Matching with Invariant Features

Sheng Li

sheng.li@seu.edu.cn

School of Cyber Science and Engineering
Southeast University

July 7, 2022

Overview

- SIFT
  - Scale-space extrema detection
  - Keypoint localization
  - Orientation assignment
  - Keypoint descriptor
- Image Registration
  - Keypoint matching
  - Computing homograhy
  - Robust fitting with RANSAC

Introduction to SIFT

SIFT, aka scale invariant feature transform is a method for extracting **distinctive** invariant features from images that can be used to perform reliable matching between different views of an object or scene.

- Invariant to **image scale and rotation**
- Robust matching across a a substantial range of **affine distortion, change in 3D viewpoint, addition of noise, and change in illumination**
- Match with **high probability** against a large database of features from many images

Workflow

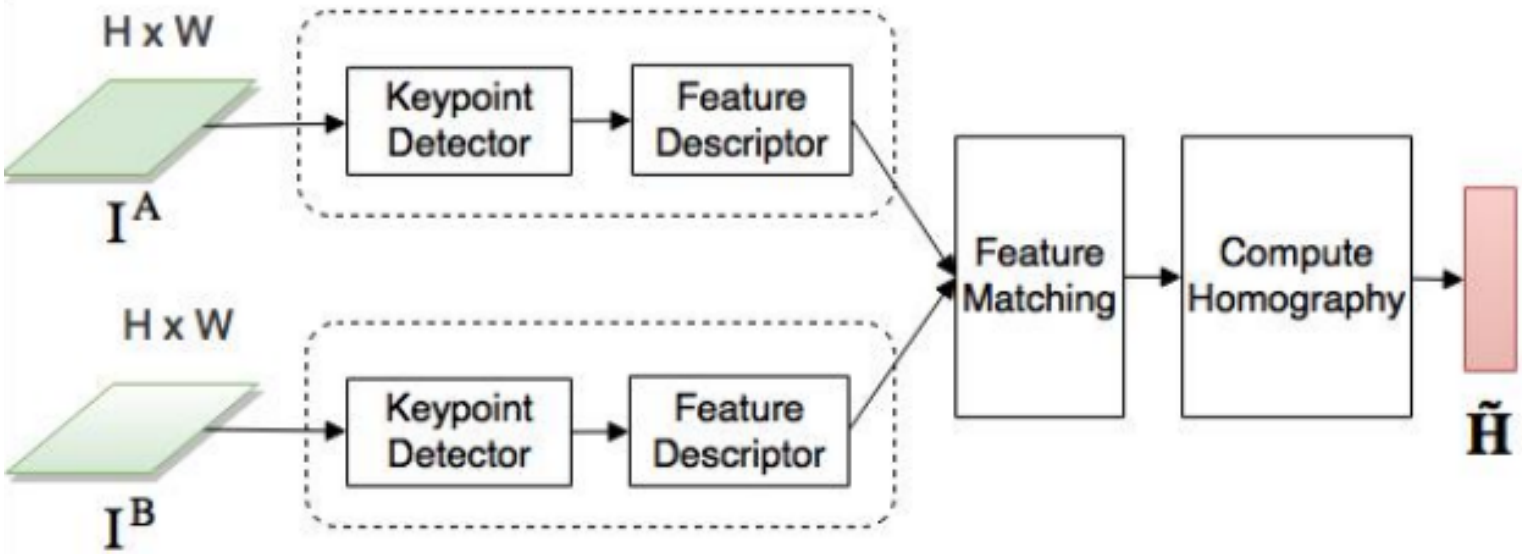

Figure 1: Generic Feature-based Approach

1. Find a set of distinctive keypoints
2. Define a region around each keypoint
3. Compute a local descriptor from the region
4. Match descriptors

