

part 2: feature & object descriptors

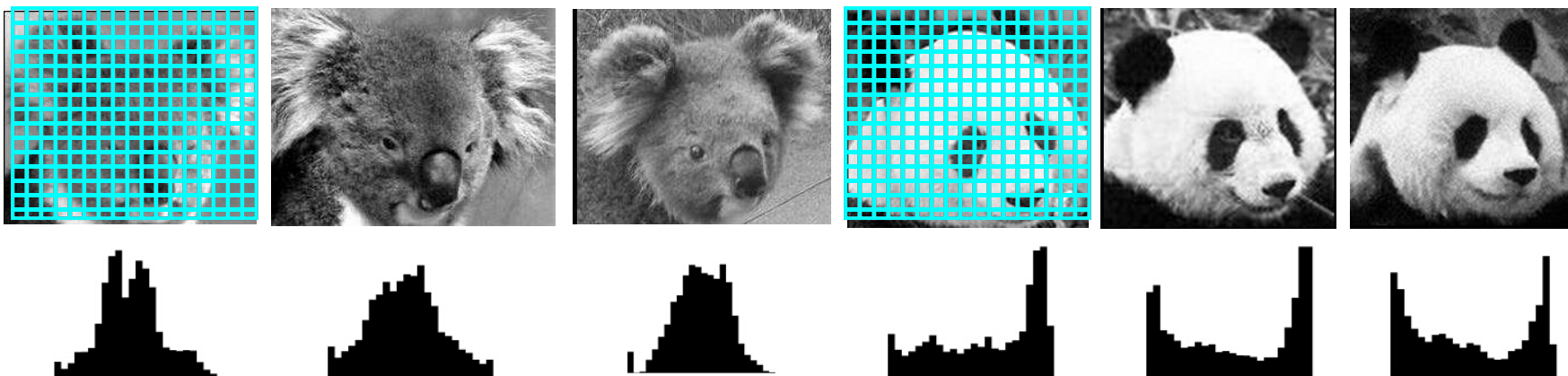
1. Recognition with global representation

- sliding window



Car/non-car
Classifier

Descriptors suitable for global appearance



Simple holistic descriptions of image content

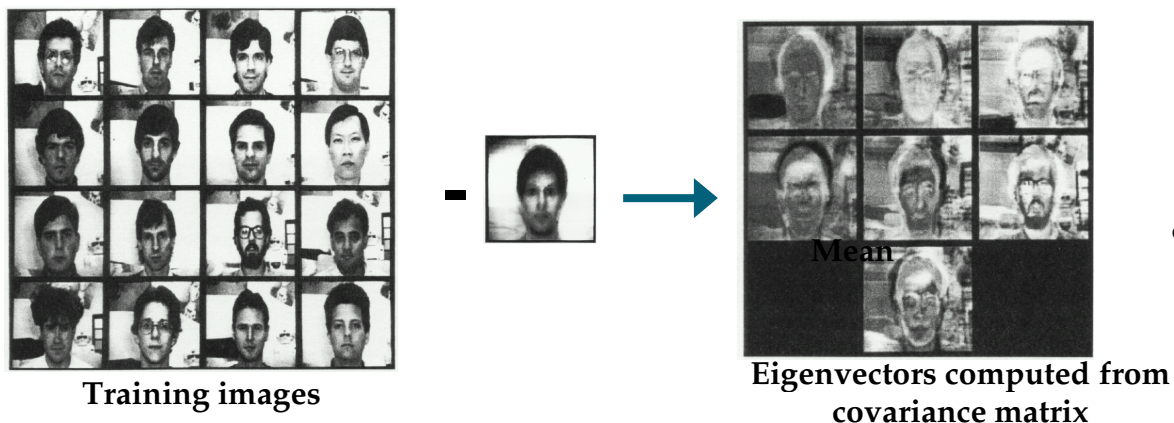
grayscale / color histogram

or

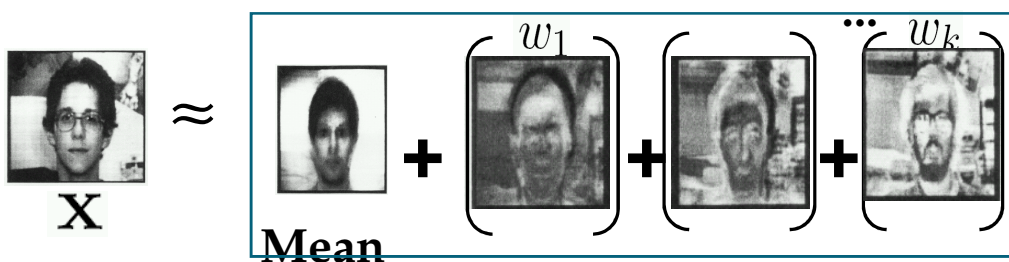
vector of pixel intensities

Eigenfaces: global appearance description

An early appearance-based approach to face recognition



Generate low-dimensional representation of appearance with a linear subspace.



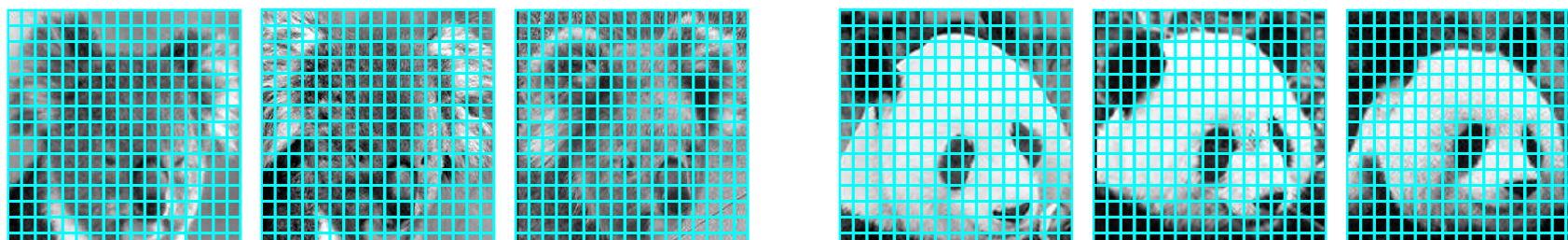
Project new images to "face space".

