Features and Image Matching

Feature detection and matching is an important task in many computer vision applications, such as structure-from-motion, image retrieval, object detection, and more.

1 Feature
- A feature is a piece of information which is relevant for solving the computational task related to a certain application. Features may be specific structures in the image such as points, edges or objects. The features can be classified into two main categories:
+ Keypoint features: localized features such as the features that are in specific locations of the images (mountain peaks, building corners, doorways, or interestingly shaped patches of snow) are often called keypoint features. Key point features are often described by the appearance of patches of pixels surrounding the point location.
+ Edges: the features that can be matched based on their orientation and local appearance (edge profiles). Edges can be good indicators of object boundaries and occlusion events in the image sequence.

2 Application of Feature detection and matching
- Automate object tracking
- Point matching for computing disparity
- Stereo calibration(Estimation of the fundamental matrix)
- Motion-based segmentation
- Recognition
- 3D object reconstruction
- Robot navigation
- Image retrieval and indexing

3 Main component of Feature detection and matching
- Detection: Identify the Interest Point.
- Matching: Descriptors are compared across the images, to identify similar features. For two images we may get a set of pairs (***Xi, Yi***) ↔ (***Xi`, Yi`***), where (***Xi, Yi***) is a feature in one image and (***Xi`, Yi`***) its matching feature in the other image.

4 Algorithm for Feature detection and Matching
- Find a set of distinctive keypoints.
- Define a region around each keypoint.
- Extract and normalize the region content.

- Compute a local descriptor from the normalized region.
- Match local descriptors.

5 Techniques

5.1 Harris Corner Detector

- Harris Corner Detector is a corner detection operator that is commonly used in computer vision algorithms to extract corners and infer features of an image.
- The idea is to consider a small window around each pixel p in an image. Uniqueness of all such pixel windows can be measured by shifting each window by a small amount in a given direction and measuring the amount of change that occurs in the pixel values.
- We take the sum squared difference (SSD) of the pixel values before and after the shift and identify pixel windows where the SSD is large for shifts in all 8 directions.
- Let define the change function E(u,v) as the sum of all the sum squared differences (SSD) , where u,v are the x,y coordinates of every pixel in 3 x 3 window and I is the intensity value of the pixel. The features in the image are all pixels that have large values of E(u,v), as defined by some threshold.

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

We have to maximize this function *E*(*u*,*v*) for corner detection. That means, we have to maximize the second term. Applying Taylor Expansion, we get the final equation.

$$E(u,v) \approx [u \quad v]\left(\sum \begin{bmatrix} {I_x}^2 & I_x I_y \\ I_x I_y & {I_y}^2 \end{bmatrix}\right)\begin{bmatrix} u \\ v \end{bmatrix}$$

Rename the summed-matrix M.

$$M = \sum w(x,y) \begin{bmatrix} {I_x}^2 & I_x I_y \\ I_x I_y & {I_y}^2 \end{bmatrix}$$

The equation becomes:

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

A score, R, is calculated for each window and identifies whether the pixel is corner or not.

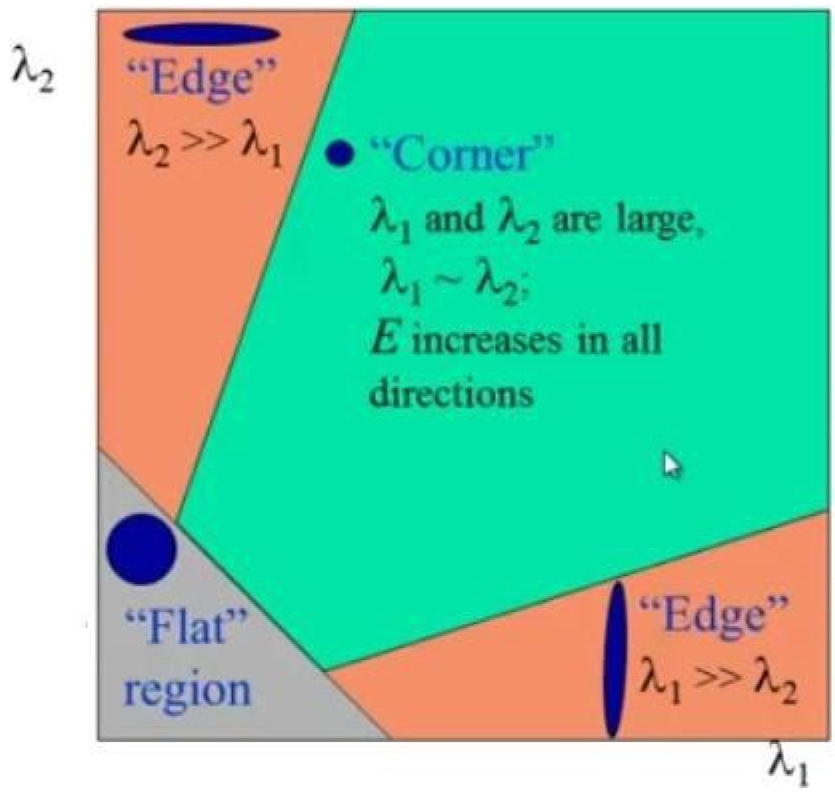
$$R = \det M - k(trace\, M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$trace\, M = \lambda_1 + \lambda_2$$

λ1 and λ2 are the eigenvalues of M. So the values of these eigenvalues decide whether a region is a corner, edge or flat.

+ When |R| is small, which happens when λ1 and λ2 are small, the region is flat.
+ When R<0, which happens when λ1>>λ2 or vice versa, the region is an edge.
+ When R is large, which happens when λ1 and λ2 are large and λ1~λ2, the region is a corner.


λ_2
"Edge"
$\lambda_2 >> \lambda_1$
"Corner"
λ_1 and λ_2 are large,
$\lambda_1 \sim \lambda_2$;
E increases in all
directions
"Flat"
region
"Edge"
$\lambda_1 >> \lambda_2$
λ_1