
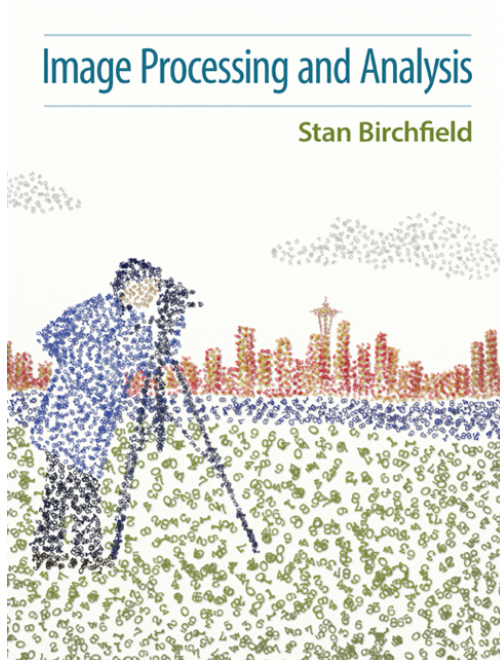


ELE510 Image processing and computer vision

Feature point detection and description, (SIFT) (chap 7.4 (7.3) Birchfield) 2020





Parts in course presentations is material from Cengage learning. It can be used for teaching, and it can be share it with students on access controlled web-sites (Canvas) for use in THIS course. **But it should not be copied or distributed further in any way** (by you or anybody).

