

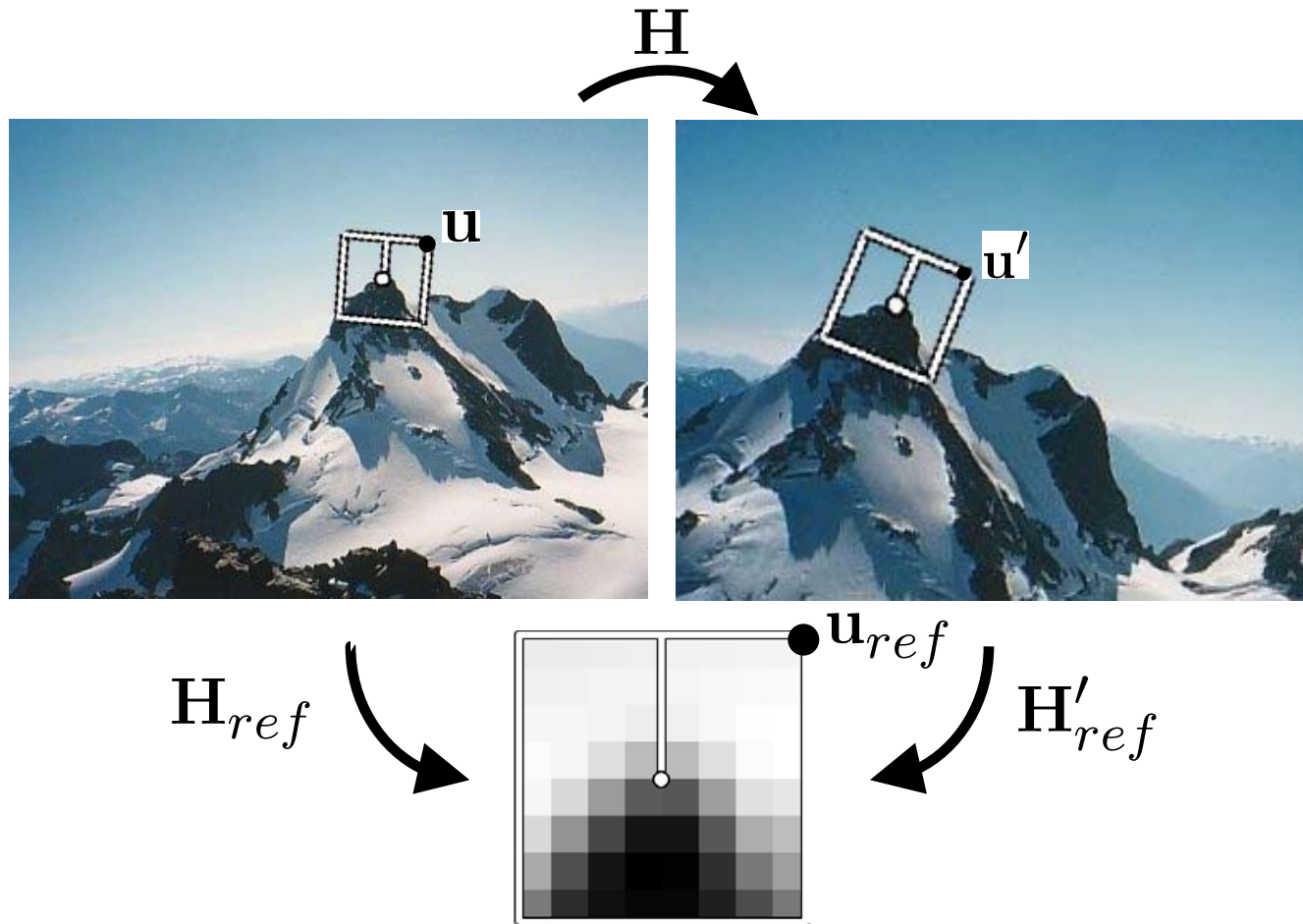
---

# The SIFT (Scale Invariant Feature Transform) Detector and Descriptor

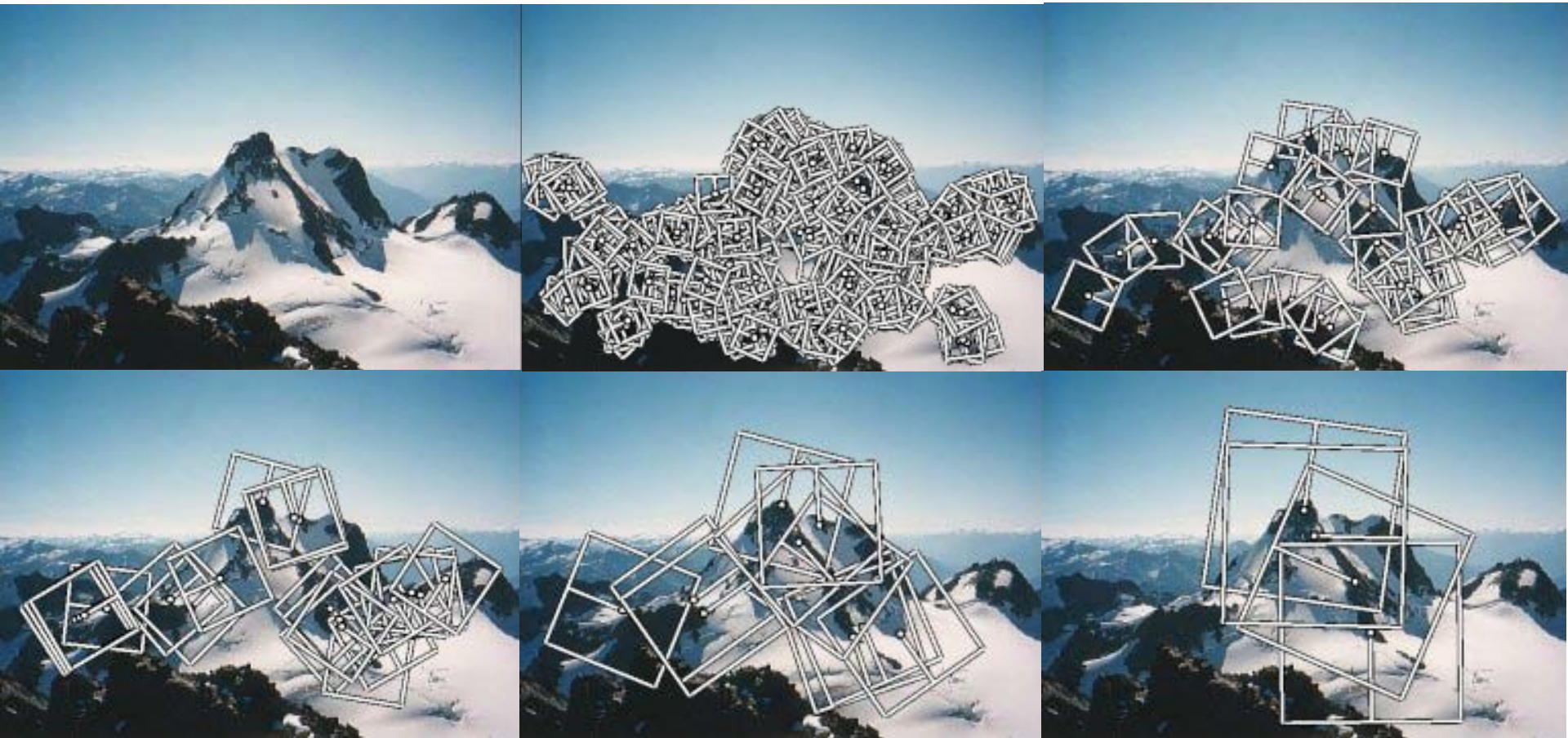
---

developed by David Lowe  
University of British Columbia  
Initial paper ICCV 1999  
Newer journal paper IJCV 2004

# Review: Matt Brown's Canonical Frames



# Multi-Scale Oriented Patches



- Extract oriented patches at multiple scales

# Application: Image Stitching



# Ideas from Matt's Multi-Scale Oriented Patches

- 1. Detect an interesting patch with an interest operator. Patches are translation invariant.
- 2. Determine its dominant orientation.
