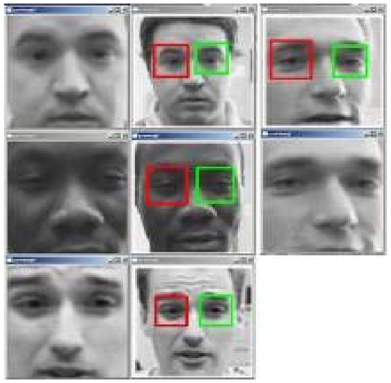
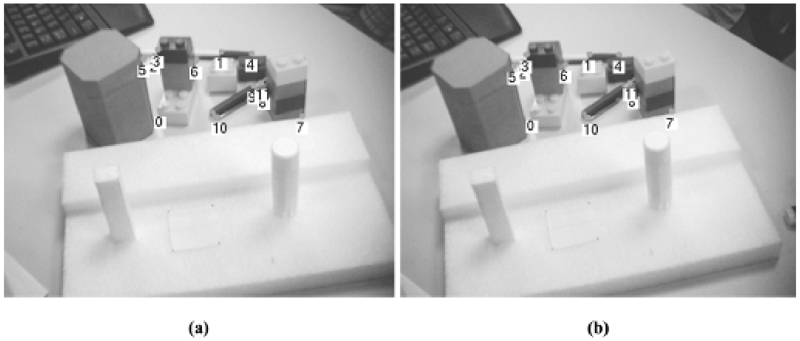
Scale Invariant Feature Transform (SIFT)

Why do we care about matching features?

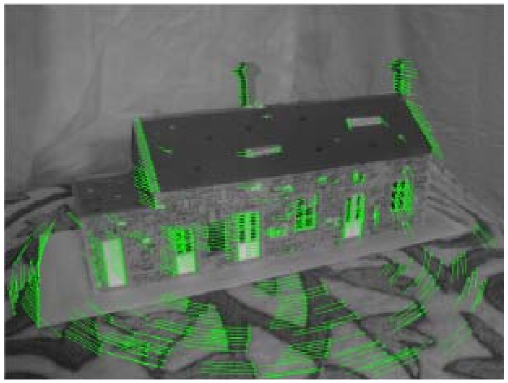
- Camera calibration
- Stereo
- Tracking/SFM
- Image moiaicing
- Object/activity Recognition
- …





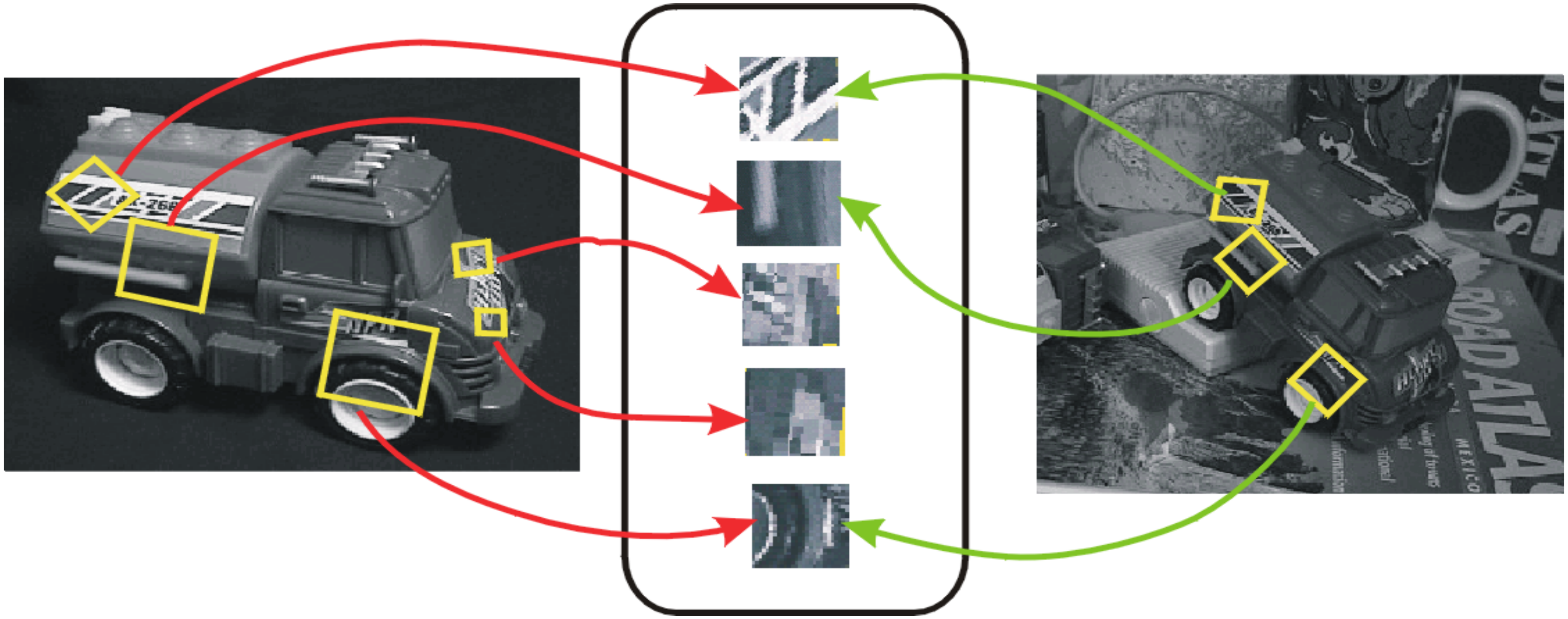
(a)

(b)

