# SIFT FEATURES

Guido Gerig

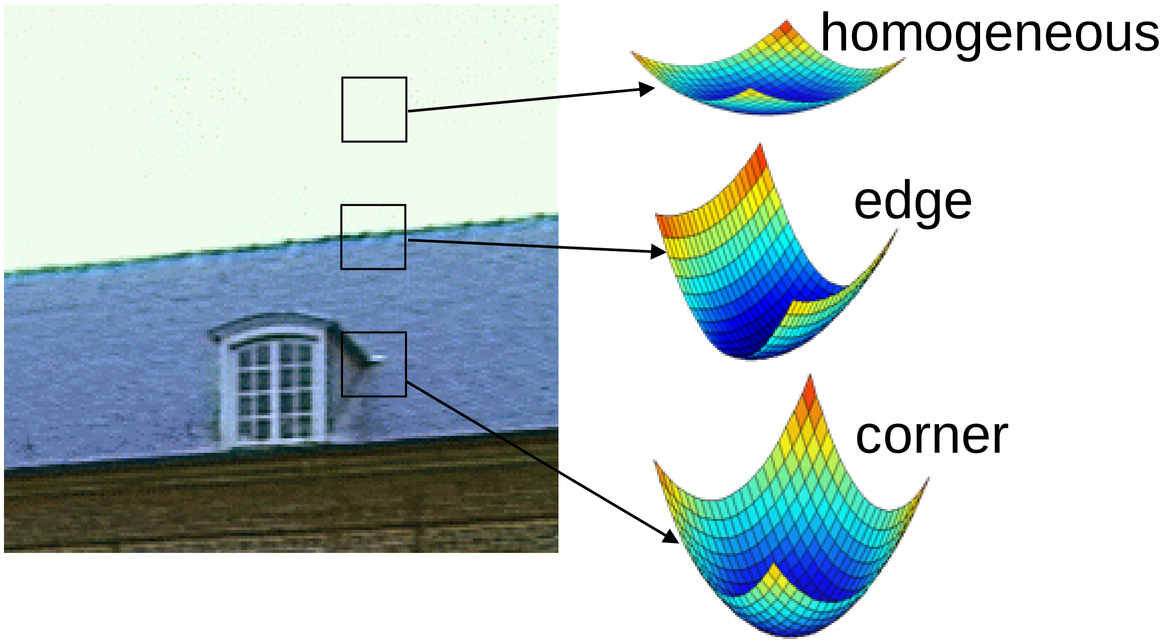
Additioal Materials Chapter 3/4

Credit: slides modified from J.M. Frahm/ M. Pollefeys UNC Chapel Hill)