Overall picture of keypoint detection

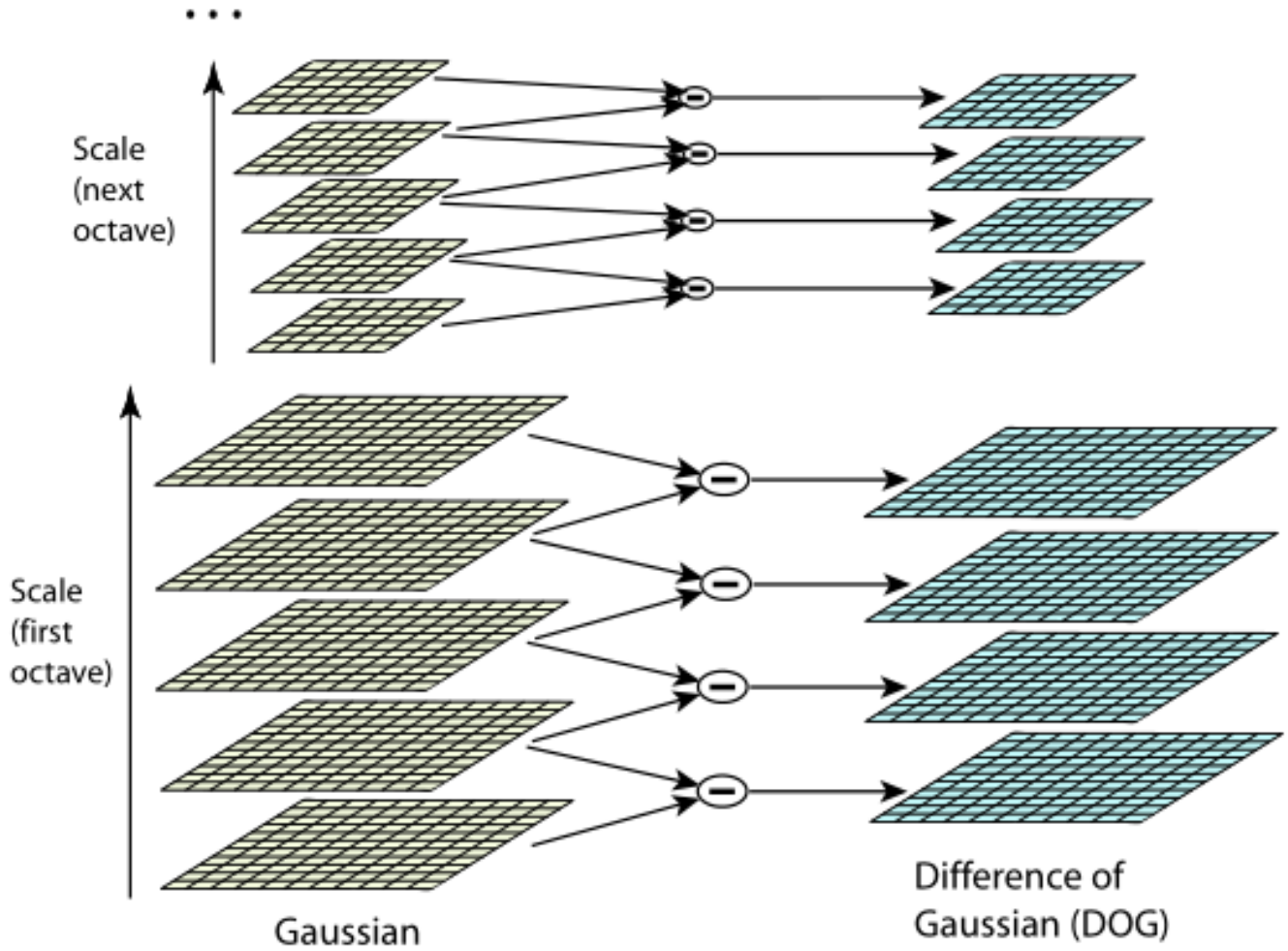

Figure 2: Scale space construction [1]

