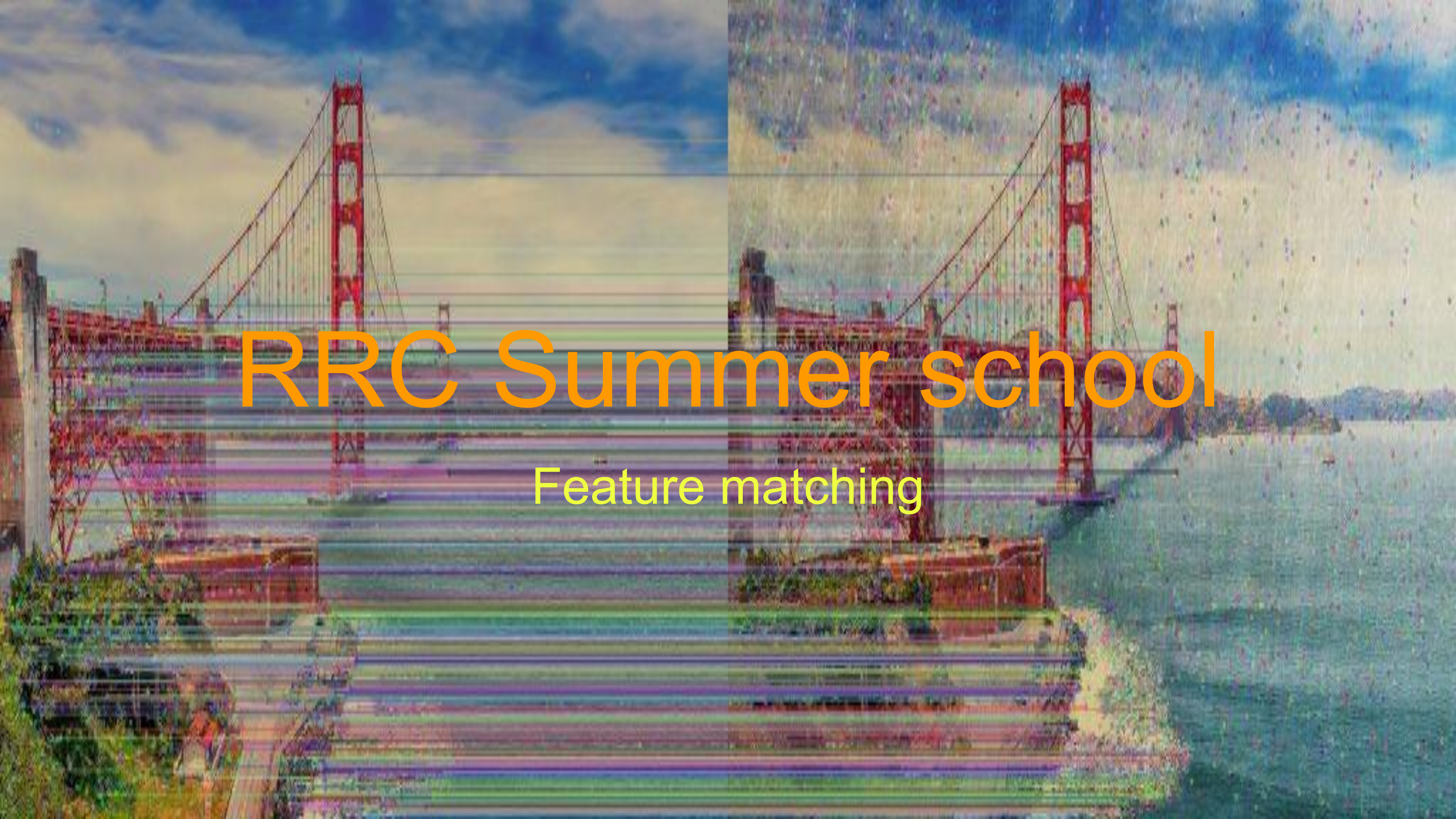


RRC Summer school

Feature matching

Topics of discussion

1.) **Feature matching:**
- **Intro to feature matching and applications.**
- **Traditional feature matching algorithms.**
- **Techniques for computing correspondences.**

2.) Homography

3.) Image stitching.

What is feature matching?

Feature: Keypoint + descriptor (1D vector in most cases).

In below image, colored lines are correspondences (there may be outliers).

Image 1 Image 2

Key properties of interest points:

a.) Translation Invariance.

b.) Rotation invariance.

c.) Scale invariance

d.) Robust to changes in illumination.

e.) Repeatability

Image courtesy: https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj2/html/zsun311/index.html

What’s the use?

a.) Image stitching.

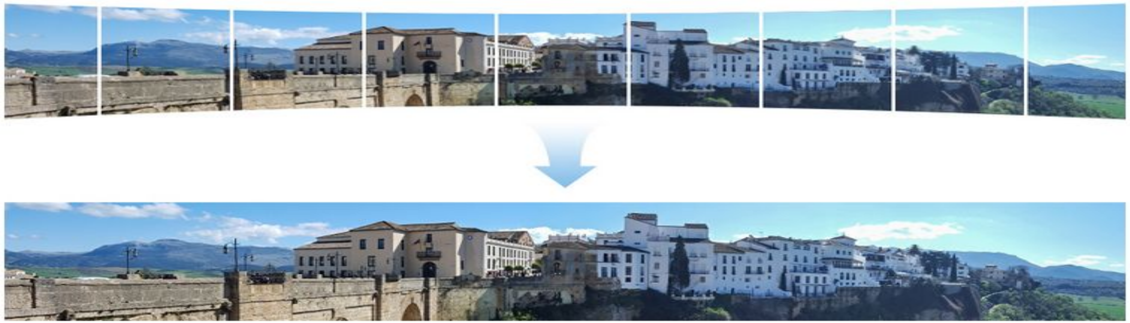


Image stitching

b.) SfM/SLAM.