# Feature Tracking

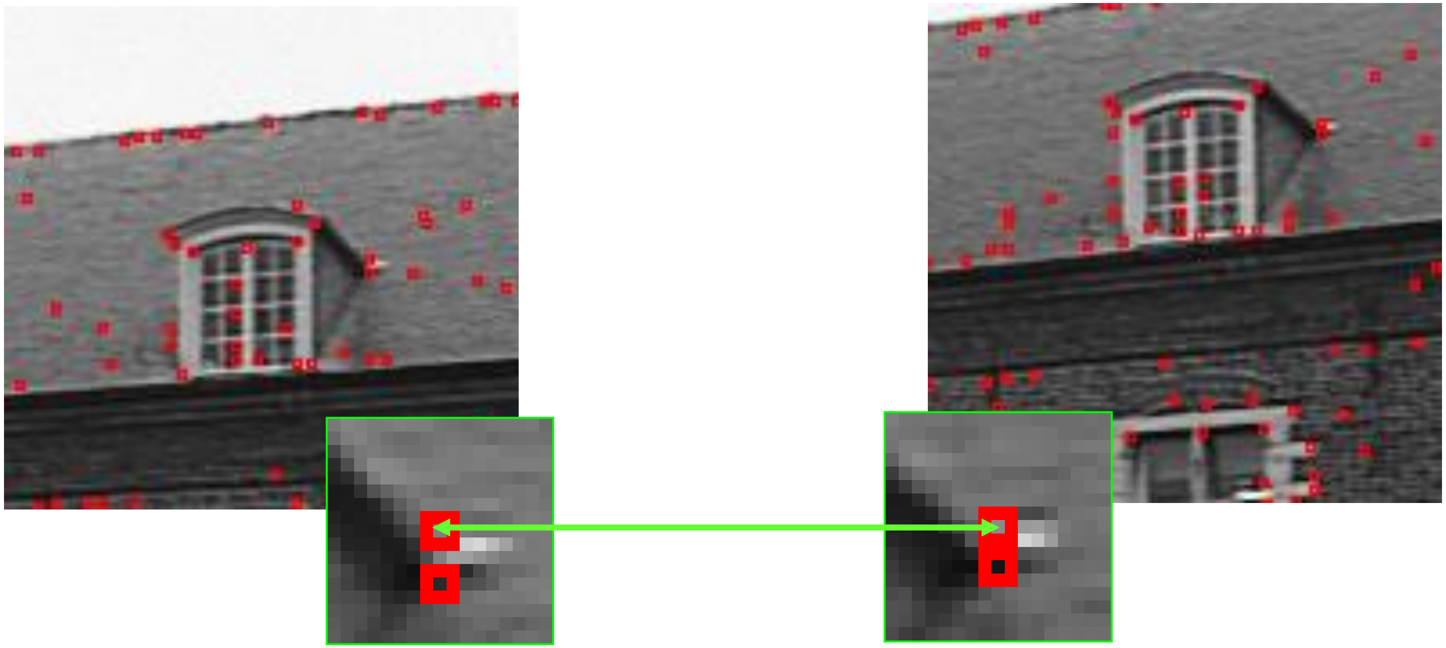
- Tracking of “good” features & efficient search for subsequent positions.
- What are good features?
- Required properties:
  - Well-defined
    - (i.e. neigboring points should all be different)
  - Stable across views
    - (i.e. same 3D point should be extracted as feature for neighboring viewpoints)

# Feature point extraction

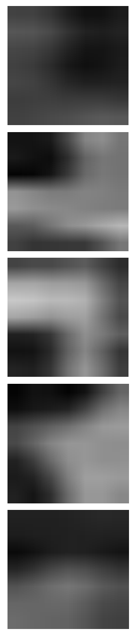


# Simple matching

- for each corner in image 1 find the corner in image 2 that is most similar (using SSD or NCC) and vice-versa
- Only compare geometrically compatible points
- Keep mutual best matches



# Feature matching: example

| | | | | |
|---|---|---|---|---|
| 0.96 | **-0.40** | **-0.16** | **-0.39** | **0.19** |
| **-0.05** | 0.75 | **-0.47** | **0.51** | **0.72** |
| **-0.18** | **-0.39** | 0.73 | **0.15** | **-0.75** |
| **-0.27** | **0.49** | **0.16** | 0.79 | **0.21** |
| **0.08** | **0.50** | **-0.45** | **0.28** | 0.99 |


What transformations does this work for?

