UNIT-II

Point and Patches

Feature Detectors

Detection : Identify the Interest Point.

Description :

Matching.

Interest point

It is the point at which the direction of the boundary of the object changes abruptly or intersection point between two or more edge segments.

Algorithms for Identification

* Harris Corner.
* SIFT
* SURF
* FAST
* ORB

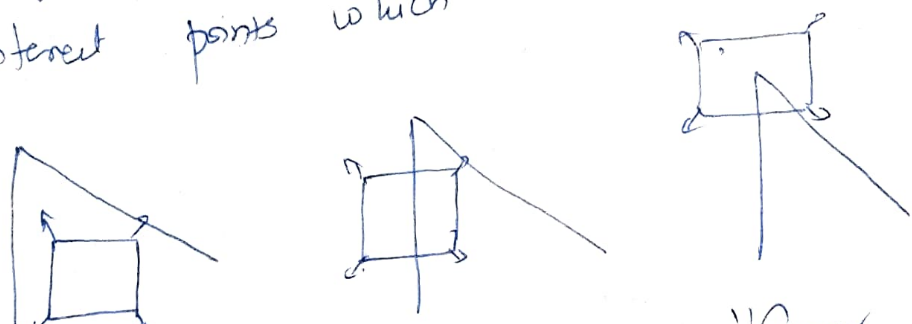
Application of Feature Detection and Matching

* Automotive object tracking
* Point matching
* Stereo Calibration
* Motion-based Segmentation

* Recognition
* 3D object reconstruction
* Robot Navigation
* Image retrieval & Indexing

Harris Corner detector

* Corner can be interpreted as the junction of two edges, where an edge is a sudden change in image brightness.
* Corners are the important features in the image, and they are generally termed as interest points which are invar



"Flat" Region

"Edge"

"Corner"

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

We have to maximize this function $E(u,v)$ for corner detection

Taylor Series for 2D Functions

$$f(x+u, y+v) = f(x,y) + u f_x(x,y) + v f_y(x,y) +$$

$$\frac{1}{2!}\left[u^2 f_{xx}(x,y) + uv f_{xy}\, x,y + v^2 f_{yy}(x,y)\right] +$$

$$\frac{1}{3!}\left[u^3 f_{xxx}(x,y) + u^2 v f_{xxy}(x,y) + uv^2 f_{xyy}(x,y) + v^3 f_{yyy}(x,y)\right] + \text{Higher order terms}$$

Harris Corner Derivation

$$\sum \left[I(x+u, y+v) - I(x,y)\right]^2$$

$$\approx \sum \left[I(x,y) + uI_x + vI_y - I(x,y)\right]^2$$

$$= \sum u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2$$

$$= \sum [u \ v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

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

Corner Response Measure

$$R = \det M - k(\operatorname{trace} M)^2$$

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

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

Feature Description.

A feature description is an algorithm which takes an image and outputs feature descriptors/feature vectors.

Descriptions can be categorized into two classes

* Local Descriptors:

  It is a compact representation of a point's local neighbourhood.

* Global Descriptors:

  A global descriptor describes the whole image. They are generally not very robust as a change in part of the image may cause it to fail as it will affect the resulting descriptor.

Two descriptor

MOPS

SIFT

Multiscale Oriented Patches descriptor (MOPS)

① Take a 40x40 square window around detected feature

② Scale to 1/5 size to get 8x8 square window

③ Rotate to horizontal

④ Normalize the window by subtracting the mean, dividing by the standard deviation in the window.

<u>Scale</u> <u>Invariant</u> <u>Feature</u> <u>Transform</u> (SIFT)

① Take 16x16 square window around detected interest point

② Compute edge orientation for each pixel

③ Throw out weak edges

④ Create histogram of surviving edge orientation

angle histogram

<u>Select</u> <u>Canonical</u> <u>Orientation</u>

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).

Properties of SIFT-based Matching

* Can handle changes in viewpoint
* Can handle significant changes in illumination
* Fast and efficient.

Feature Matching

Given a feature in I_1, how to find the best match in I_2?

1. Define distance function that compares two descriptors
2. Test all the features in I_2, find the one with min distance

→ Simple approach is $SSD(f_1, f_2)$
- Sum of square differences between entities of the two descriptors.