* Harris Stephens corner / feature detector

key points

Other feature detectors/ descriptors

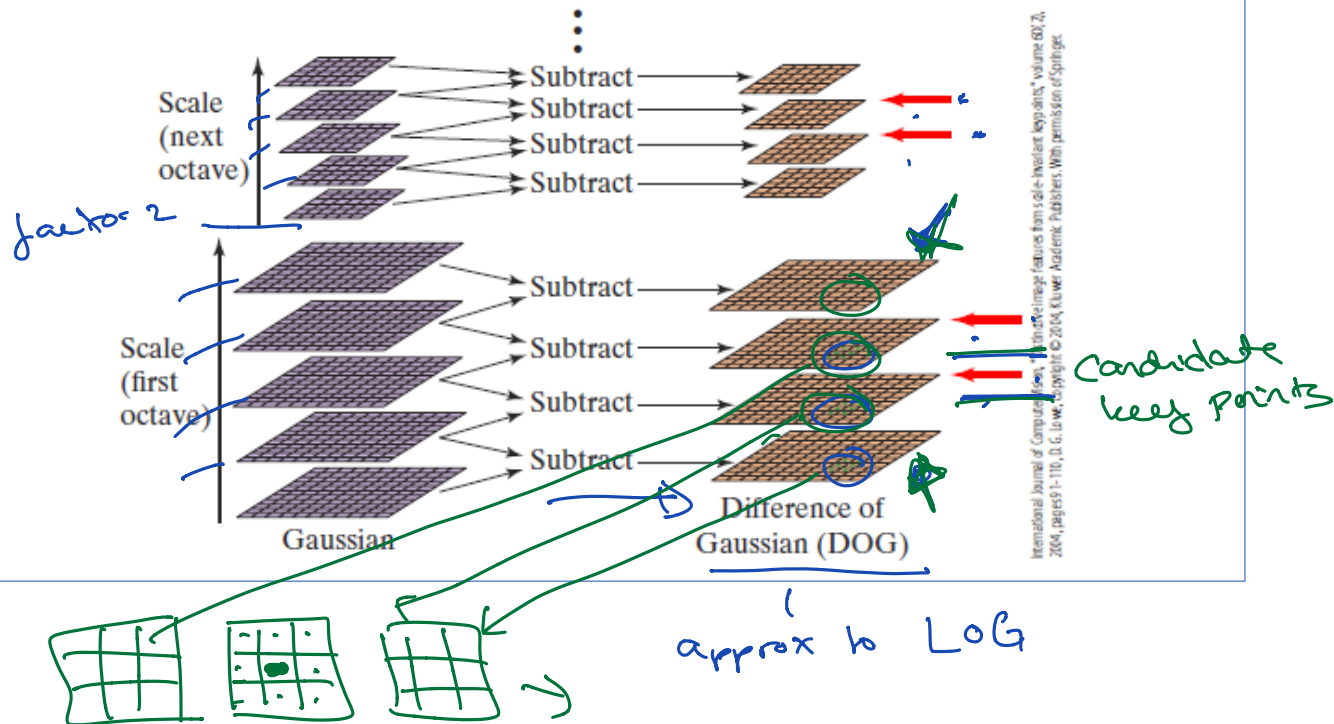
- Some examples are

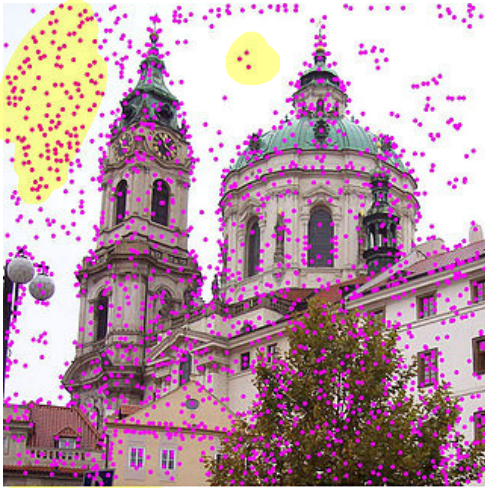
- ▪ SIFT, Scale Invariant Feature Transform
- ▪ FAST, Features from Accelerated Segment Test
- ▪ SURF, Speeded-Up Robust Features

SIFT feature detector

- SIFT- **Scale Invariant** Feature Transform was published by David Lowe in 1999
1. The first step is to construct a scale space
 2. Thereafter approximate a Laplacian pyramid of the image using DoG using the scale space images.
 3. Determine, for every pixel and for every scale, whether the pixel is a local maximum among its 26 neighbors. (8 in same scale, 9 in neighbor larger scale, 9 from neighbor smaller scale)
candidate key points
 4. Discard **bad key-points**, i.e. untextured areas or along intensity edges. A technique similar to **Harris corner detection** is used here. The **Hessian** is found from the DoG images already calculated
 - Discard keypoints with low DoG value (**low contrast**)
 - Discard keypoints on edges based on finding **"edgeness"** from the Hessian matrix

Figure 7.17 SIFT features are detected by computing a Laplacian pyramid, then looking for local maxima among the 26 neighbors of a pixel. For each octave in this drawing there are 5 Gaussians, 4 LoGs (approximated by DoGs), and two scales (indicated by red arrows); the remaining two LoGs are used only in the neighborhood computation.





By Lukas Mach at English Wikipedia, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=11510369>

Left: After scale space extrema are detected.

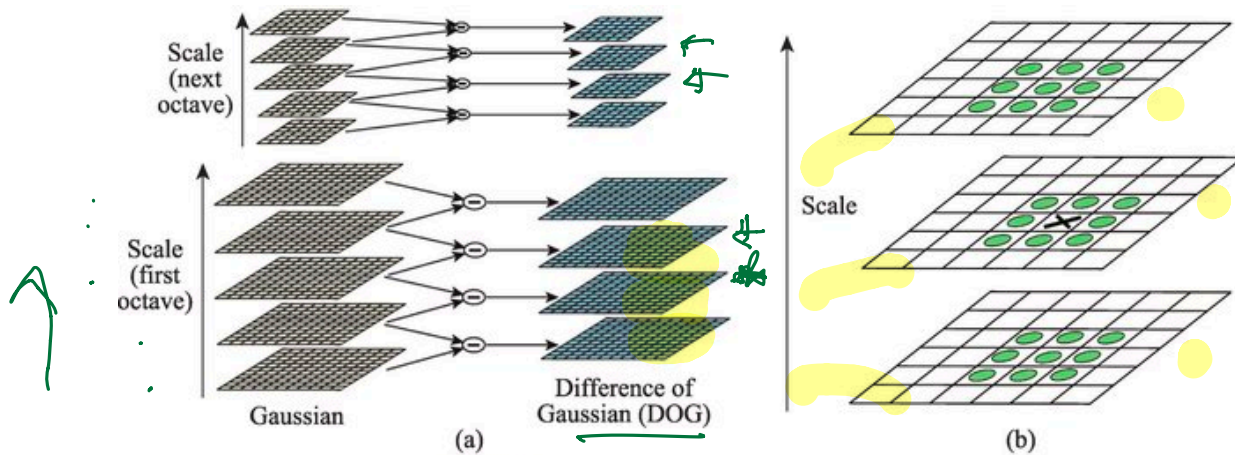
Middle: the SIFT algorithm discards low-contrast keypoint.

