SIFT: SCALE INVARIANT FEATURE TRANSFORM BY DAVID LOWE

Presented by: Jason Clemons

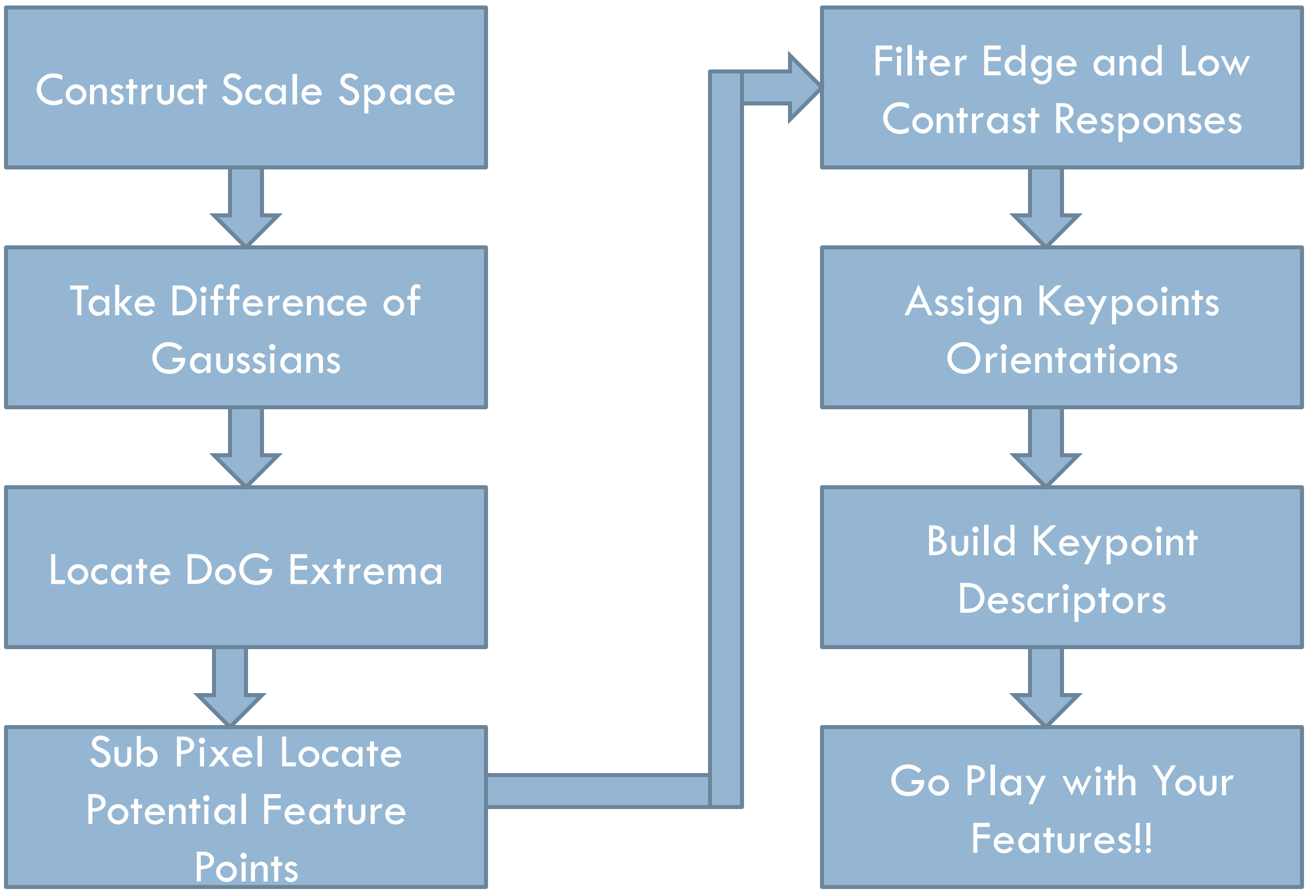
Overview

- Motivation of Work
- Overview of Algorithm
- Scale Space and Difference of Gaussian
- Keypoint Localization
- Orientation Assignment
- Descriptor Building
- Application

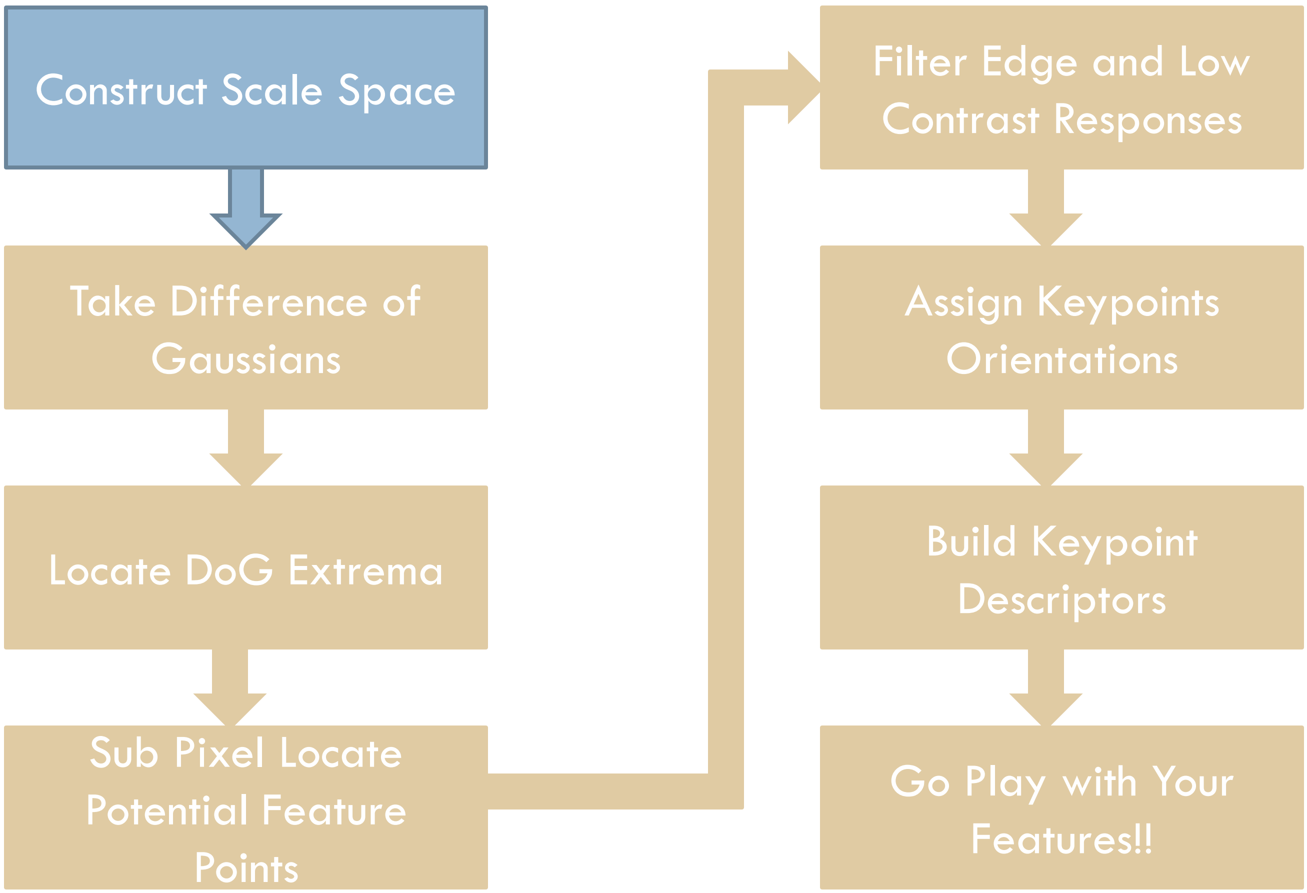
Motivation

- Image Matching
  - Correspondence Problem
- Desirable Feature Characteristics
  - Scale Invariance
  - Rotation Invariance
  - Illumination invariance
  - Viewpoint invariance

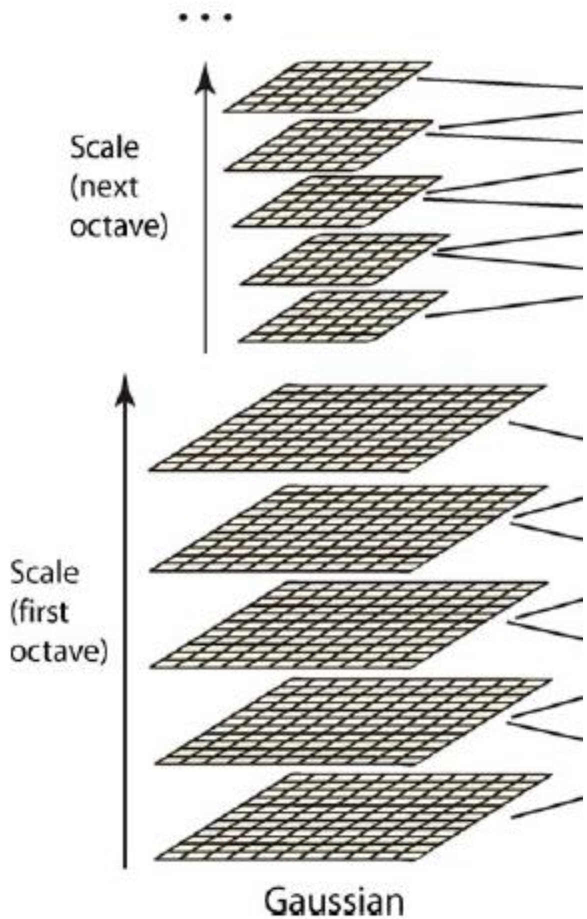
Overview Of Algorithm



Constructing Scale Space



Scale Space



Constructing Scale Space

- Gaussian kernel used to create scale space
  - Only possible scale space kernel (Lindberg '94)