- 3. Rotate the patch so that the dominant orientation points upward. This makes the patches rotation invariant.
- 4. Do this at multiple scales, converting them all to one scale through sampling.
- 5. Convert to illumination “invariant” form

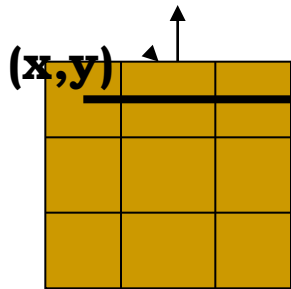
---

## Implementation Concern:

### How do you rotate a patch?

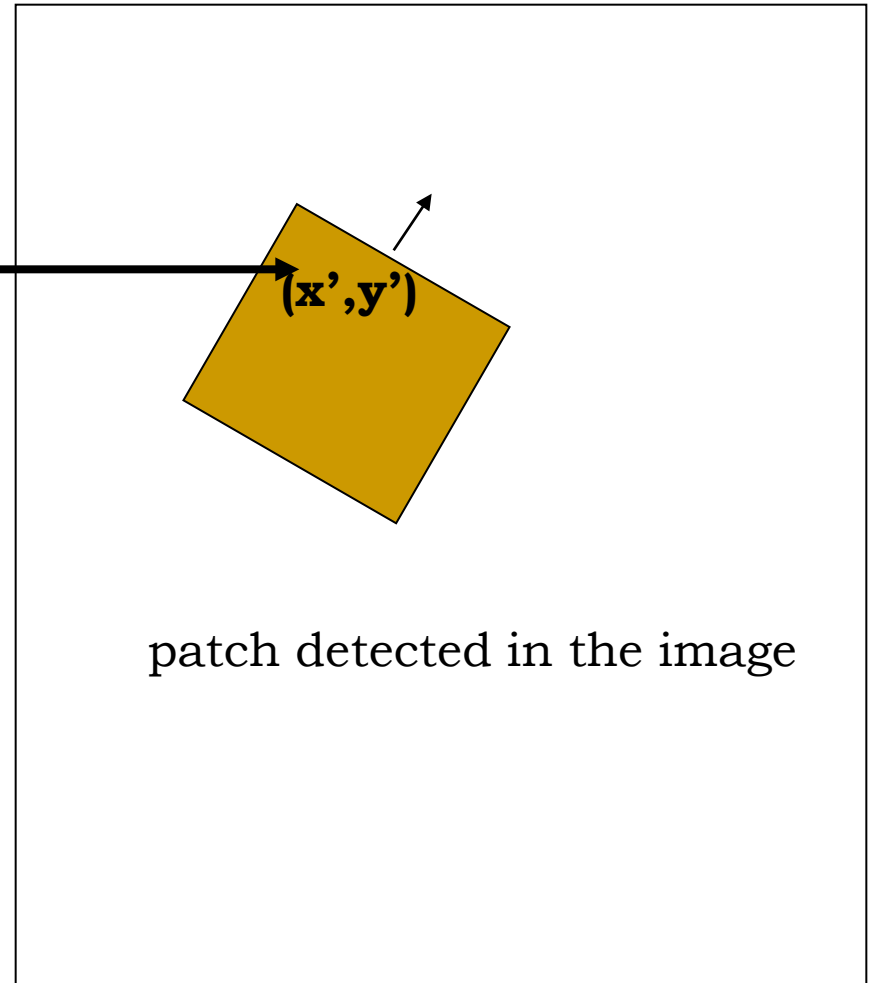
- Start with an “empty” patch whose dominant direction is “up”.
- For each pixel in your patch, compute the position in the detected image patch. It will be in floating point and will fall between the image pixels.
- Interpolate the values of the 4 closest pixels in the image, to get a value for the pixel in your patch.