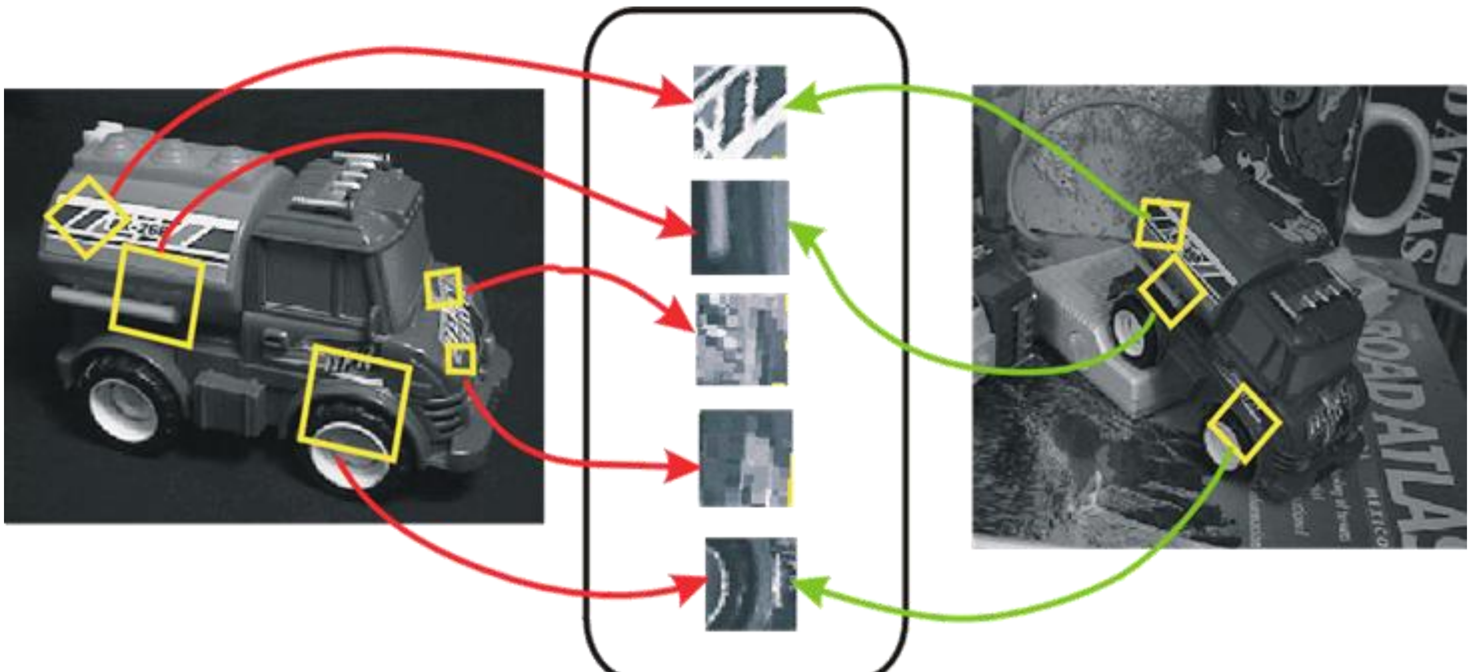
What level of transformation do we need?

# Lowe’s SIFT features

(Lowe, ICCV99)

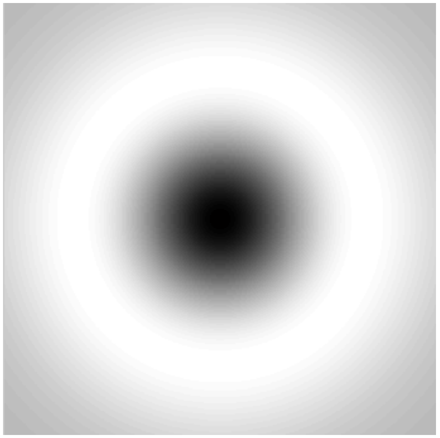
SIFT: Scale Invariant Feature Transform

Recover features with change of position, orientation and scale



# Position

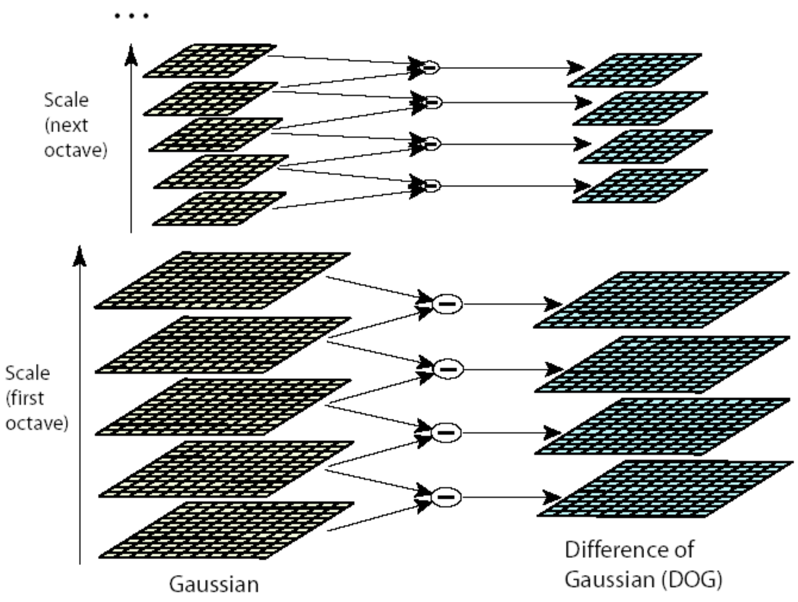
- Look for strong responses of DOG filter (Difference-Of-Gaussian)
- Only consider local maxima

