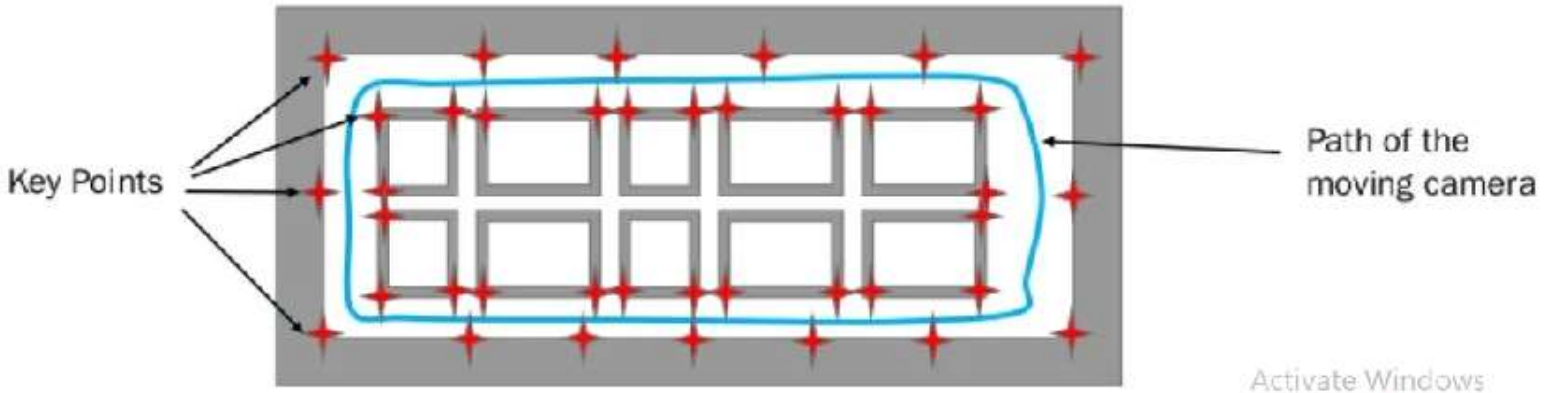
VISUAL SLAM

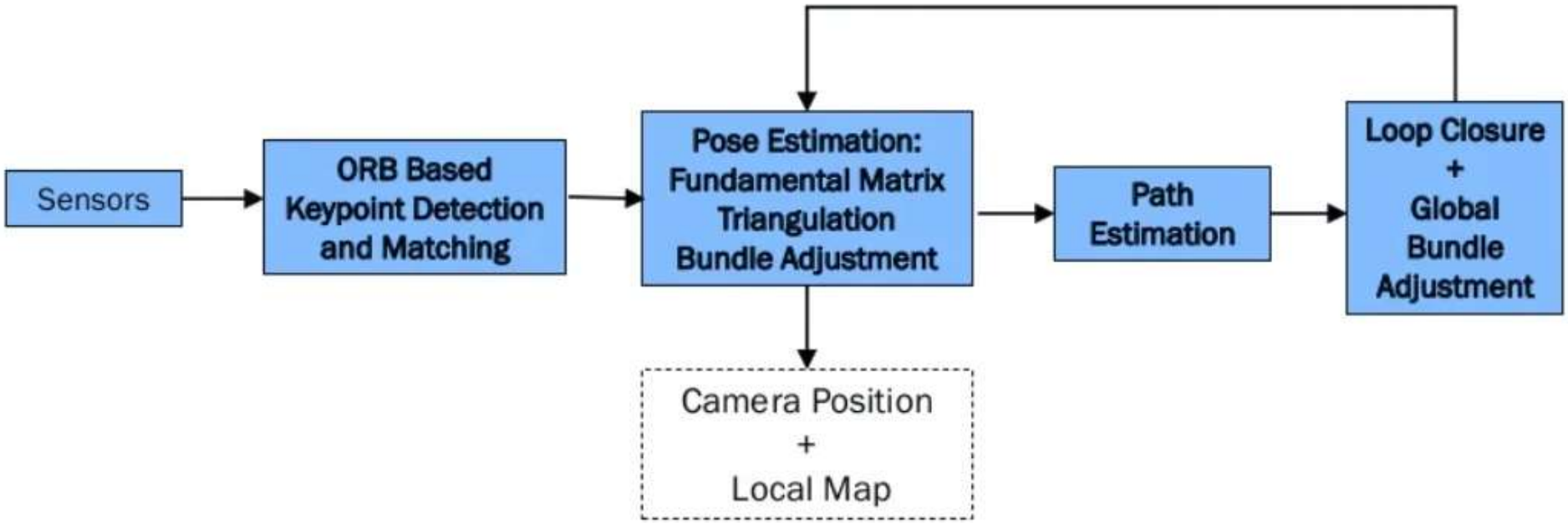
Some slides were adapted/taken from various sources, including 3D Computer Vision of Prof. Hee, NUS, Air Lab Summer School, The Robotic Institute, CMU, Computer Vision of Prof. Mubarak Shah, UCF, Computer Vision of Prof. William Hoff, Colorado School of Mines, Coursera Visual Odometry, Robotics: Perception, University of Pennsylvania, "Fundamentals of Monocular SLAM," a Presentation from Cadence and many more. We thankfully acknowledge them. Students are requested to use this material for their study only and **NOT** to distribute it.