# Rotating a Patch



empty canonical patch

T



patch detected in the image

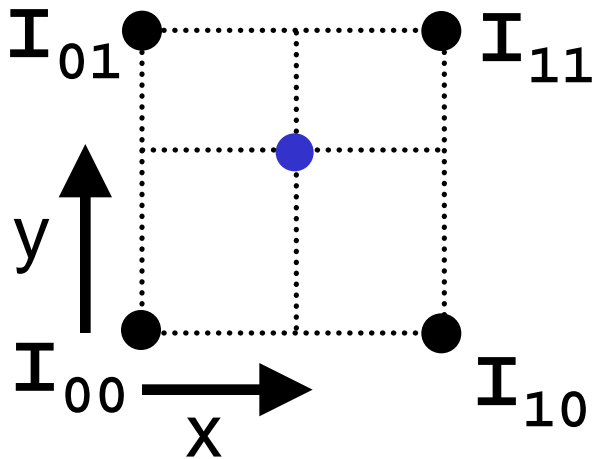
T

$$\begin{cases} x' = x \cos\theta - y \sin\theta \\ y' = x \sin\theta + y \cos\theta \end{cases}$$

counterclockwise rotation

# Using Bilinear Interpolation

- Use all 4 adjacent samples





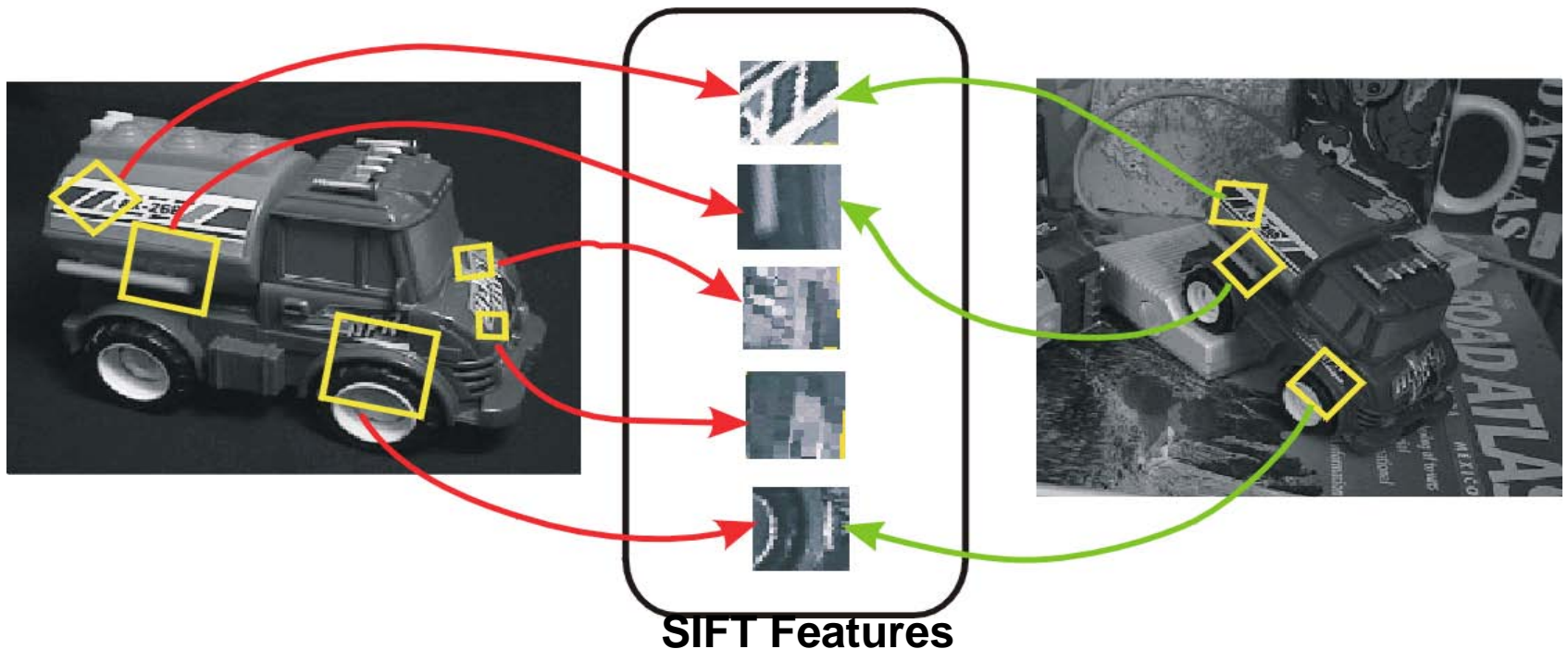
# SIFT: Motivation

- The Harris operator is not invariant to scale and correlation is not invariant to rotation<sup>1</sup>.
- For better image matching, Lowe's goal was to develop an interest operator that is invariant to scale and rotation.
- Also, Lowe aimed to create a **descriptor** that was robust to the variations corresponding to typical viewing conditions. **The descriptor is the most-used part of SIFT.**

<sup>1</sup>But Schmid and Mohr developed a rotation invariant descriptor for it in 1997.

# Idea of SIFT

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



---

# Claimed Advantages of SIFT

- **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
- **Extensibility:** can easily be extended to wide range of differing feature types, with each adding robustness

---

# Overall Procedure at a High Level

## 1. Scale-space extrema detection

Search over multiple scales and image locations.

## 2. Keypoint localization

Fit a model to determine location and scale.

Select keypoints based on a measure of stability.

## 3. Orientation assignment

Compute best orientation(s) for each keypoint region.