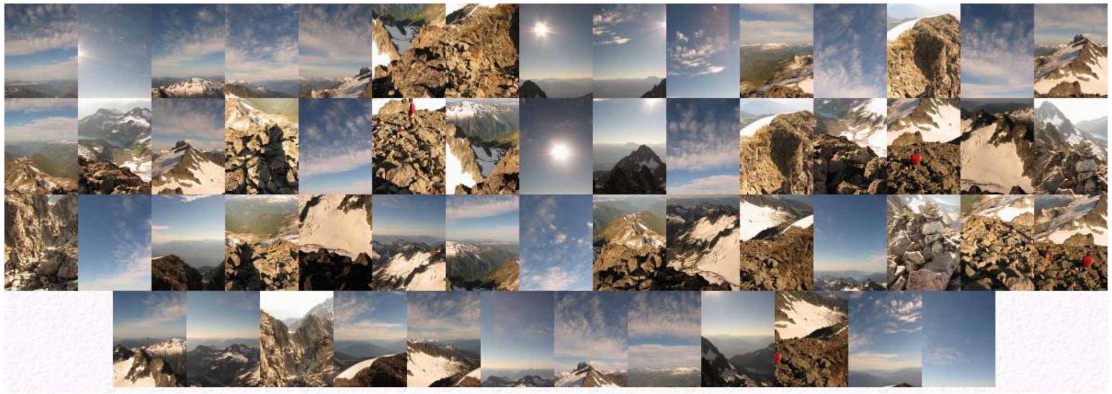


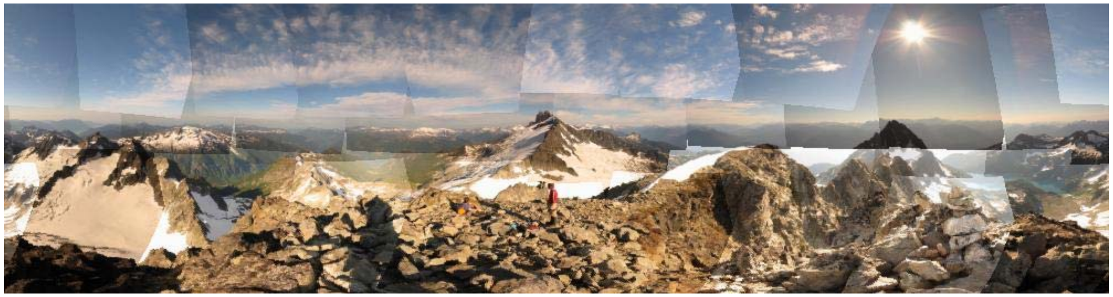
Objection representation and recognition

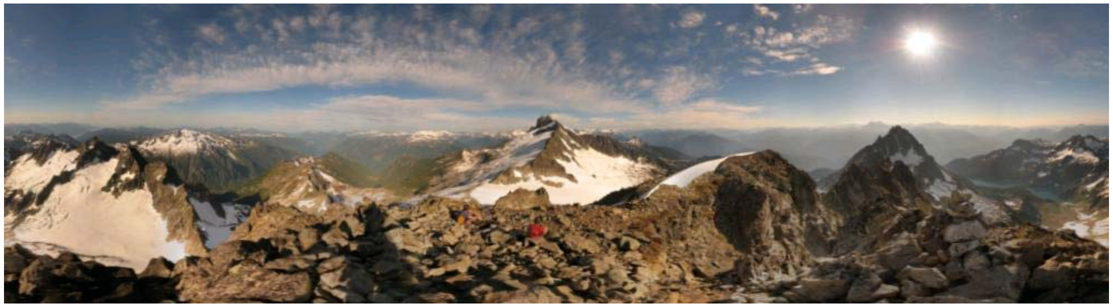
- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



- Automatic Mosaicing
- http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html





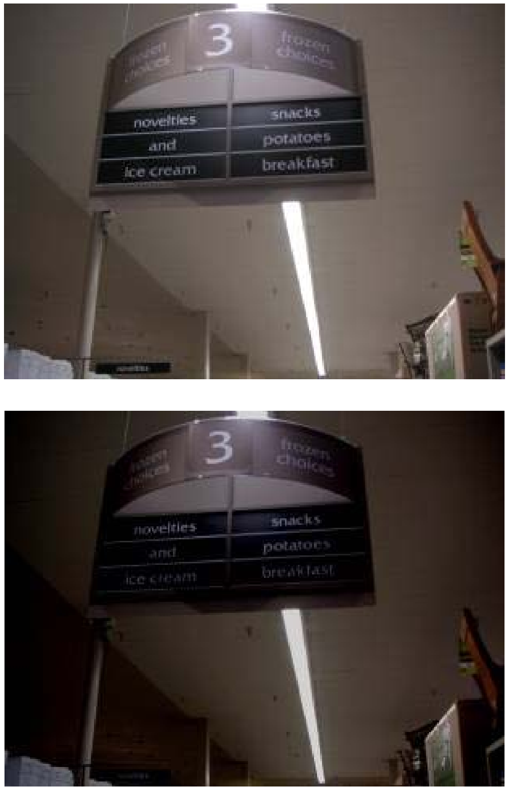


We want invariance!!!

- To illumination
- To scale
- To rotation
- To affine
- To perspective projection