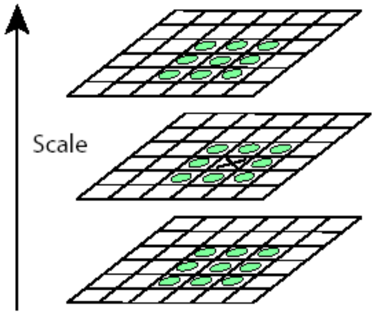


$$\mathrm{DOG}(x,y) = \frac{1}{k} e^{-\frac{x^2+y^2}{(k\sigma)^2}} - e^{-\frac{x^2+y^2}{\sigma^2}} \qquad k = \sqrt{2}$$

# Scale

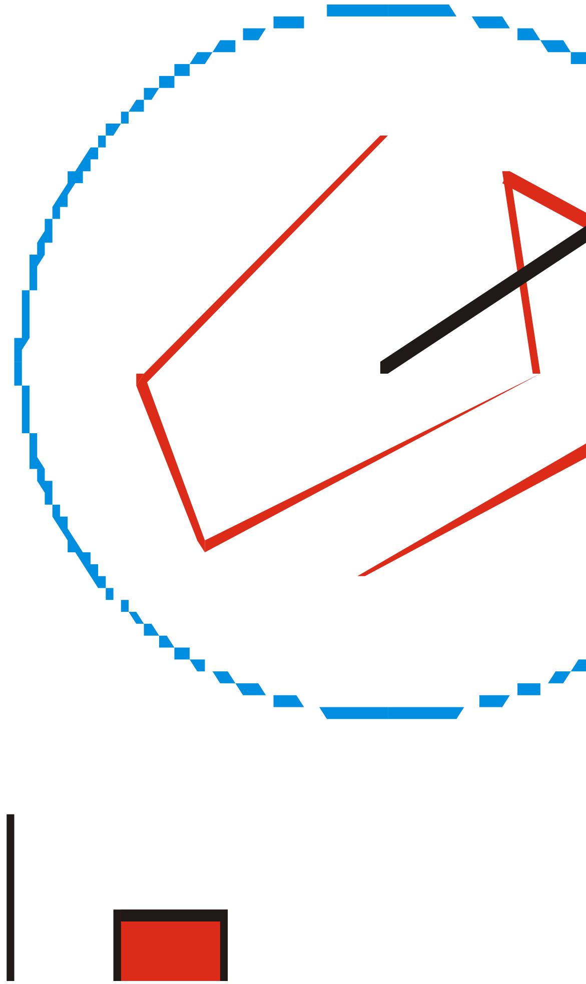
- Look for strong responses of DOG filter (Difference-Of-Gaussian) over scale space
- Only consider local maxima in both position and scale
- Fit quadratic around maxima for subpixel accuracy