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

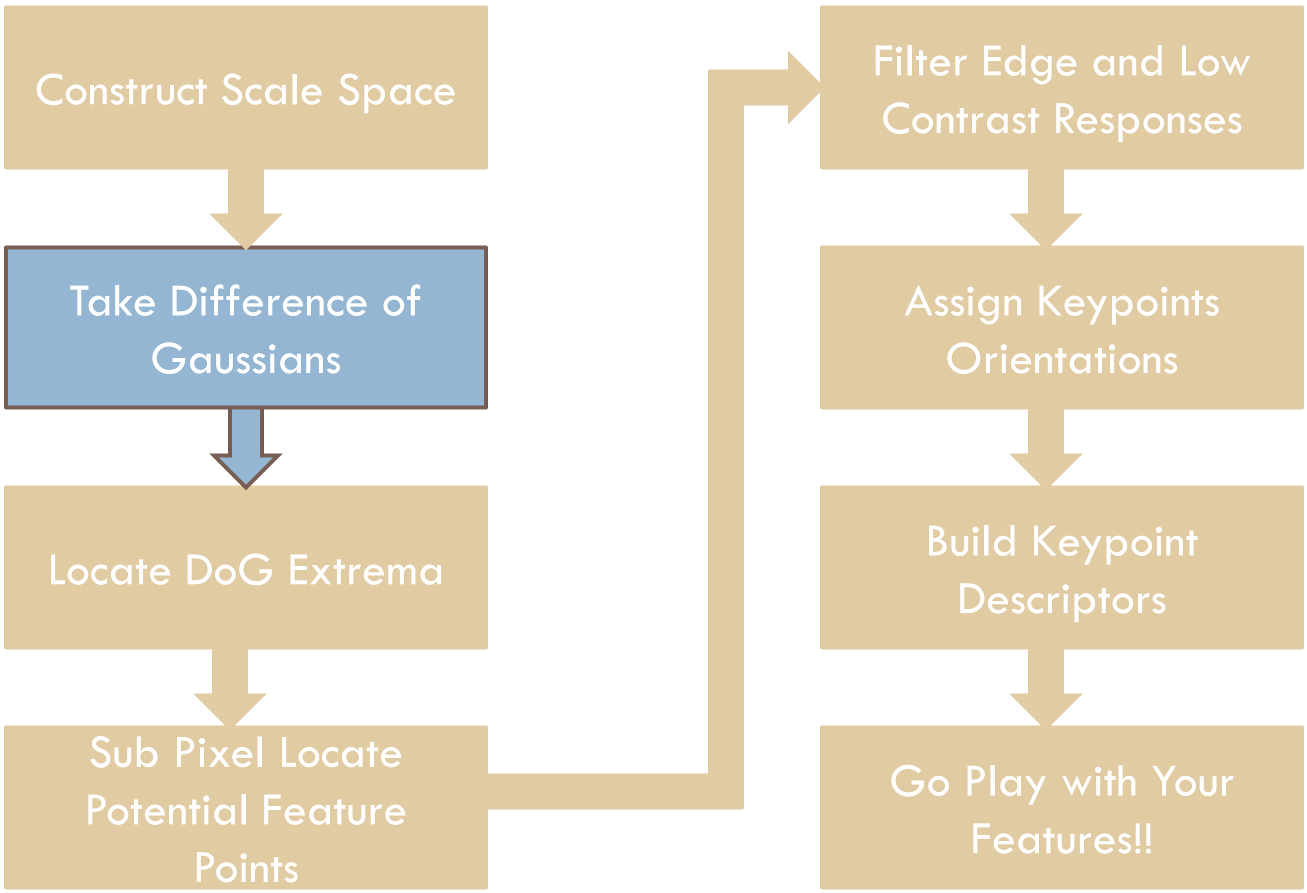
where

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

Laplacian of Gaussians

- LoG - $\sigma^2\Delta^2 G$
- Extrema Useful
  - Found to be stable features
  - Gives Excellent notion of scale
- Calculation costly so instead….

Take DoG



Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

Difference of Gaussian

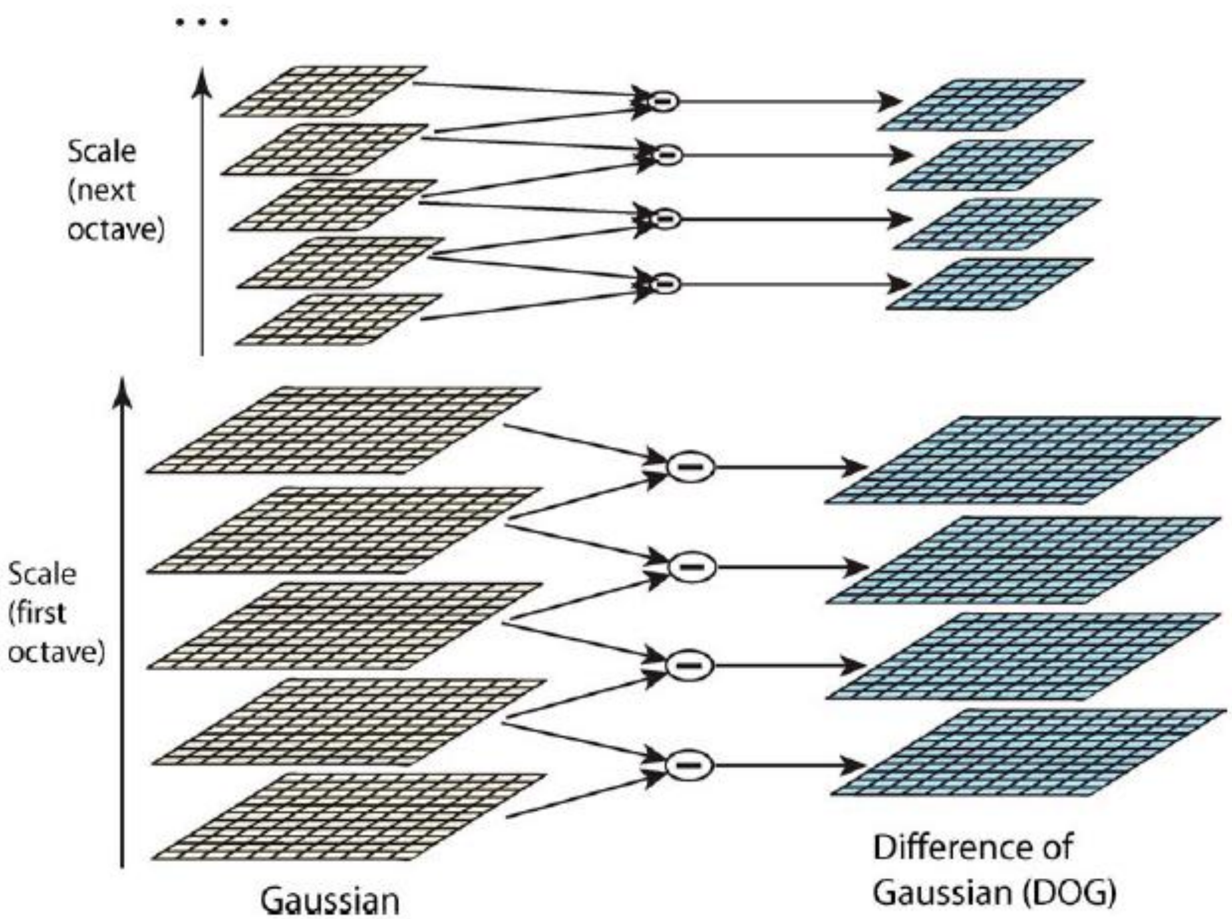
- Approximation of Laplacian of Gaussians