SLAM: Problem description

- A camera mounted on a mobile device (cell phone, robot, drone) is moving in an unknown region, capturing an image sequence of the surroundings. By analyzing the captured image sequence, we would like to:
  - Identify the distinct features (key points) in the surroundings, along with their locations (Mapping)
  - Trace the path taken by the moving camera in this region (Localization)
- Simultaneous Localization And Mapping = SLAM



SLAM: Block Diagram

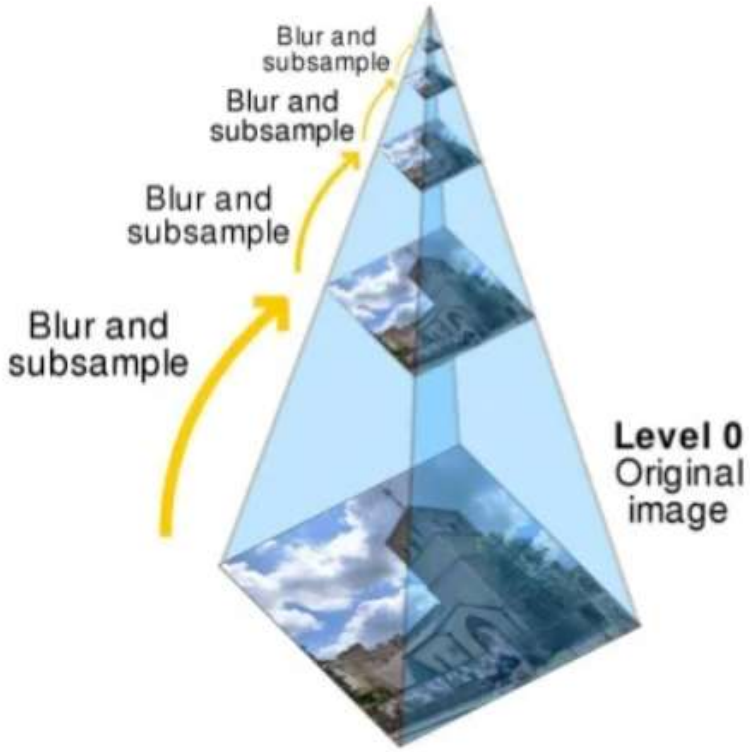


Part I: Key point estimation and matching

- Given a sequence of images,
- Find the important key points: (2D Location + Quantitative descriptor) per key point
- Match Key Points in different frames : Minimum “distance” between descriptors
- Final Goal:
  - Locate key points
  - Track the 2D movement of the key points across the image sequence
- We use ORB (Oriented fast, Rotated BRIEF) for key point detection and descriptor computation
- Match Key-Points via minimum Hamming distance between BRIEF descriptors

ORG1: Image Pyramid

- Create a scaled image pyramid for each frame
- Basic Idea: Different key points (Features) may be best detected at different image scales.
- Example: Consider Image scaling needed to best detect the following Key-Points
  - Corner of room, corner of a table, corner of a laptop, corner of a cell phone
- Typical Image Pyramid:
  - 8 level scaling
  - Down scale the image by a factor of 1.2