The difference between two features f_1, f_2

Better approach: ratio distance $= SSD(f_1, f_2) / SSD(f_1, f_2')$

4

- f_2 is best SSD match to f_1 in I_2
- f_2' is 2nd best SSD Match to f_1 in I_2

Decision rule: Accept match if $SSD < T$

where T is a threshold

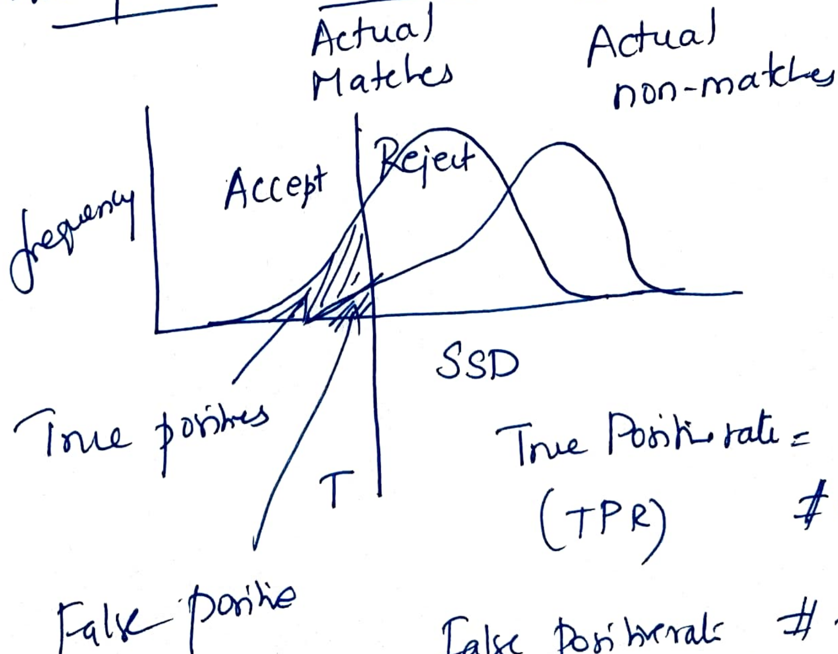
ex: Large T

$T = 250 \Rightarrow a, b, c$ are all accepted as matches

a and b are true matches ("true positives")

c is false match ("false positive")

True positives and False Positives



$$\text{True Positive rate (TPR)} = \frac{\#\text{ true positives}}{\#\text{ actual matches}}$$

$$\text{False positive rate (FPR)} = \frac{\#\text{ false positives}}{\text{actual non matches}}$$

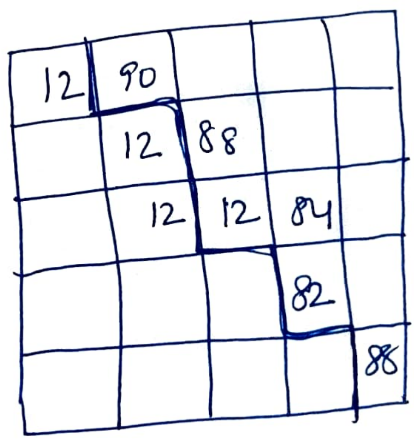
Application of features

1. Image alignment
2. Object recognition.
3. 3D reconstruction
4. Motion Tracking
5. Indexing & database retrieval
6. Robot Navigation.

Edges detection

Edges and lines are used in

* Object recognition.
* Image Matching
* Document analysis
* horizon detection
* line following robots



Look for a neighbourhood with strong signs of change.

Image gradient

The gradient of an image

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

The gradient direction is given by

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

The Sobel Operator

The definition of the sobel operator omits the 1/8 term

- doesn't make a difference for edge detection.

→ $\frac{1}{8}$ term is needed to get the right gradient value

2D edge detection filters

Gaussian

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{\frac{-u^2+v^2}{2\sigma^2}}$$

Derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Laplacian of Gaussian

$$\nabla^2 h_\sigma(u,v)$$

∇^2 is laplacian operator

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Canny Edge operator

1. Smoothing
2. Gradient
3. Thresholding
4. Non-maximum suppression
5. Tracing edges