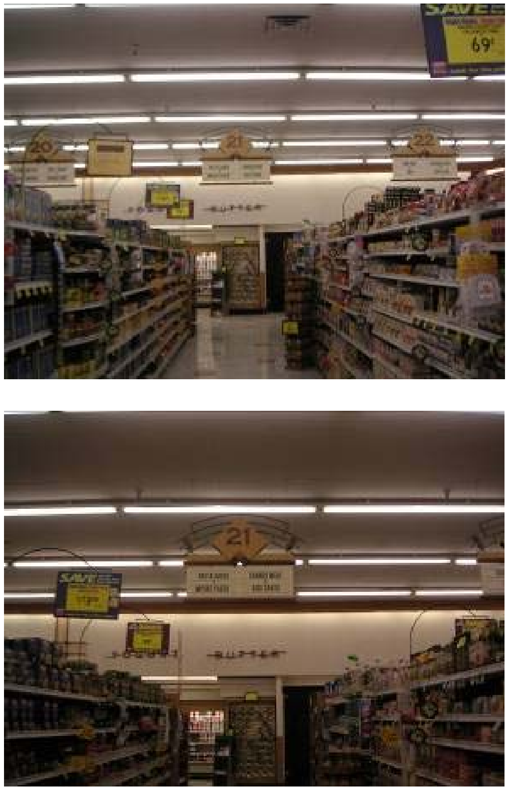
Types of invariance

- Illumination



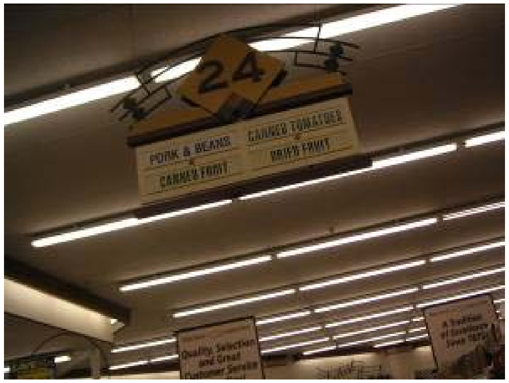
Types of invariance

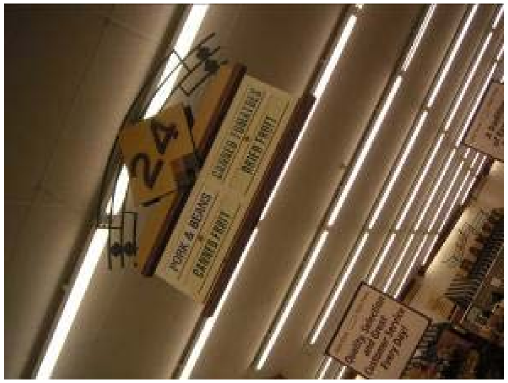
- Illumination
- Scale



Types of invariance

- Illumination
- Scale
- Rotation



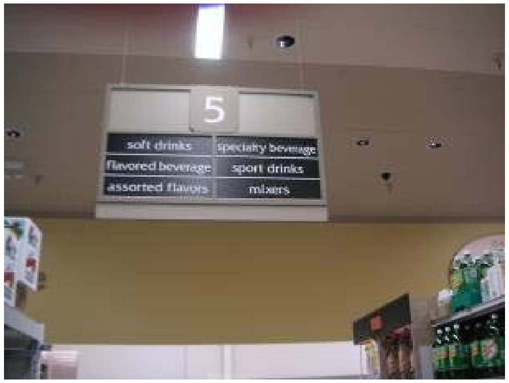


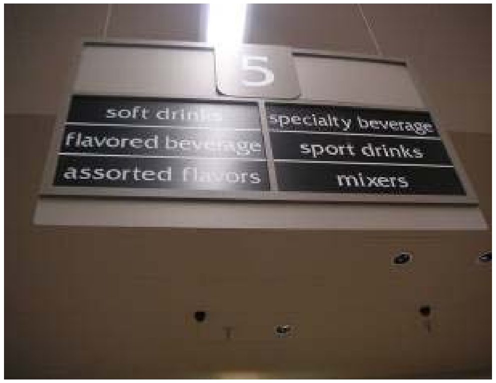
Types of invariance

- Illumination
- Scale
- Rotation
- Affine (view point change)

Types of invariance

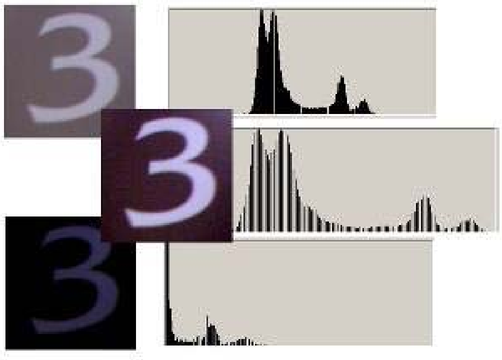
- Illumination
- Scale
- Rotation
- Affine
- Full Perspective





How to achieve illumination invariance

- The easy way (normalized)
- Difference based metrics (random tree, Haar, and sift, gradient)



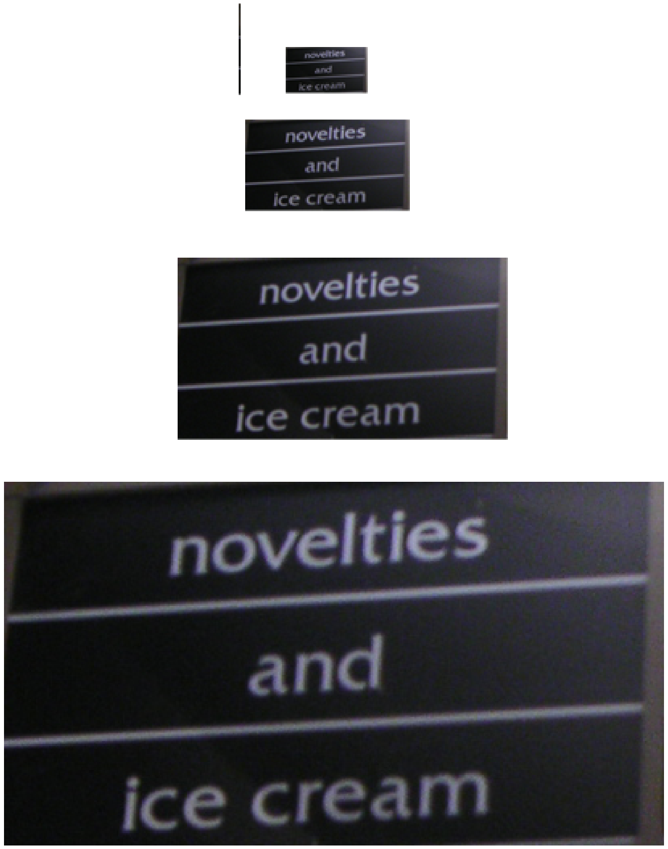
How to achieve scale invariance

- Pyramids

- Scale Space (DOG method)