Recognition via nearest neighbors in face space

Turk & Pentland, 1991

Feature extraction: global appearance

- Pixel-based representations sensitive to small shifts

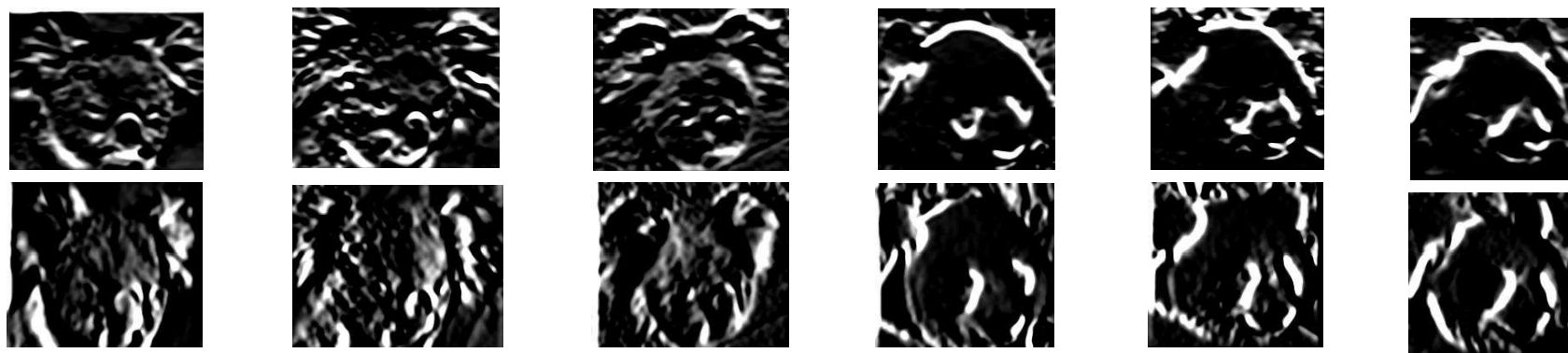


Cartoon example:
an albino koala

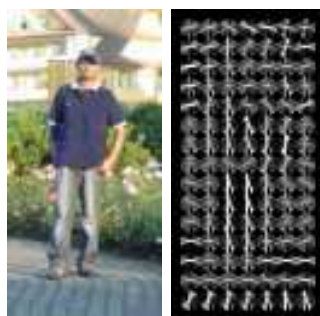
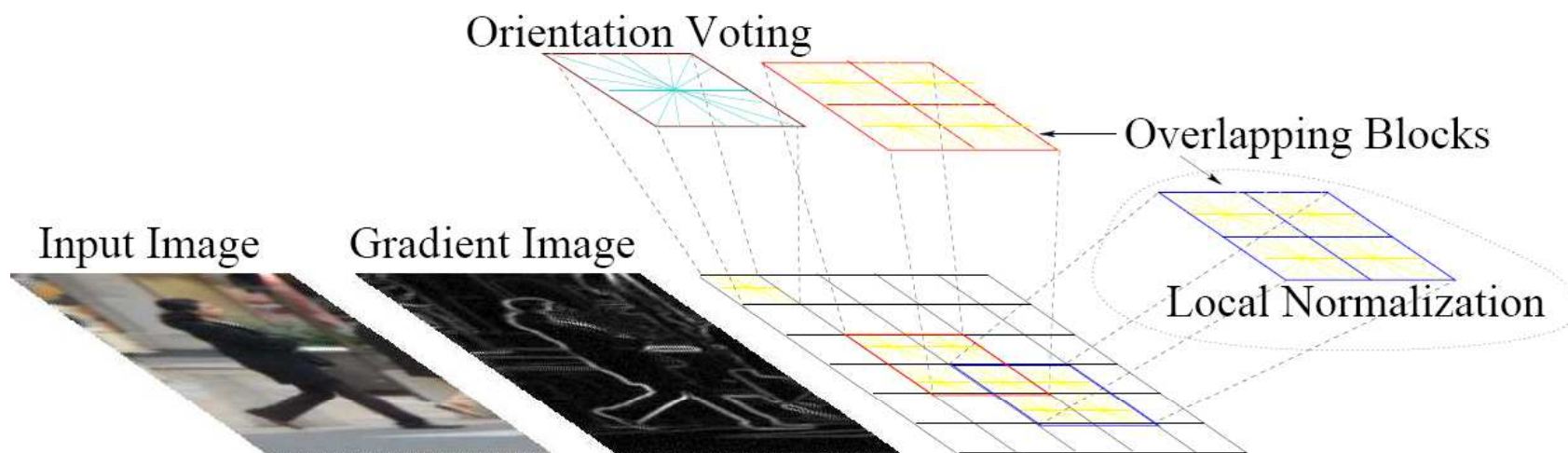
- Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation

Gradient-based representations

- Consider edges, contours, and (oriented) intensity gradients



Gradient-based representations: Histograms of oriented gradients (HoG)



Map each grid cell in the input window to a histogram counting the gradients per orientation.

Code available:

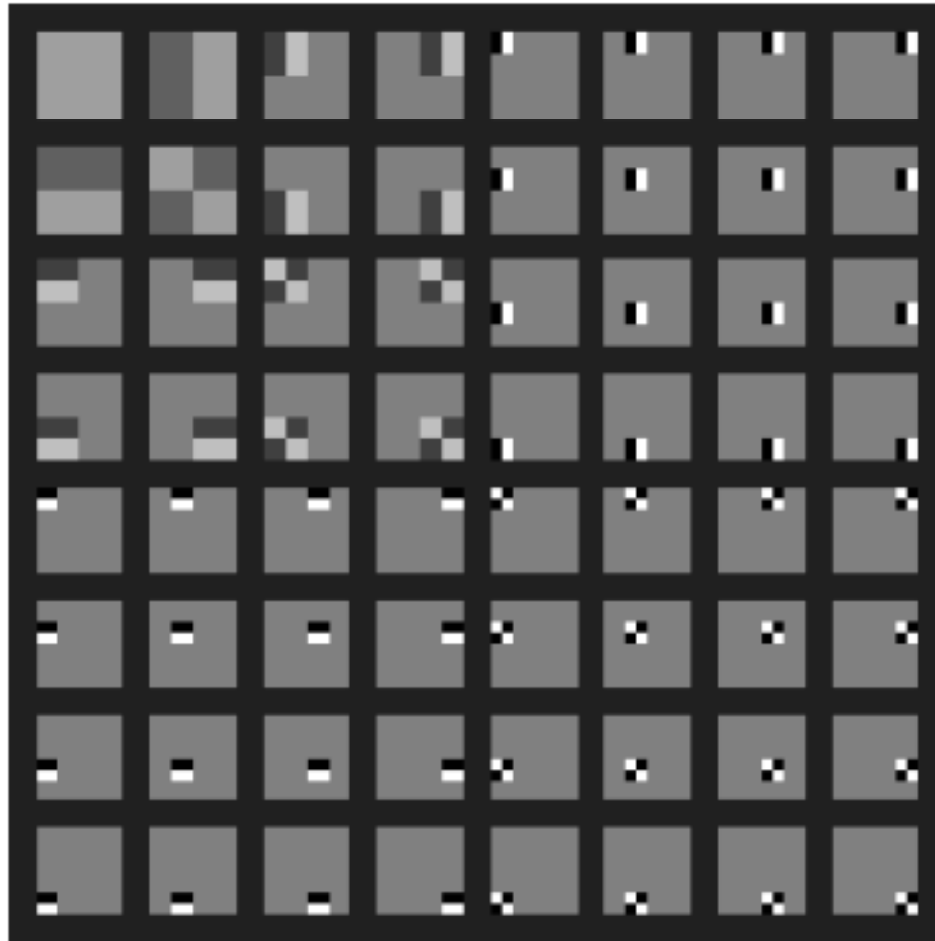
<http://pascal.inrialpes.fr/soft/olt/>

Dalal & Triggs, CVPR 2005

Representation based on convolution with some orthonormal basis

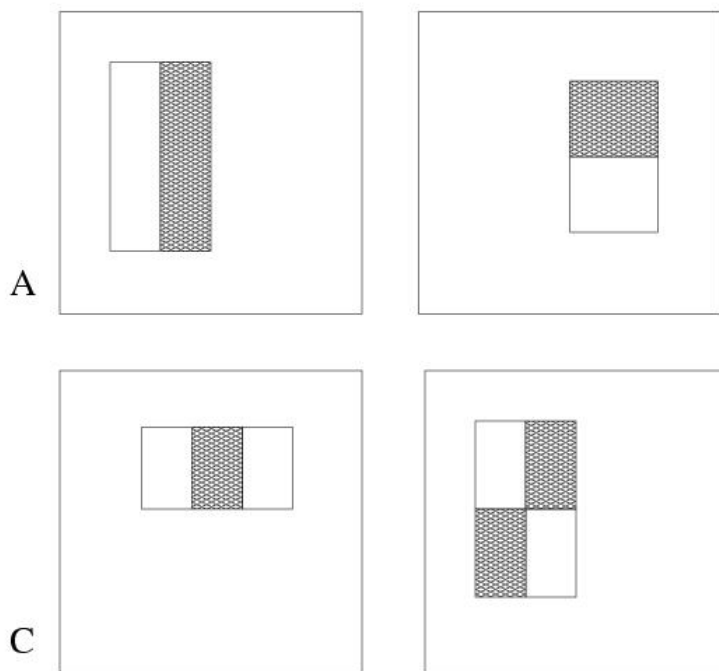
2D orthonormal Haar basis:

8x8 image



Object Detection with Many Simple Features

- Viola Jones, face detection

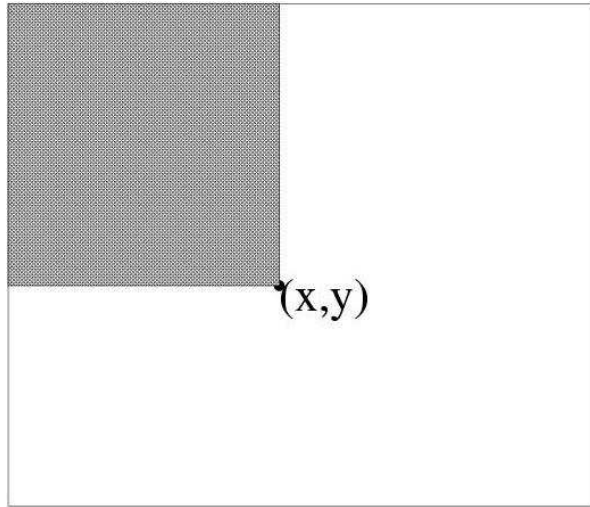


(Generalized) Haar Features:

- rectangular blocks, white or black
- 3 types of features:
 - two rectangles: horizontal/vertical
 - three rectangles
 - four rectangles
- in 24x24 window: 180,000 possible features

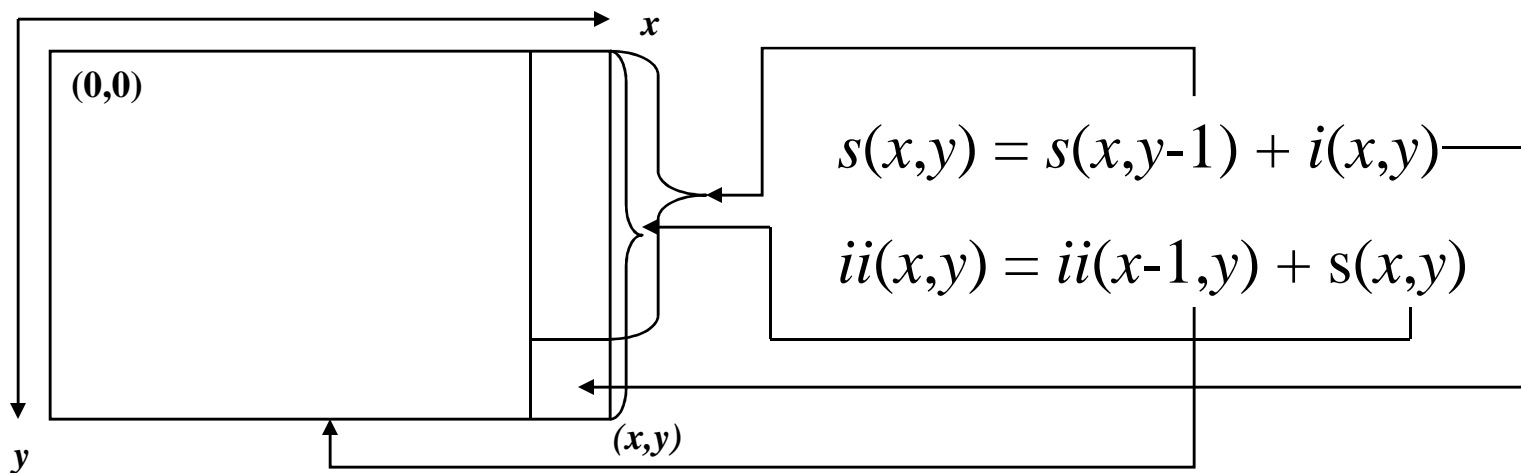
- Basic idea: learn those features which are good for the task; specifically, face detection

Integral Image

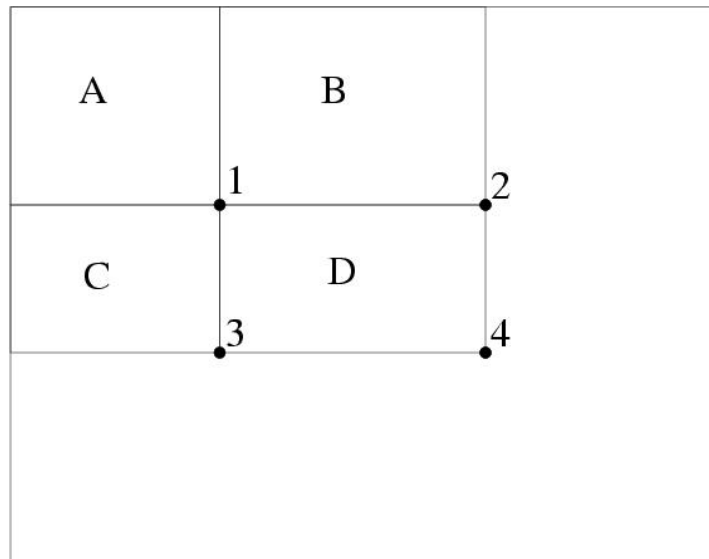


Definition: The *integral image* at location (x,y) , is the sum of the pixel values above and to the left of (x,y) , inclusive.

We can calculate the integral image representation of the image in a single pass.



Efficient Computation of Rectangle Value



Using the integral image representation one can compute the value of any rectangular sum in constant time.