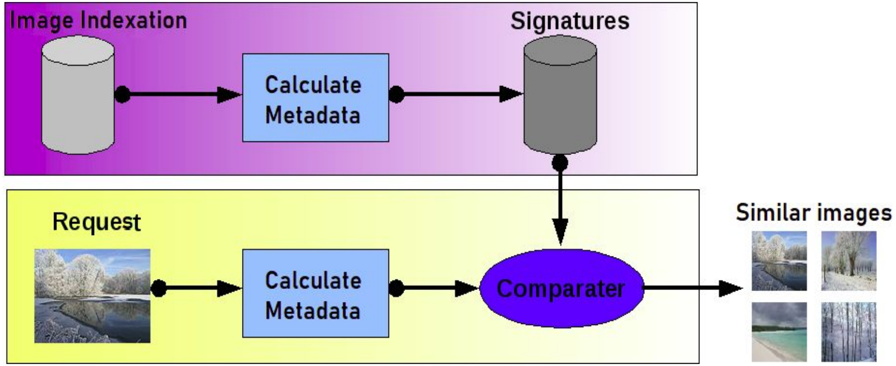
Building Rome in a day

More applications:

c.) Object tracking.

Remember this match? :)

d.) Image retrieval.



Given query image, return most similar images from DB.

So on…..

Image source: Wikipedia

Famous traditional image descriptors

a.) Harris corner detector

b.) SIFT, SURF

c.) FAST

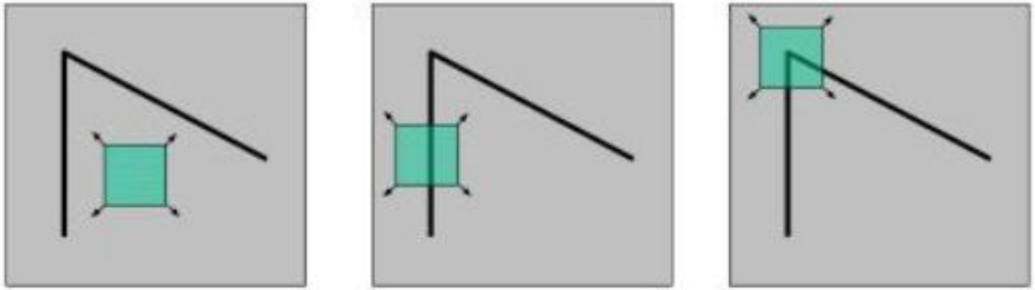
d.) BRIEF

e.) ORB

So on …..

Harris corner detection

- Key idea: Sliding window in any direction around the corner results in huge variation in intensity values.



- Consider a pixel of interest:

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{window function}} \; [\underbrace{I(x+u,y+v)}_{\text{shifted intensity}} - \underbrace{I(x,y)}_{\text{intensity}}]^2$$

(u, v) is shift in pixels along x and y directions.

Harris corner detection …

$$E(u,v)=\sum_{x,y}[\underbrace{I(x+u,y+v)}_{\text{shifted intensity}}-\underbrace{I(x,y)}_{\text{intensity}}]^2$$

$$E(u,v)=\sum_{x,y}[\underbrace{I(x,y)+uI_x+vI_y}_{\text{shifted intensity}}-\underbrace{I(x,y)}_{\text{intensity}}]^2$$

Taylor Series

$$E(u,v)=\sum_{x,y}[uI_x+vI_y]^2$$

$$E(u,v)=\sum_{x,y}\left[\begin{pmatrix}u & v\end{pmatrix}\begin{pmatrix}I_x\\ I_y\end{pmatrix}\right]^2$$

$$E(u,v)=\sum_{x,y}\begin{pmatrix}u & v\end{pmatrix}\begin{pmatrix}I_x\\ I_y\end{pmatrix}\begin{pmatrix}I_x & I_y\end{pmatrix}\begin{pmatrix}u\\ v\end{pmatrix}$$

