Image Features: Descriptors

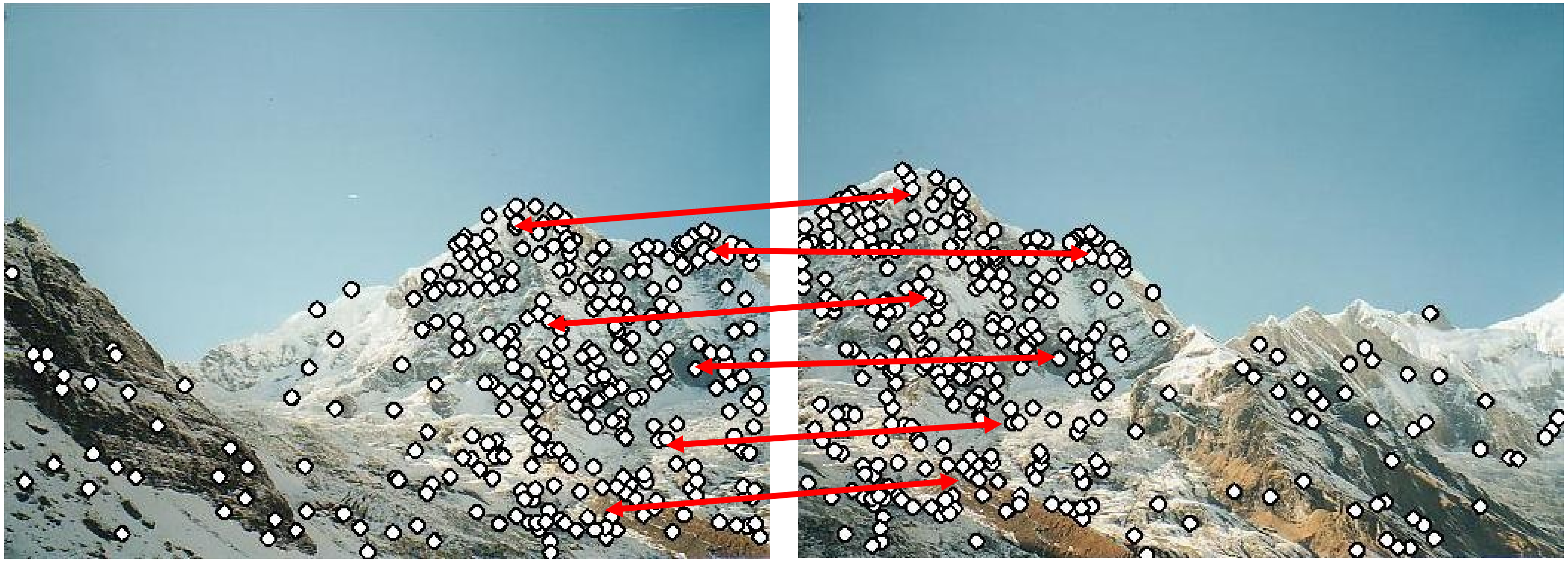
Slides adapted from Lana Lazebnik, Kirill Dyagilev, Ayelet Dominitz, Matthew Brown and Robert Pless

Recall: Image Feature Motivation

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Image Feature Motivation

- Motivation: panorama stitching
  - We have two images – how do we combine them?



Step 1: find features

Step 2: match features (how?)

Image Features

- One characteristic of good image features is *locality*
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion
- We describe the feature by a patch of the image centered at the interest point

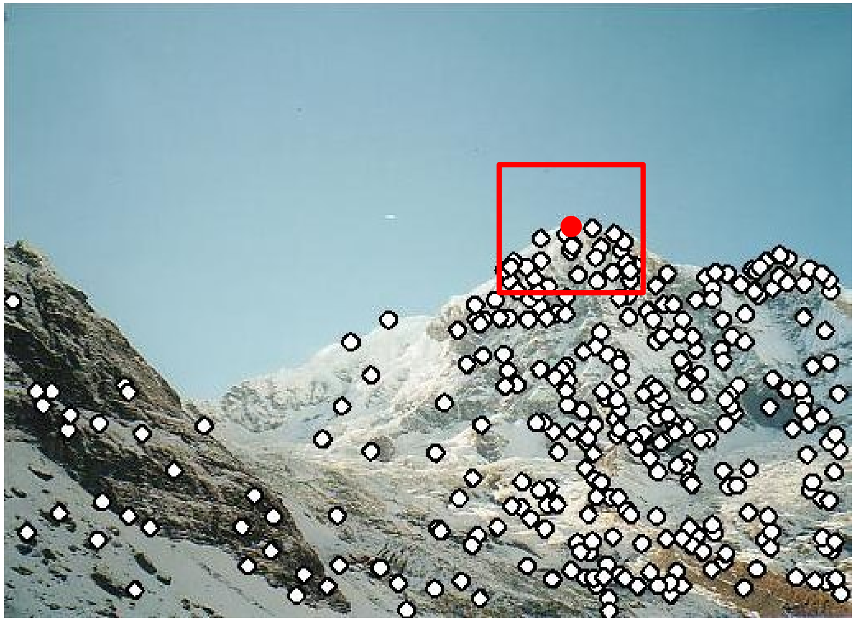


Image Features

- Sparse image features have:
  - a location (interest point)
  - descriptor
- So far, we've focused on location
  - edge points
  - corner points
  - blobs
- Today, we'll cover 2 feature descriptors
  - MOPS (Brown et al., 2005)
  - SIFT (Lowe, 2004)

But First…
Scale Invariance → Affine Invariance

- “Same” region may be rotated or stretched in two images
  - In addition to translation & scale invariance, we want *affine invariance*

Recall: 2nd Moment Matrix

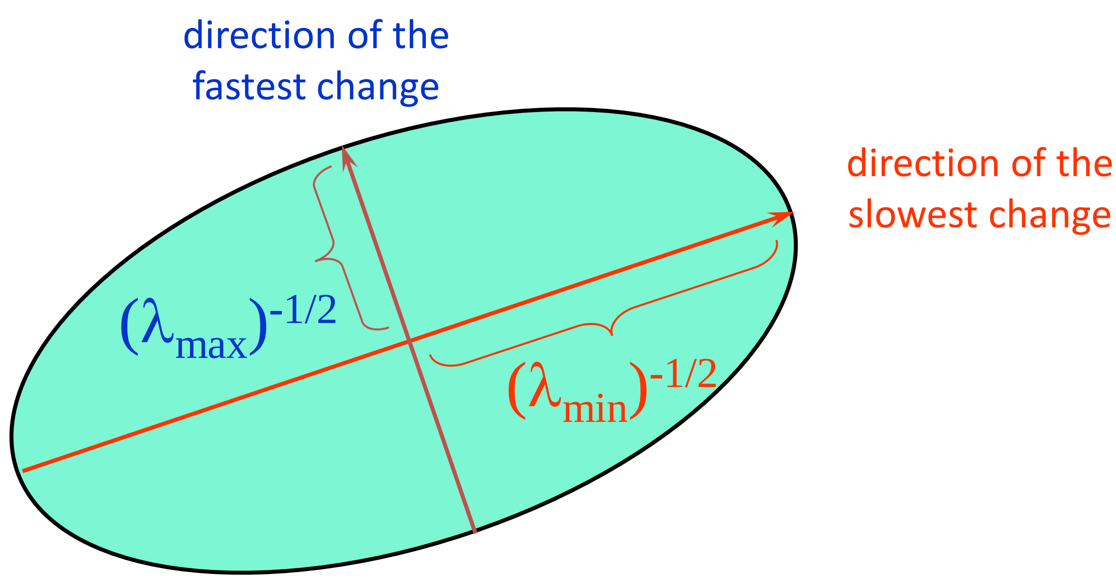
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

We can visualize M as an ellipse with axis lengths determined by the eigenvalues and orientation determined by R