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G.$$

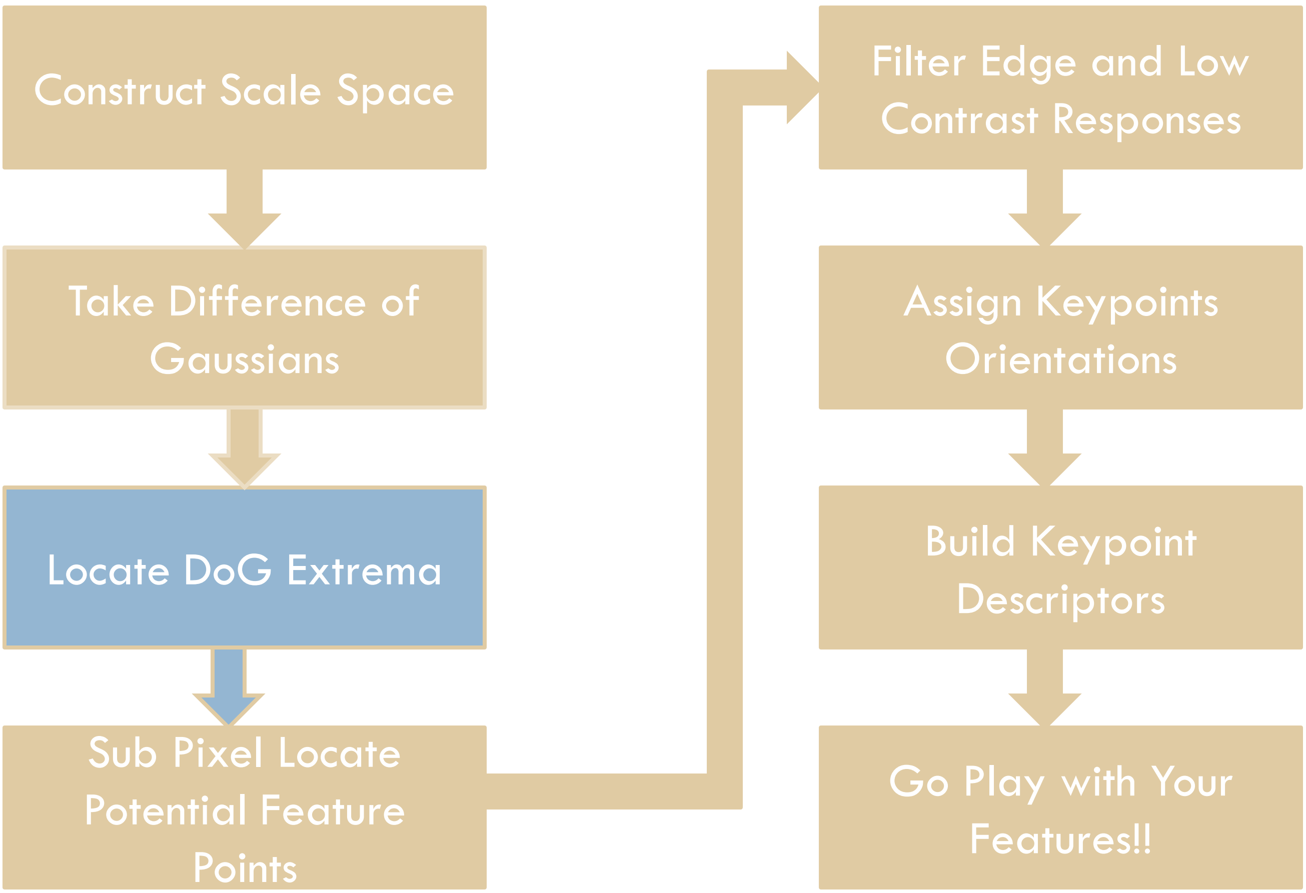
$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \qquad (1)$$

DoG Pyramid



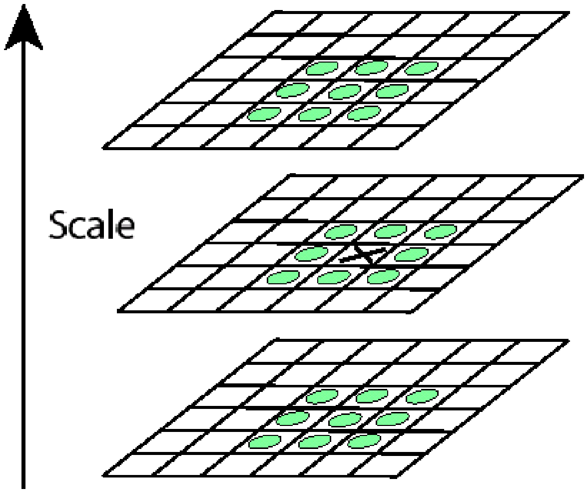
...
Scale
(next
octave)
Scale
(first
octave)
Gaussian
Difference of
Gaussian (DOG)

DoG Extrema

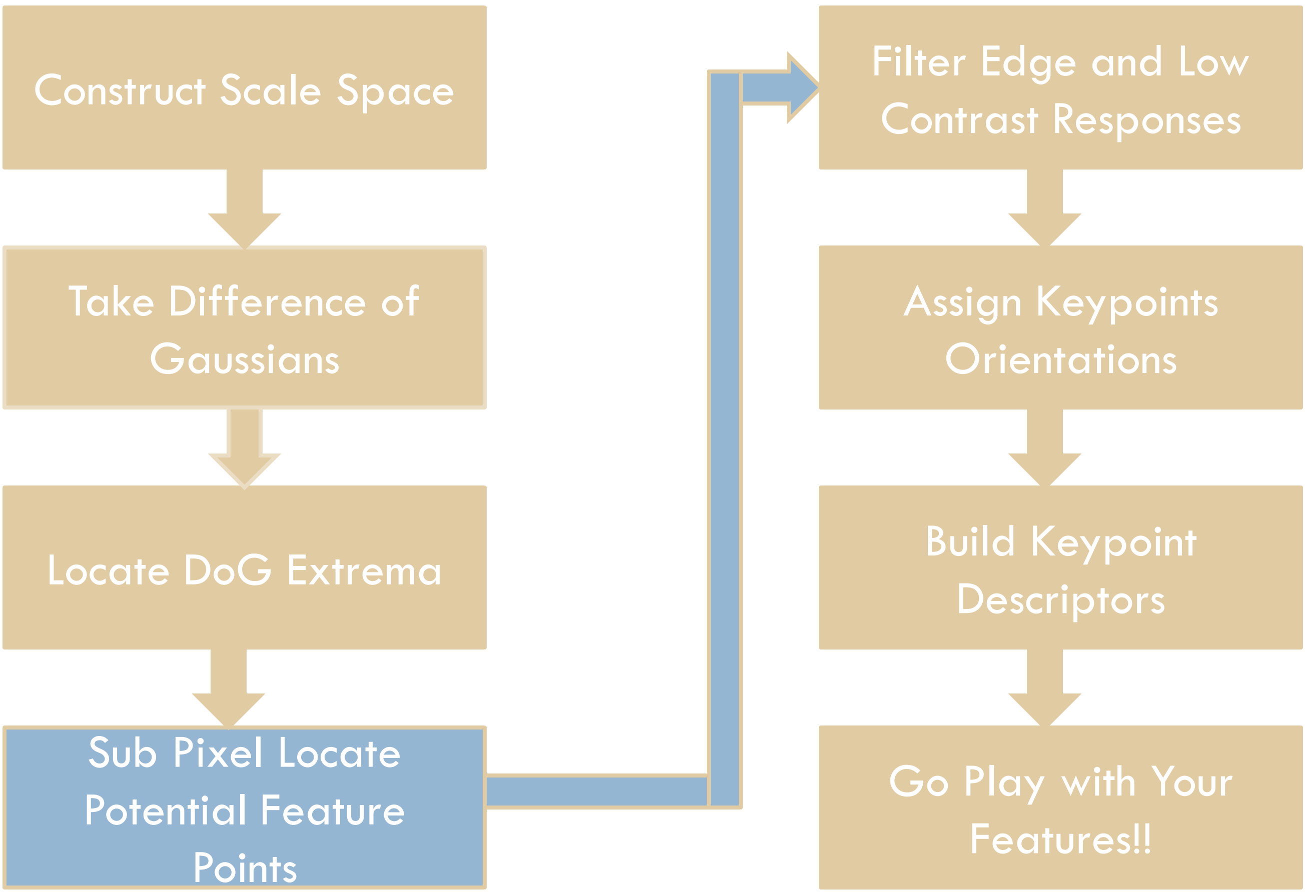

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

Locate the Extrema of the DoG

- Scan each DOG image
  - Look at all neighboring points (including scale)
  - Identify Min and Max
    - 26 Comparisons