Gaussian pyramid

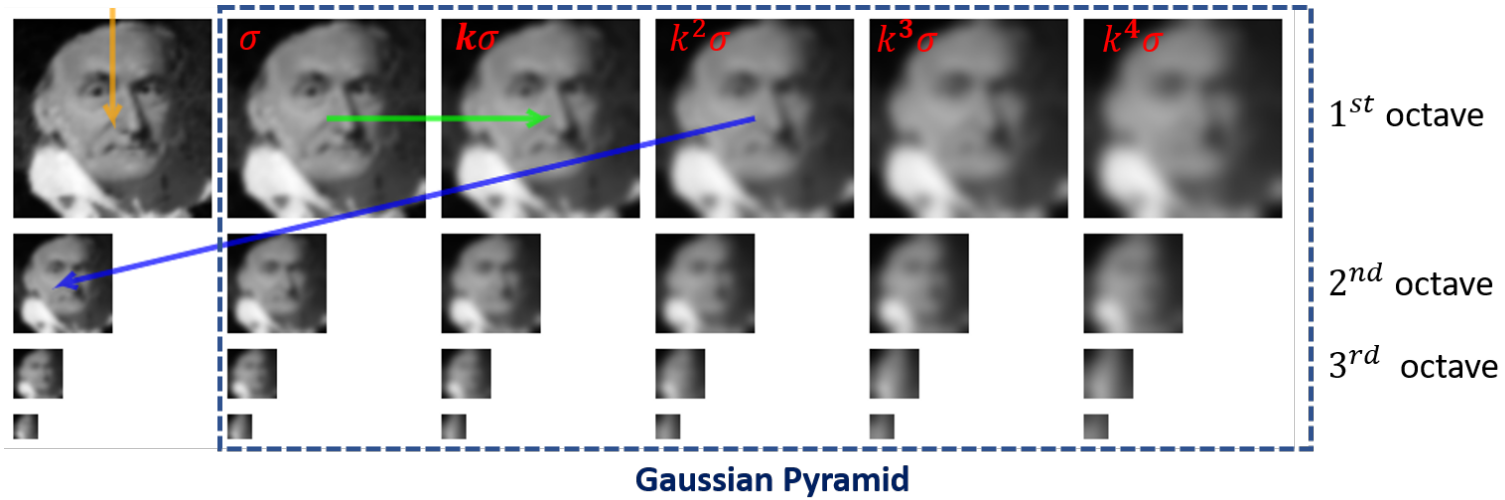

Figure 3: Gaussian pyramid construction

Convolve input image I with Gaussian G of various scale σ, following the green arrow.

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

where $k = \sqrt{2}$ and $l \in 0, 1, 2, 3, 4$

After each octave, the Gaussian image is down-sampled by a factor of 2, following the blue arrow.

Approximation to LoG

The relationship between D and $\nabla^2 G$ can be understood from the heat diffusion equation:

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

$\nabla^2 G$ can be computed from the finite difference approximation to $\frac{\partial G}{\partial \sigma}$, using the difference of nearby scales at $k\sigma$ and σ:

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

and therefore,

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

This shows that when the DoG function has scales differing by a constant factor, it already incorporates the σ^2 scale normalization required for the scale-invariant Laplacian.