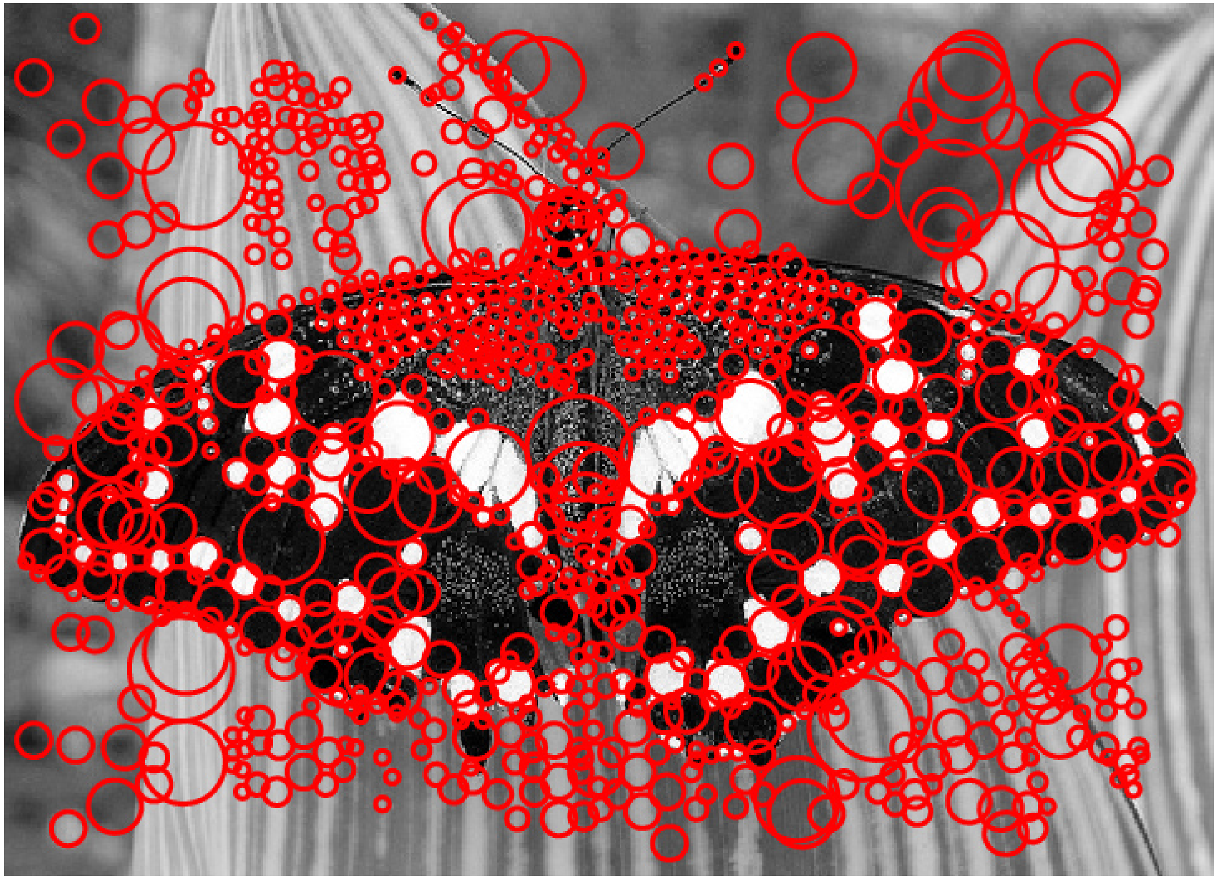
Rotation Matrix

Ellipse equation:

$$[u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$

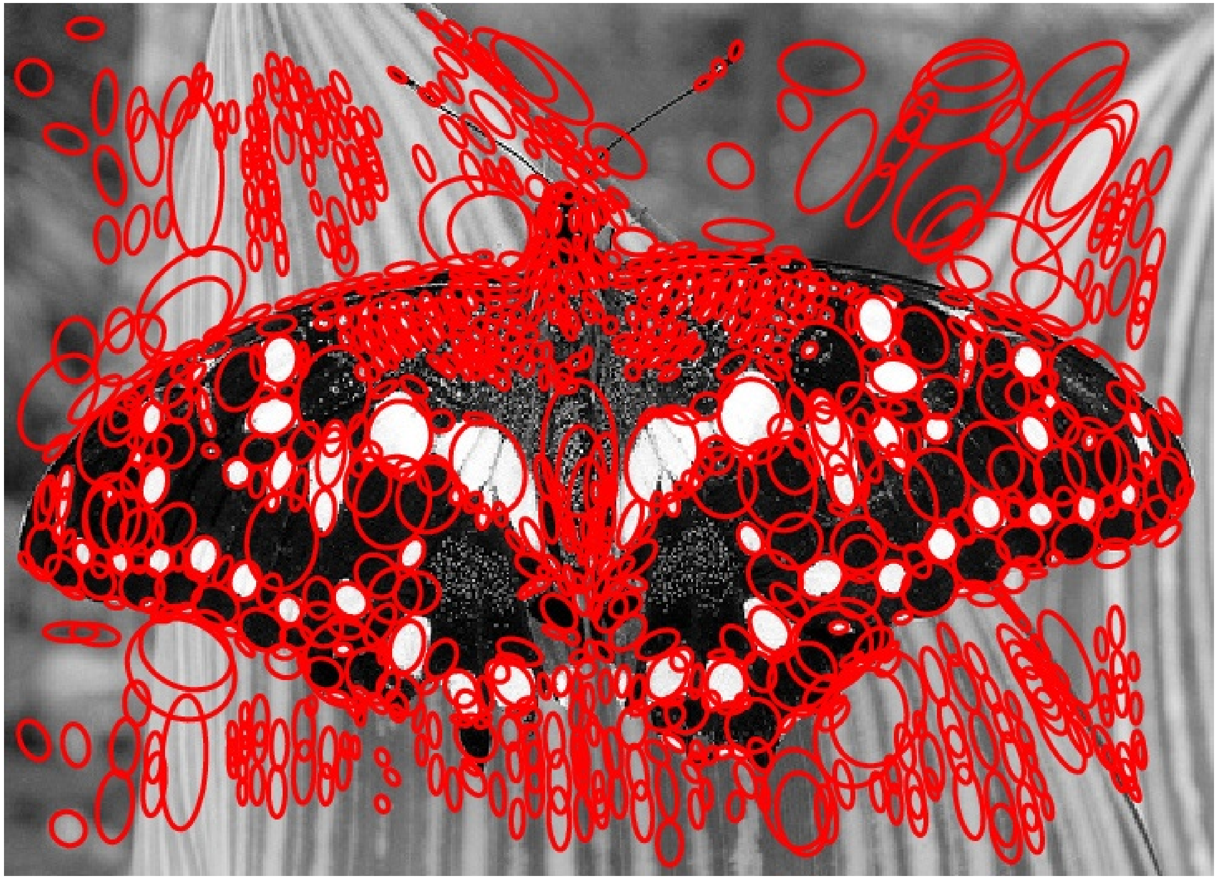


Affine adaptation example



Scale-invariant regions (blobs)