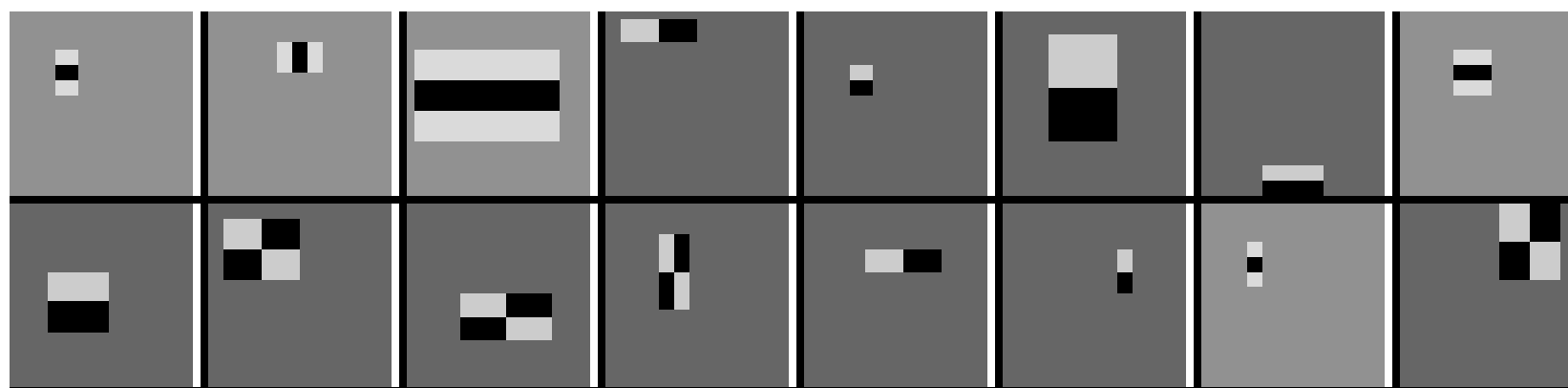
Example: Rectangle D

$$ii(4) + ii(1) - ii(2) - ii(3)$$

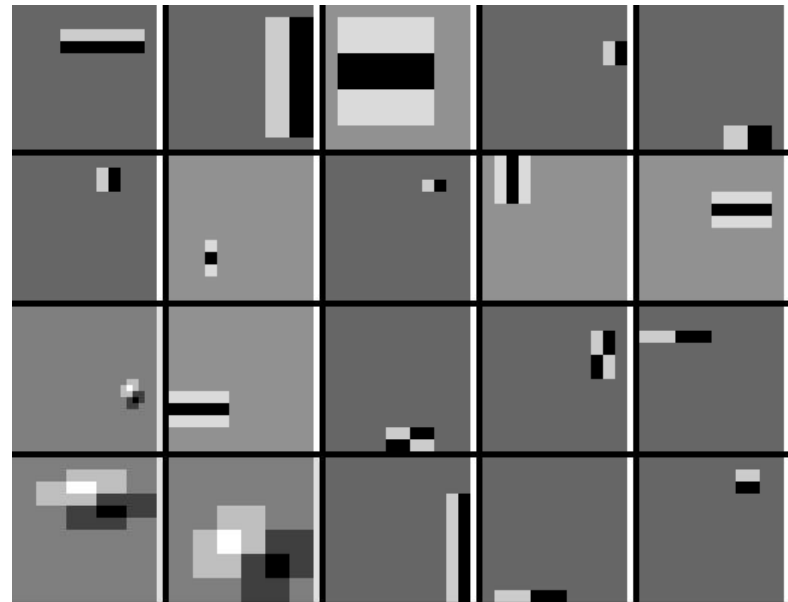
As a result two-, three-, and four-rectangular features can be computed with 6, 8 and 9 array references respectively.

Face Localization Features

- Learned features reflect the task

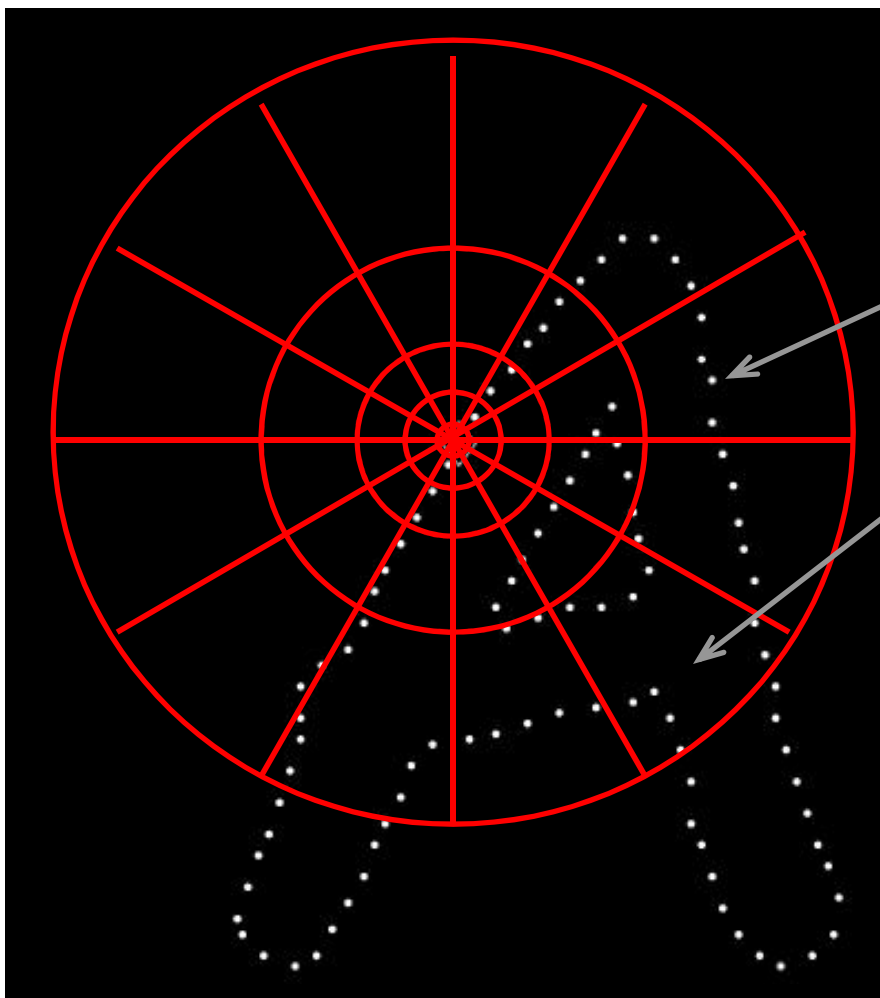


Face Profile Features



Shape Context

Given a set of points, captures the relative distribution of points in the plane relative to each keypoint on the shape



Count the number of points inside each bin, e.g.:

Count = 4

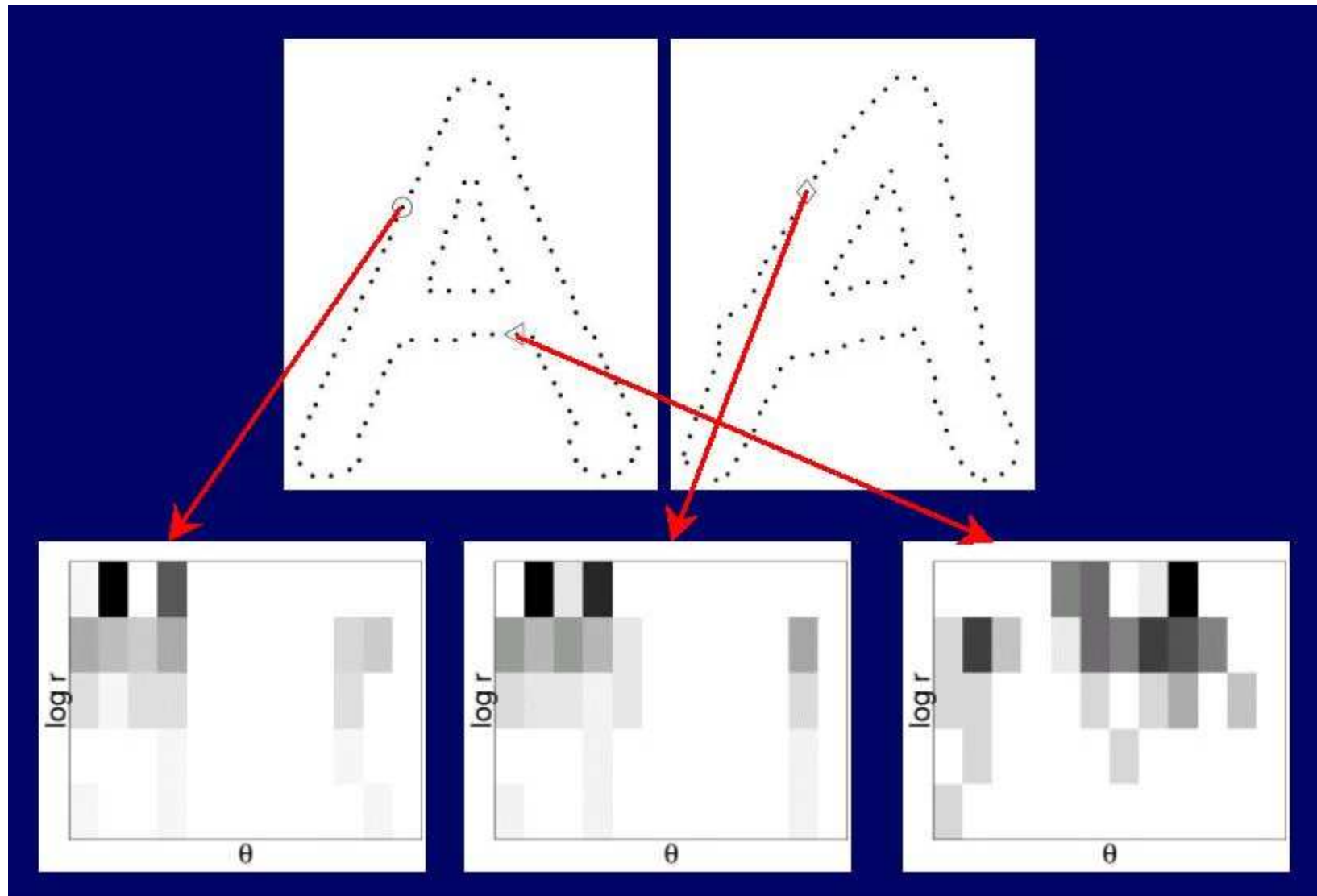
⋮

Count = 10

Log-polar binning: more precision for nearby points, more flexibility for farther points.