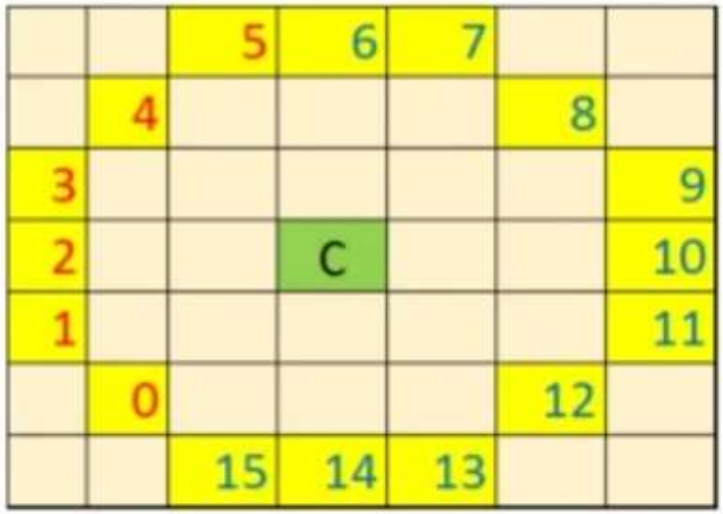


ORB2: FAST9: Features from accelerated Segment test

- Consider the center pixel – C in the window.
- Consider a circle of 16-pixels around C

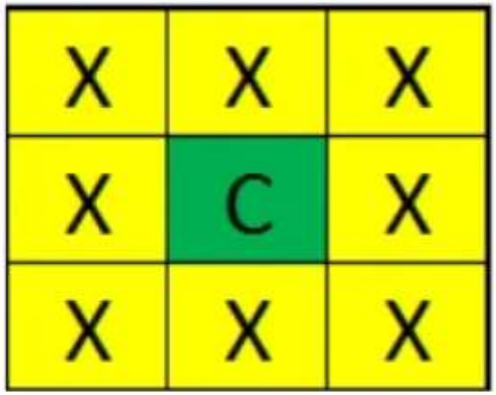
To classify the pixel C as a key point:

- 9 or more contiguous pixels out of the 16 should be darker than pixel C, with intensity difference > Threshold
- Or, 9 or more contiguous pixels out of the 16 should be brighter than pixel C, with intensity difference > Threshold
- Threshold can be say 20% of pixel value at C



ORB3: Non Maximal Suppression

- Typically each corner tends to be detected at several adjacent locations/ pixels
- Non-Max suppression is used to retain the most dominant corner pixel
- Simple NMS on a 3x3 window: If the FAST score is maximum for the center pixel C compared 8 neighbors then the pixel is considered as corner.



ORB4: Rotation Invariance

- Problem: As the camera moves, there can a non trivial camera rotation
- The orientation of the surroundings for the same key point can be significantly different from frame to frame.
- BRIEF descriptors are NOT rotationally invariant
- Need to explicitly correct for Key Point rotation:

- Consider a Circular patch of fixed radius around the key point
- Find an angle θ, such that if patch is rotated θ, the x-moment of the pixel values about the center pixel is = 0
- $\theta = \tan^{-1}(\text{y-moment}/\text{x-moment})$
  - x-moment = $\sum (x - x_0) P(x,y)$
  - y-moment = $\sum (y - y_0) P(x,y)$
  - (x_0,y_0) are center pixel coordinates, P(x,y) is pixel value at (x, y)

ORB5: BRIEF: Binary Robust Independent Elementary Feature

- BRIEF is 256 bit descriptor that characterizes the key point surrounding area
- BRIEF descriptor allows for key-point matching using Hamming Distance.
- Consider a patch of size 31x31 around the key-point

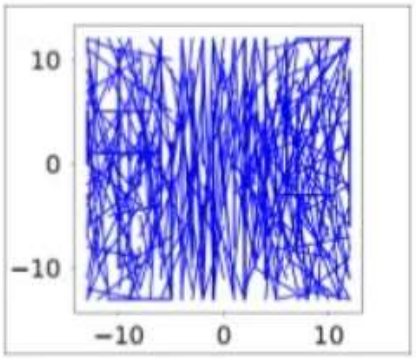
  BRIEF descriptor : (b0, b1, ..., b255)