Affine adaptation example

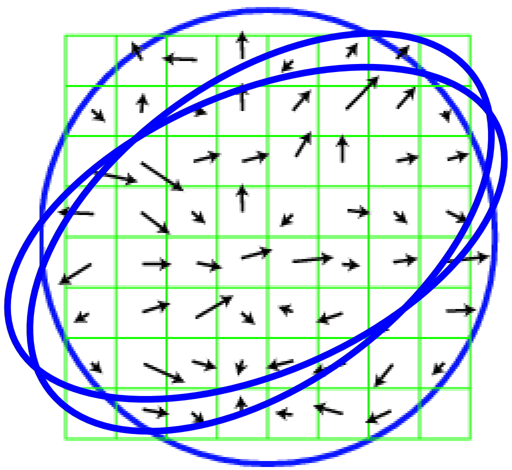


Affine-adapted blobs

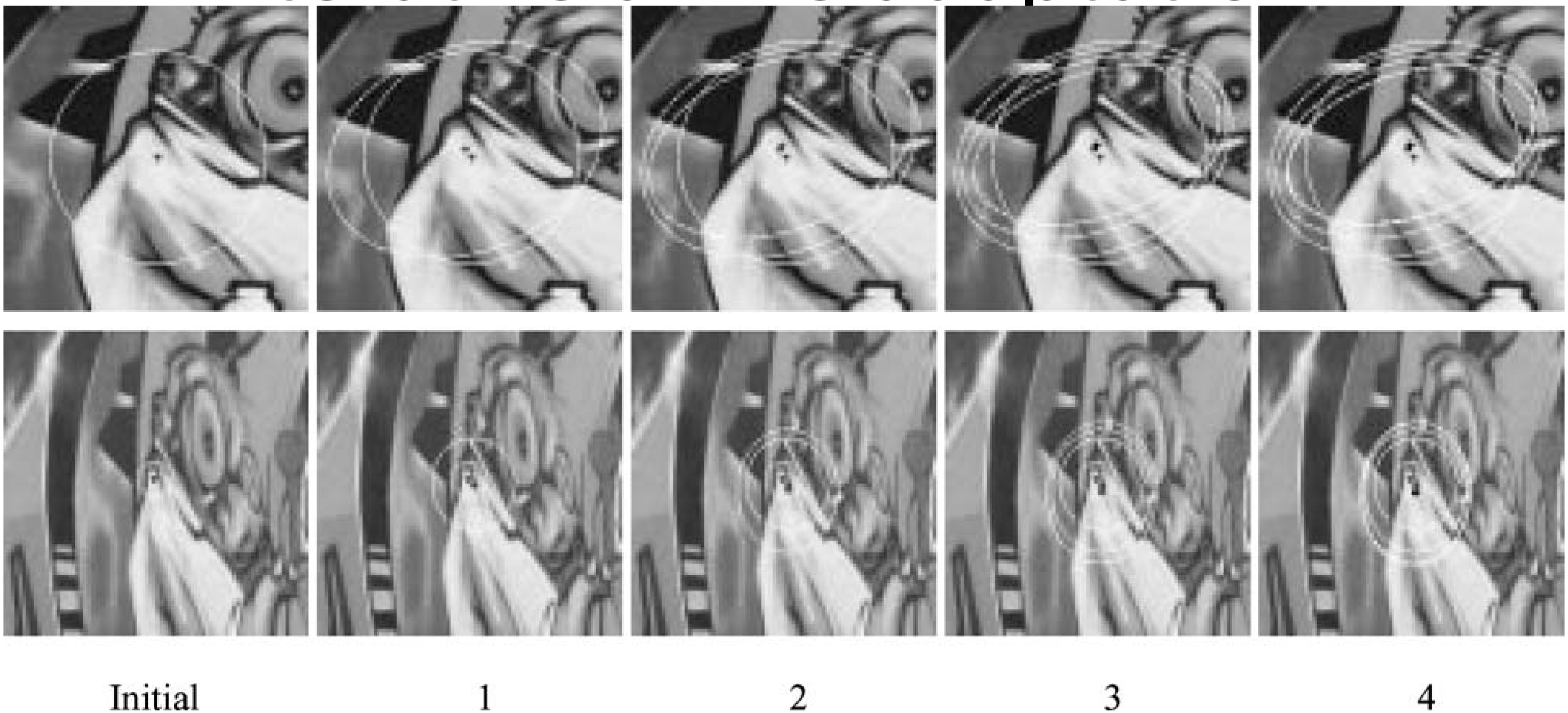
Affine adaptation

- Problem: the second moment "window" determined by weights $w(x,y)$ must match the characteristic shape of the region
  - *"Chicken an egg" problem: Pixels "vote" for shape. Shape determines which pixels get to vote.*

- Solution: iterative approach
  – Use a circular window to compute second moment matrix
  – Perform affine adaptation to find an ellipse-shaped window
  – Recompute second moment matrix using new window and iterate



Iterative affine adaptation



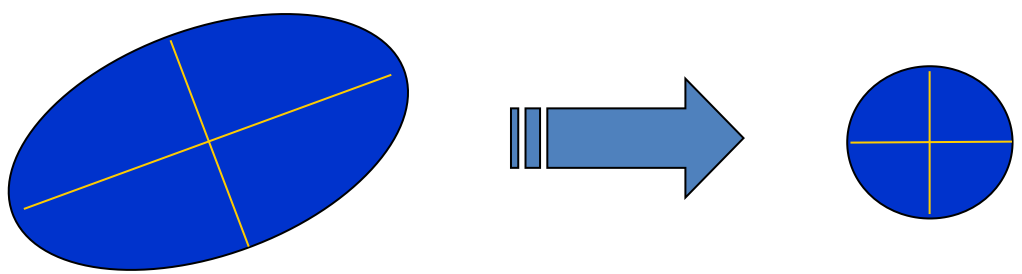
K. Mikolajczyk and C. Schmid, [Scale and Affine invariant interest point detectors](), IJCV 60(1):63-86, 2004.

[http://www.robots.ox.ac.uk/~vgg/research/affine/]()

Affine normalization

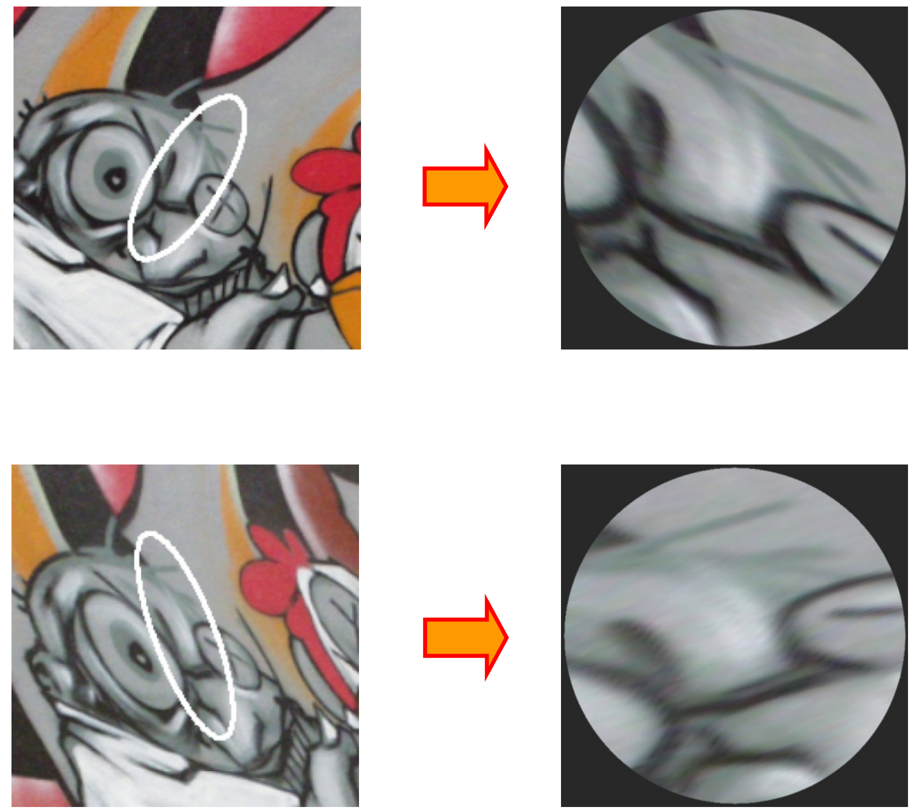
- The second moment ellipse can be viewed as the “characteristic shape” of a region
- How do we compare two regions of different shape?