Pyramids

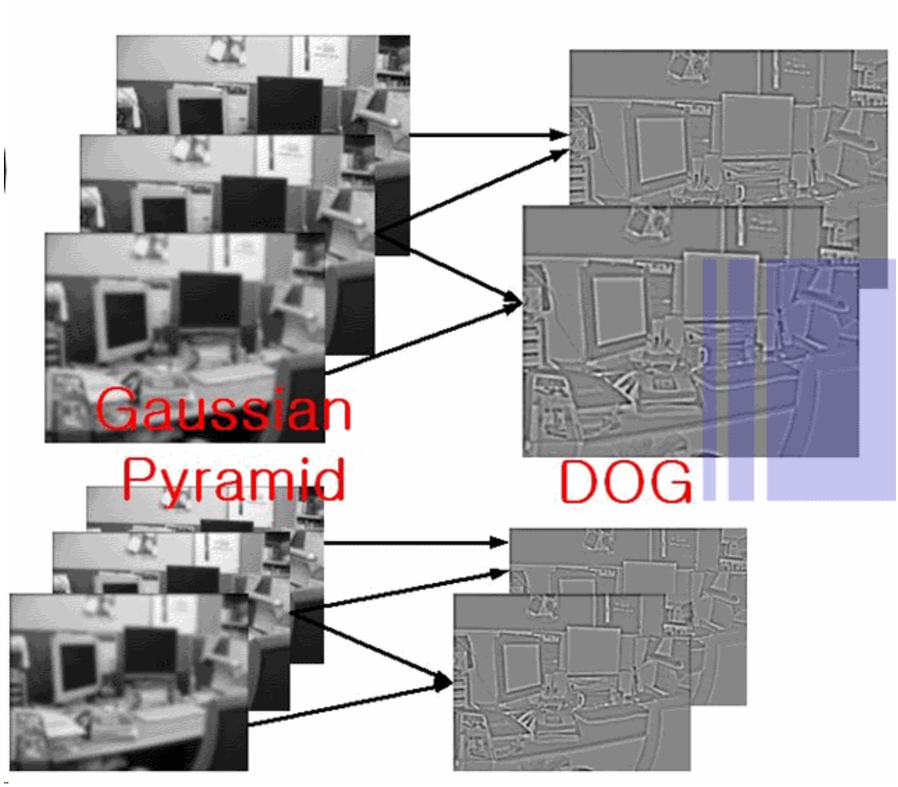
- Divide width and height by 2
- Take average of 4 pixels for each pixel (or Gaussian blur with different σ)
- Repeat until image is tiny
- Run filter over each size image and hope its robust



How to achieve scale invariance

- Scale Space: Difference of Gaussian (DOG)
  - Take DOG features from differences of these images-producing the gradient image at different scales.
  - If the feature is repeatedly present in between Difference of Gaussians, it is Scale Invariant and should be kept.

Differences Of Gaussians

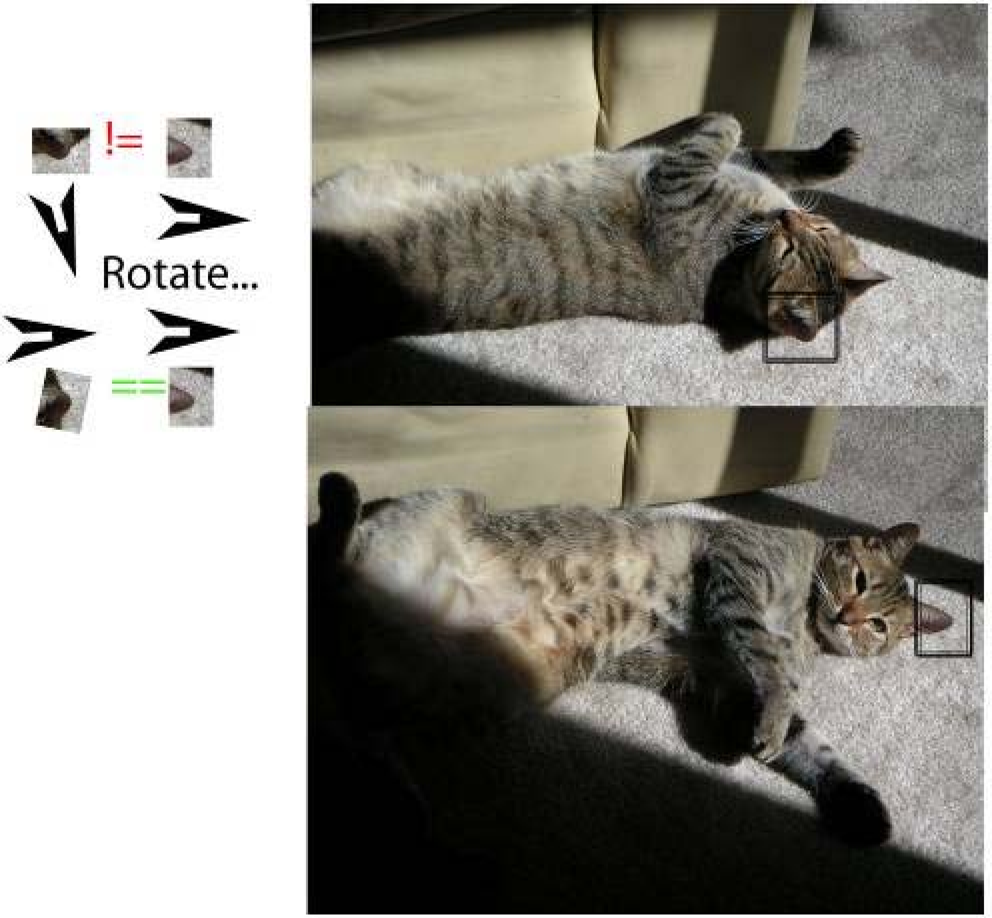


Rotation Invariance

- Compute histogram of gradient directions to identify the most dominant direction. Rotate image to the most dominant direction.