$$M=\sum_{x,y}\begin{bmatrix}I_xI_x & I_xI_y\\ I_xI_y & I_yI_y\end{bmatrix}$$

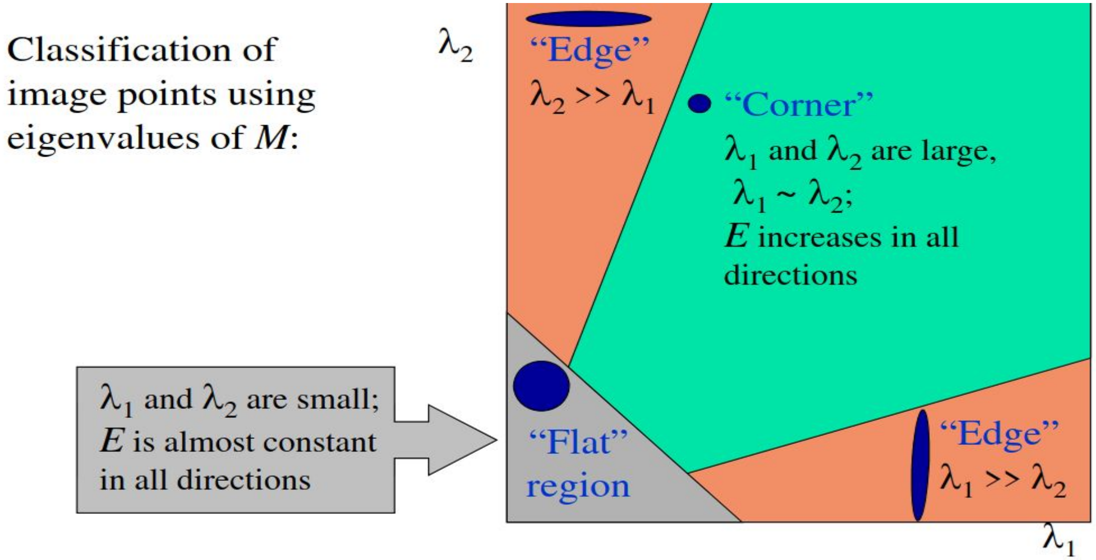
$$E(u,v)=\begin{pmatrix}u & v\end{pmatrix}\left[\sum_{x,y}\begin{pmatrix}I_x\\ I_y\end{pmatrix}\begin{pmatrix}I_x & I_y\end{pmatrix}\right]\begin{pmatrix}u\\ v\end{pmatrix}$$

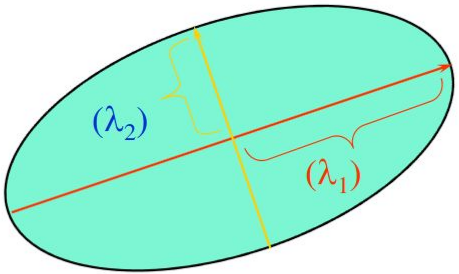
$$E(u,v)=\begin{pmatrix}u & v\end{pmatrix}M\begin{pmatrix}u\\ v\end{pmatrix}$$

Source: https://www.crcv.ucf.edu/wp-content/uploads/2019/03/Lecture-4-Harris.pdf

Harris corner detection…

- E(u, v) to be maximized at interest point. This is eqn of ellipse. And shape of ellipse is determined by eigenvalues of M.





Should we even compute Eigenvalues? Not efficient to do for many pixels

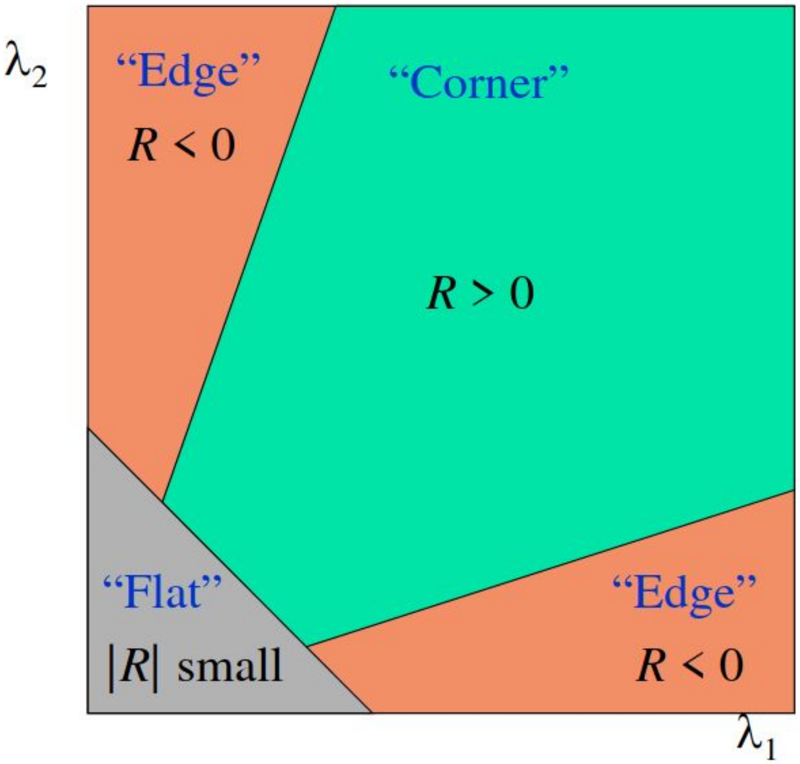
Source: https://www.crcv.ucf.edu/wp-content/uploads/2019/03/Lecture-4-Harris.pdf

Harris response value

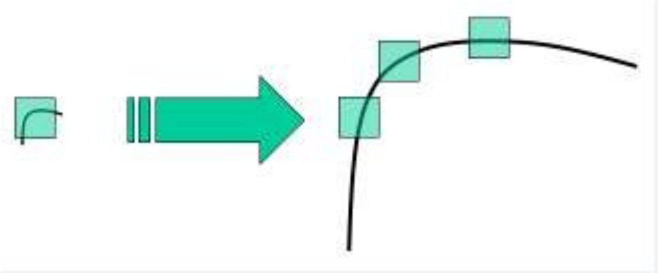
Denoted by R

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

- Get R for all pixels and select corners if they cross a threshold value.
- Which properties violate?