- We can normalize the region by transforming the ellipse into a unit circle



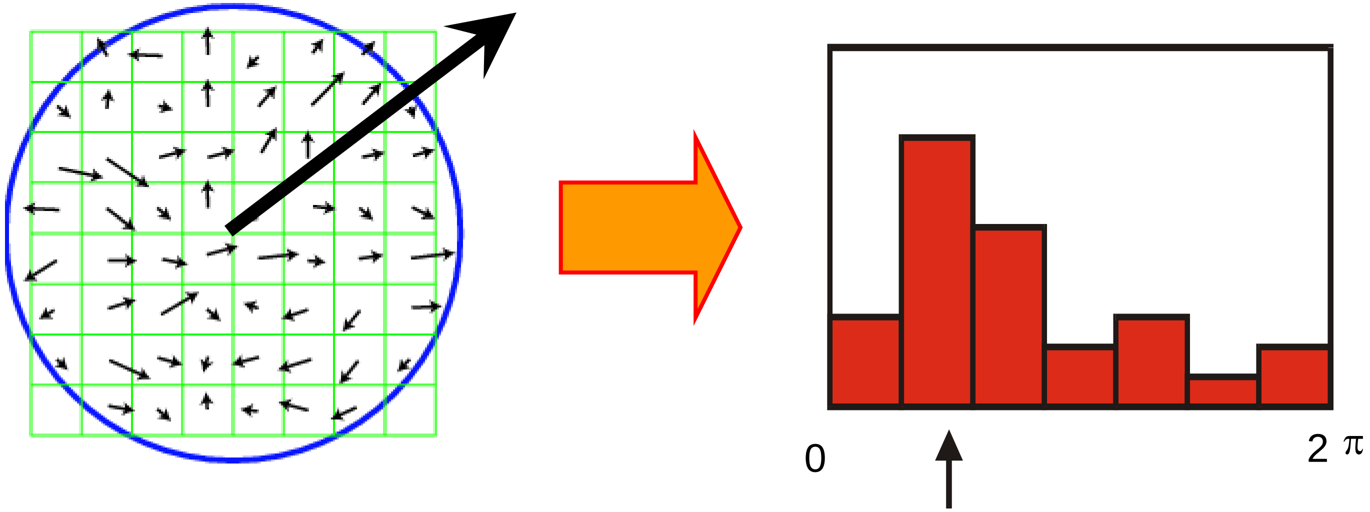
Orientation ambiguity

- There is no unique transformation from an ellipse to a unit circle
  - We can rotate or flip a unit circle, and it still stays a unit circle



Orientation ambiguity

- There is no unique transformation from an ellipse to a unit circle
  - We can rotate or flip a unit circle, and it still stays a unit circle
- So, to assign a unique orientation to keypoints:
  - Create histogram of local gradient directions in the patch
  - Assign canonical orientation at peak of smoothed histogram



Ok, What's Left?

- We have:
  - Interest point location
  - Scale
  - Orientation

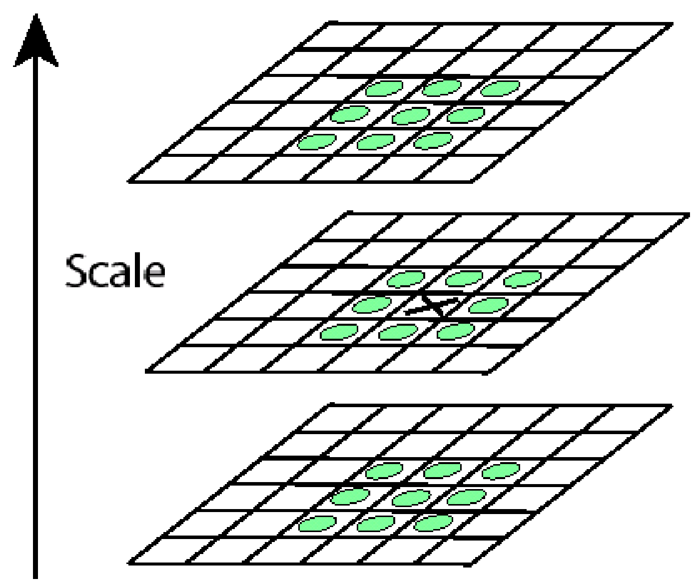
- We need to describe the image patch in that region
- Two methods: MOPS & SIFT

Scale-Invariant Feature Transform (SIFT) Algorithm

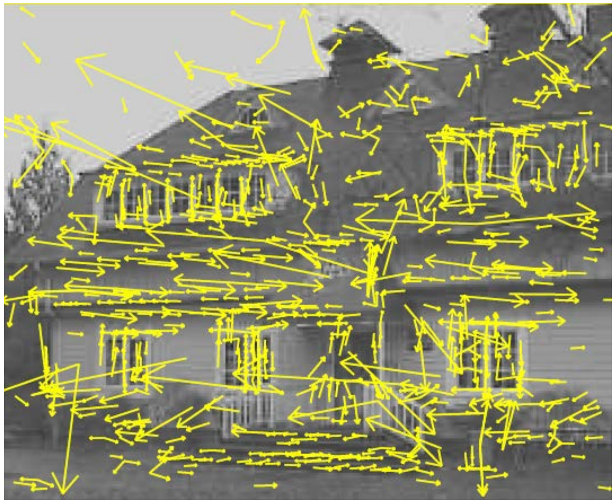
1. Interest-point detection
2. Feature localization
3. Orientation assignment
4. Feature descriptor

Step 1: Interest Point Detection

- Calculate Difference of Gaussian at varying scales
- X is selected if it is larger or smaller than all 26 neighbors

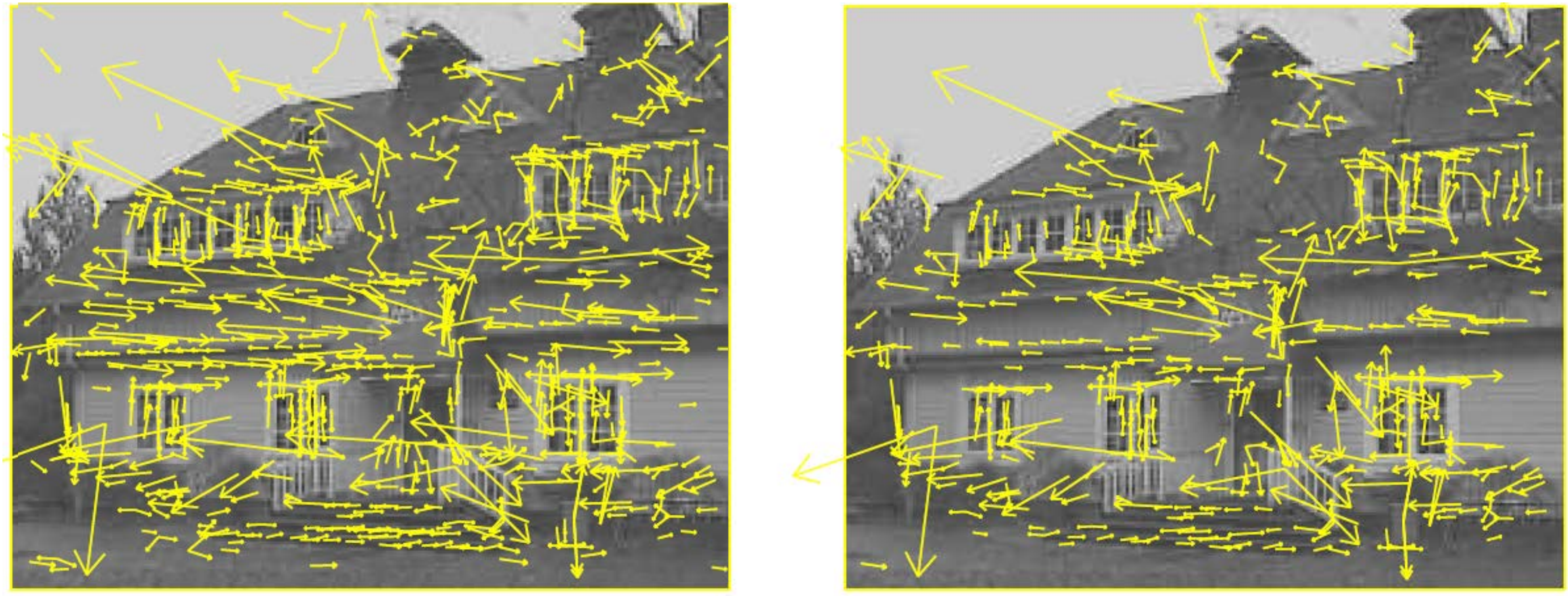


Step 2: Feature Localization



- Too many keypoints, some are unstable:
  - points with low contrast (sensitive to noise)
  - points that are localized along an edge
    - prefer points on corners