Sub pixel Localization



Sub-pixel Localization

- 3D Curve Fitting

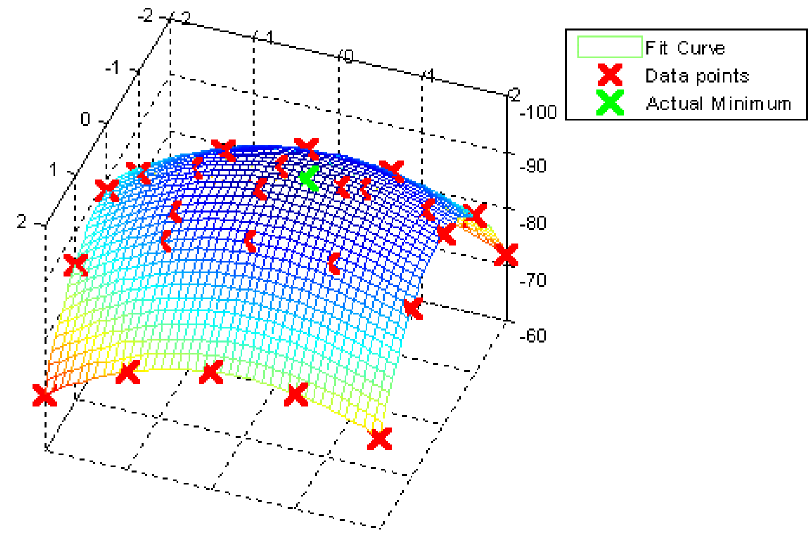
  Taylor Series Expansion

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

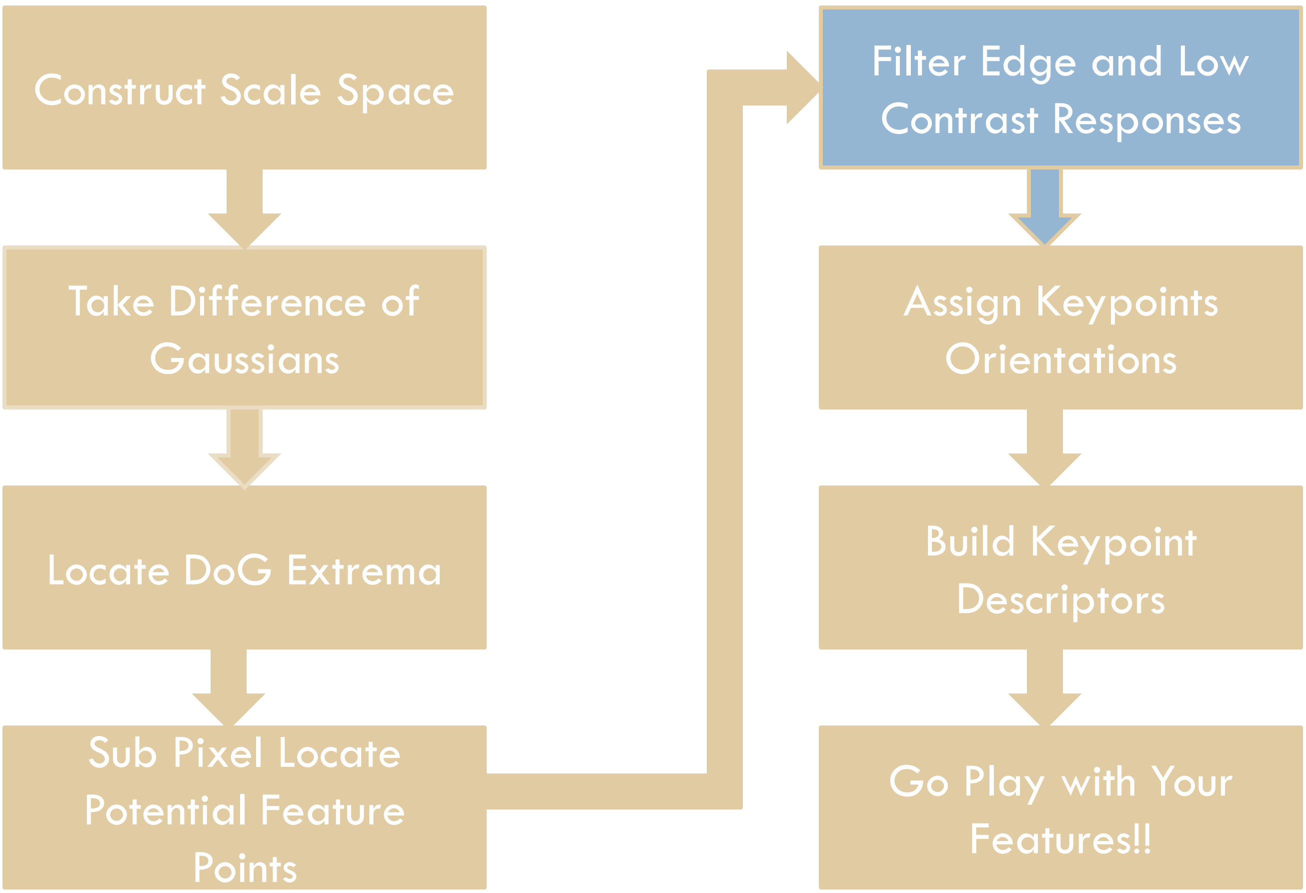
  Differentiate and set to 0

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}.$$

  to get location in terms of (x,y,σ)



Filter Responses

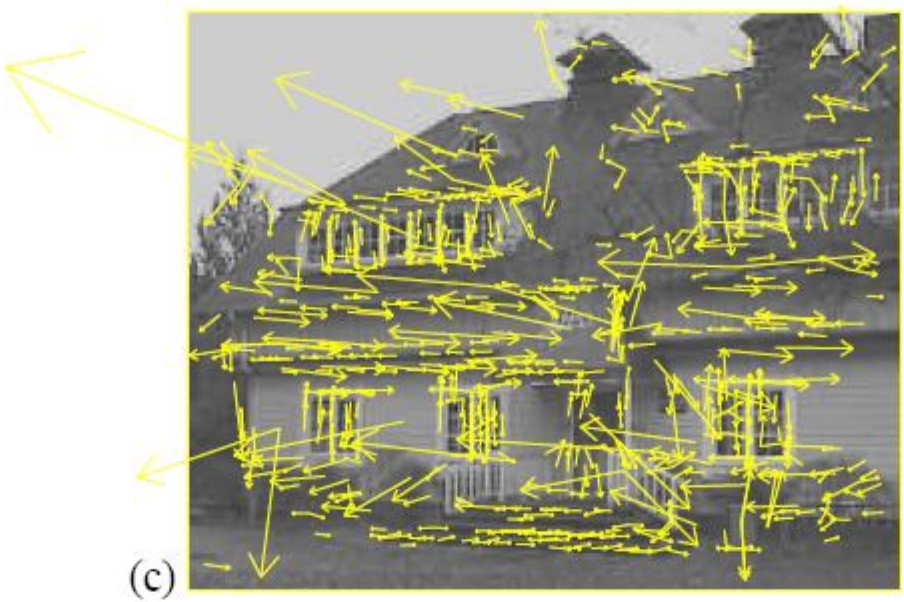


Filter Low Contrast Points

- Low Contrast Points Filter
  - Use Scale Space value at previously found location

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}}.$$

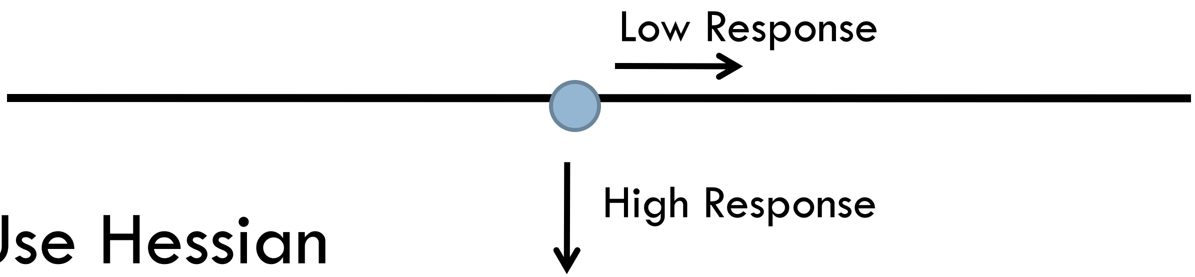
The House With Contrast Elimination



(c)

Edge Response Elimination

- Peak has high response along edge, poor other direction



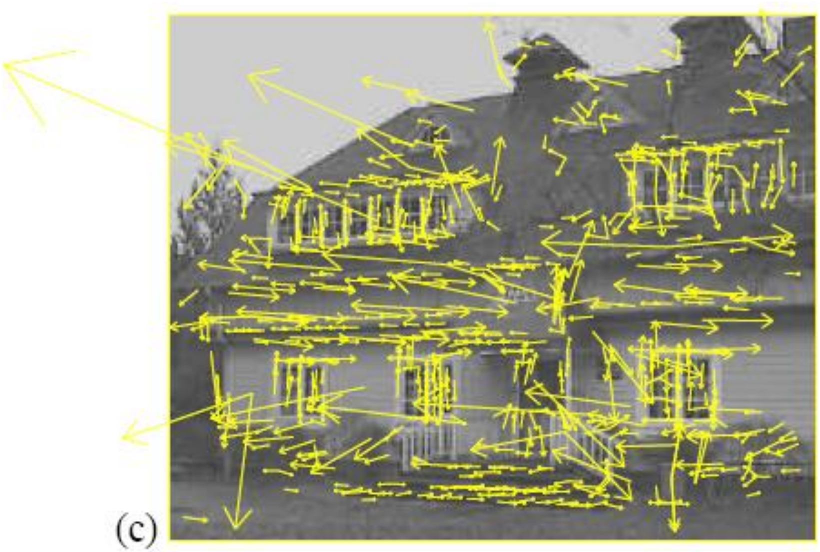
- Use Hessian
  - Eigenvalues Proportional to principle Curvatures
  - Use Trace and Determinant

