Lecture #6-7: Features and Fitting/Feature Descriptors

Trevor Danielson, Wesley Olmsted, Kelsey Wang, Ben Barnett
Department of Computer Science
Stanford University
Stanford, CA 94305
{trevord, wolmsted, kyw, ben.barnett}@cs.stanford.edu

1 RANSAC

1.1 Goal

RANSAC comes from the phrase RANdom SAmple Consensus. It is a model fitting method for line detection in an image, which can be extremely useful for object identification, among other applications. It is often more helpful than pure edge detection because unless an image is noise-free, detected edges are likely to contain extra points, while also leaving out certain points that are in fact part of the real edge in the image.

1.2 Motivation

One of the primary advantages of using RANSAC is that it is relatively efficient and accurate even when the number of parameters is high. However, it should be noted that RANSAC is likely to fail or produce inaccurate results in images with a relatively large amount of noise.

1.3 General Approach

The intuition for RANSAC is that by randomly sampling a group of points in an edge and applying a line of best fit to those points *many times*, we have a high probability of finding a line that fits the points very well. Below is the general process "RANSAC loop":

1. Randomly select a seed group of points from the overall group of points you are trying to fit with a line.
2. Compute a line of best fit among the seed group. For example, if the seed group is only 2 distinct points, then it is clear to see that there is only one line that passes through both points, which can be determined with relative ease from the points' coordinates.
3. Find the number of **inliers** to this line by iterating over each point in the data set and calculating its distance from the line; if is less than a (predetermined) threshold value, it counts as an inlier. Otherwise, it is counted as an outlier.
4. If the number of inliers is sufficiently large, conclude that this line is "good", and that at least one line exists that includes the inliers tallied in the previous step. To further improve the line, re-compute the line using a least-squares estimate using all of the inliers that were within the threshold distance. Keep this transformation as the line that best approximates the data.

1.4 Drawbacks

The biggest drawback to RANSAC is that the greater fraction of outliers among the data points to be fitted, the more samples that are required to get a line of best fit. More importantly, the noisier an image is, the less likely it is for a line to *ever* be considered sufficiently good at fitting the data. This is a significant problem because most real world problems have a relatively large proportion of noise/outliers.

2 Local Invariant Features

2.1 Motivation

The purpose of local invariant features in image detection is motivated by its usefulness in detection under a wide range of circumstances that pose issues for previously discussed methods, such as cross-correlation. This method works by finding local, distinctive structures within an image and uses the surrounding region as small patches as opposed to using "global" representations to find correspondences. By doing so, this allows for a more robust image detection strategy invariant to object rotations, point of view translations, scale changes, etc.

2.2 General Approach

Below is the general approach associated with employing local invariant features in image detection.

1. Find and define a set of distinctive key points.
2. Define a local region around the keypoint.
3. Extract and normalize the regional content from the area.
4. Compute a local descriptor from the normalized region (ie. function of pixel color).
5. Match local descriptors.

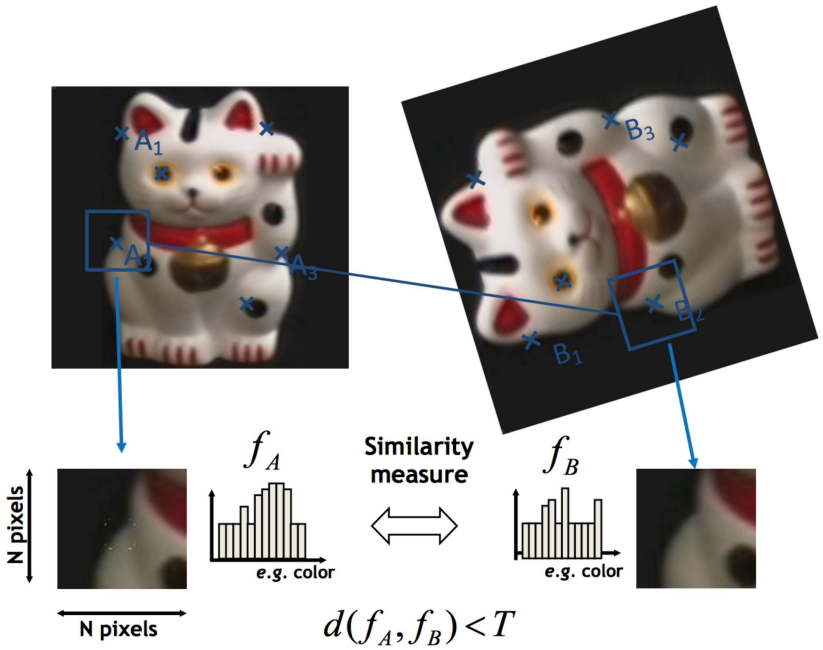

Figure 1: Example of local features within similar pictures being identified and normalized, then defined by a descriptor and compared to each other for similarity. Source: Lecture 6 slides.

2.3 Requirements

Good local features should have the following properties.

1. *Repeatability*: Given the same object or scene under different image conditions such as lighting or viewpoint change, a high amount of features should be detectable in both images being compared. In other words, should be robust to lighting changes, noise, blur, etc, as well as invariant to rotations and viewpoint changes.
2. *Locality*: Features should be local to avoid issues caused by occlusion and clutter.
3. *Quantity*: There need to be enough features chosen to sufficiently identify the object.
4. *Distinctiveness*: Features need to contain "interesting" features that show a large amount of variation in order to ensure the features can be distinguished.
5. *Efficiency*: Feature matching in new images should be conducive to real-time applications.