  Each bit bi, is computed as:

  bi = 1 if P(xi,yi) > P(xi',yi'), else = 0

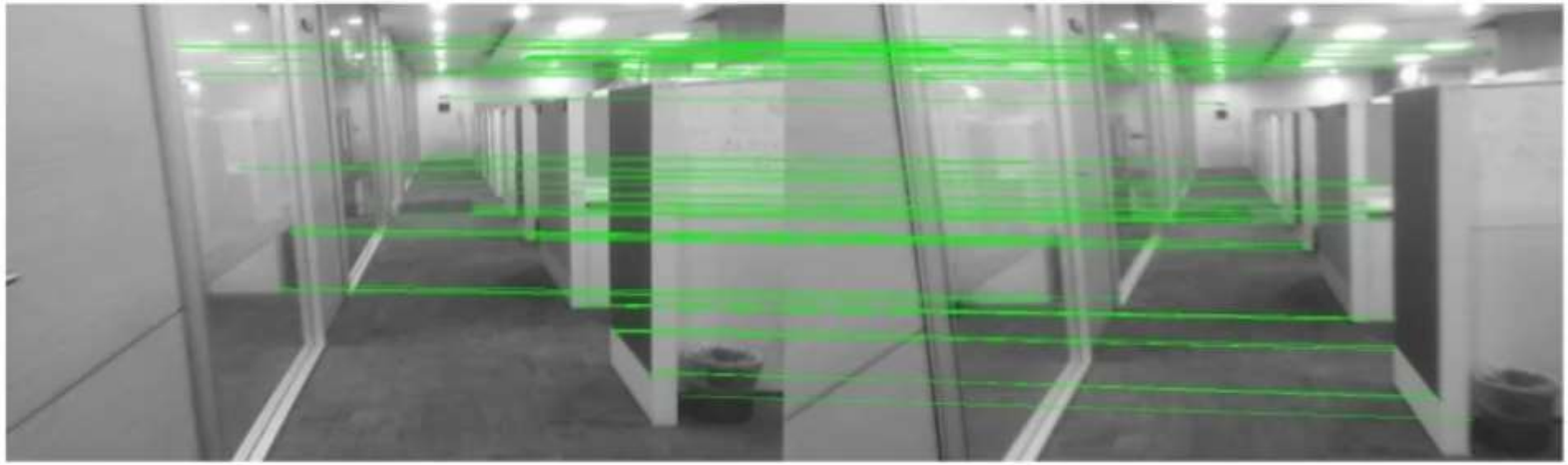
  (xi, yi), and (xi', yi') are predetermined test locations

- Test patterns are chosen empirically.
- Plot of brief pattern used by ORB SLAM.