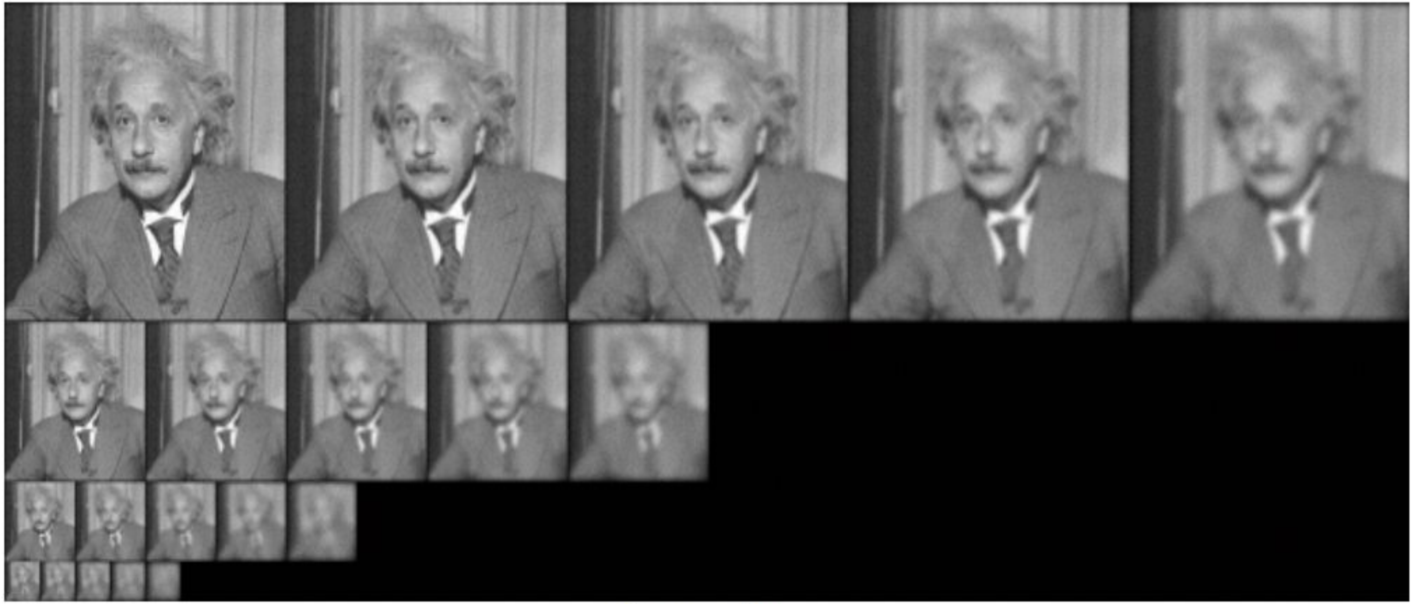


- Any drawbacks with Harris corner detection?



SIFT…

- **Step 1 : Scale space extrema detection (keypoint detection)**



First octave

Second octave

Third octave

Fourth octave

Gaussian

$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$	B = I * G(k*σ)

In each octave, blurring is increased by a factor of k.

Ref: http://www.cs.cmu.edu/~16385/s17/Slides/7.3_SIFT_Detector_and_Descriptor.pdf