- Rotate all features to go the same way in a determined manner

Rotation Invariance



SIFT algorithm overview

- Scale-space extrema detection
  - Get tons of points from maxima+minima of DOGS
- Keypoint localization
  - Threshold on simple contrast (low contrast is generally less reliable than high for feature points)
  - Threshold based on principal curvatures to remove linear features such as edges
  - Orientation assignment
- Keypoint descriptor
  - Construct histograms of gradients (HOG)

Scale-space extrema detection

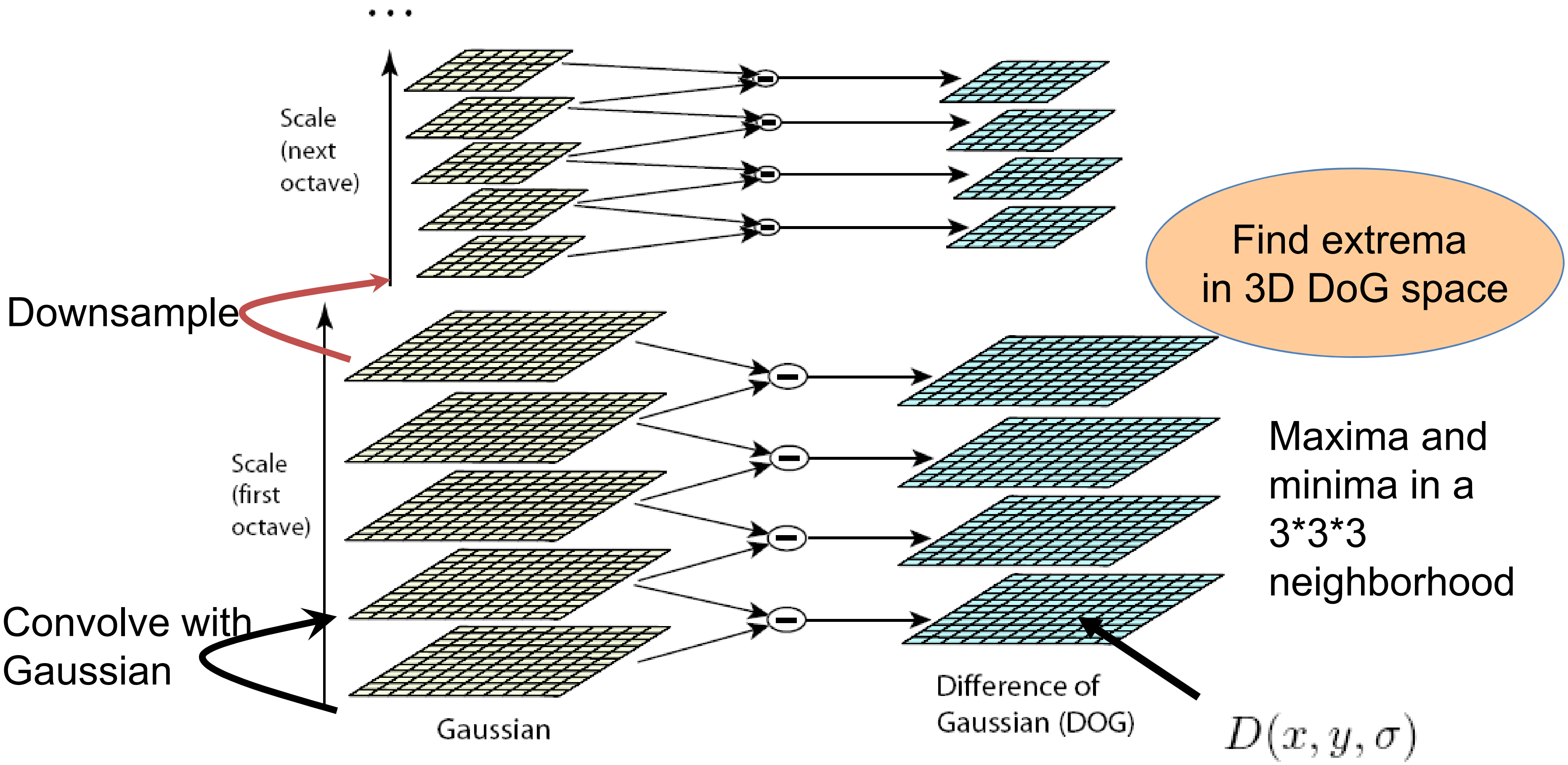
- Find the points, whose surrounding patches (with some scale) are distinctive
- An approximation to the scale-normalized Difference of Gaussian

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

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

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

Extreme Point Detection



Downsample

Convolve with Gaussian

Find extrema in 3D DoG space

Maxima and minima in a 3*3*3 neighborhood

$D(x, y, \sigma)$

At each image size, convolve with DoG of different scales.