Difference of Gaussian

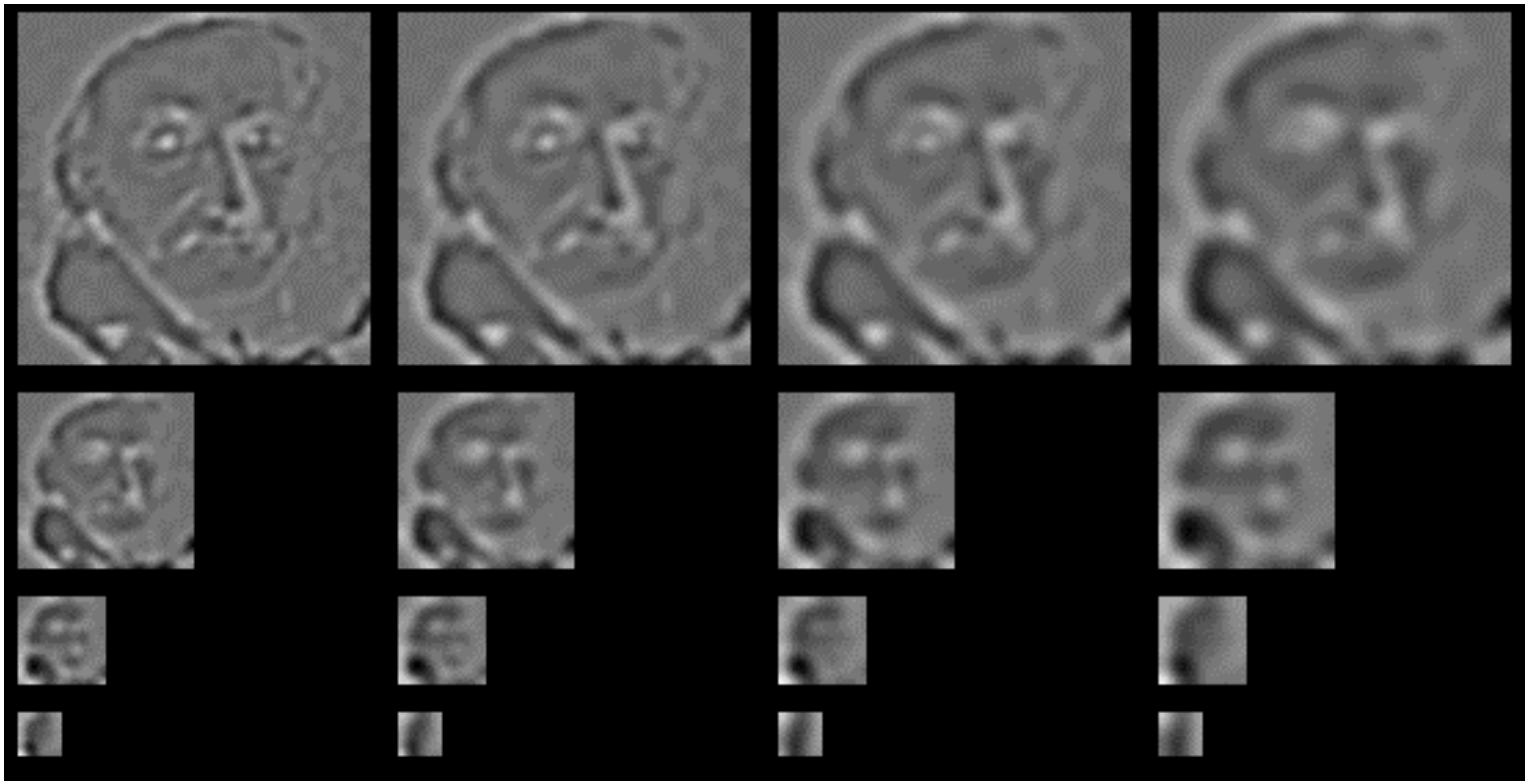

Figure 4: DoG pyramid construction

To detect stable keypoint, convolve image I with difference of Gaussian:

$$D(x, y, k^l\sigma) = L(x, y, k^{l+1}\sigma) - L(x, y, k^l\sigma)$$

where $k = \sqrt{2}$ and $l \in 0, 1, 2, 3$

Local extrema detection

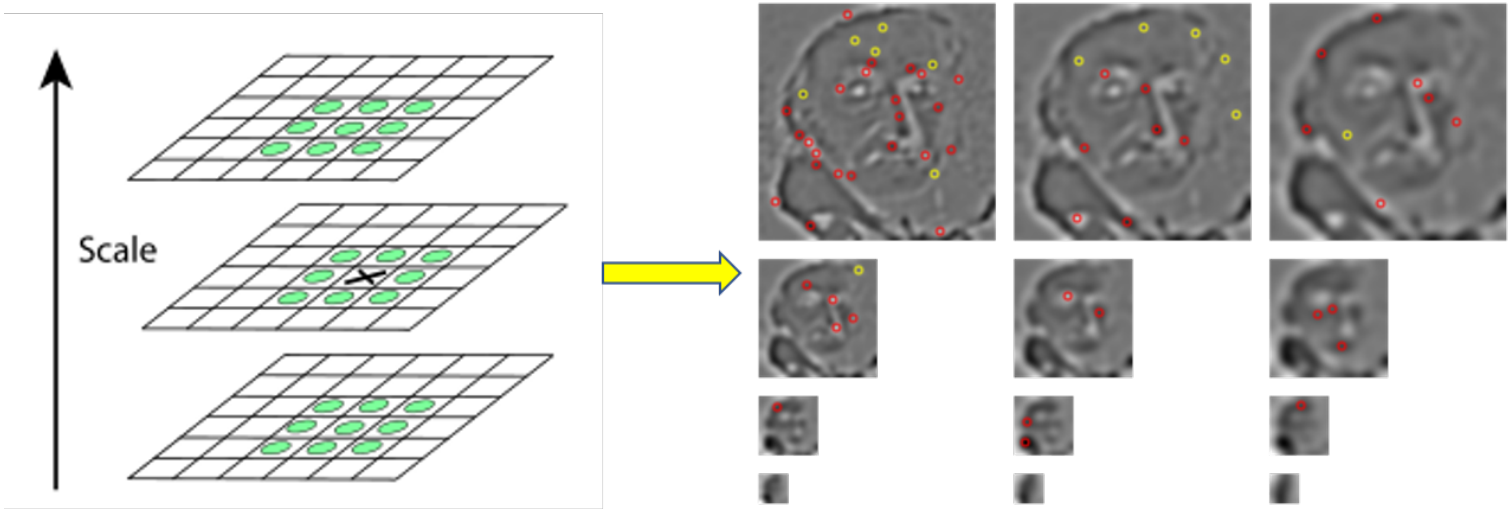

Figure 5: 26 comparisons in 3x3 regions at the current and adjacent scales

Each sample point is compared to its 8 neighbors in the current image and 9 neighbors in the scale above and below.

To **remove weak extrema**: check against $|D(x, y, \sigma)| > 0.03$

Principal curvature ratio

The principal curvatures can be computed from a 2x2 Hessian matrix [2]:

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

Let α be the larger eigenvalue and β be the smaller one($\alpha = r\beta$):

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

Curvature ratio R for the corresponding point in the DoG pyramid:

$$R = \frac{\mathrm{Tr}(H)^2}{\mathrm{Det}(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r+1)^2}{r}$$