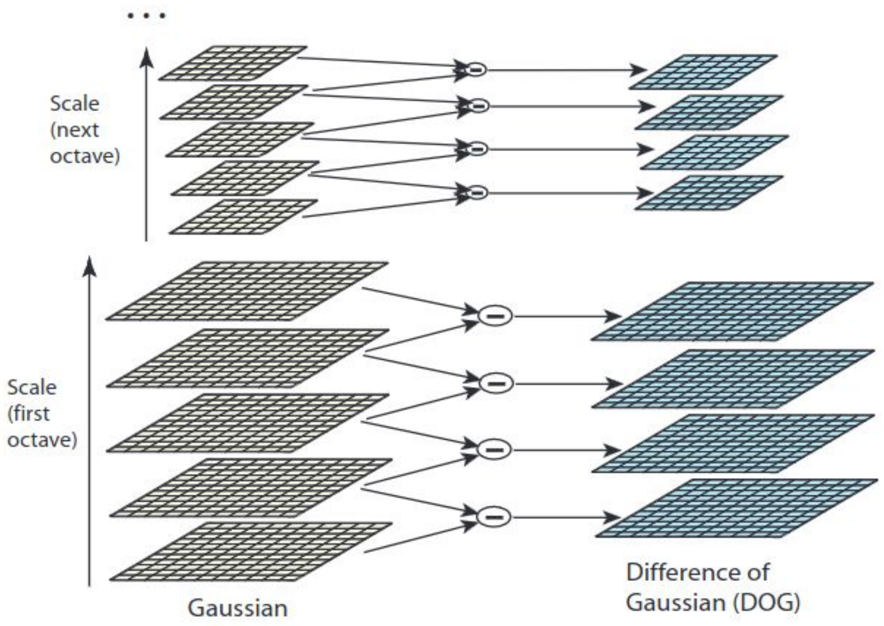
SIFT…

- **Approximate LoG using scale space**

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

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

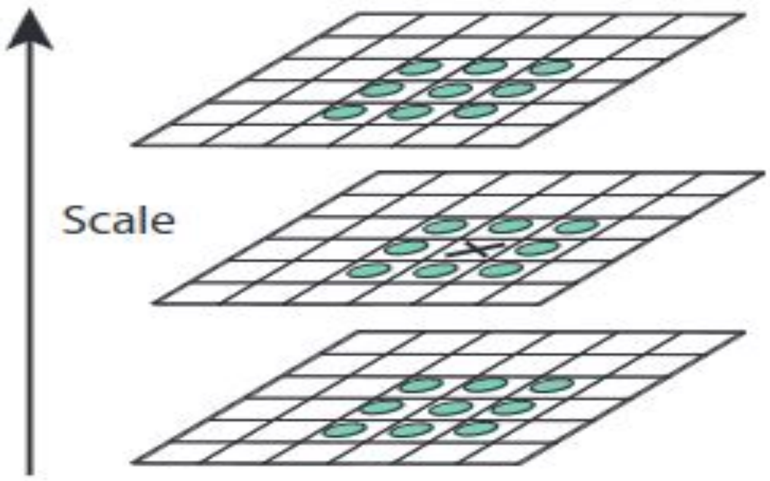
$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2 \nabla^2 G.$$



Ref: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

SIFT…

- **Find the extrema using DoG images**



These are not exact optima in LoG, needs to be refined.

Ref: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

SIFT…

- **Step 2 : Local Extrema detection**

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

$$\downarrow$$

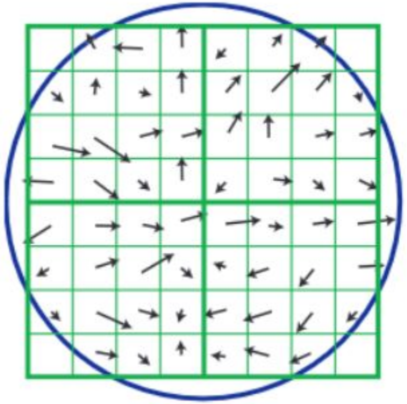
$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}.$$

Ref: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

SIFT…

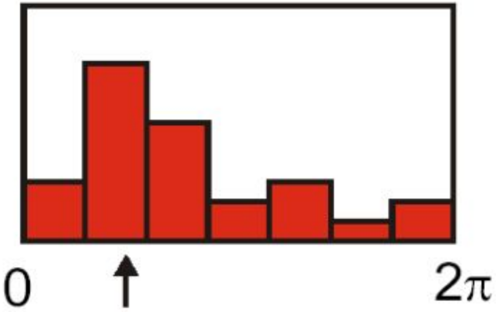
- **Step 3 : Estimate orientation of keypoint**

  Why are we estimating direction of keypoint?



$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

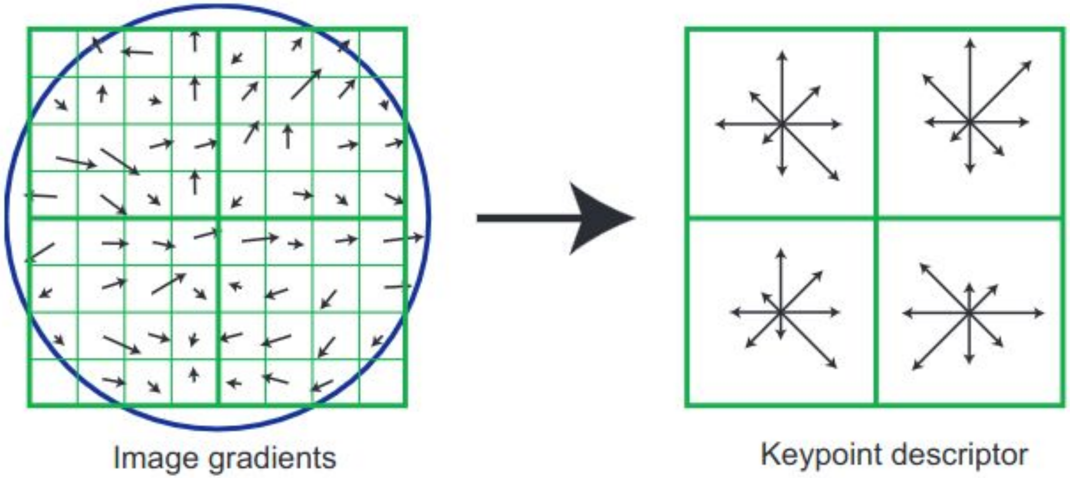
$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$



Ref: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

SIFT…

- **Step 4 : Compute descriptor**



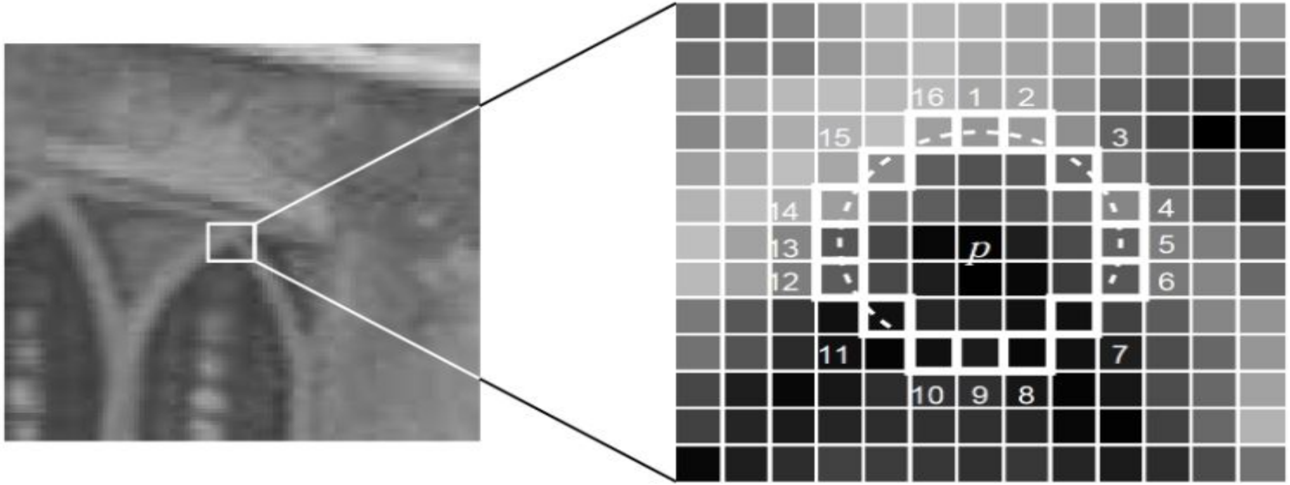
Divide into 8 bins (Total descriptor len = 4*4*8)