Keypoint localization

- Eliminating extreme points with local contrast

- Eliminating edge points
  - Similar to Harris corner detector

Eliminating edge points

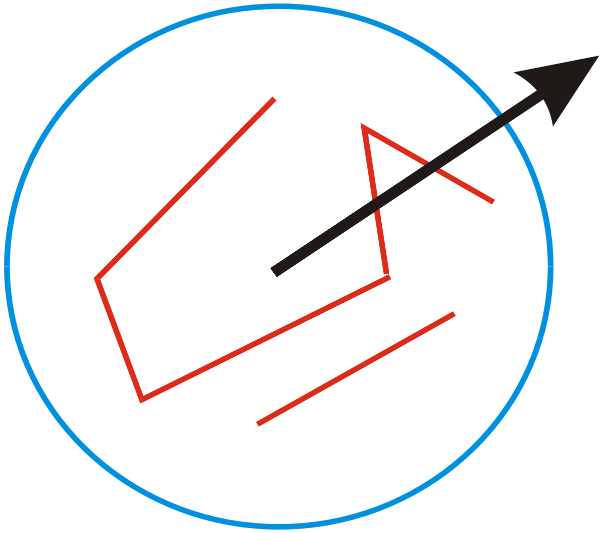
- Such a point has large principal curvature across the edge but a small one in the perpendicular direction
- The principal curvatures can be calculated from a Hessian function or covariance matrix of gradient (Harris detector)

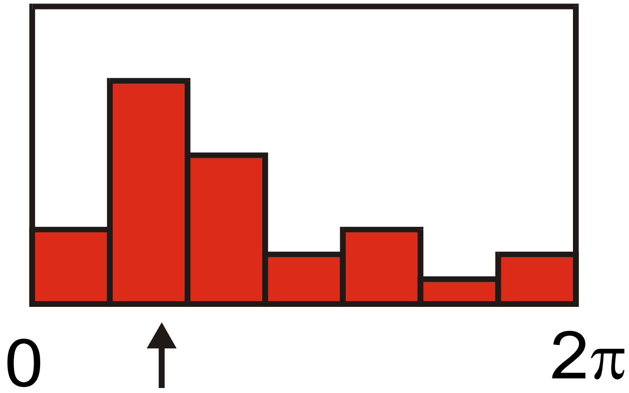
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \qquad C = \begin{bmatrix} \sum I_c^2 & \sum I_c I_r \\ \sum I_c I_r & \sum I_r^2 \end{bmatrix}$$

- The eigenvalues of H or C are proportional to the principal curvatures, so two eigenvalues shouldn’t differ too much.

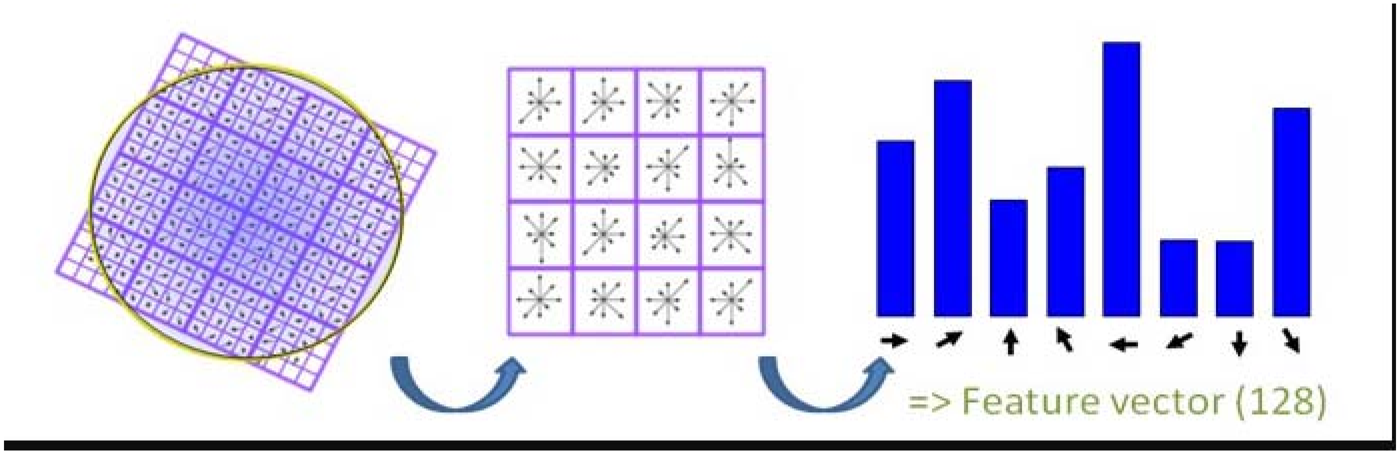
Finding Keypoints – Orientation

- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram, achieving invariance to image rotation
- Each key point specifies stable 2D coordinates (x, y, scale, orientation)