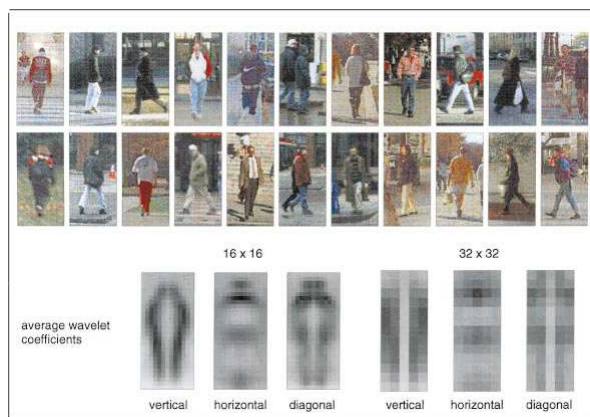
Shape Context

we get descriptors that are similar for homologous (corresponding) points and dissimilar for non-homologous points



Examples: Pedestrian detection

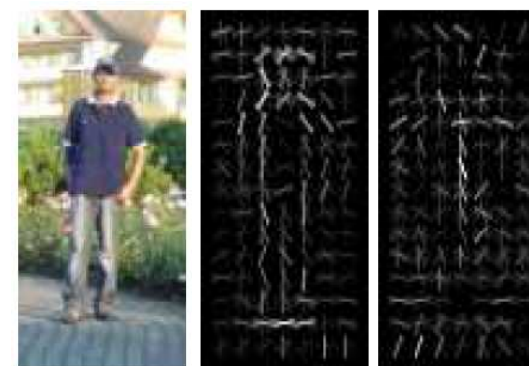
- Detecting upright, walking humans also possible using sliding window's appearance/texture; e.g.,



SVM with Haar wavelets
[Papageorgiou & Poggio, IJCV 2000]



Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]



SVM with HoGs [Dalal & Triggs, CVPR 2005]

Highlights

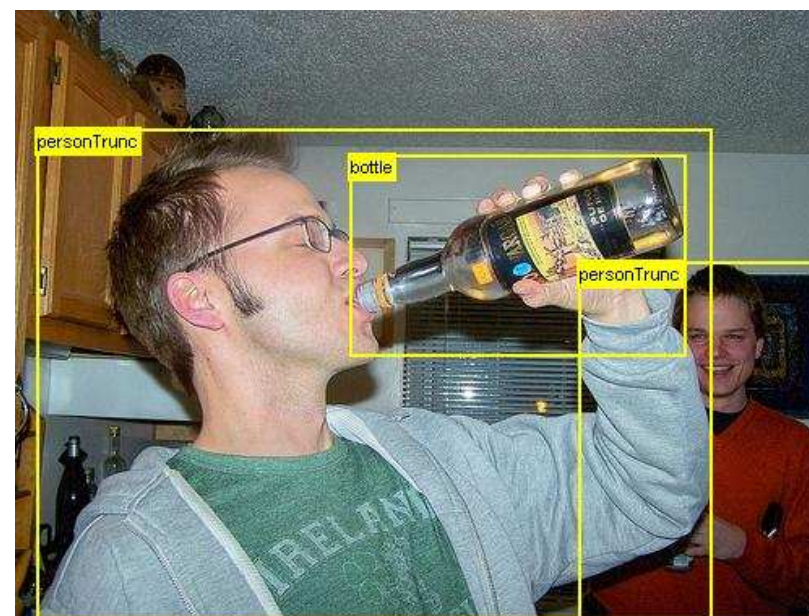
- Sliding window detection and global appearance descriptors:
 - Simple detection protocol to implement
 - Good feature choices critical
 - Past success for certain classes

Limitations

- High computational complexity
 - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
 - If training binary detectors independently, means cost increases linearly with number of classes
- With so many windows, false positive rate better be low

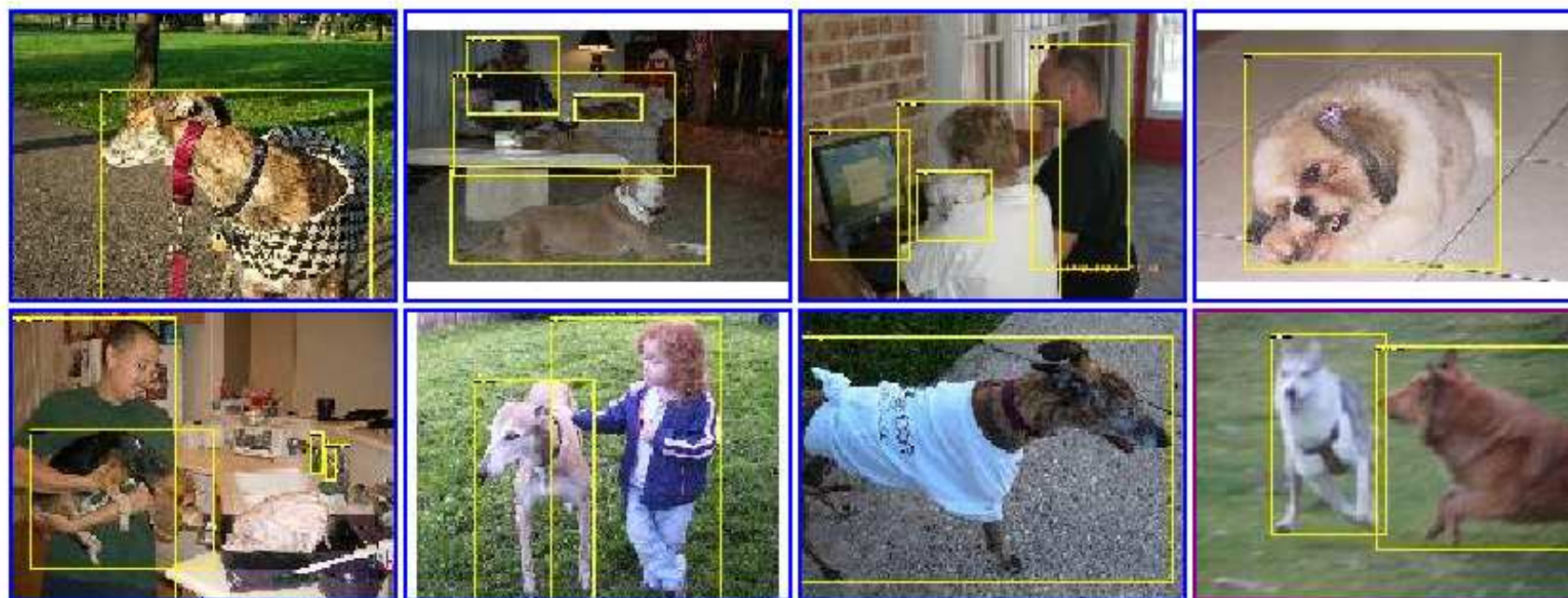
Limitations (continued)

- Not all objects are “box” shaped



Limitations (continued)

- Non-rigid, deformable objects not captured well with representations assuming a fixed 2D structure; or must assume fixed viewpoint



Limitations (continued)

- If considering windows in isolation, context is lost



Sliding window



Detector's view

Limitations (continued)

- In practice, often entails large, cropped training set (expensive)
- Requiring good match to a global appearance description can lead to sensitivity to partial occlusions