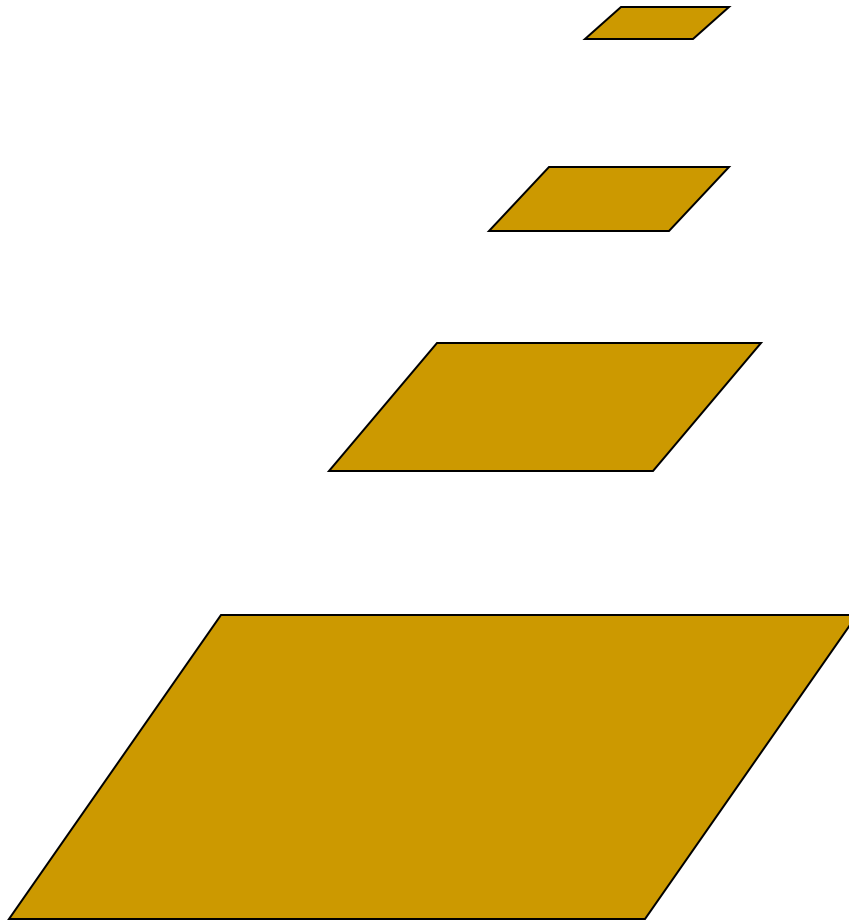
## 4. Keypoint description

Use local image gradients at selected scale and rotation to describe each keypoint region.

# 1. Scale-space extrema detection

- **Goal:** Identify locations and scales that can be repeatably assigned under different views of the same scene or object.
- **Method:** search for stable features across multiple scales using a continuous function of scale.
- **Prior work** has shown that under a variety of assumptions, the best function is a **Gaussian function**.
- **The scale space of an image is a function  $L(x,y,\sigma)$**  that is produced from the convolution of a Gaussian kernel (at different scales) with the input image.

# Aside: Image Pyramids



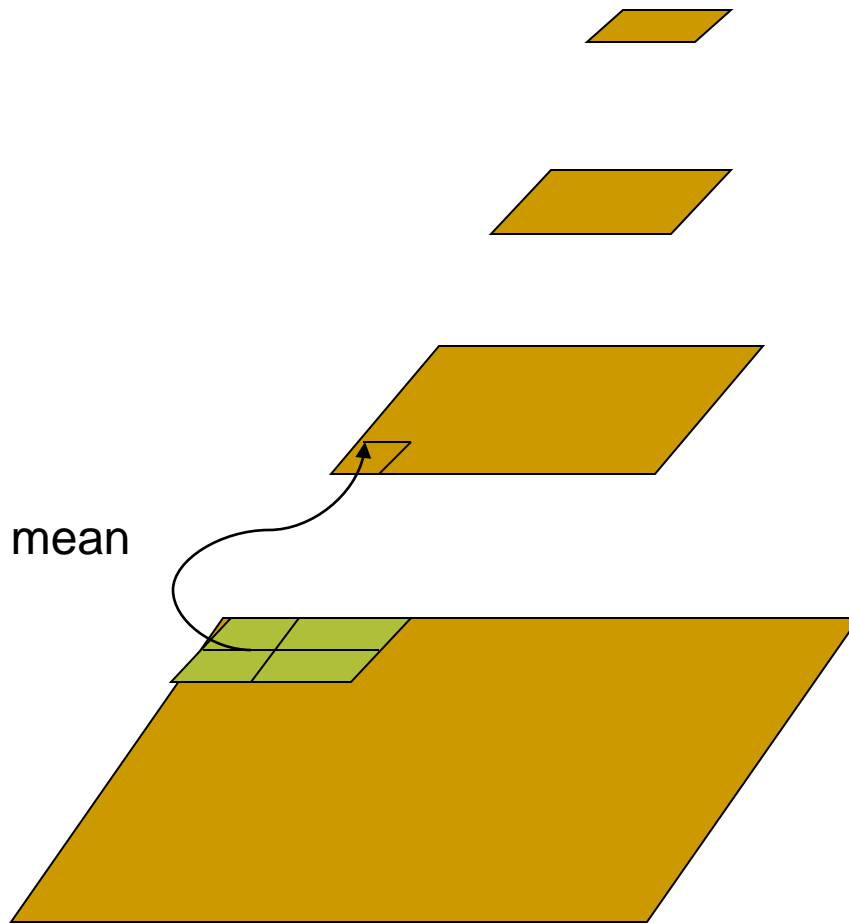
And so on.

3<sup>rd</sup> level is derived from the 2<sup>nd</sup> level according to the same function

2<sup>nd</sup> level is derived from the original image according to some function

Bottom level is the original image.

# Aside: Mean Pyramid



And so on.

At 3<sup>rd</sup> level, each pixel is the mean of 4 pixels in the 2<sup>nd</sup> level.

At 2<sup>nd</sup> level, each pixel is the mean of 4 pixels in the original image.

Bottom level is the original image.

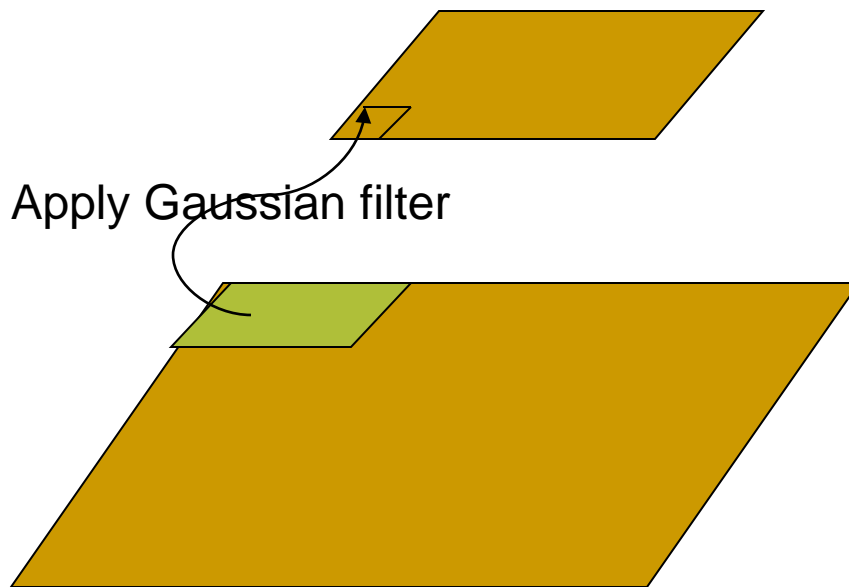
# Aside: Gaussian Pyramid

At each level, image is smoothed and reduced in size. 