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

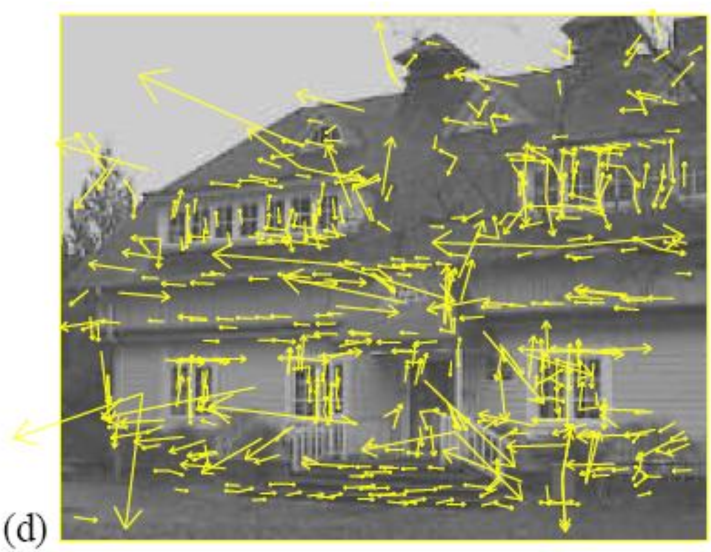
$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$$

Results On The House

(c)



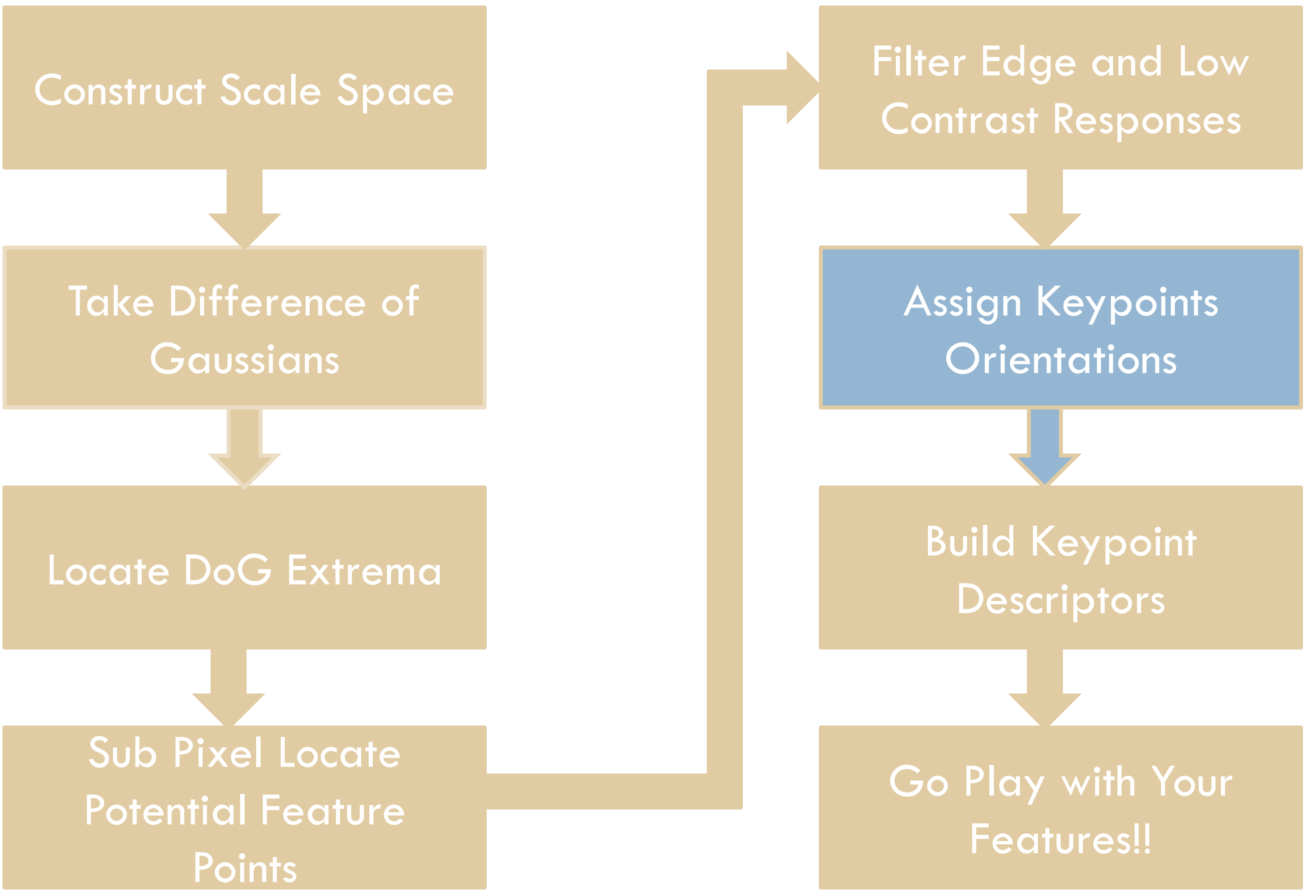
(d)



Apply Contrast Limit

Apply Contrast and Edge Response Elimination

Assign Keypoint Orientations



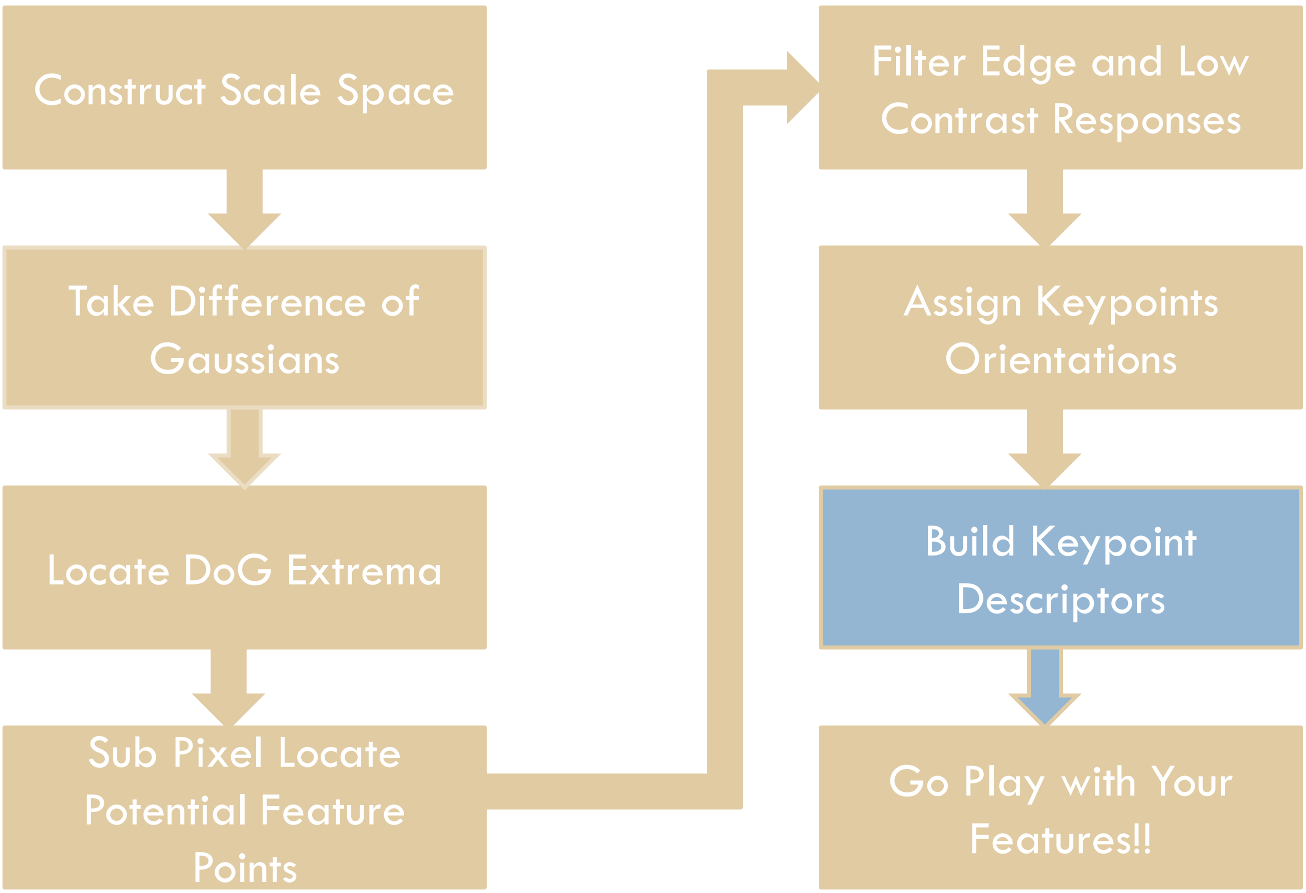
Orientation Assignment

- Compute Gradient for each blurred image

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

- For region around keypoint
  - Create Histogram with 36 bins for orientation
  - Weight each point with Gaussian window of 1.5σ
  - Create keypoint for all peaks with value>=.8 max bin
    - Note that a parabola is fit to better locate each max (least squares)