3 Keypoint Localization

3.1 Motivation

The goals of keypoint localization are to detect regions consistently and repeatably, to allow for more precise localization, and to find interesting content within the image.

3.2 General Approach

We will look for corners, as they are repeatable and distinctive in the majority of images. To find corners, we will look for large changes in intensity in all dimensions. To provide context, a "flat" region would not have change in any direction and an edge would have no change along the direction of the edge. We will find these corners using the Harris technique.

3.3 Harris Detector

To calculate the change of intensity for the shift [u,v]:

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

To find corners, we must maximize this function. Taylor Expansion is then used in the process to get the following equation:

$$E(u,v) = [u \quad v]\, M \begin{bmatrix} u \\ v \end{bmatrix}$$

where we have M defined as:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

This matrix reveals:

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Corners will have both large and similar eigenvalues, whereas edges will have one significantly greater eigenvalue and flat regions will have both small eigenvalues. The Corner Response Function computes a score for each window:

$$\theta = det(M) - \alpha \text{trace}(M)^2$$

where α is a constant around .04-.06.

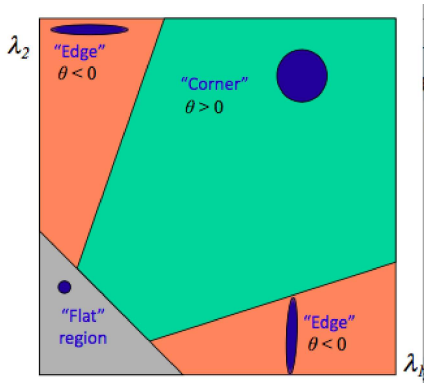

Figure 2: Visualization of theta thresholds from Corner Response Function indicating Corner, Edge, or Flat regions. Source: Lecture 6 slides

This is not rotation invariant. To allow for rotation invariance, we will smooth with Gaussian, where the Gaussian already performs the weighted sum:

$$M = g(\sigma) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

Ultimately, this is an example of keypoints that the Harris detector can identify:

Figure 3: Example of Harris keypoint detection on an image. Source: Lecture 6 slides

4 Scale Invariant Keypoint Detection

4.1 Motivation

Earlier we used the Harris detector to find keypoints or corners. The Harris detector used small windows in order to maintain good locality. Since the Harris detector uses this small window, if an image is scaled up, the window now no longer sees the same gradients it had with a smaller image.

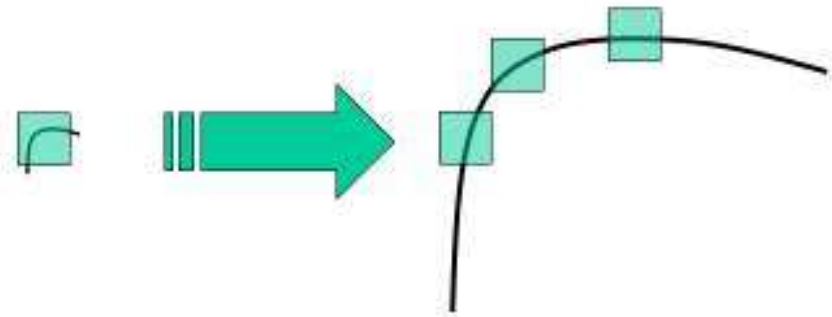

Figure 4: Harris detector windows on an increased scale. Source:
$https://docs.opencv.org/3.1.0/sift_scale_invariant.jpg$

Figure 4: Looking at the above image, we can see how the three windows on the right no longer see the sharp gradient in the x and y direction. All three now would classify that space as being edge. In order to address this problem, we need to normalize the scale of the detector.

4.2 Solution

We can design a function to be scalable meaning that corresponding regions are the same regardless of the scale. We can use a circle to represent this scalable function. A point on the circle is a function of the region size of the circle's radius.

4.3 General Approach

We can find the local max of a function. Relative to the local max, the region size should be the same regardless of the scale. This also means that the region size is co-variant with the image scale. A "good" function means that there is a single, distinct local max. This means in general, we should use a function that responds well to stark contrasts in intensity.

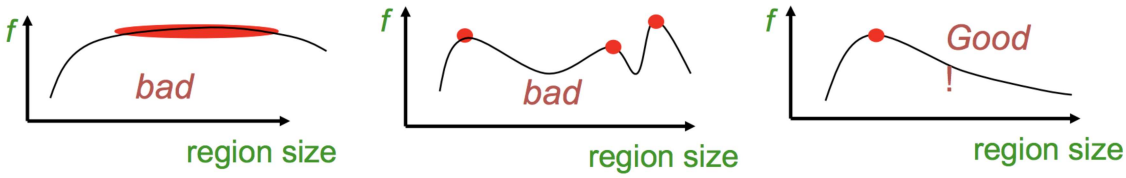

Figure 5: Examples of different functions for finding local maxes. Source: Lecture 7 slides.

Our function is defined as: $f = kernel * image$. Two kernels we could use are Laplacian or Difference of Gaussians.

$$L = \sigma^2(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$$

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

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

Both of these kernels are also scale and rotation invariant.

References