# Orientation

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



# SIFT descriptor

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

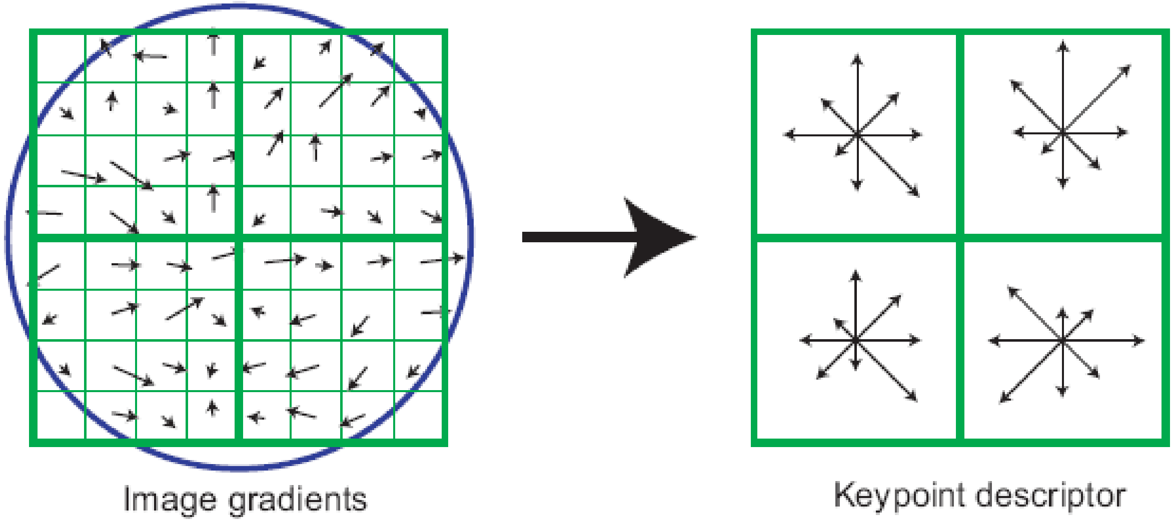
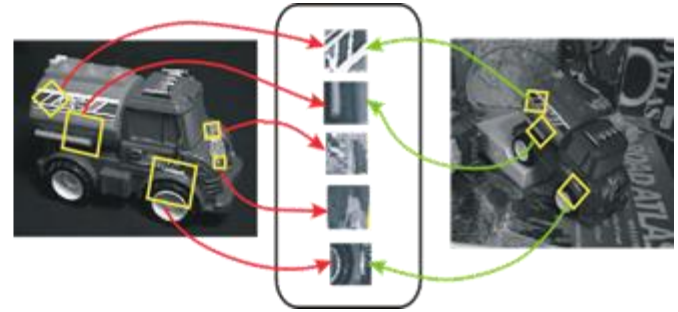
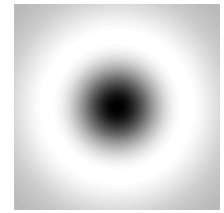