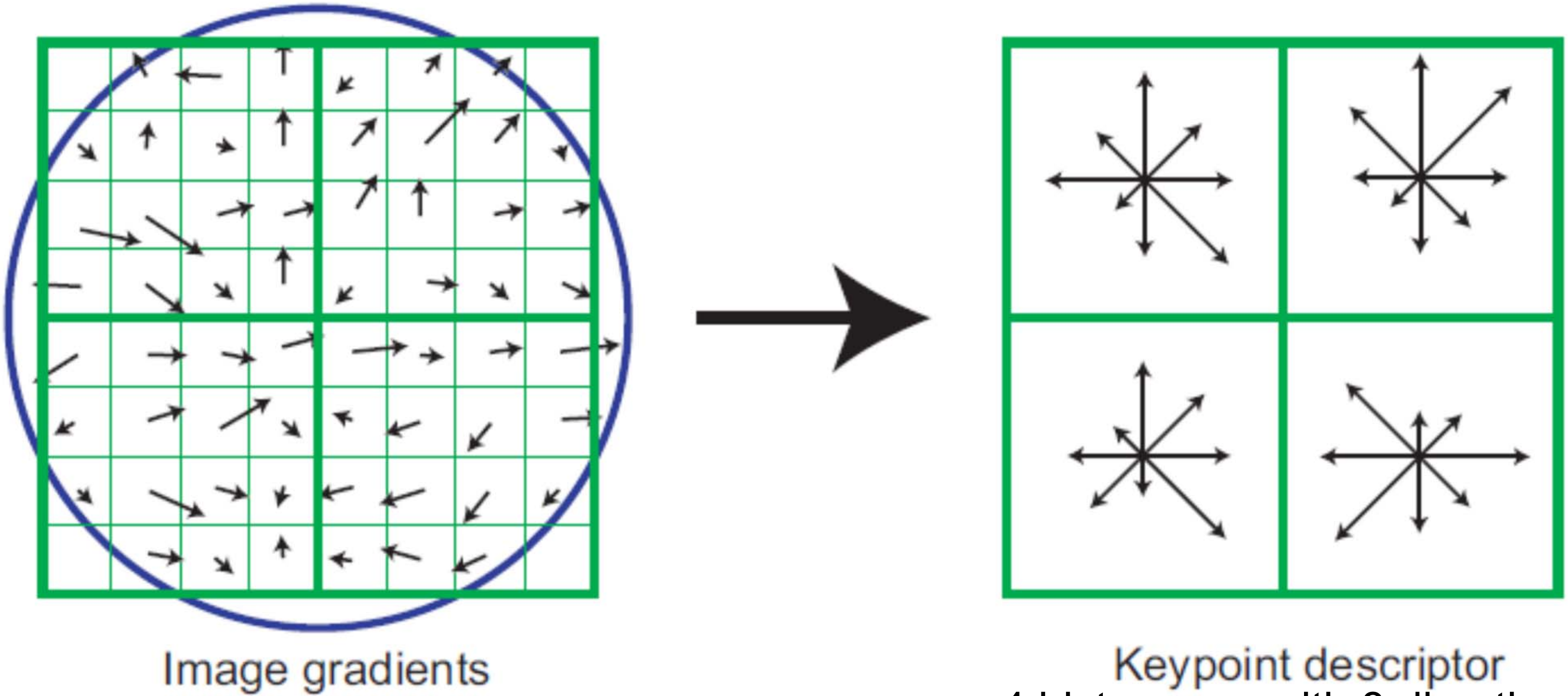
SIFT Key Feature descriptor



Take a 16x16 window of "in-between" pixels around the key point. Split that window into sixteen 4x4 windows. From each 4x4 window, generate a histogram of 8 bins, producing a total of 4x4x8=128 feature vector.

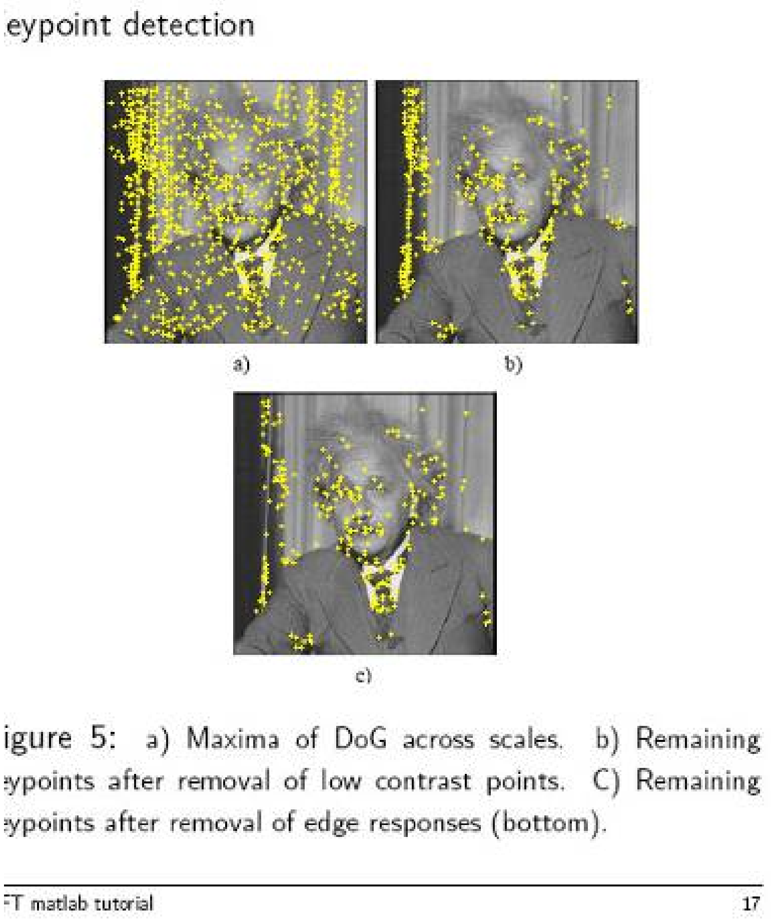
SIFT Key Feature descriptor

Create the features for each SIFT key point. Given 8 orientations and 4x4 histograms, each SIFT key is described by a vector of 128 (8x4x4) as shown below.