To **remove edge points**, check against a threshold

$$R > 12(r = 10)$$

Principal orientation

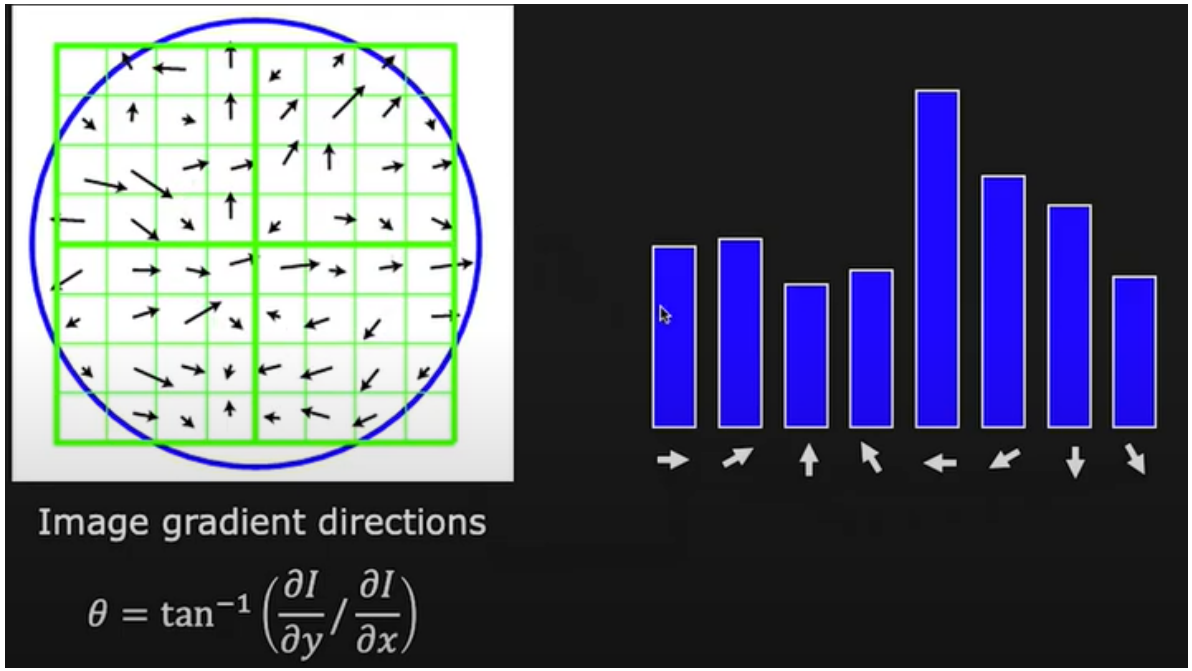

Figure 6: Use the histogram of gradient directions

The orientation histogram has 36 bins covering the 360 degree range of orientations. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with **a** σ **that is 1.5 times that of the scale of the keypoint**. Peaks in the orientation histogram correspond to dominant directions of local gradients.