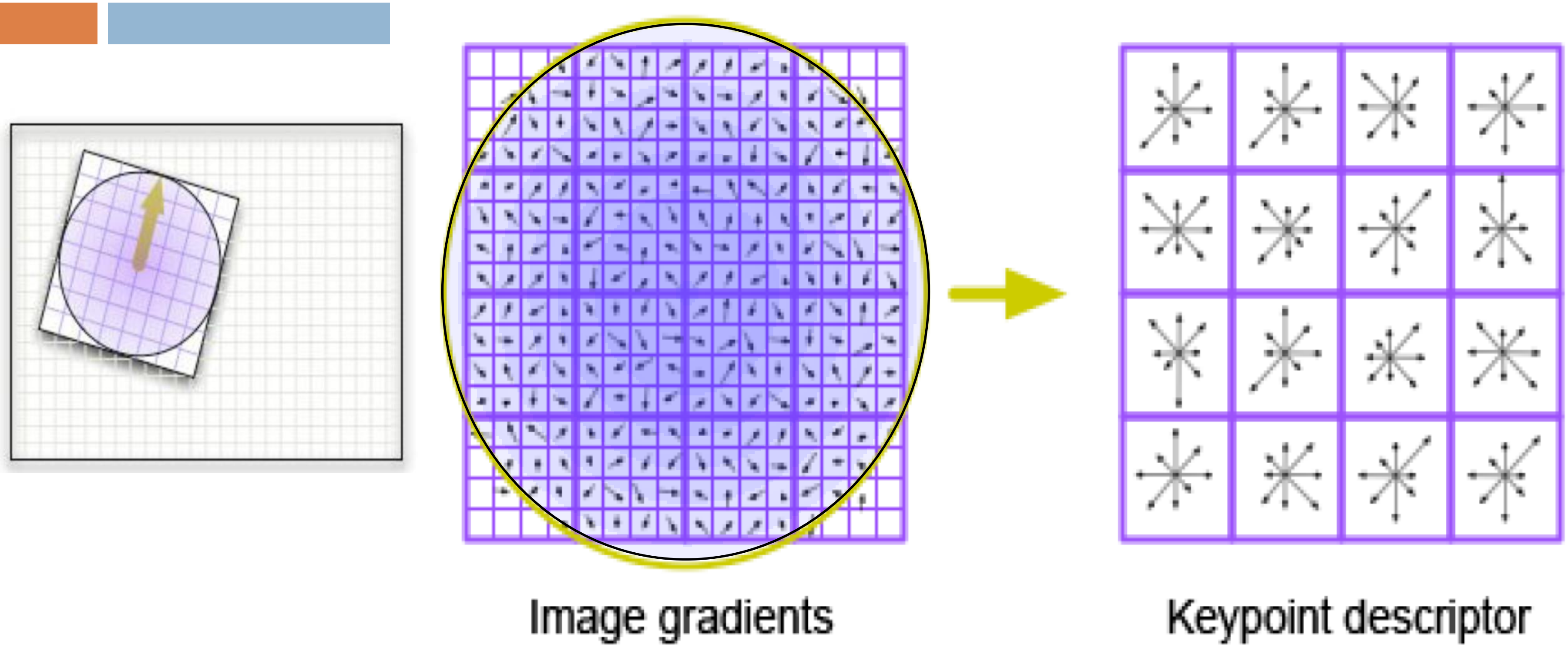
Build Keypoint Descriptors



Building the Descriptor

- Find the blurred image of closest scale
- Sample the points around the keypoint
- Rotate the gradients and coordinates by the previously computer orientation
- Separate the region in to sub regions
- Create histogram for each sub region with 8 bins
  - Weight the samples with N(σ) = 1.5 Region width
  - Trilinear Interpolation (1-d factor) to place in histogram bins

Building a Descriptor



- Actual implementation uses 4x4 descriptors from 16x16 which leads to a 4x4x8=128 element vector

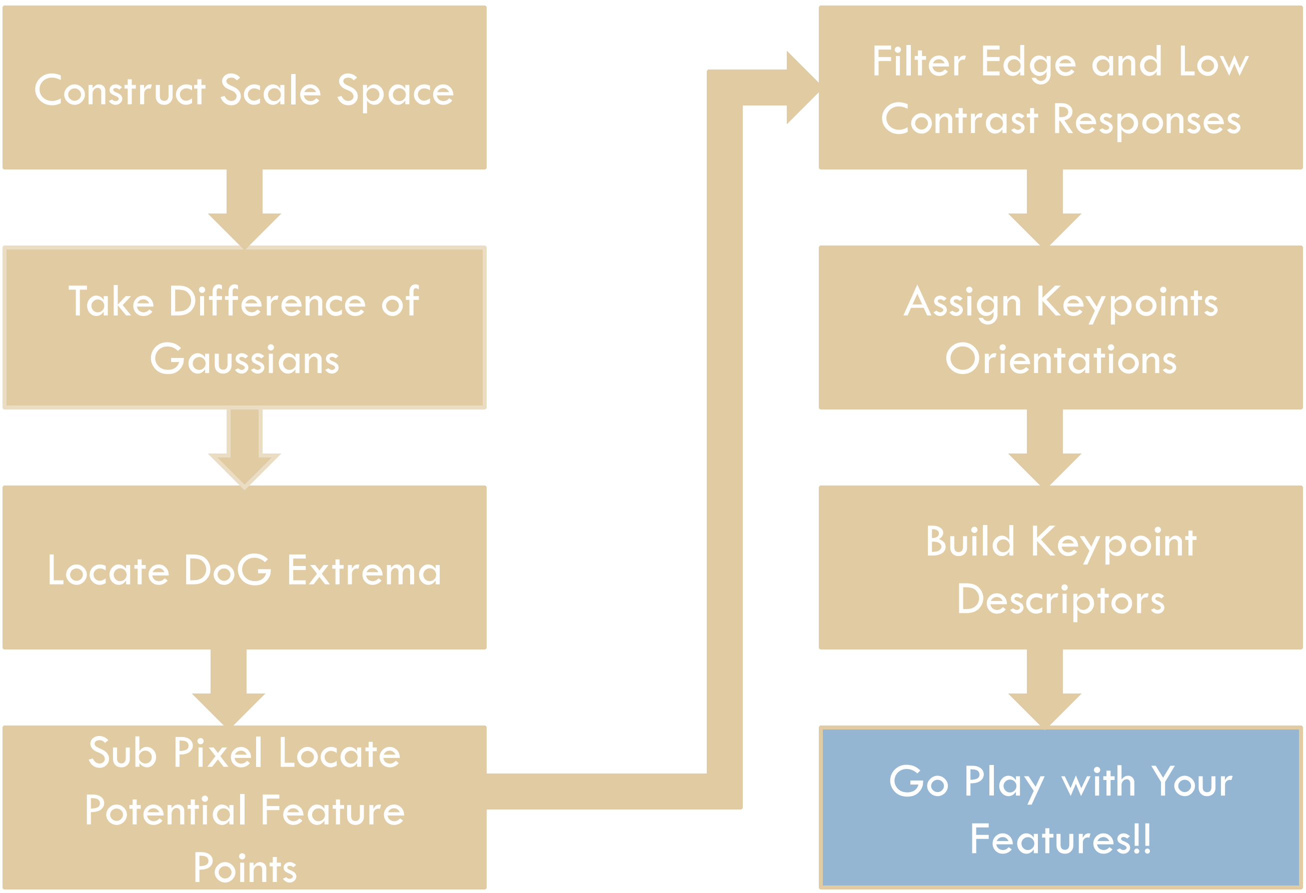
Illumination Issues

- Illumination changes can cause issues
  - So normalize the vector
- Solves Affine but what non-linear sources like camera saturation?
  - Cap the vector elements to .2 and renormalize
- Now we have some illumination invariance