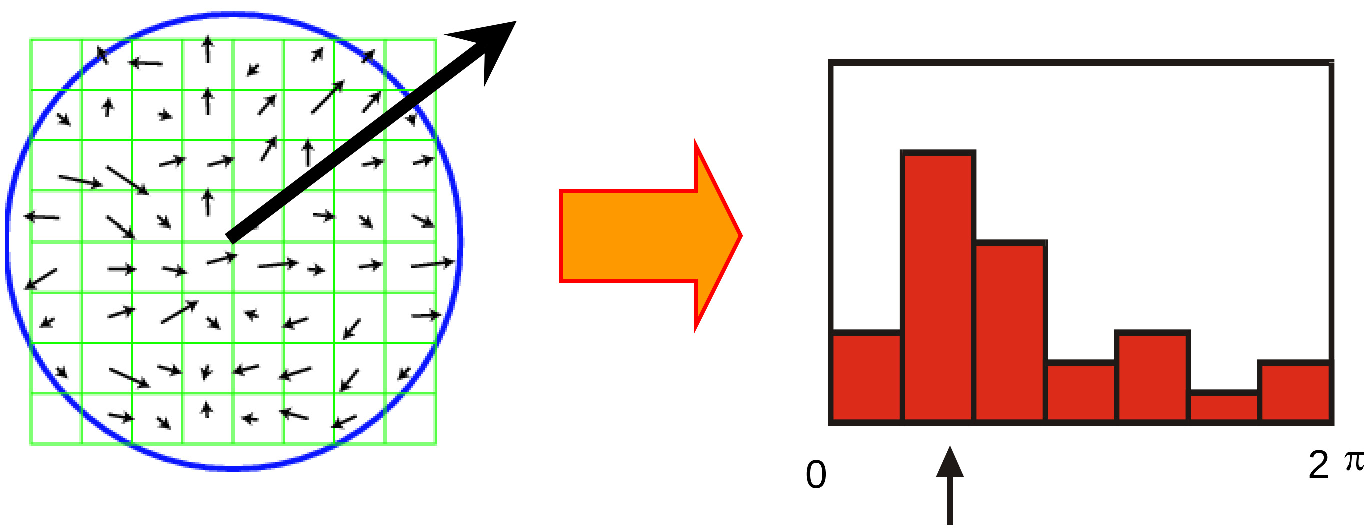
Step 2: Feature Localization



536 out of 832 are left after contrast thresholding and cornerness thresholding

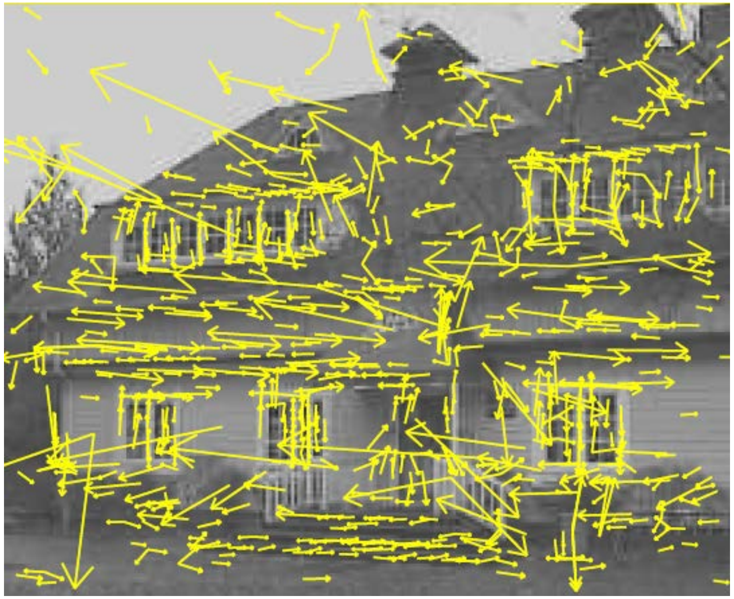
Step 3: Orientation assignment

- Required: Rotation invariance of features
- Solution:
  - Assign orientation to feature based on local gradients
  - Transform relative data accordingly
  - For location of multiple peaks multiply key point



Step 4: Feature Descriptor

- So far, we have assigned location, scale, and orientation to each keypoint:
  - Impose a repeatable local 2D coordinate system
  - Provide invariance to these parameters

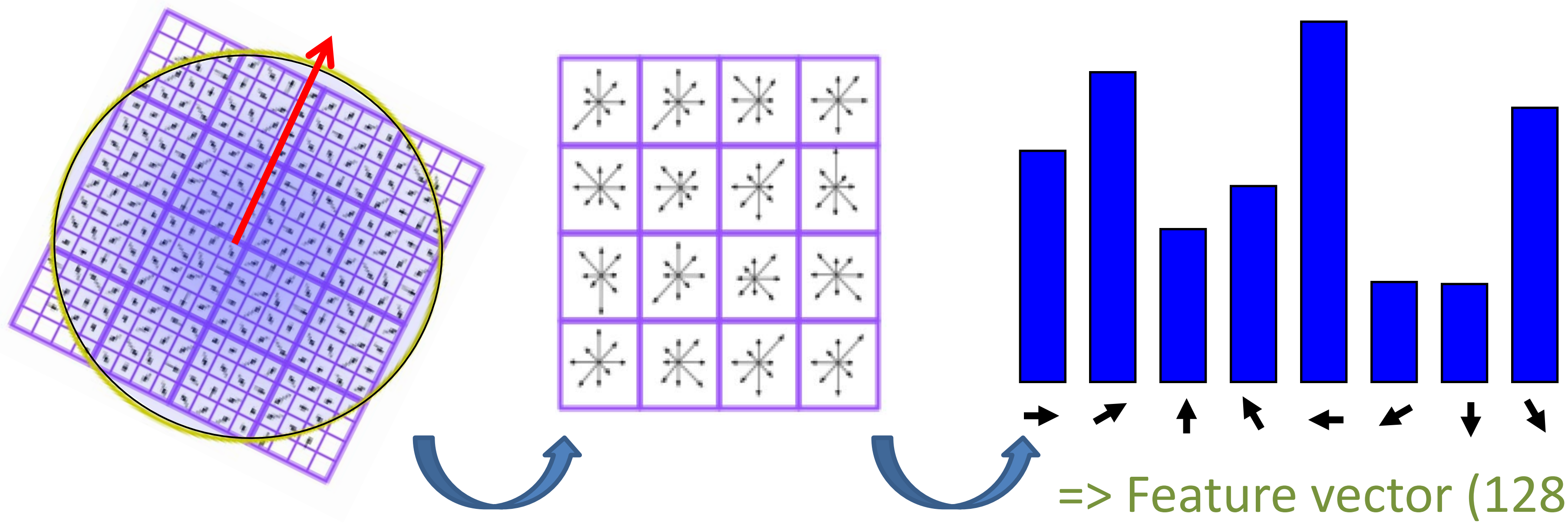


Remaining goal:

- Define local descriptor invariant to remaining variations:
  - Illumination
  - 3D Viewpoint

Step 4: Keypoint descriptor

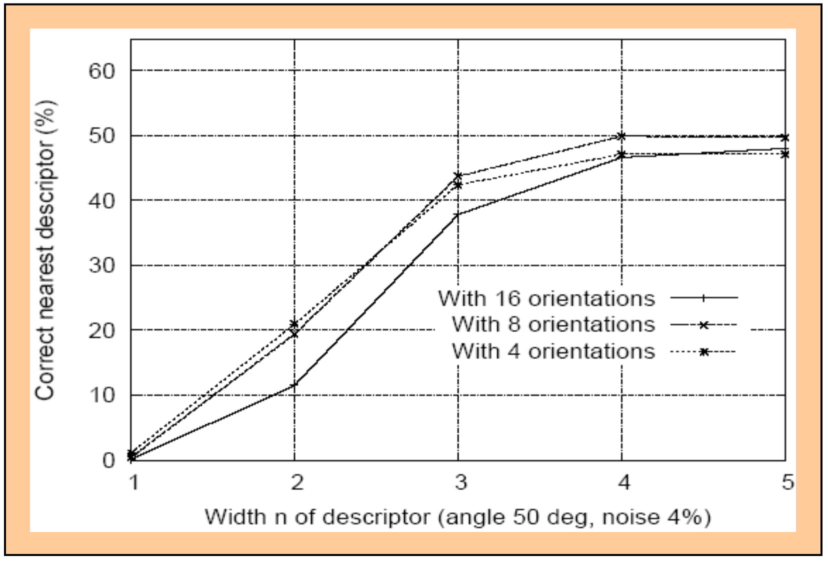
- Create 16 gradient histograms
  - 8 bins each
  - Weighted by magnitude
  - Histogram and gradient values are interpolated and smoothed