Ref: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

FAST (**Features from accelerated segment test)**

- Works in real time (useful in applications like SLAM, tracking etc).
- Key terms in algo: Thresh (T), bresenham circle (rad 3).

- **Threshold criterion**: N(12) out of 16 pixel intensities should lie outside [I_p-T, I_p+T].
- **Qualifying test**: At least 3 out of 4 pixels (1, 5, 8, 13) should satisfy threshold criterion.
- If pixel p passes Qualifying test, go for Threshold criterion.



Ref: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/AV1FeaturefromAcceleratedSegmentTest.pdf

Ref 2 : https://ieeexplore.ieee.org/document/4674368

BRIEF **(Binary Robust Independent Elementary Features)**

- Binary descriptor unlike prev ones. So, matching correspondences is very fast.

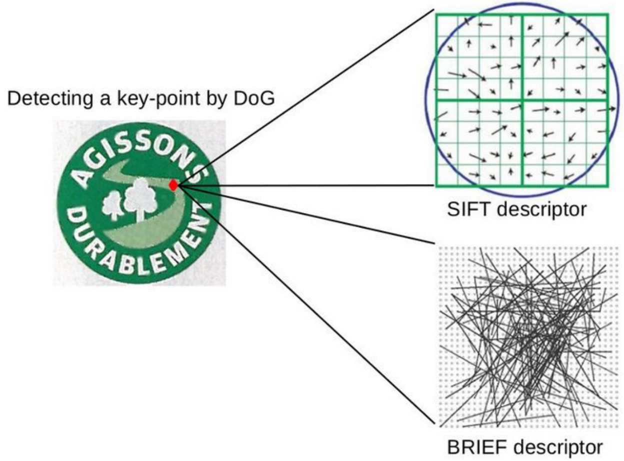
Steps:

1. Smooth image using Gaussian kernel (9*9). Why?
2. Get the keypoints using FAST or any other algo.
3. Compute descriptors for each kp:
    - Consider patch of S*S around kp.
    - Select n_d (x,y) pixel pair locations in patch. How do we do this?
    - Perform test T on each pixel pair.

$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} ,$$

    - Descriptor is computed as:

$$f_{n_d}(\mathbf{p}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \, \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i) \ .$$



Note: BRIEF-k (k is no of bytes per descriptor = n_d/8)

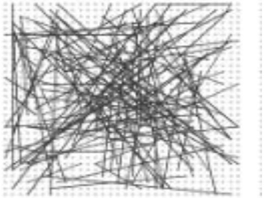
Ref: https://www.cs.ubc.ca/~lowe/525/papers/calonder_eccv10.pdf

Selecting test locations in BRIEF:

There are 3 main sampling strategies for test locations:

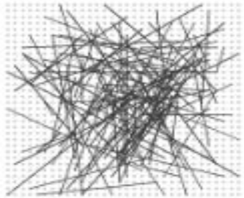
1.) Uniform:

$$(\mathbf{X}, \mathbf{Y}) \sim \text{i.i.d. Uniform}(-\frac{S}{2}, \frac{S}{2})$$



2.) Gaussian:

$$(\mathbf{X}, \mathbf{Y}) \sim \text{i.i.d. Gaussian}(0, \frac{1}{25}S^2)$$



3.) Gaussian with second pixel centered around first pixel:

$$\mathbf{X} \sim \text{i.i.d. Gaussian}(0, \frac{1}{25}S^2)$$

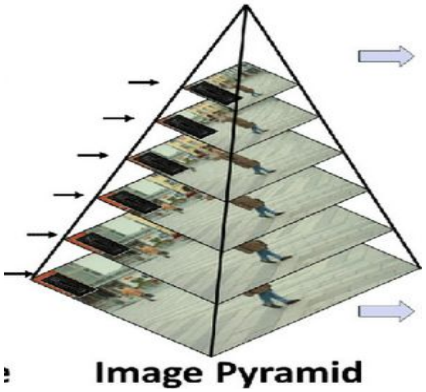
$$\mathbf{Y} \sim \text{i.i.d. Gaussian}(\mathbf{x}_i, \frac{1}{100}S^2)$$

ORB (Oriented FAST and Rotated BRIEF):

- Combines idea of FAST kp detector and BRIEF descriptor.
- Any issues with BRIEF and FAST?