4 histograms with 8 direction bins for each histogram

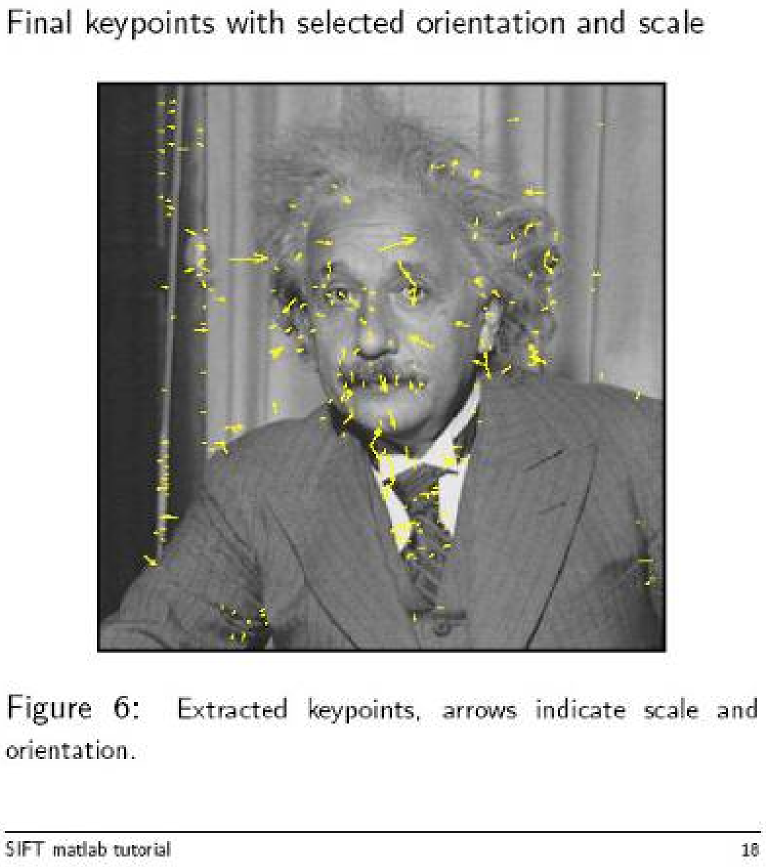
Actual SIFT stage output



eypoint detection

igure 5: a) Maxima of DoG across scales. b) Remaining ypoints after removal of low contrast points. C) Remaining ypoints after removal of edge responses (bottom).

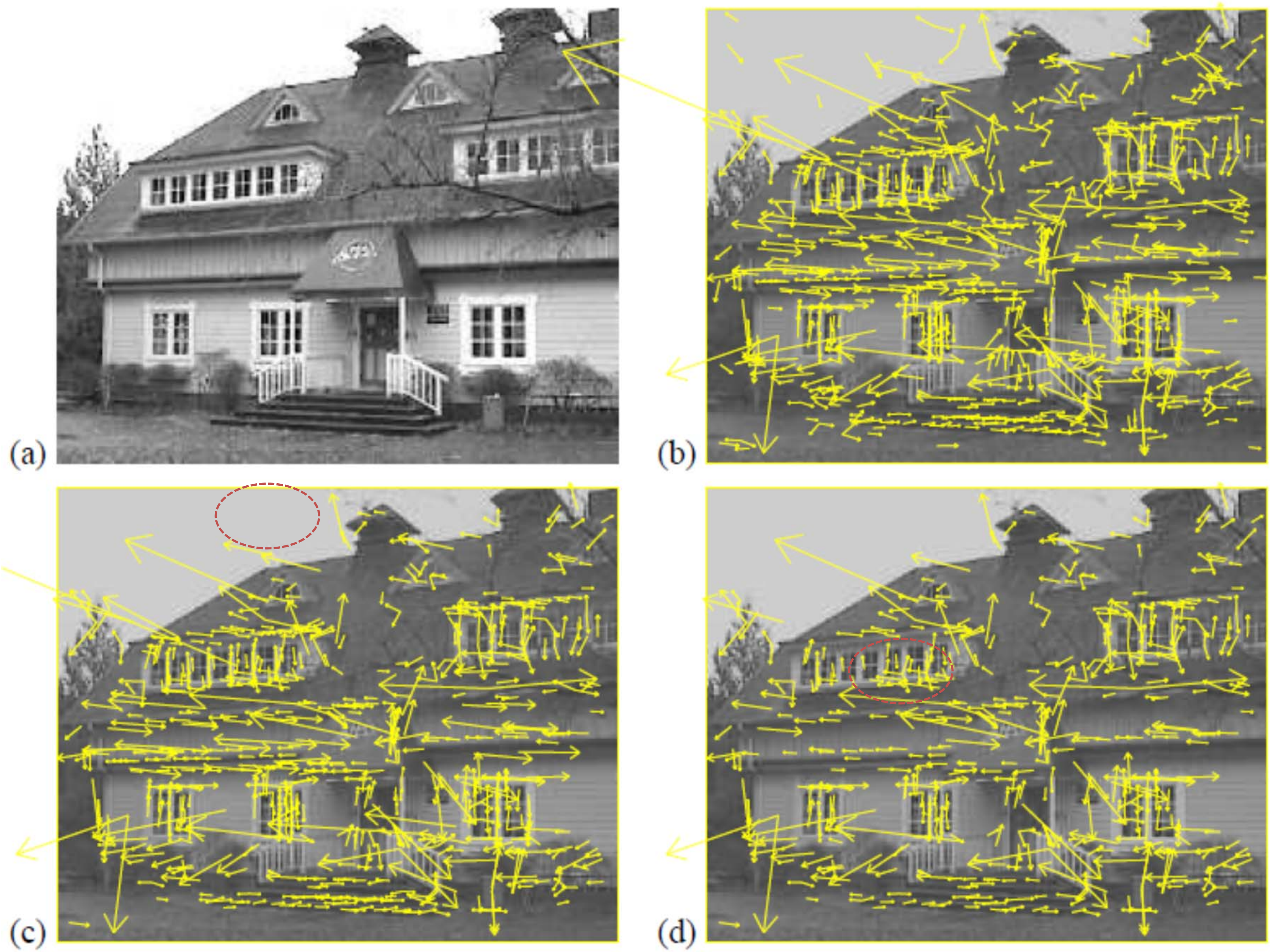
FT matlab tutorial 17



Final keypoints with selected orientation and scale

Figure 6: Extracted keypoints, arrows indicate scale and orientation.

SIFT matlab tutorial 18



After remove low contrast features
e.g. in the sky region

After removing linear features (e.g. window edges)

Image matching with SIFT features

- Nearest neighbor matching
  - Distance could be L2 norm on histograms
  - Match by (nearest neighbor distance)/(2nd nearest neighbor distance) ratio
- Use Hough transform like voting: each SIFT key point votes for an object and the object receives the most votes is the recognized object

Application: object recognition

- The SIFT key features of training images are extracted and stored
- For a query image

1. Extract SIFT key points and their features
2. Nearest neighbor matching or HT voting

Conclusion

- A novel method for detecting interest points. The most successful feature in computer vision
- Histogram of Oriented Gradients are becoming more popular
- SIFT may not be optimal for general object classification