And so on.

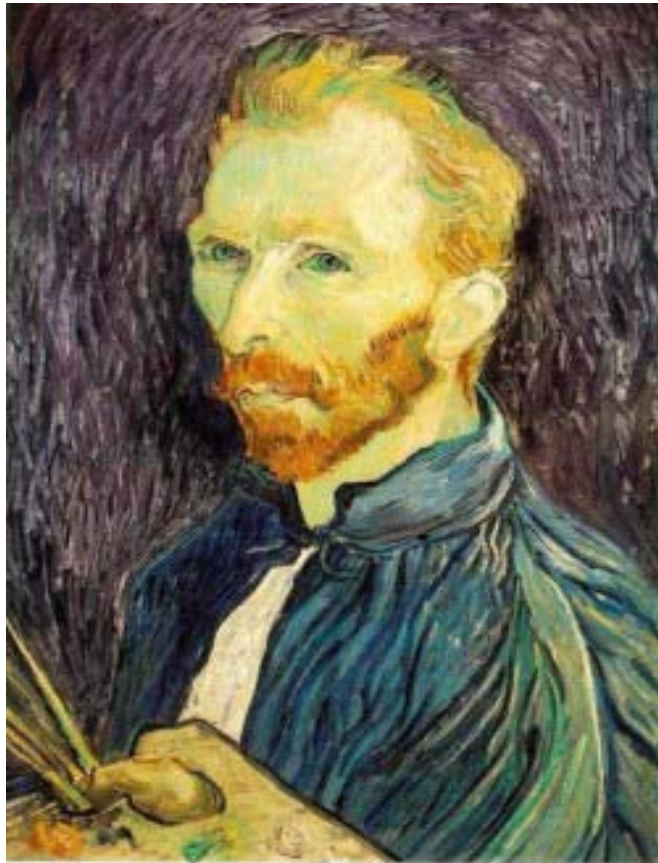
At 2<sup>nd</sup> level, each pixel is the result of applying a Gaussian mask to the first level and then subsampling to reduce the size.



Bottom level is the original image.



# Example: Subsampling with Gaussian pre-filtering



Gaussian 1/2



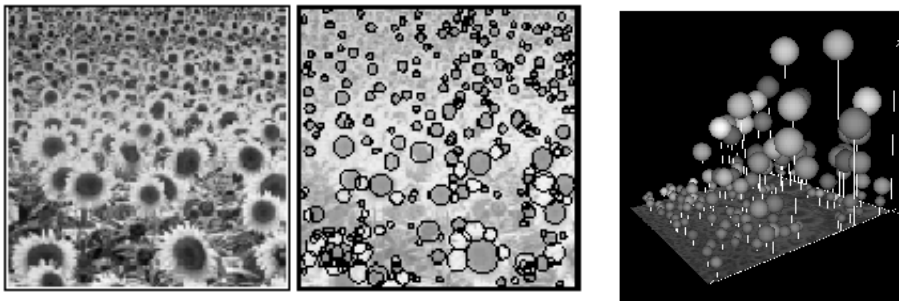
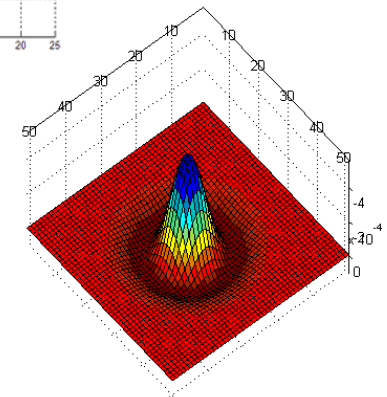
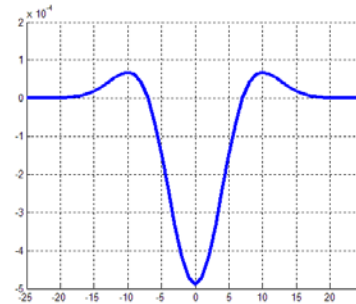
G 1/4



G 1/8

# Lowe's Scale-space Interest Points

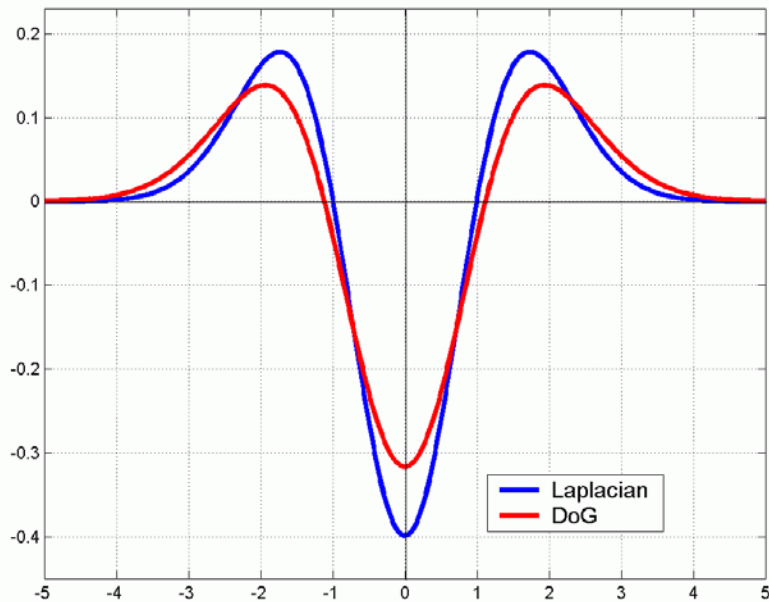
- **Laplacian of Gaussian kernel**
  - Scale normalised (x by  $\text{scale}^2$ )
  - Proposed by Lindeberg
- **Scale-space detection**
  - Find local maxima across scale/space
  - A good “blob” detector