Descriptor representation

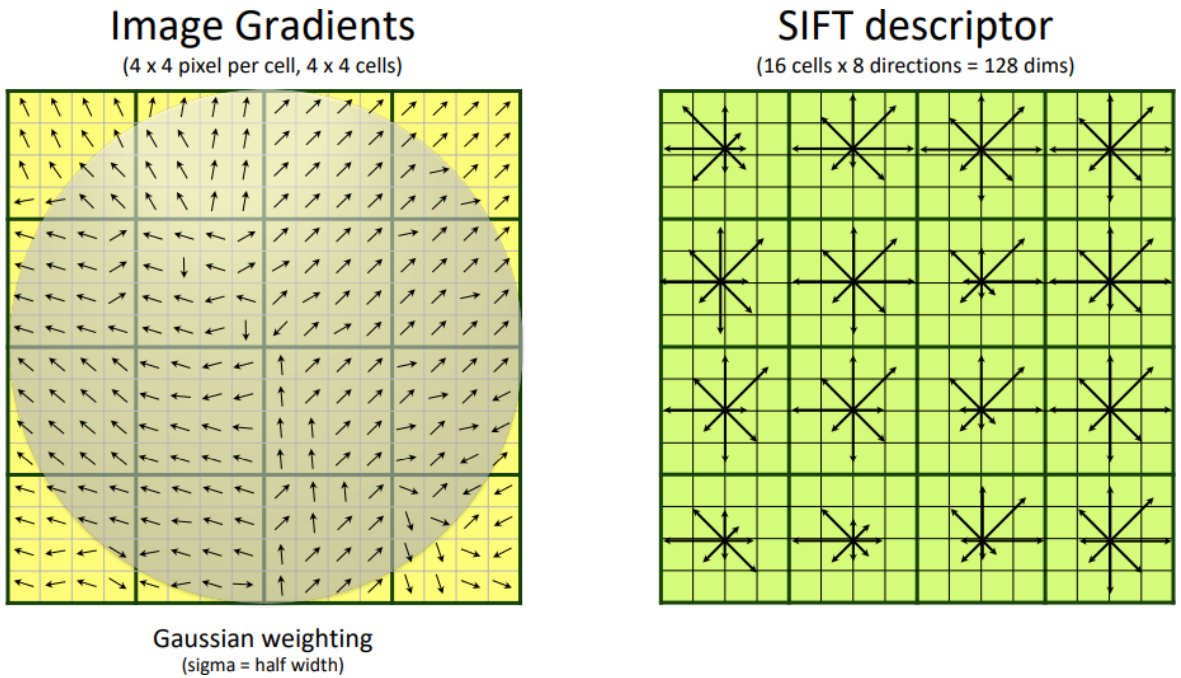

Figure 7: Descriptor representation

- Rotation invariance: relate with the keypoint principal orientation
- Collect into 4×4 orientation histograms with 8 orientation bins
- Bin value = sum of gradient magnitudes near that orientation
- Normalize feature vector to unit length to reduce effect of linear illumination change.

Image Registration

Image registration is the process of transforming different images of one scene into the same coordinate system. These images can be taken at different times (multi-temporal registration) and/or from different viewpoints. The spatial relationships between these images can be rigid (translations and rotations), affine (shears for example), homographies.

Keypoints matching

1. The **best candidate** match for each keypoint is found by identifying its **nearest neighbor** in the database of keypoints from training images.
2. The nearest neighbor is defined as the keypoint with minimum Euclidean distance between feature vectors:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

3. Efficient nearest neighbor indexing with k-d tree [3].

Normal equation

The affine transformation of $[x\ y]^T$ to $[u\ v]^T$ can be written as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

where the model translation is $[t_x\ t_y]^T$.

To solve for the transformation parameters, the equation above can be rewritten to gather the unknowns into a column vector:

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

x can be determined by solving the normal equations,

$$\mathbf{x} = \left[\mathbf{A}^\mathbf{T}\mathbf{A}\right]^{-1}\mathbf{A}^\mathbf{T}\mathbf{b}$$

Estimating homography using RANSAC

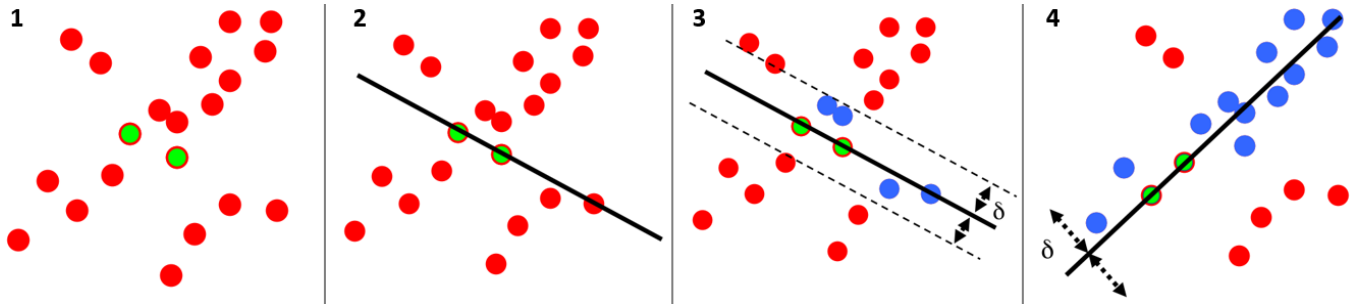

Figure 8: RANSAC algorithm[4] for fitting lines

1. Randomly choose s samples. Typically s is the minimum samples to fit a model.
2. Fit the model to the randomly chosen samples, i.e. compute H.
3. Count inliers that fit the model within a measure of error ε.
4. Repeat Steps N times.
5. Choose the model that has the largest number inliers and recompute H using all inliers.

Where $s = 4$ pairs of feature points and ε is acceptable alignment error in pixels for homography.

Bibliography

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] C. Harris, M. Stephens, *et al.*, "A combined corner and edge detector," in *Alvey vision conference*, Citeseer, vol. 15, 1988, pp. 10–5244.

[3] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.