Right: Then keypoints located on edges are removed after looking at the edginess using the Hessian.

Resulting set of keypoints is shown on last image.

(7.5) feature descriptor (SIFT)

- Around a feature point (found by the SIFT feature detector), in the same scale: Find the image gradient magnitudes and orientation in the neighborhood. Scale decides size of neighborhood and gradient smoothing.
- Find dominant gradient orientation, make all orientations relative to that. (to make it **rotational invariant**)
- Weight by a Gaussian to give more emphasize to the orientation close to the feature point (center)
- Make into 3D histogram over position and orientation
- Values in histogram are concatenated to form a vector that describes the feature point.



good
key points

8x8

64

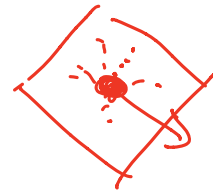
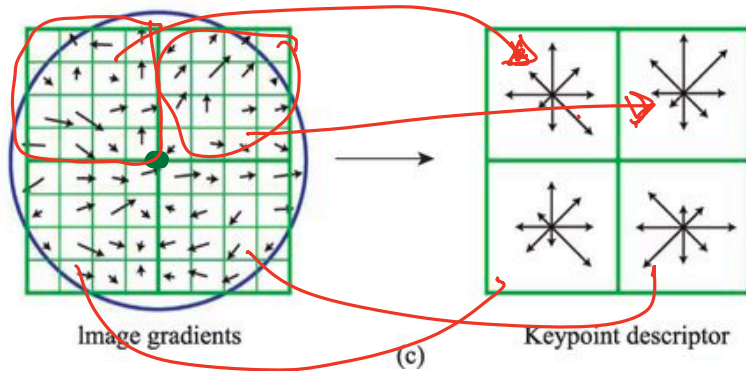
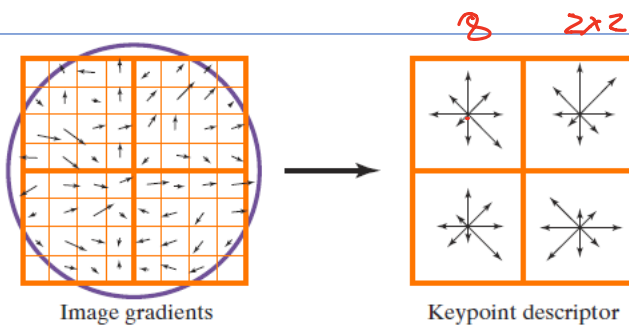


Figure 7.18 The SIFT feature descriptor is computed by accumulating the orientations of the gradient vectors in a neighborhood of the feature point into a 3D array over position and orientation.



$$4 \times 4 \times 8 = 128 \approx \text{normalised}$$

Figure 7.19 SIFT feature matching results. SIFT feature descriptors from the query image (middle) are matched against descriptors from the database (left) to detect objects at various poses and lighting conditions, and even with severe occlusion (right).



International Journal of Computer Vision, "Distinctive image features from scale-invariant keypoints", Volume 60(2), 2004, pages 91-110. D. G. Lowe, Copyright © 2004, Kluwer Academic Publishers. With permission of Springer.

Other feature descriptors

- **GLOH** – Gradient location and orientation Histogram
 - Extension of SIFT, slightly better than SIFT in some situations
- **HOG** – Histogram of oriented Gradients, usually meant as calculating the descriptor on **all points in the image**, not just **corner/feature** points.
- **SURF** – speeded up robust features
 - Much faster than SIFT, using square filters instead of gaussian etc. Both feature detector and descriptor..