=> Feature vector (128)

Step 4: Keypoint descriptor

- Justification:
  - Inspired by the human visual system
  - Parameters r (# of bins) and $n \times n$ (# of histograms) chosen empirically



Matching SIFT Features

- Nearest Neighbor algorithm based on L2 distance
  - Euclidean distance in 128-D

- How to discard bad matches?
  - Threshold on L2 → bad performance
  - Solution: threshold on ratio

$$\frac{\textbf{best match}}{\textbf{second best match}}$$

SIFT for Recognition / Detection

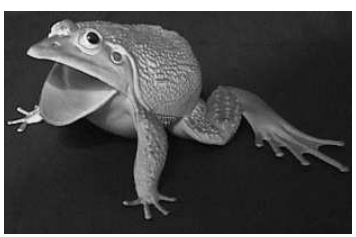
For set of database images:

1. Compute SIFT features
2. Save descriptors to database

For query image:

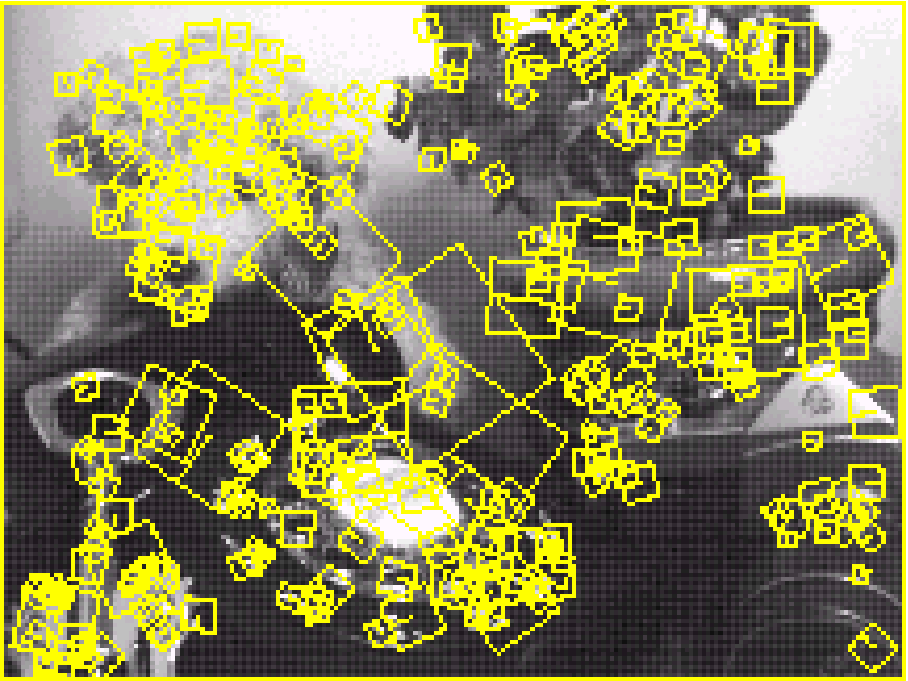
1. Compute SIFT features
2. For each descriptor find its match in the database
3. Check to see if you have "enough" matches at the correct relative scale, orientation, and location
   - *We'll see how to do this next class*
4. Verify (with affine transform)

Recognition under occlusion



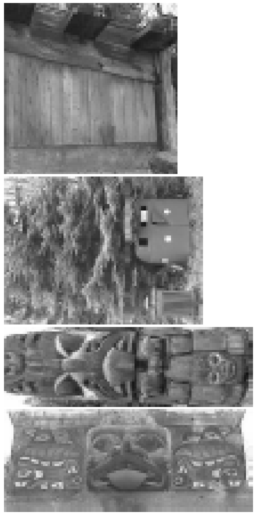
Test of illumination Robustness

- Same **image** under differing illumination



273 keypoints verified in final match

Location recognition



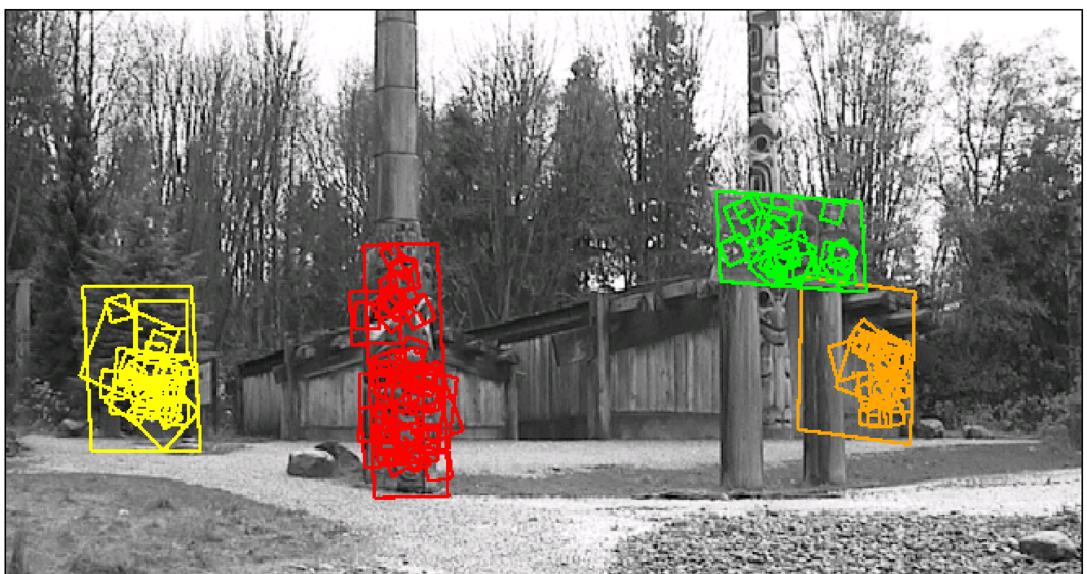
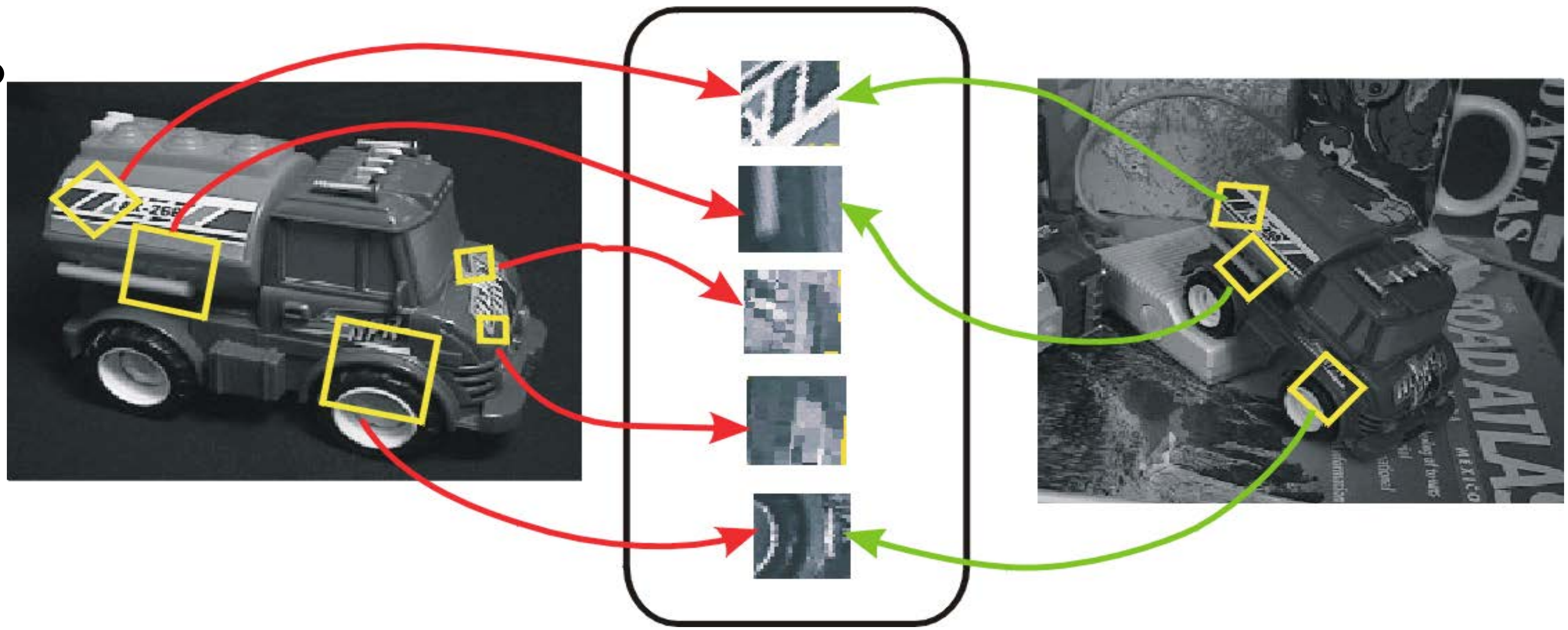


