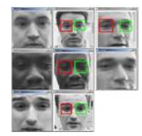
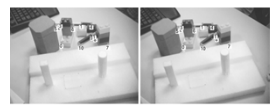
Scale Invariant Feature Transform

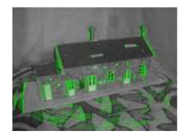
Why do we care about matching features?

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



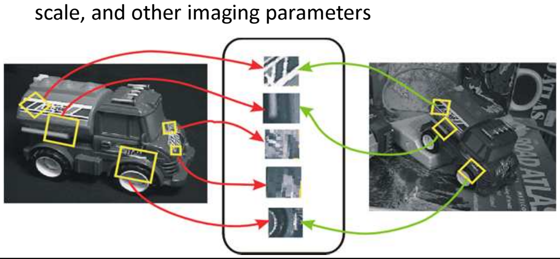


(a) (b)



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

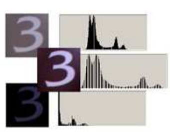
- Illumination
- Scale
- Rotation
- Affine

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

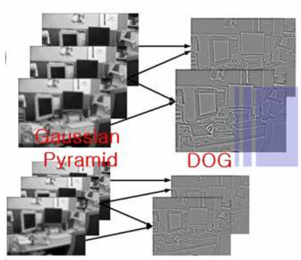
- Pyramids
  - Divide width and height by 2
  - Take average of 4 pixels for each pixel (or Gaussian blur with different σ)
  - Repeat until image is tiny
  - Run filter over each size image and hope its robust
- Scale Space (DOG method)

Pyramids

How to achieve scale invariance

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

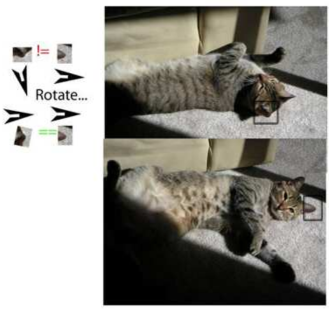
Differences Of Gaussians



Rotation Invariance

- Rotate all features to go the same way in a determined manner
- Take histogram of Gradient directions. Rotate to most dominant (maybe second if its good enough, sub-Bin accuracy)

Rotation Invariance



Affine Invariance

- Easy way: Warp your training and hope
- Fancy way: design your feature itself to be robust against affine transformations (SIFT method)

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
  - Orientation assignment
- Keypoint descriptor
  - Construct histograms of gradients (HOG)

Scale-space extrema detection

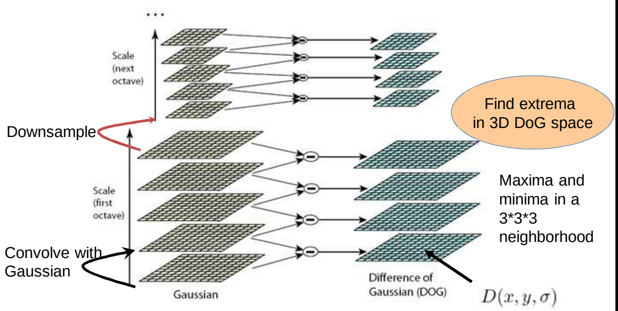
- Find the points, whose surrounding patches (with some scale) are distinctive
- An approximation to the scale-normalized Difference of Gaussian

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

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

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

Extreme Point Detection



Keypoint localization

- Eliminating extreme points with local contrast
- Eliminating edge points
  - Similar to Harris corner detector

Eliminating edge points

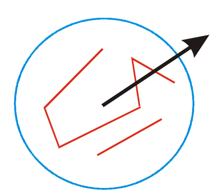
- Such a point has large principal curvature across the edge but a small one in the perpendicular direction
- The principal curvatures can be calculated from a Hessian function or covariance matrix of gradient (Harris detector)

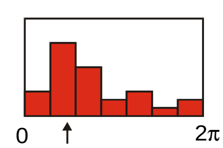
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \qquad C = \begin{bmatrix} \sum I_c^2 & \sum I_c I_r \\ \sum I_c I_r & \sum I_r^2 \end{bmatrix}$$

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

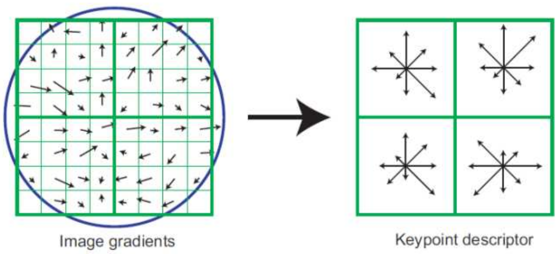
Finding Keypoints – Orientation

- 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)

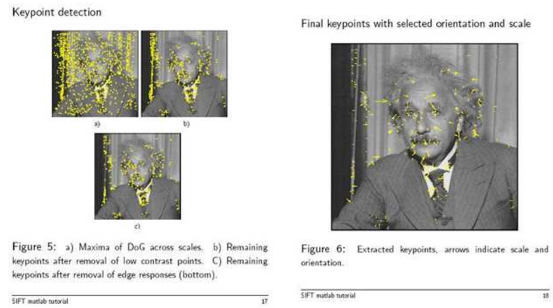




Feature descriptor



Actual SIFT stage output


Keypoint detection

a) b) c)

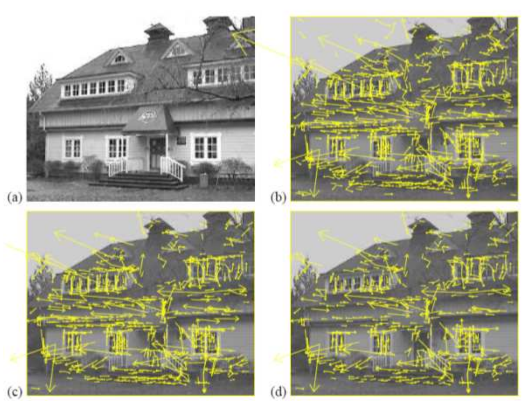
Figure 5: a) Maxima of DoG across scales. b) Remaining keypoints after removal of low contrast points. C) Remaining keypoints after removal of edge responses (bottom).

SIFT matlab tutorial 17

Final keypoints with selected orientation and scale

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

SIFT matlab tutorial 18



How to use these features?

- Distance could be L2 norm on histograms
- Match by (nearest neighbor distance)/(2nd nearest neighbor distance) ratio

Application: object recognition

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

1. Extract SIFT feature
2. Nearest neighbor matching

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