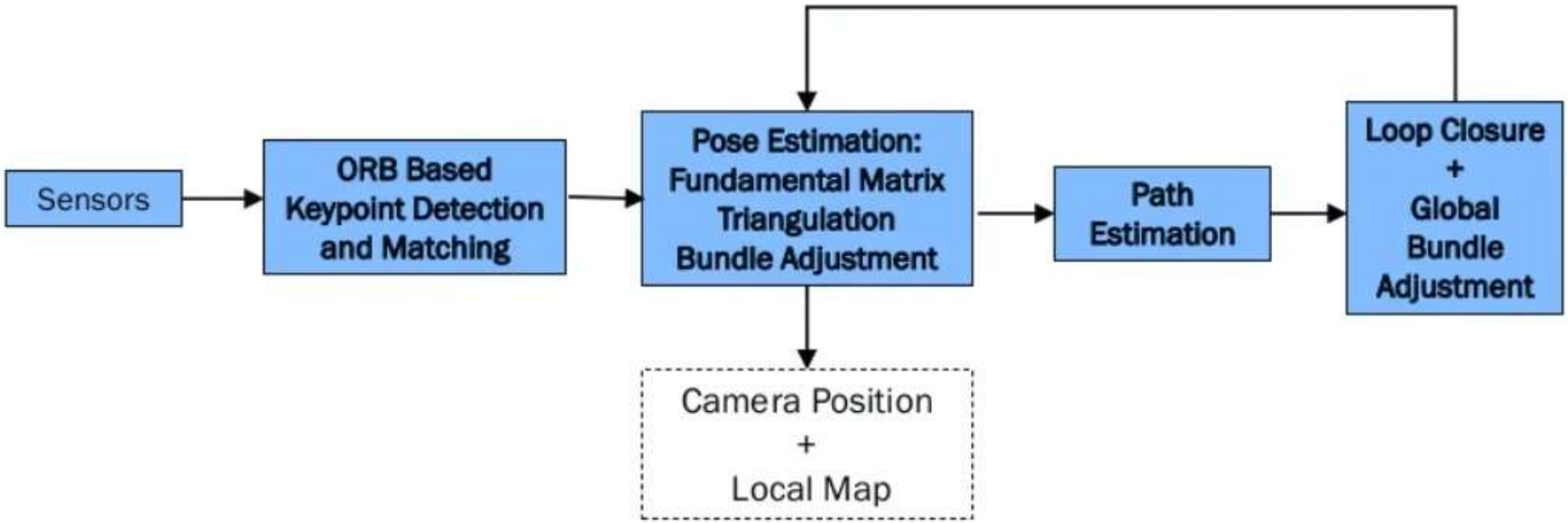


ORB6 : BRIEF Descriptor Matching

- The Hamming distance between two BRIEF descriptors is the number of positions at which the corresponding symbols are different.
- The BRIEF descriptor of each key point in the current frame is compared with the BRIEF descriptor of the corner points in the previous frame (within local vicinity)
- The two points with minimum Hamming distance are considered as matching.
- Heuristics used to eliminate incorrect matches



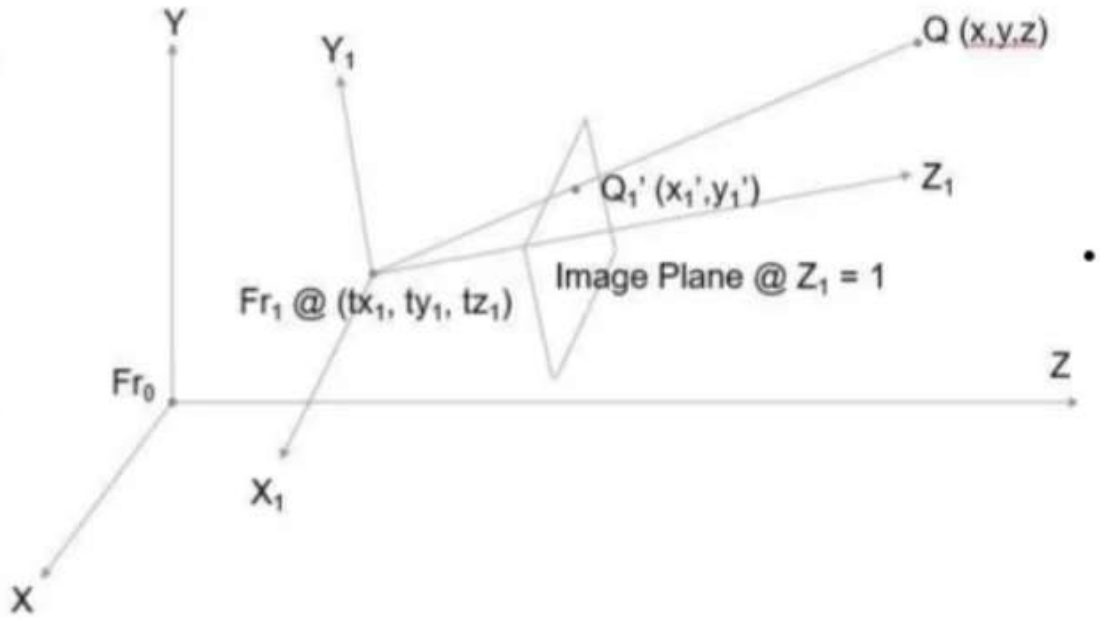
SLAM: Block Diagram



Part2: Mapping and Localization

- What have we achieved till now :
    - Located key points in the image sequence
    - Tracked the 2D coordinates of each key point in the image sequence
- We now analyze the 2D location data of the key point in the image sequence to estimate the 3D coordinates of each key point, and the 3D location and orientation of the camera (Pose of the Camera) in each frame.
- Topics Covered:
    - Basics of Projection
    - Relative Pose Estimation
    - Global Pose Estimation
    - Loop Closure