2. Descriptors suitable for local appearance

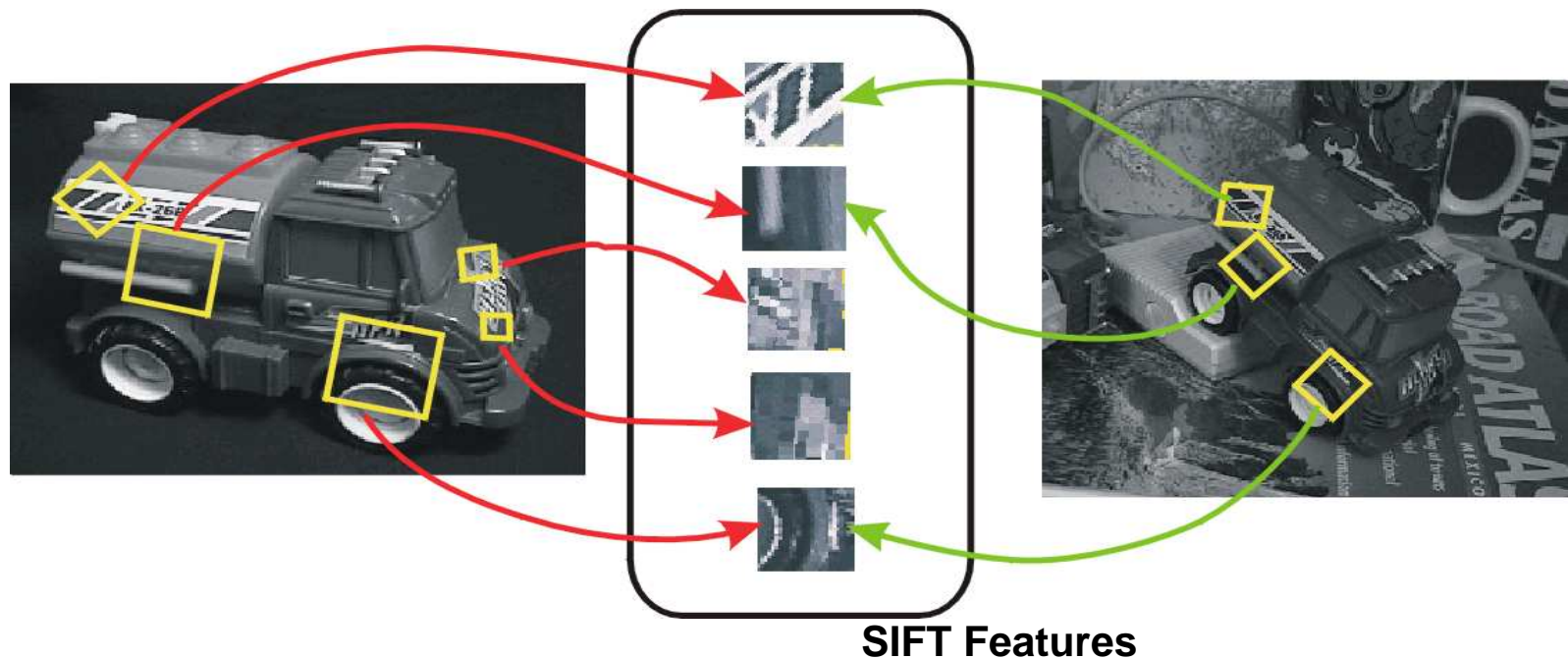
note: any global descriptor can be used to characterize the region around local feature

Advantages of invariant local features

- **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness:** individual features can be matched to a large database of objects
- **Quantity:** many features can be generated for even small objects
- **Efficiency:** close to real-time performance

SIFT: Scale Invariant Feature Transform

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Distinctive image features from scale-invariant keypoints. David G. Lowe, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