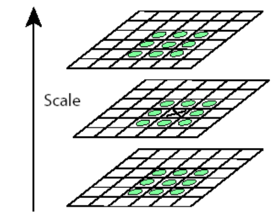

Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

# SIFT features

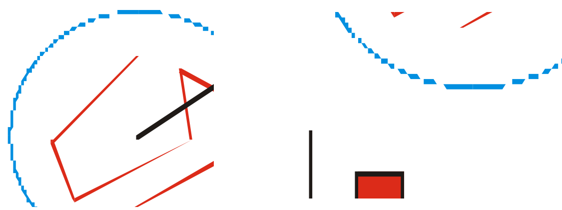


- Scale-space DoG maxima






- Verify minimum contrast and “cornerness”
- Orientation from dominant gradient



- Descriptor based on gradient distributions

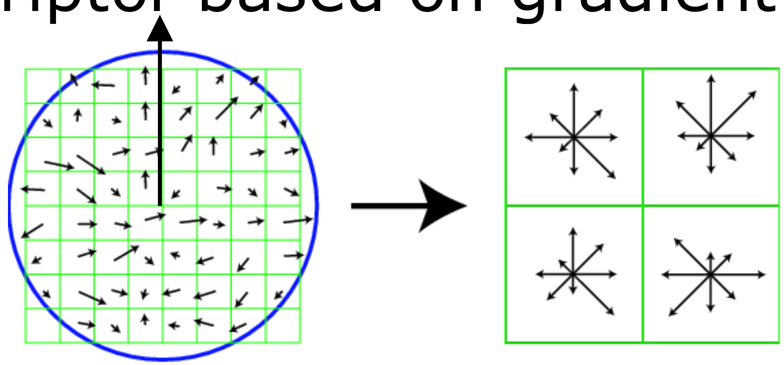


Image gradients

Keypoint descriptor

# Minimum contrast and "cornerness"

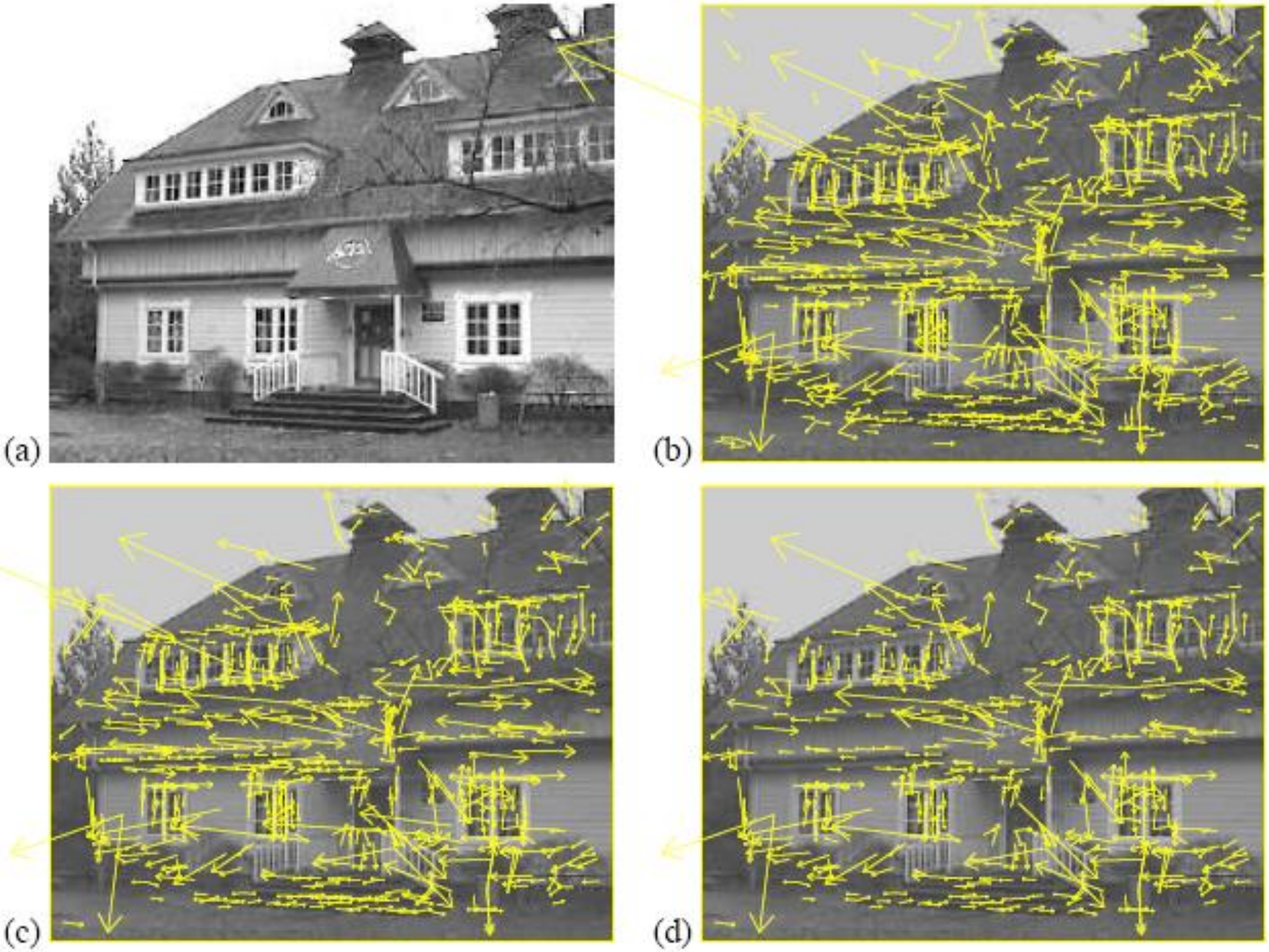


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

**Example 1**

*Input images* (zip 1.1Mb)

*Output panorama 1*

*Output panorama 2* · Image | Panorama

# Free code available

SIFT & SIFT ++

SIFT implementation by Andrea Vedaldi
http://vision.ucla.edu/~vedaldi/code/sift/sift.html
http://www.xmarks.com/site/vision.ucla.edu/~vedaldi/code/sift/sift.html

Ning Xu: http://xuning-cn.blogspot.com/2007/11/sift-implementation_30.html
(also SIFT, SURF, Keypoints etc.)

David Lowe: Keypoints: http://www.cs.ubc.ca/~lowe/keypoints/