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

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

# Lowe's Scale-space Interest Points: Difference of Gaussians



- Gaussian is an ad hoc solution of heat diffusion equation

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G.$$

- Hence

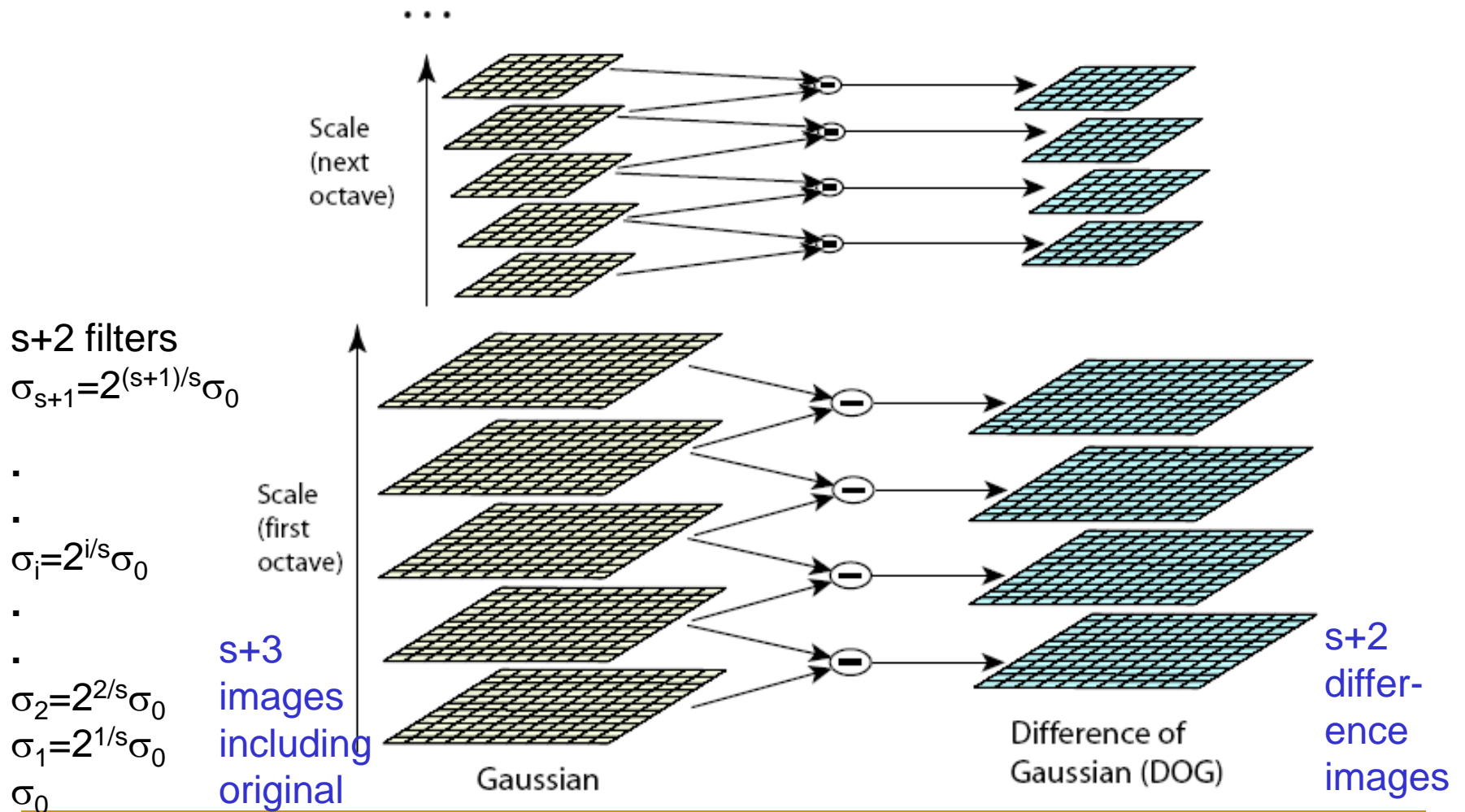
$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G.$$

- $k$  is not necessarily very small in practice

# Lowe's Pyramid Scheme

- Scale space is separated into **octaves**:
  - Octave 1 uses scale  $\sigma$
  - Octave 2 uses scale  $2\sigma$
  - etc.
- In each octave, the initial image is repeatedly convolved with Gaussians to produce a set of scale space images.
- Adjacent Gaussians are subtracted to produce the DOG
- After each octave, the Gaussian image is down-sampled by a factor of 2 to produce an image  $\frac{1}{4}$  the size to start the next level.

# Lowe's Pyramid Scheme

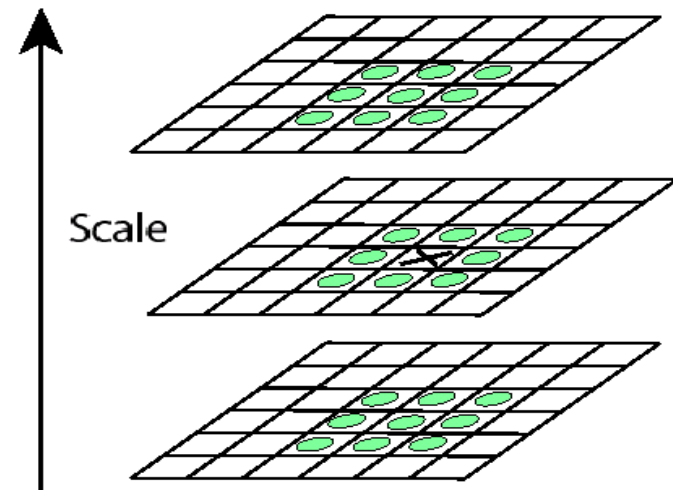


The parameter **s** determines the number of images per octave.