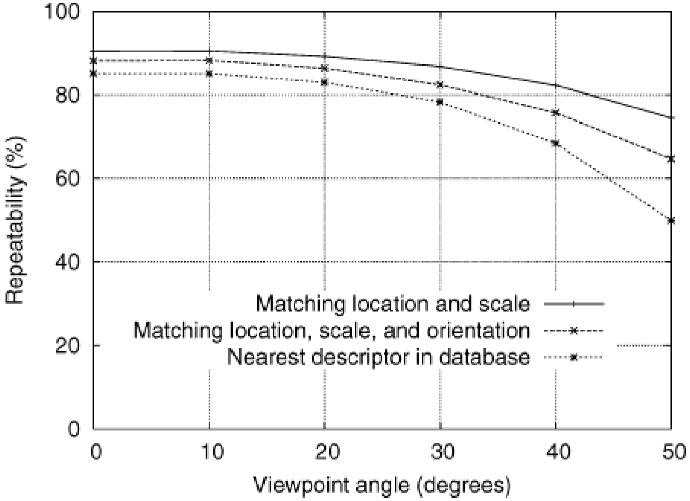
Results Check

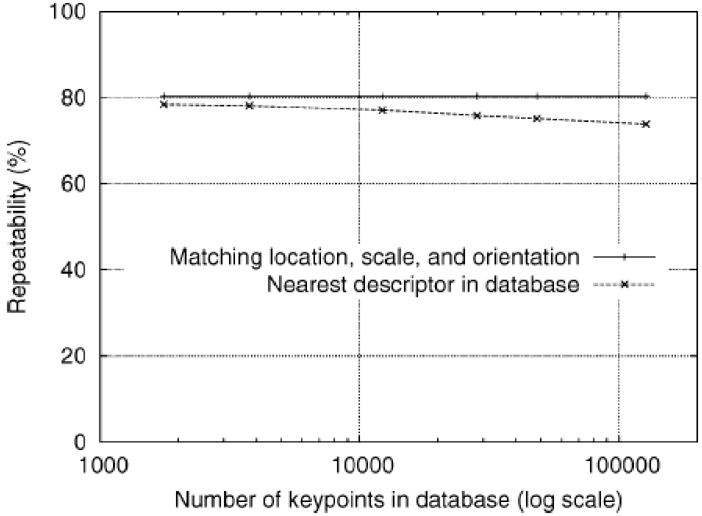
- Scale Invariance
  - Scale Space usage – Check
- Rotation Invariance
  - Align with largest gradient – Check
- Illumination Invariance
  - Normalization – Check
- Viewpoint Invariance
  - For small viewpoint changes – Check (mostly)

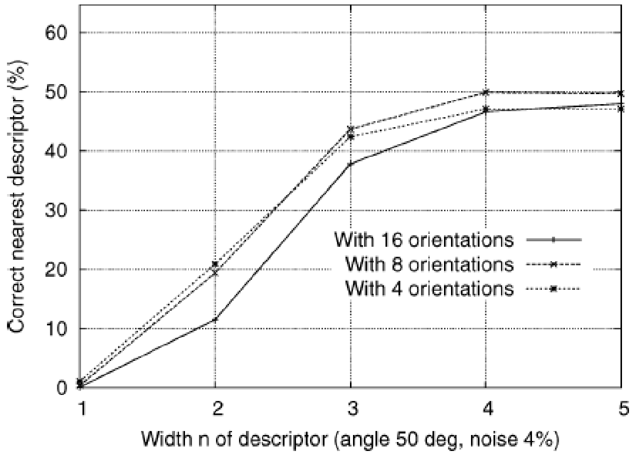
Constructing Scale Space



Supporting Data for Performance





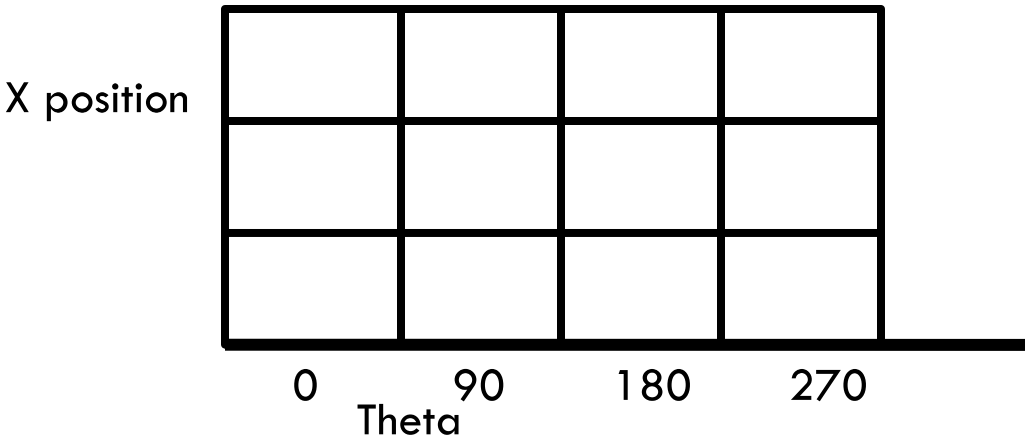


About matching…

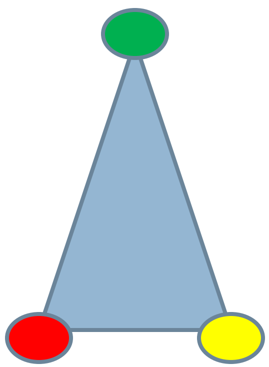
- Can be done with as few as 3 features.
- Use Hough transform to cluster features in pose space
  - Have to use broad bins since 4 items but 6 dof
  - Match to 2 closest bins
- After Hough finds clusters with 3 entries
  - Verify with affine constraint

Hough Transform Example (Simplified)

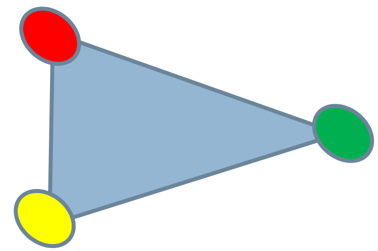
- For the Current View, color feature match with the database image
- If we take each feature and align the database image at that feature we can vote for the x position of the center of the object and the theta of the object based on all the poses that align



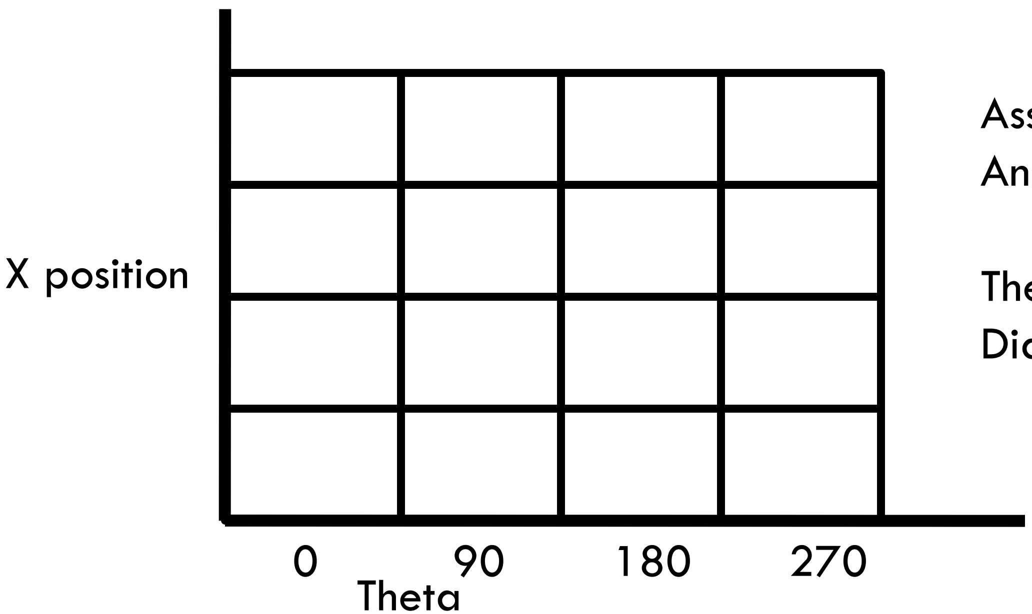
Hough Transform Example (Simplified)