Image Registration Results

[Brown & Lowe 2003]

Invariance vs. Covariance

- **Invariance:**
  – features(transform(image)) = features(image)



Covariant detection => invariant description

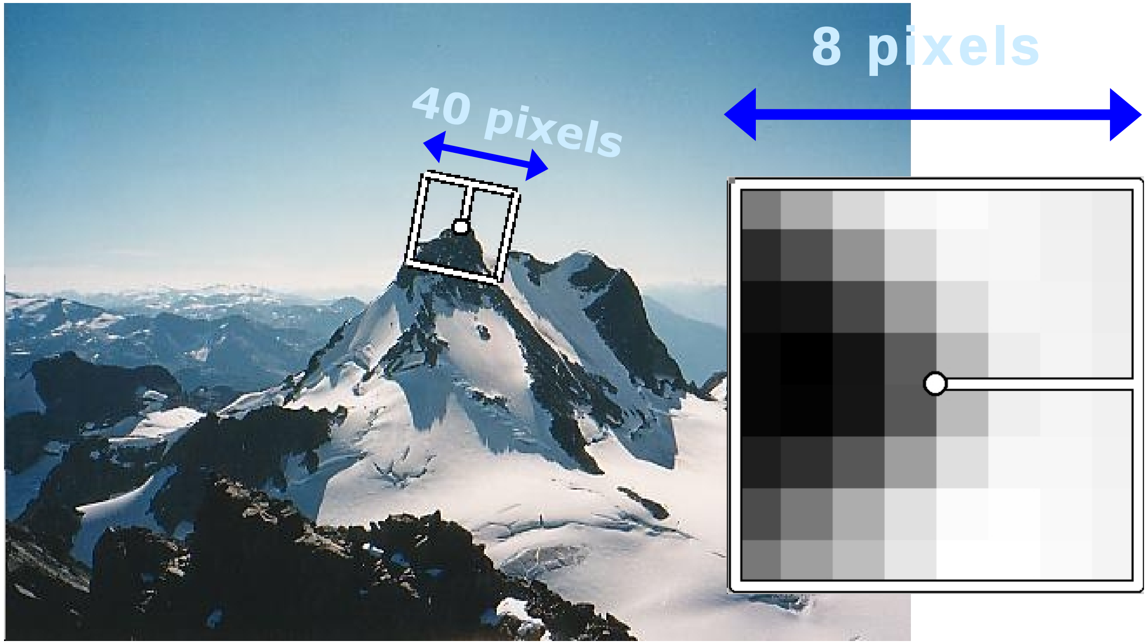
SIFT Summary

- Most widely used image feature in computer vision
  - >36,000 citations (as of 2016)
- Robust to Viewpoint, Lighting, Scale
- Many Extensions
  - PCA-SIFT
  - SURF (More efficient implementation)
- Weak point [Lowe]: key point detection
  - Many papers use a different interest point operator + SIFT feature descriptor

Multiscale Oriented PatcheS Descriptor

Take square window around detected feature

- Rotate to horizontal
- Sample 8x8 square window centered at feature
- Intensity normalize the window by subtracting the mean, dividing by the standard deviation in the window



Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder. CVPR2005. pages 510-517

Detections at multiple scales

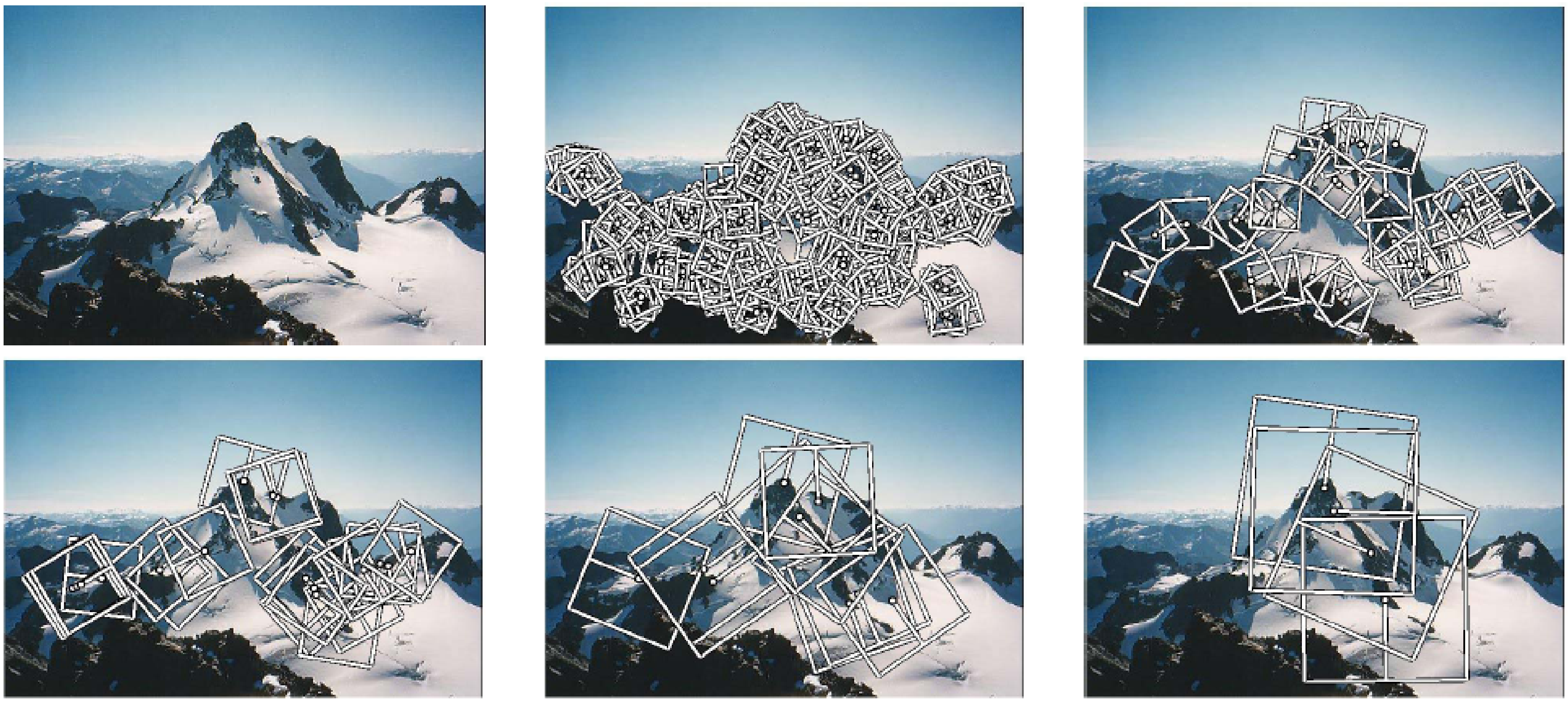


Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matier images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.