# Key point localization

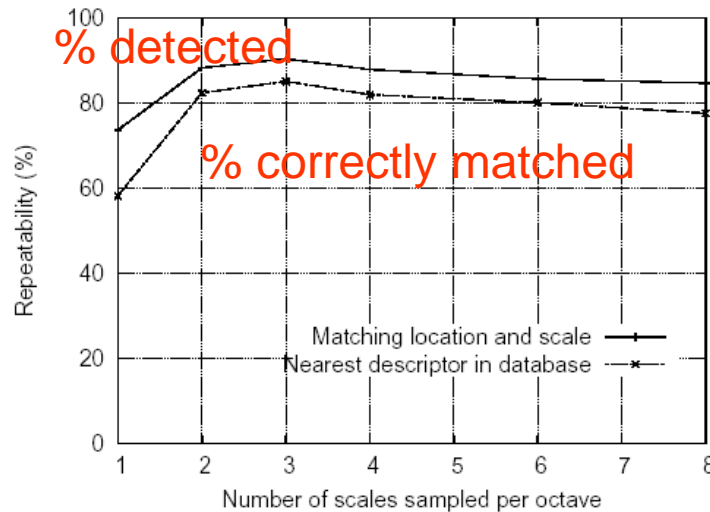
s+2 difference images.  
top and bottom ignored.  
s planes searched.

- Detect maxima and minima of difference-of-Gaussian in scale space
- Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below

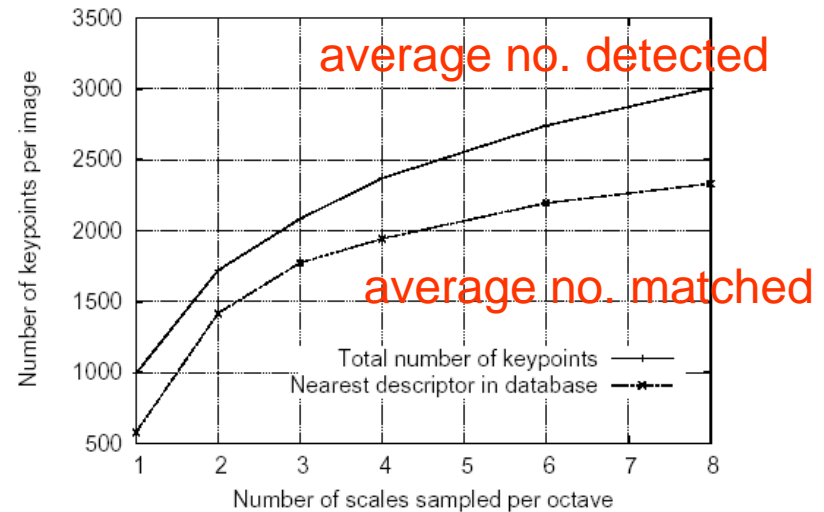


For each max or min found, output is the **location** and the **scale**.

# Scale-space extrema detection: experimental results over 32 images that were synthetically transformed and noise added.



Stability



Expense

- Sampling in scale for efficiency
  - How many scales should be used per octave?  $S=?$ 
    - More scales evaluated, more keypoints found
    - $S < 3$ , stable keypoints increased too
    - $S > 3$ , stable keypoints decreased
    - $S = 3$ , maximum stable keypoints found

# Keypoint localization

- Once a keypoint candidate is found, perform a detailed fit to nearby data to determine
  - location, scale, and ratio of principal curvatures
- In initial work keypoints were found at location and scale of a central sample point.
- In newer work, they fit a 3D quadratic function to improve interpolation accuracy.
- The Hessian matrix was used to eliminate edge responses.



# Eliminating the Edge Response

- Reject flats:

- $|D(\hat{\mathbf{x}})| < 0.03$

- Reject edges:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Let  $\alpha$  be the eigenvalue with larger magnitude and  $\beta$  the smaller.

$$\text{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

Let  $r = \alpha/\beta$ .  
So  $\alpha = r\beta$

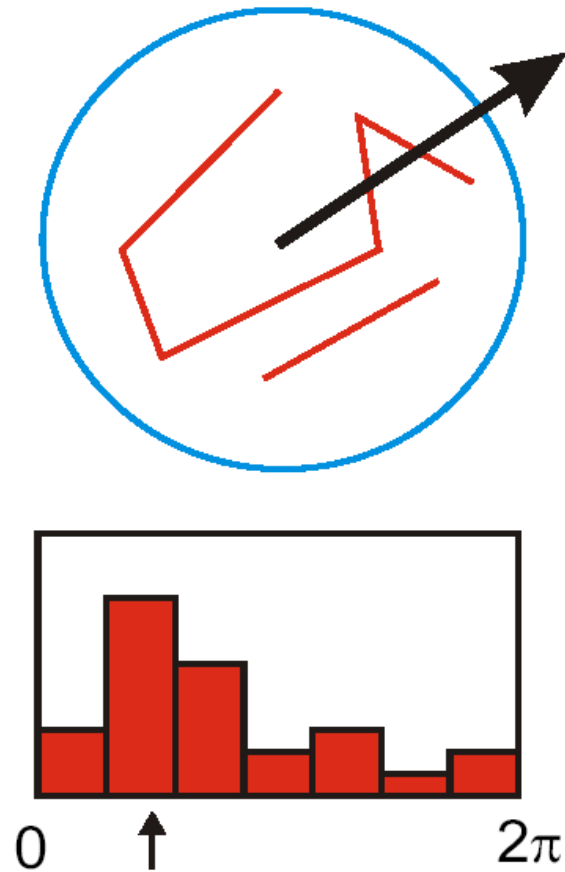
$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r},$$

$(r+1)^2/r$  is at a min when the 2 eigenvalues are equal.

- $r < 10$

- What does this look like?

# 3. Orientation assignment



- Create histogram of local gradient directions at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)

If 2 major orientations, use both.