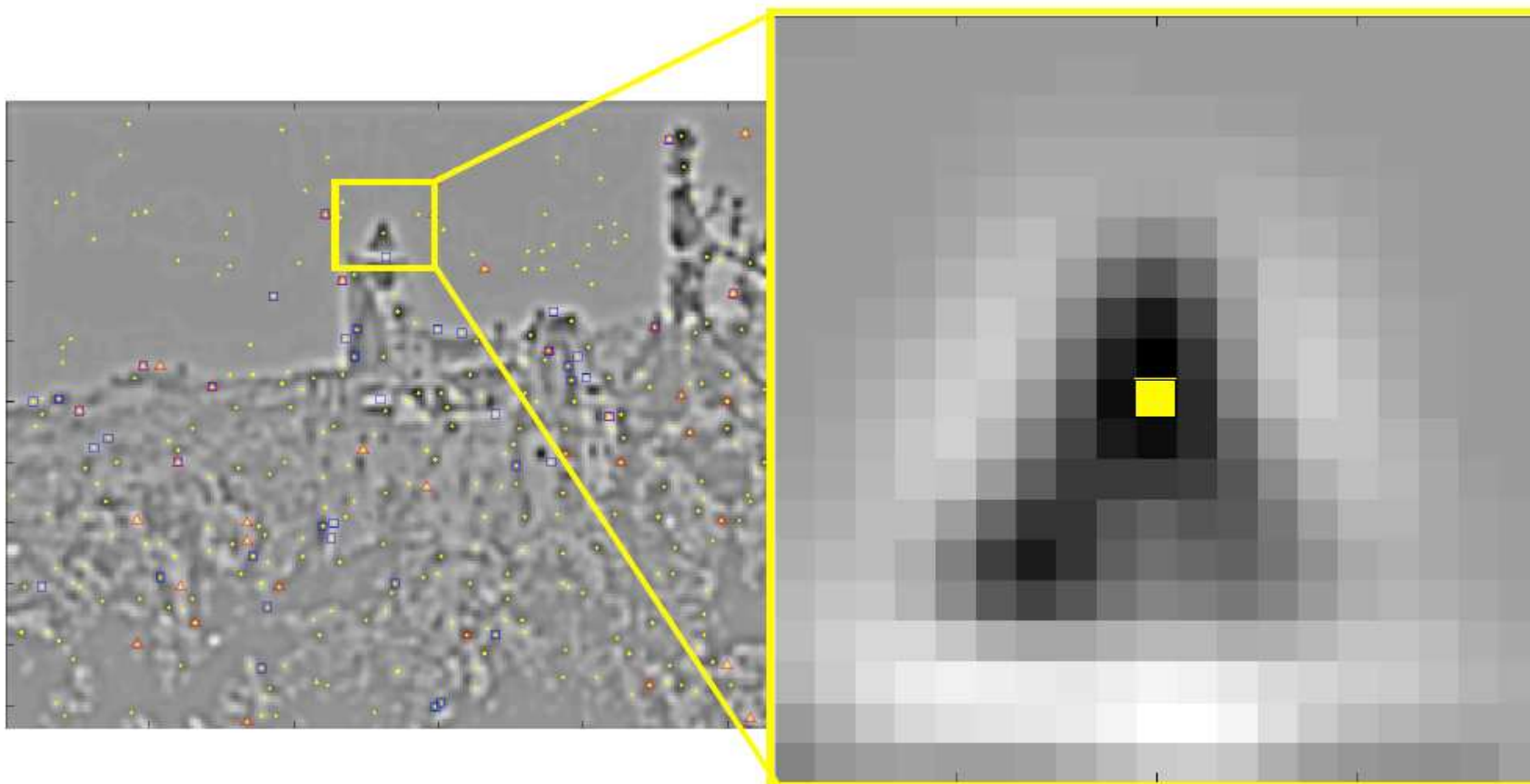
SIFT - algorithm

1. Keypoint selection, Enforce invariance to scale

2. Point description

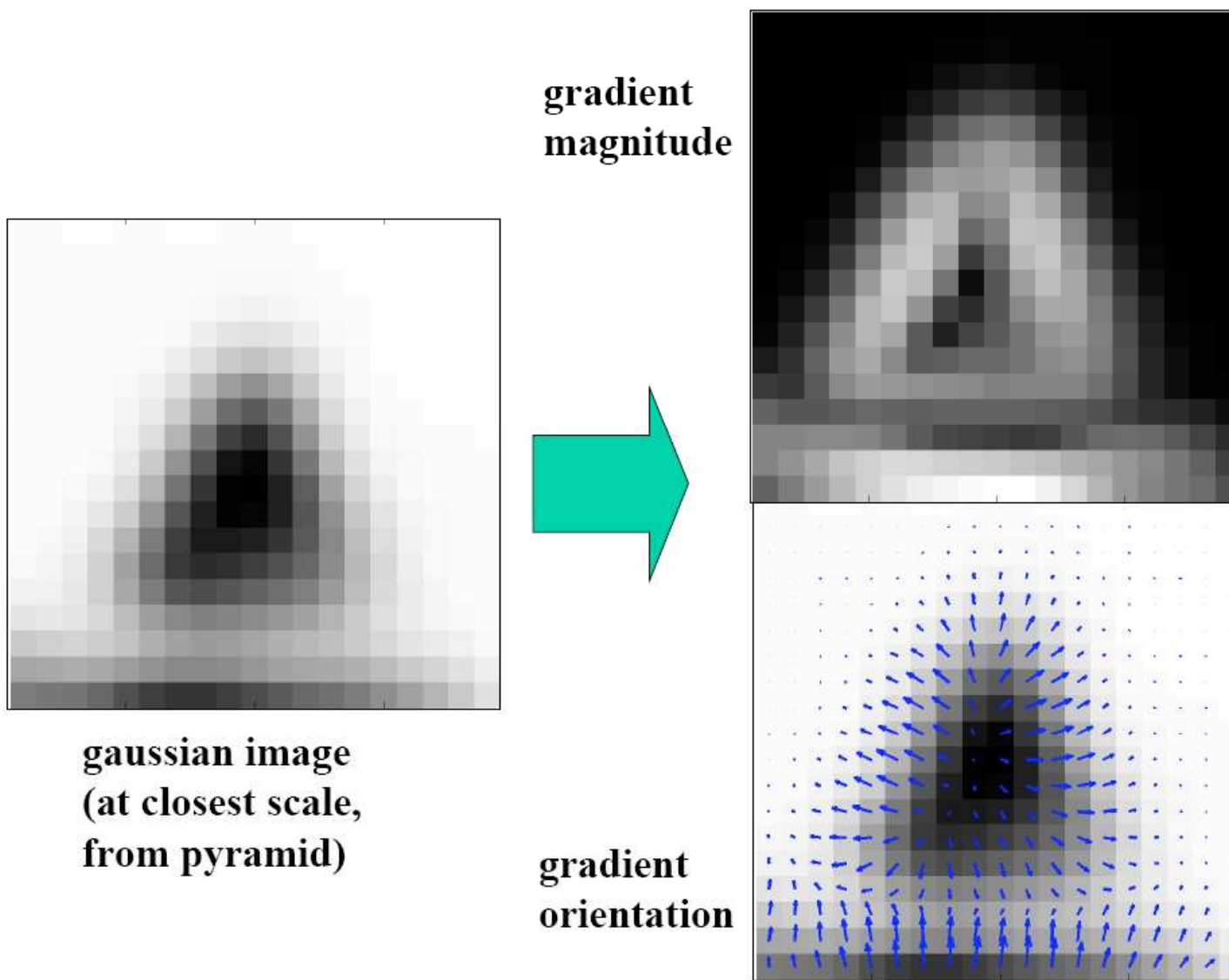
- o **Enforce invariance to orientation:** Compute orientation, to achieve rotation invariance, by finding the strongest gradient direction in the smoothed image; rotate patch so that orientation points up.
- o **Compute feature signature:** Compute a "gradient histogram" of the local image region in a 4x4 pixel region; do this for 4x4 regions of that size; orient so that largest gradient points up. Result: feature vector with 128 values (16 fields, 8 gradients).
- o **Enforce invariance to illumination change and camera saturation:** Normalize to unit length to increase invariance to illumination; threshold all gradients, to become invariant to camera saturation.

Select canonical orientation



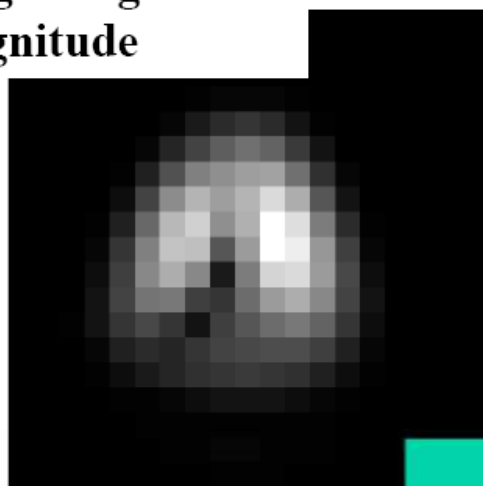
- **Keypoint location = extrema location**
- **Keypoint scale is scale of the DOG image**

Orientation assignment

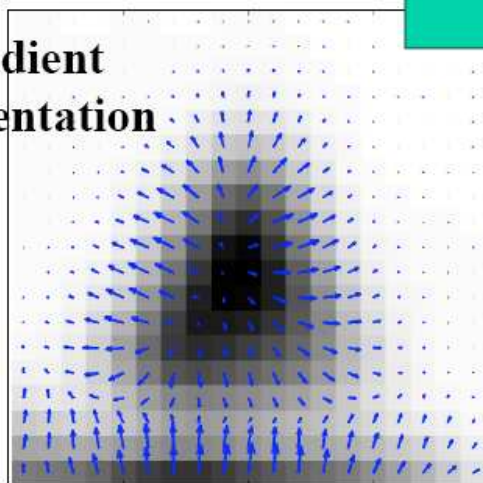


Orientation assignment

weighted gradient
magnitude

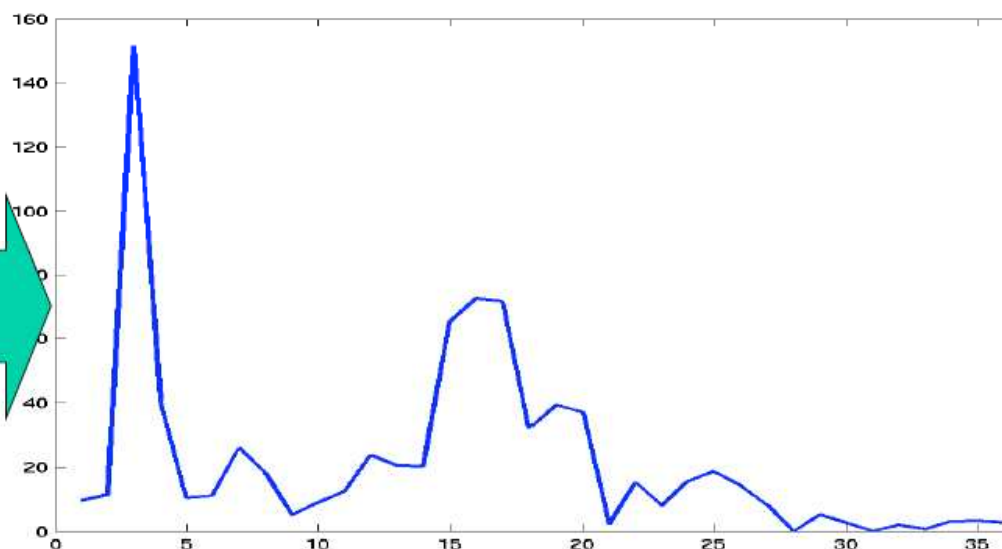


gradient
orientation



weighted orientation histogram.

Each bucket contains sum of weighted gradient magnitudes corresponding to angles that fall within that bucket.



36 buckets

10 degree range of angles in each bucket, i.e.