Steps:

1. Detect Feature points/keypoints (based on improved FAST):
    - Apply standard FAST to get feature points and compute Harris response values.
    - Sort the feature points based on response values and pick largest N.
    - ORB uses image pyramid technique (to address scale invariance).



ORB …

- Intensity Centroid method is used to compute direction of FAST features/kps.

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y), \longrightarrow C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \longrightarrow \theta = \operatorname{atan2}(m_{01}, m_{10}),$$

2. Computing descriptors for the oriented FAST keypoints:

- Standard BRIEF is not rotation invariant. So, improved version called steerBRIEF is used.
- We already have theta(orientation) of each kp from improved FAST.

$$R_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \qquad Q = \begin{bmatrix} x_1, x_2, \cdots, x_N \\ y_1, y_2, \cdots, y_N \end{bmatrix} \longrightarrow Q_\theta = R_\theta Q$$

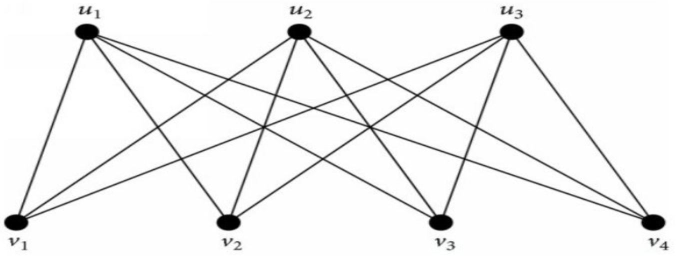
$$g_{N(p,\theta)} = f_N(p) | (x_i, y_i) \in Q_\theta$$

3. Matching/Correspondence estimation: Based on FLANN.

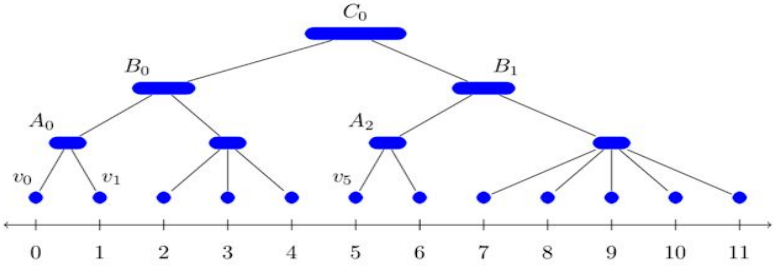
Ref: https://iopscience.iop.org/article/10.1088/1742-6596/1237/3/032020/pdf

Find Correspondences

- **Brute force**: Every descriptor in first image is matched with every descriptor in second image and we check for the closest one based on some distance metric.



- **FLANN** (Fast Library for Approximate Nearest Neighbors): Builds a KDtree/ Hierarchical k means tree from the descriptors of second image (inverted file index). For a descriptor in first image, we query the tree and find the leaf node which is nearest.`



Ref: https://ai.googleblog.com/2018/03/balanced-partitioning-and-hierarchical.html

Next topic:

1.) Feature matching

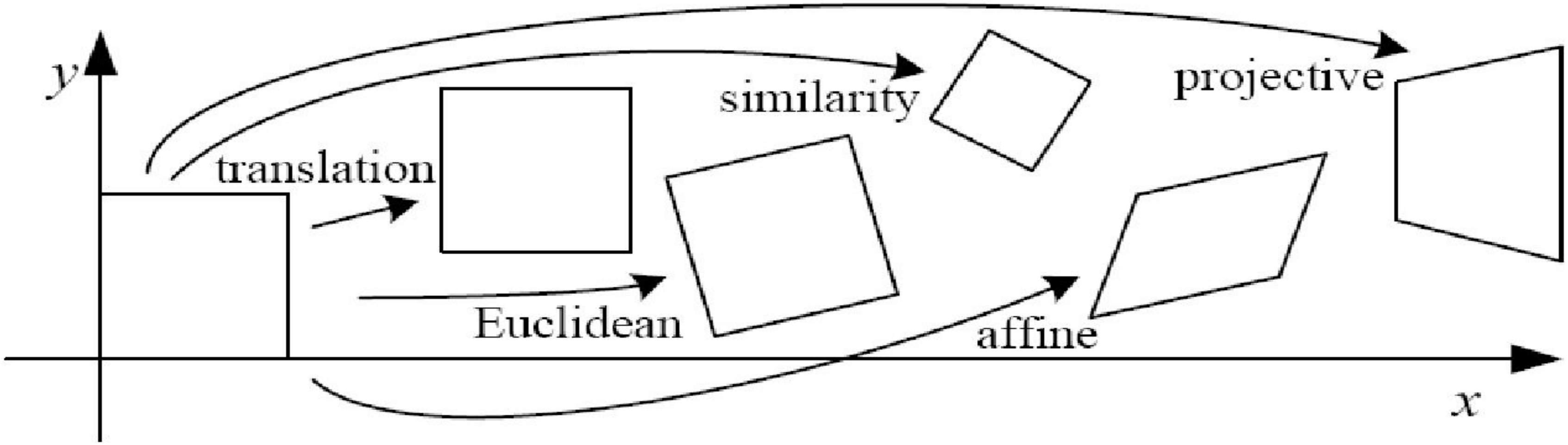
2.) **Homography and Image stitching:**

- **Introduction to Homography.**
- **Computing H using DLT.**
- **Application of Homography (Panorama)**

3.) Code demo.