# Keypoint localization with orientation

233x189



832

initial keypoints

729

keypoints after  
gradient threshold



536

keypoints after  
ratio threshold

---

# 4. Keypoint Descriptors

- At this point, each keypoint has
  - location
  - scale
  - orientation
- Next is to compute a descriptor for the local image region about each keypoint that is
  - highly distinctive
  - invariant as possible to variations such as changes in viewpoint and illumination

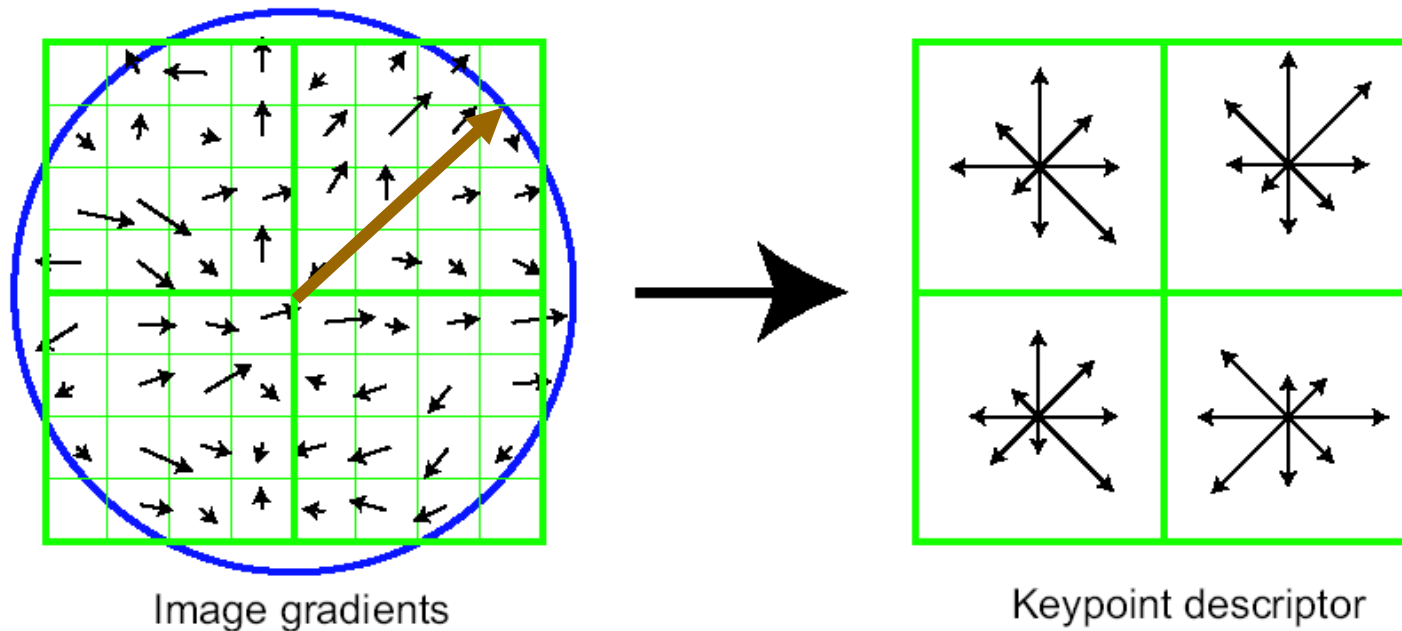
---

# Normalization

- Rotate the window to standard orientation
- Scale the window size based on the scale at which the point was found.

# Lowe's Keypoint Descriptor

(shown with 2 X 2 descriptors over 8 X 8)



In experiments, 4x4 arrays of 8 bin histogram is used, a total of 128 features for one keypoint

# Lowe's Keypoint Descriptor

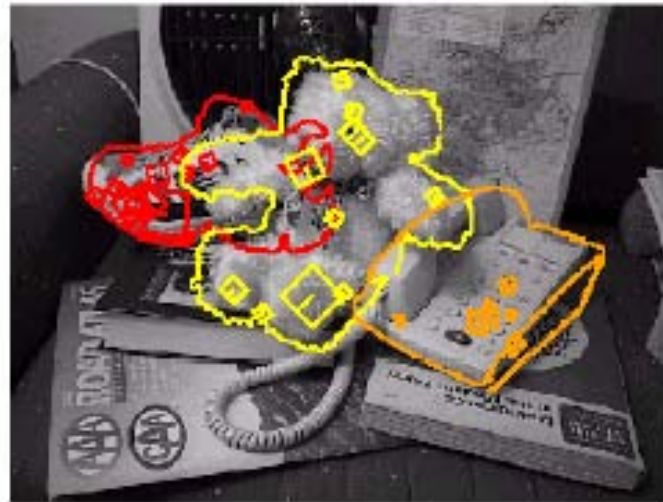
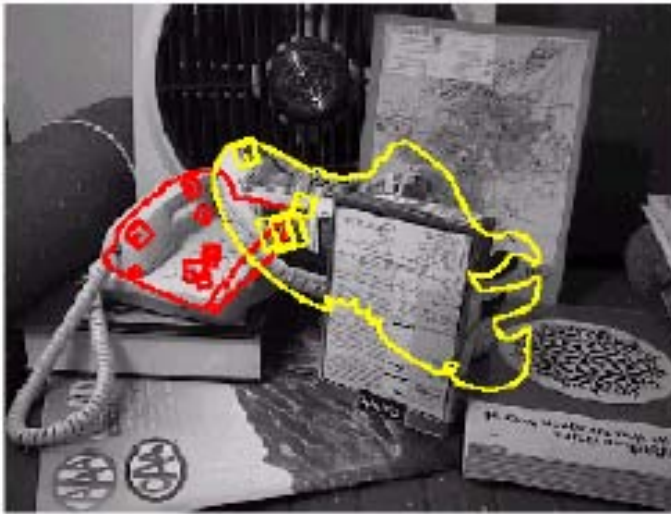
- use the **normalized** region about the keypoint
- compute gradient magnitude and orientation at each point in the region
- **weight them by a Gaussian** window overlaid on the circle
- create an **orientation histogram** over the 4 X 4 subregions of the window
- 4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 times 8 directions gives a **vector of 128 values**.



# Using SIFT for Matching “Objects”







---

# Uses for SIFT

- Feature points are used also for:
  - Image alignment (homography, fundamental matrix)
  - 3D reconstruction (e.g. Photo Tourism)
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - ... many others