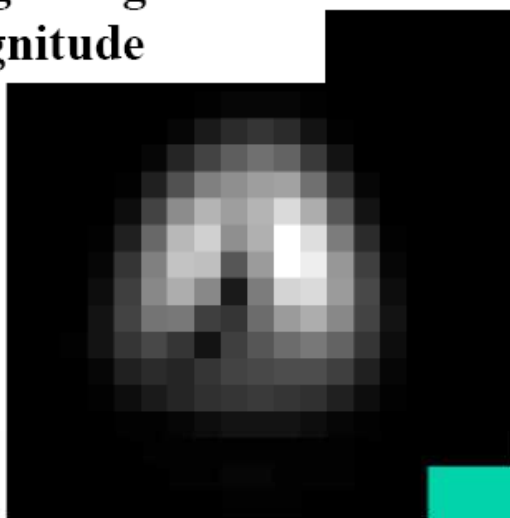
$0 \leq \text{ang} < 10$: bucket 1

$10 \leq \text{ang} < 20$: bucket 2

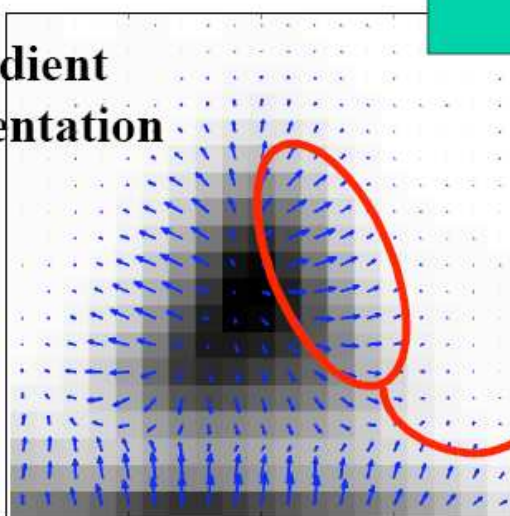
$20 \leq \text{ang} < 30$: bucket 3 ...

Orientation assignment

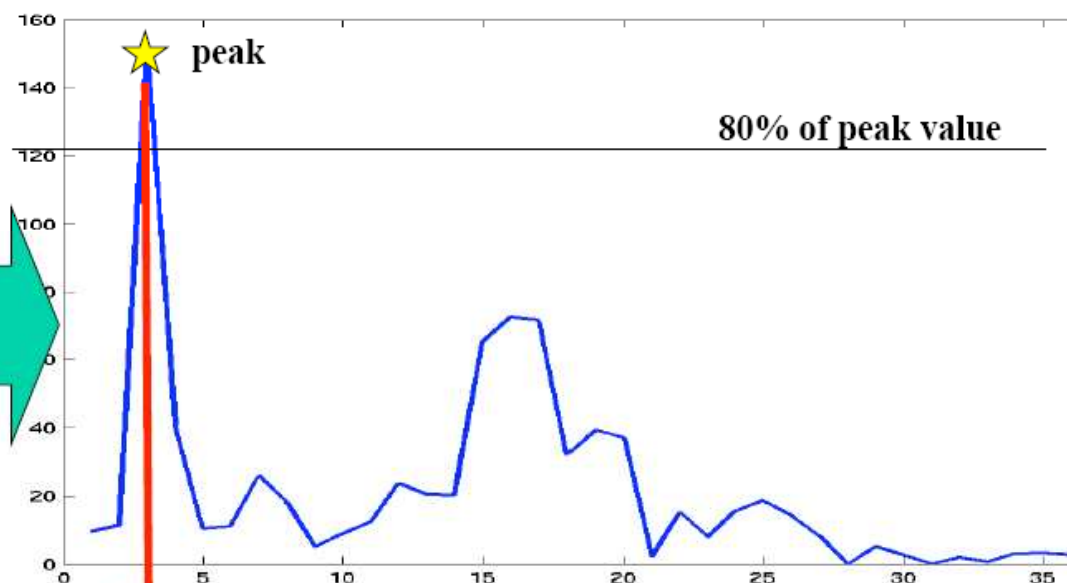
weighted gradient
magnitude



gradient
orientation



weighted orientation histogram.

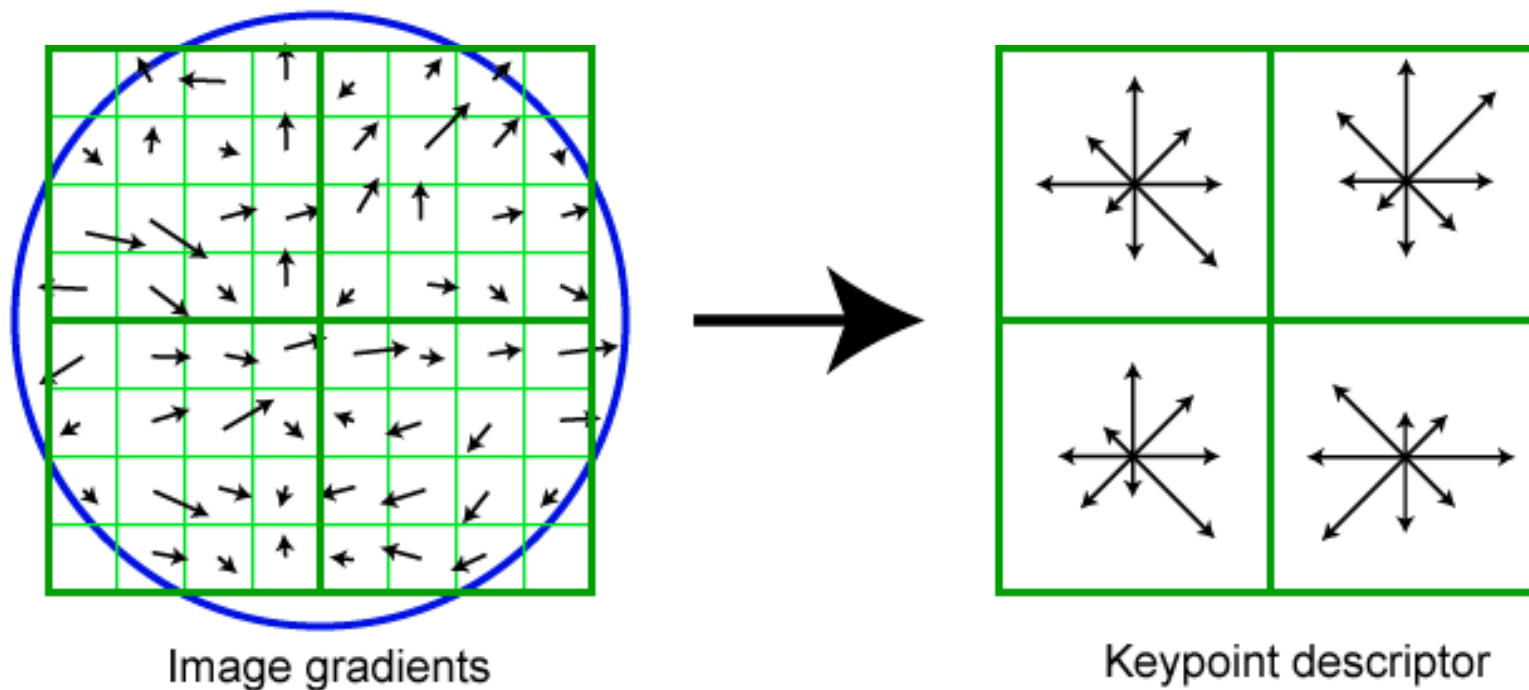


20-30 degrees

**Orientation of keypoint
is approximately 25 degrees**

SIFT vector formation

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of weighted orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions



Summary

- Goal: achieve a concise representation that is useful (and efficient) for image matching
- How to compute image features
 - Interest point detectors - try to be stable across a variety of local image changes
 - Feature descriptors – try to be descriptive and maintain meaningful information about the image patch