MOPS & SIFT Comparison

SIFT

1. **Interest-point detection:** Local max blobs in scale space.
2. **Feature localization:** Get rid of low-contrast, non-corners
3. **Orientation assignment:** Max gradient direction
4. **Feature descriptor:** Histogram of gradient directions

MOPS

1. **Interest-point detection:** multi-scale Harris corners
2. **Feature localization:** Adaptive thresholding
3. **Orientation assignment:** Pixel gradient direction
4. **Feature descriptor:** Normalized image patch

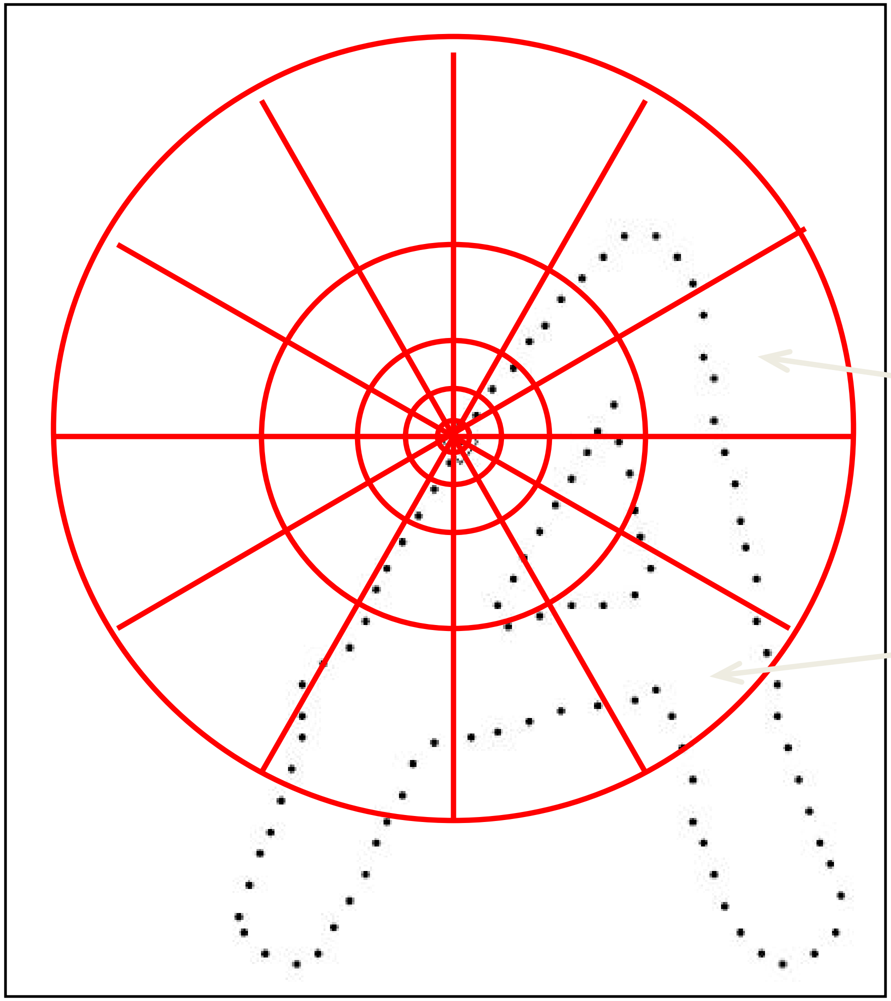
Shape Contexts

- Local image feature
  - Different Approach than SIFT / MOPS

- Interest Point: Edge points
- Feature Descriptor: Spatial histogram of neighboring edge points

"Shape Matching and Object Recognition Using Shape Contexts", Belongie et al. PAMI April 2002

Shape Context

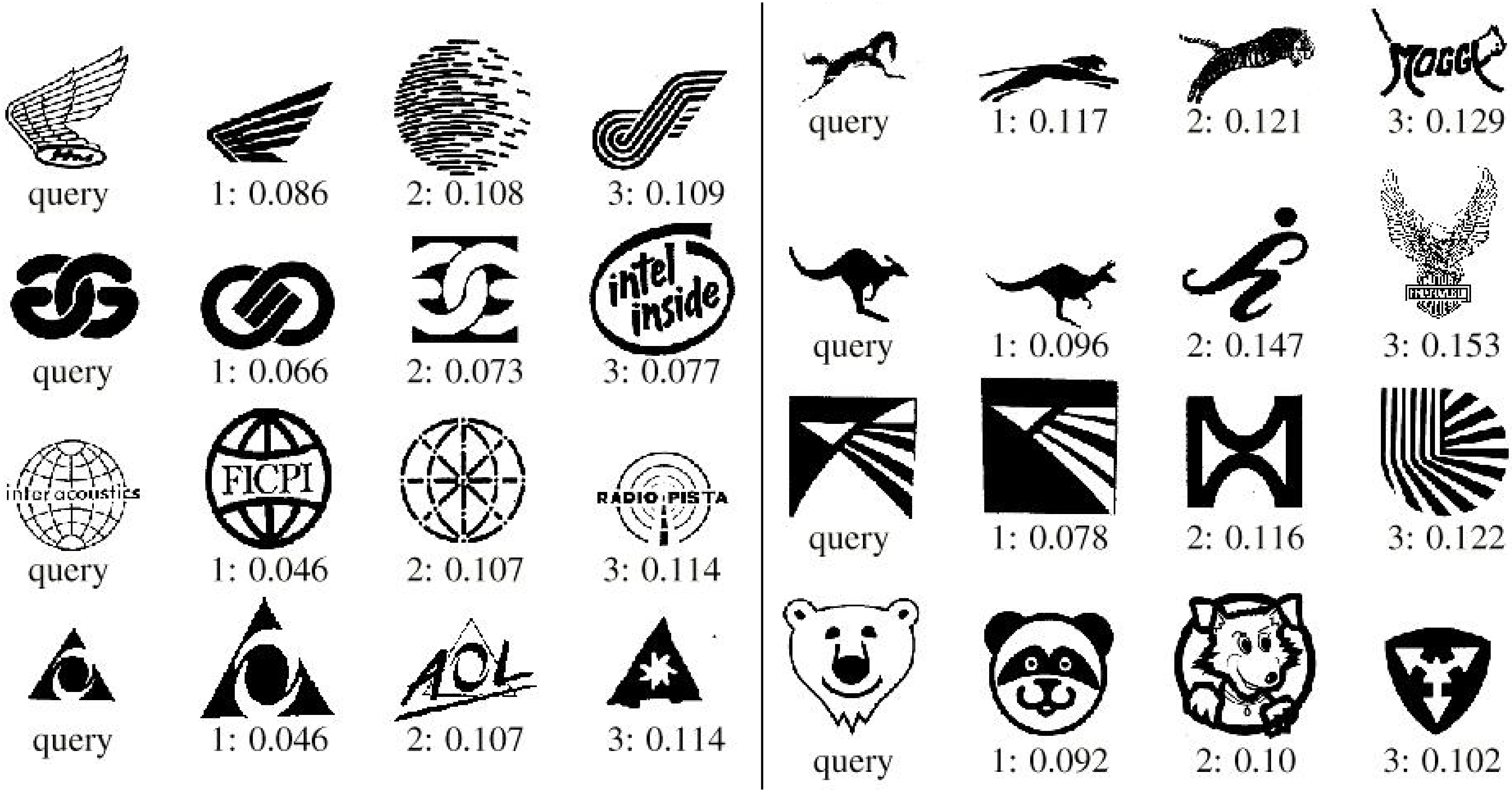


Count = 4

Count = 10

Compact representation of distribution of points relative to each point

Trademark Similarity



Shape Contexts

- Translation-Invariant: **Yes**
- Scale-Invariant: **Yes** (if radial distances are normalized)
- Rotation-Invariant: **Yes** (if angles are measured relative to common direction, e.g., tangent)

- *Robust to small deformations, noise, outliers*

- Applications:
  - Digit Recognition (.63% error on 20,000 example MNIST data)
    - Note in this case, rotation invariance is NOT desirable ‘6’ vs. ‘9’
  - Trademark Retrieval

Next time: Fitting Models to Sets of Features

- Local features describe the region around 1 pixel
- Objects take up >1 pixel (and therefore multiple local features)

- How do we match sets of features to a given model?