Homography

Review of 2D transformations:



Ref: CMU 16-385 course

2D transformations …

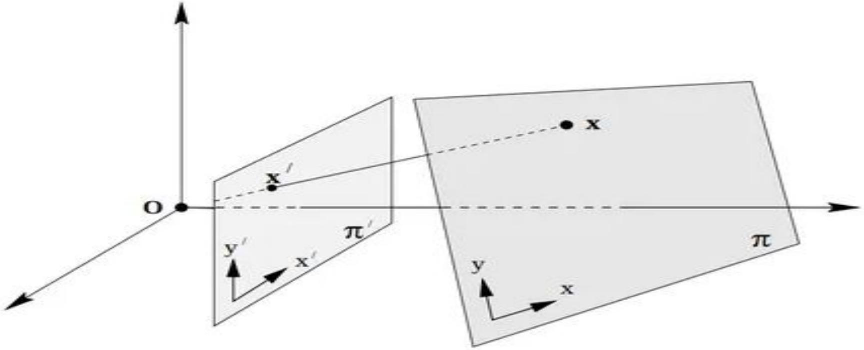
Hierarchy of 2D transformations:

Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$.
Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle. The circular points, **I**, **J** (see section 2.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

Ref: MVG book

Homography (projective transformation)

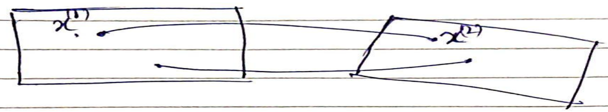
- Maps points from one plane to another plane (common centre of projection).



$$x' = Hx$$

Computing H (DLT algorithm):

- Consider a single correspondence b/w 2 images.



$$x^{(1)} = (x_1 \quad y_1 \quad 1)$$
$$x^{(2)} = (x_2 \quad y_2 \quad 1)$$

$$x^{(2)} = H\,x^{(1)} \implies \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

$$\implies x_2 = \frac{h_{11}x_1 + h_{12}y_1 + h_{13}}{h_{31}x_1 + h_{32}x_2 + 1}, \quad y_2 = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + 1}$$

Arranging in matrix form

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_2 & -y_1x_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1 & -y_1y_2 \end{bmatrix}}_{A} \overset{H}{\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix}} = \overset{b}{\begin{bmatrix} x_2 \\ y_2 \end{bmatrix}}$$

$$\implies \boxed{AH = b}$$

DLT ..

- We get 2 equations for each correspondence, how many are required to solve for H?

$$A_1 h = b_1$$
$$A_2 h = b_2$$
$$A_3 h = b_3$$
$$A_4 h = b_4$$

$$\Rightarrow \underbrace{\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix}}_{8 \times 8} h = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}}_{8 \times 1}$$

- Solve above system of linear equations using either least squares / SVD.

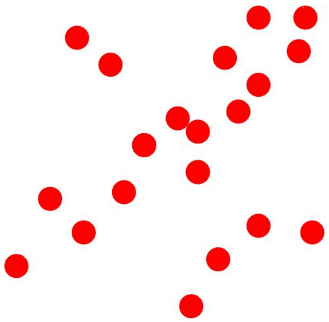
RANSAC (Random sample consensus):

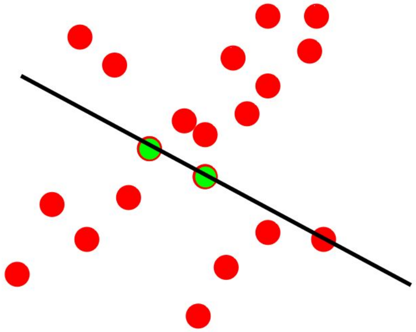
We have a problem here ..

How do we address the wrong correspondences?

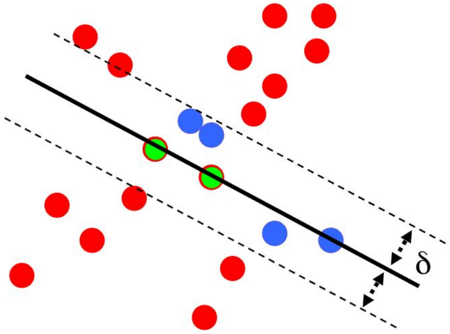
RANSAC ..

Algo:





Randomly sample points and fit model.



Set thresh and count inliers.

Repeat the above steps for N times. Choose the model which has highest inliers.

Ref: CMU 16-385 course

Adaptive RANSAC:

- What should N be? (N is no of samples - iterations of outer loop)

- $N = \infty$, sample_count$= 0$.
- While $N >$ sample_count Repeat
  - Choose a sample and count the number of inliers.
  - Set $\epsilon = 1 -$ (number of inliers)/(total number of points)
  - Set N from ϵ and (4.18) with $p = 0.99$.
  - Increment the sample_count by 1.
- Terminate.

$$N = \log(1 - p)/\log(1 - (1 - \epsilon)^s). \qquad (4.18)$$

- Epsilon = prob that random data point is outlier.
- p = prob that all points in atleast one of N samples are inliers (set to 0.99).

Ref: MVG book

Computing H using RANSAC:

Steps:

- Get keypoints and match the descriptors to get correspondences.
- RANSAC outer loop (N samples) :
    1. Choose 4 correspondences.
    2. Compute H.
    3. Find no of inliers. (based on error e = l2 norm(Hx - x’))

- Choose H_i with highest inliers.
- Recompute H using the model with highest inliers (can be further refined using LM solver).

Image stitching/Panorama