Database Image



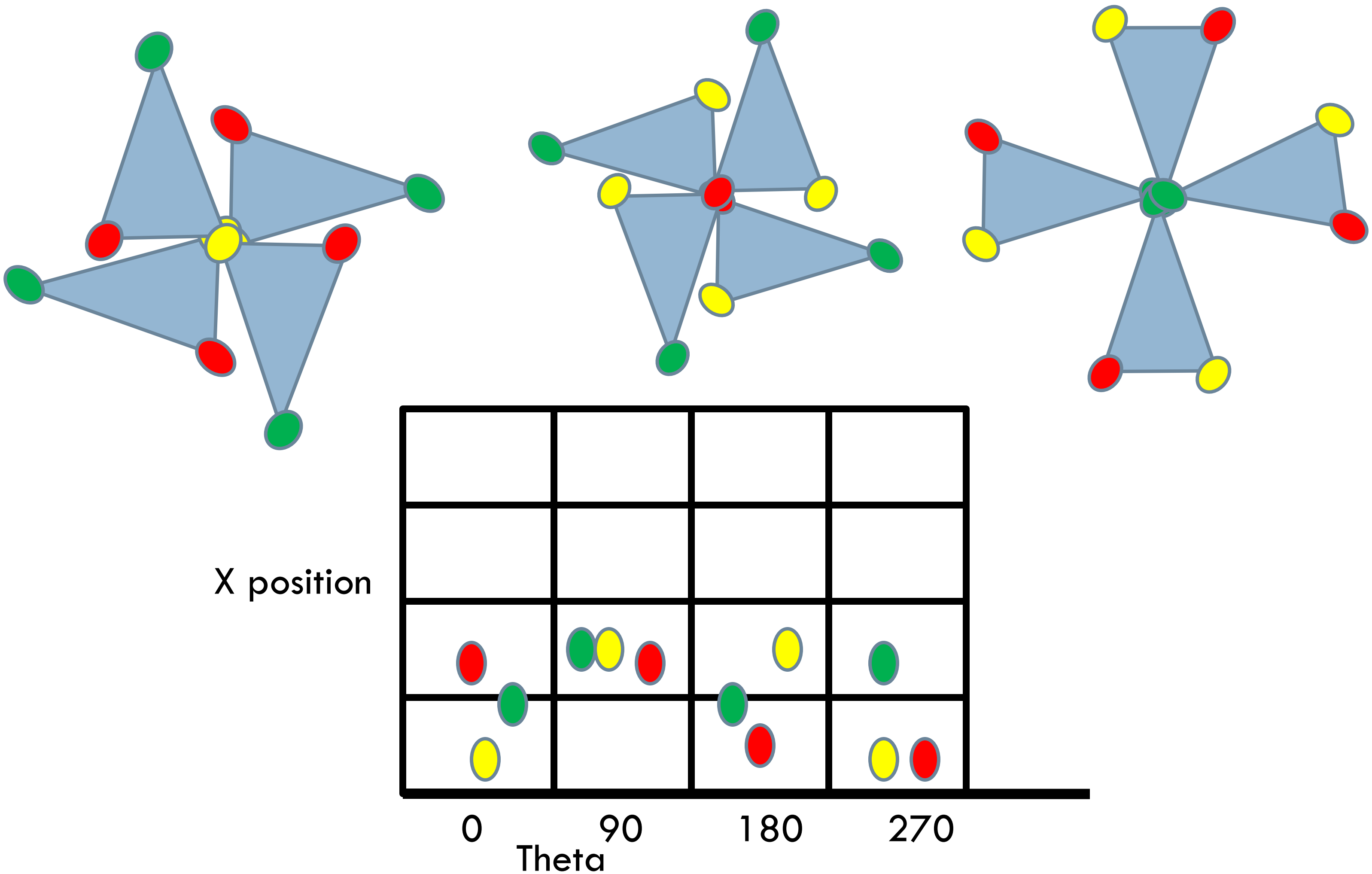
Current Item



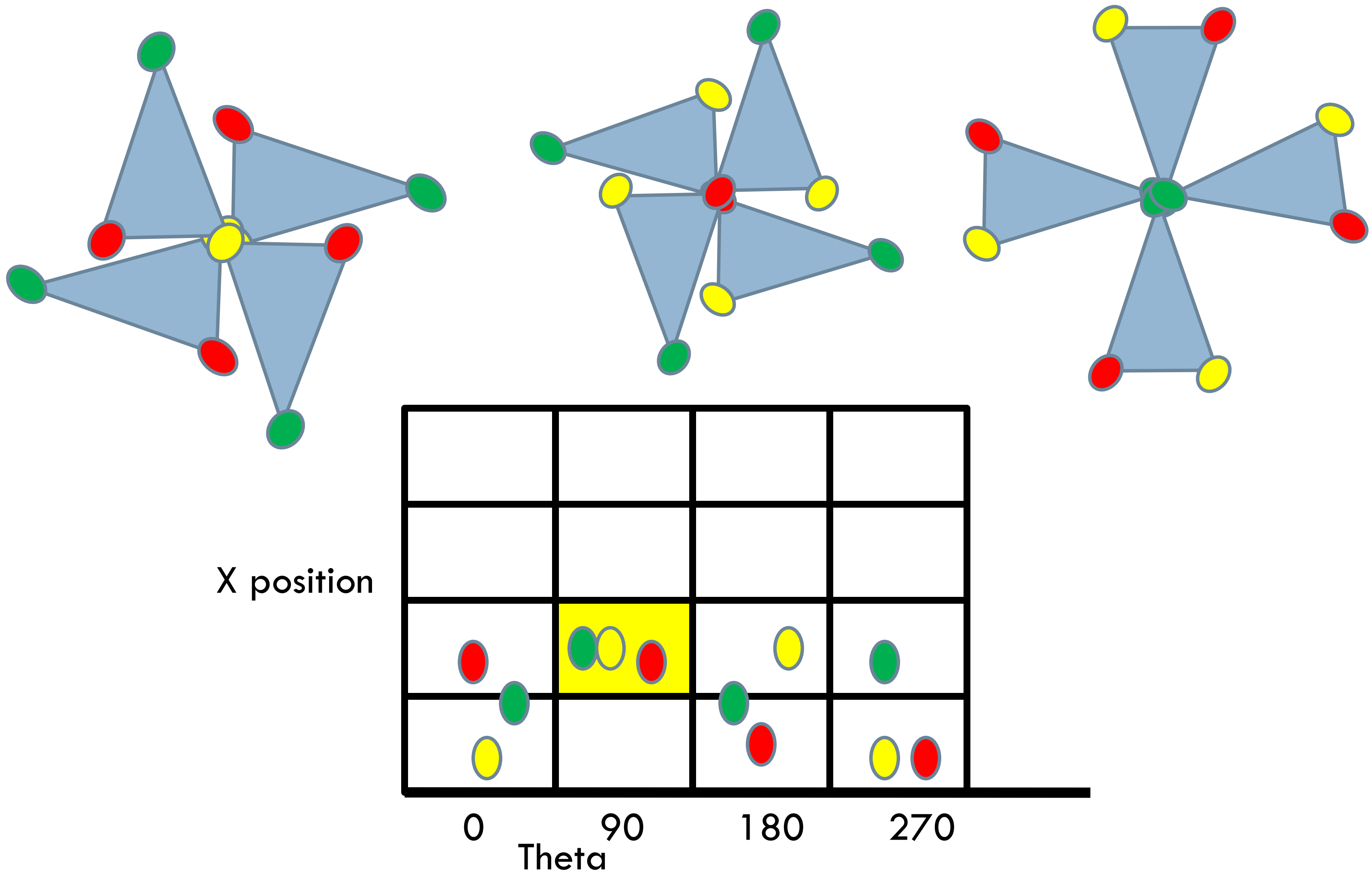
Assume we have 4 x locations
And only 4 possible rotations (thetas)

Then the Hough space can look like the
Diagram to the left

Hough Transform Example (Simplified)

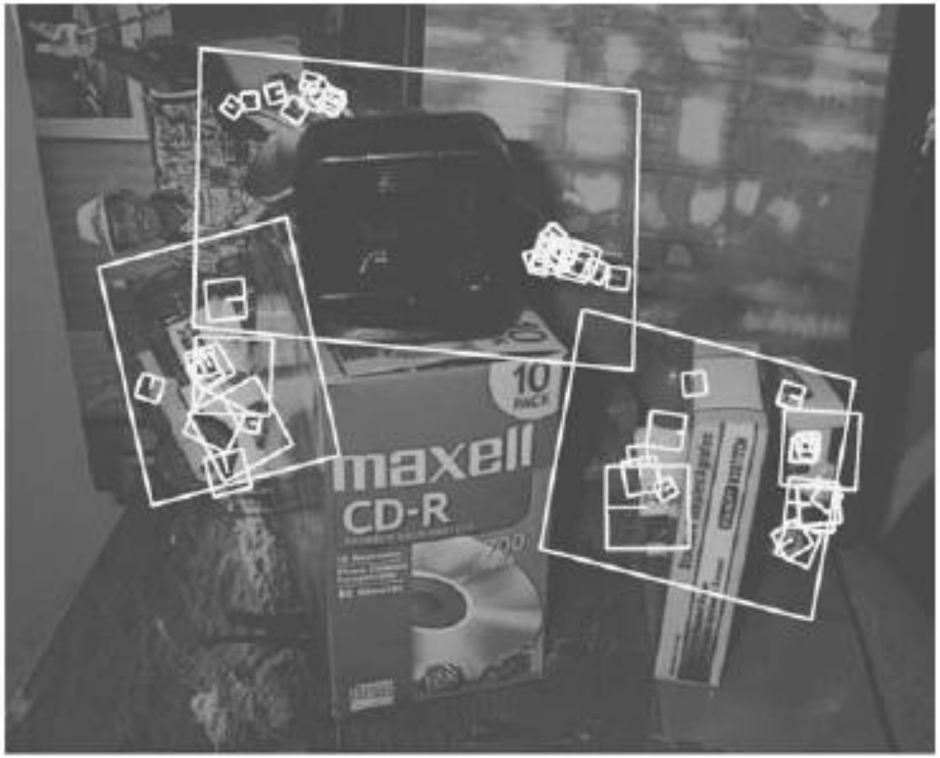


Hough Transform Example (Simplified)



Playing with our Features: Where's Traino and Froggy?

Here's Traino and Froggy!



Outdoors anyone?

Questions?

Credits

- Lowe, D. “Distinctive image features from scale-invariant keypoints” International Journal of Computer Vision, 60, 2 (2004), pp. 91-110
- Pele, Ofir. SIFT: Scale Invariant Feature Transform. Sift.ppt
- Lee, David. Object Recognition from Local Scale-Invariant Features (SIFT). O319.Sift.ppt
- Some Slide Information taken from Silvio Savarese