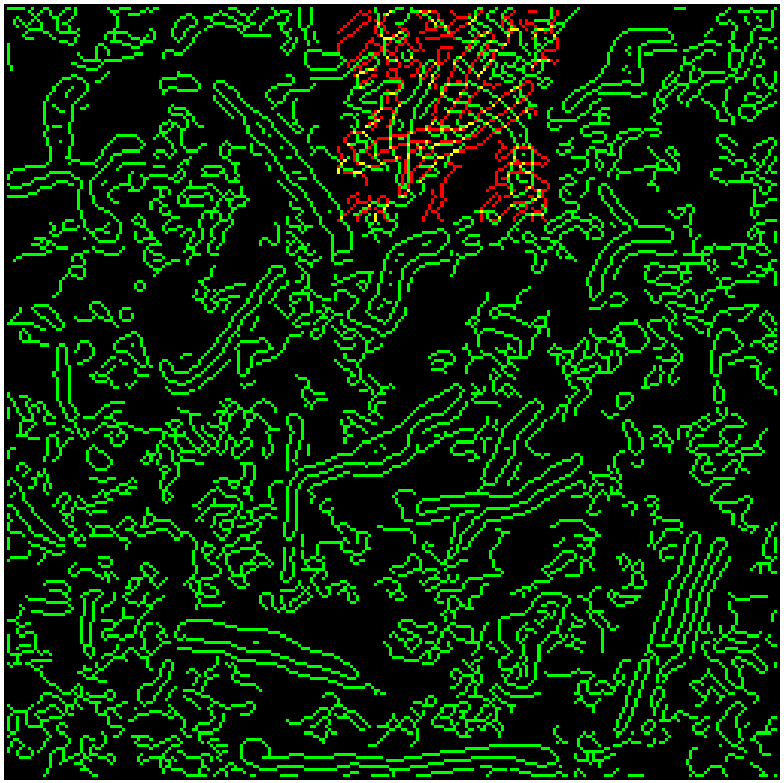
Examp

- Sample Template Image

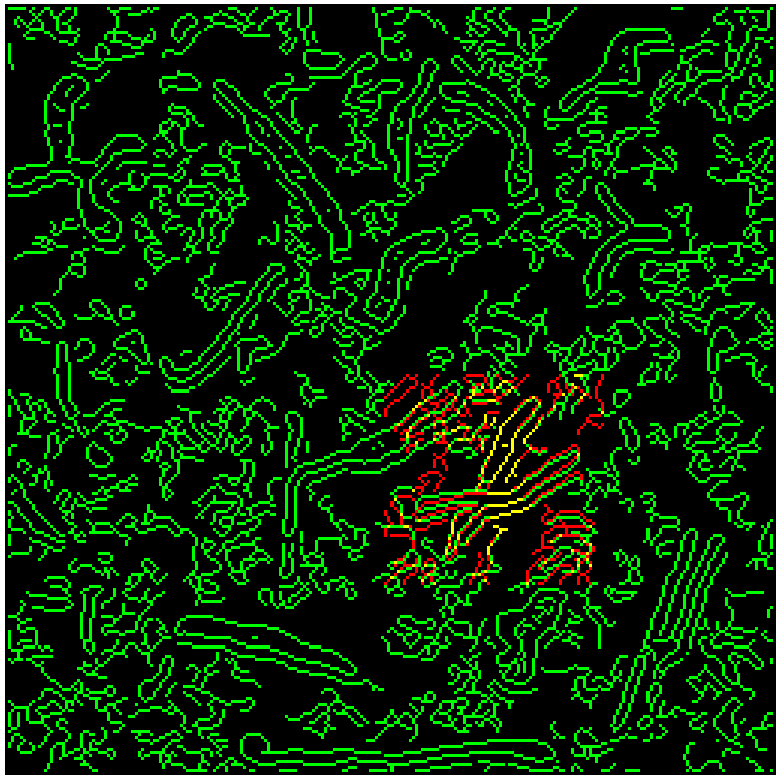
- Sa ag



Example 4 Results



Partial Hausdorff Distance Results



Enhanced Hausdorff Distance Results

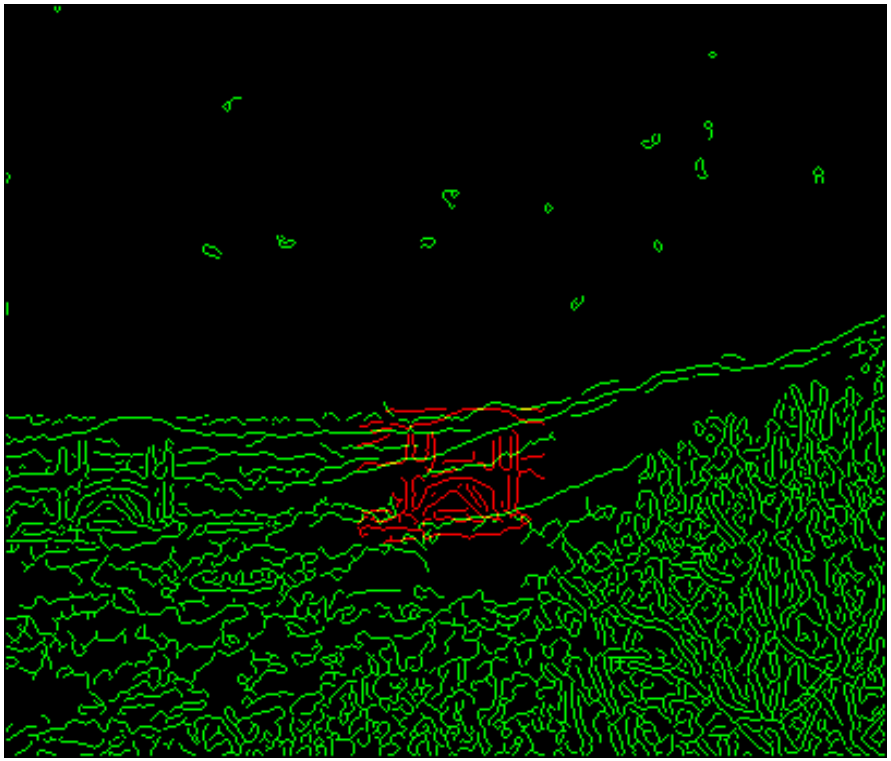
Example 5

- Sample Template Image

- Reference image

Example 5 Results

Cross Correlation Results



Conventional Hausdorff Results

Example 5 Results

Enhanced Hausdorff results