Basics of Projections



- Key Point Q(x, y, z) has coordinates (x_1, y_1, z_1) in Fr_1.
- Normalized Projected coordinates (x_1', y_1') of Q_1' on image plane located at $Z_1 = 1$
  - $x_1' = x_1/z_1$
  - $y_1' = y_1/z_1$
- Actual pixel coordinates (xp_1, yp_1) related to (x_1', y_1') via:
  - $xp_1 = fx * x_1' + cx$
  - $yp_1 = fy * y_1' + cy$
  - (fx, fy) : arbitrary location of Image Plane on Z_1, and different scaling along X_1 and Y_1
  - (cx, cy) is location of center w.r.t top left pixel

Basics of Projections -2

- To compute (x_1, y_1, z_1) from (x, y, z) – adjust for translation and rotation of frame Fr_1
- $$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix} * \begin{bmatrix} x - tx_1 \\ y - ty_1 \\ z - tz_1 \end{bmatrix}$$

- $\underline{x}_1 = R_1 * [\underline{x} - \underline{t}_1]$
  - R_1 = rotational matrix of Fr_1 w.r.t Fr_0, orthonormal
  - $\underline{t}_1$ = translation vector of frame Fr_1 w.r.t. Fr_0
  - $\underline{x}$, $\underline{x}_1$ = coordinate vectors of point Q in Fr_0 and Fr_1

- Similarly for Frames Fr_k and Fr_{k+1}:

$\underline{x}_k = R_k * (\underline{x} - \underline{t}_k)$

$\underline{x}_{k+1} = R_{k+1} * (\underline{x} - \underline{t}_{k+1})$

$\underline{x}_{k+1} = R_{k+1} * R_k^{-1} * \underline{x}_k - R_{k+1} * (\underline{t}_{k+1} - \underline{t}_k)$

- Let $R_{k+1,k} = R_{k+1} * R_k^{-1}$: relative rotation matrix of Fr_{k+1} w.r.t. Fr_k
- Let $\underline{h}_{k+1,k} = - R_{k+1} * (\underline{t}_{k+1} - \underline{t}_k)$: relative rotated translation vector of of Fr_{k+1} w.r.t. Fr_k

$\underline{x}_{k+1} = R_{k+1,k} * \underline{x}_k + \underline{h}_{k+1,k}$

Basics of Projections -3

- Expanding : $\underline{x}_{k+1} = R_{k+1,k} * \underline{x}_k + \underline{h}_{k+1,k}$ gives:
- $$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} r11_{k+1,k} & r12_{k+1,k} & r13_{k+1,k} \\ r21_{k+1,k} & r22_{k+1,k} & r23_{k+1,k} \\ r31_{k+1,k} & r32_{k+1,k} & r33_{k+1,k} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} + \begin{bmatrix} h14_{k+1,k} \\ h24_{k+1,k} \\ h34_{k+1,k} \end{bmatrix}$$
- We can rewrite this in terms of x_{k+1}', y_{k+1}', x_k', y_k' :
- $$[x'_{k+1} \quad y'_{k+1} \quad 1] * T_{K+1,k} * R_{K+1,k} * \begin{bmatrix} x'_k \\ y'_k \\ 1 \end{bmatrix} = 0$$
- Where, $T_{K+1,k} = \begin{bmatrix} 0 & -h34_{k+1,k} & h24_{k+1,k} \\ h34_{k+1,k} & 0 & -h14_{k+1,k} \\ -h24_{k+1,k} & h14_{k+1,k} & 0 \end{bmatrix}$

Basics of Projections -4

- This is the fundamental equation of Projective Geometry :

$$[x'_{k+1} \quad y'_{k+1} \quad 1] * F_{k+1,k} * \begin{bmatrix} x'_k \\ y'_k \\ 1 \end{bmatrix} = 0$$

- Where $F_{k+1,k} = T_{k+1,k} * R_{k+1,k}$ is known as the Fundamental or Essential matrix
- $R_{k+1,k}$ is orthonormal rotation matrix, and
- $T_{k+1,k}$ is skew-symmetric

Relative Pose Estimation -1

- Given set of matching Key Point image coordinates, (x'_k, y'_k) and (x'_{k+1}, y'_{k+1}), we want to estimate the Fundamental matrix : $F_{k+1,k}$
- $[x'_{k+1} \;\; y'_{k+1} \;\; 1] * F_{k+1,k} * \begin{bmatrix} x'_k \\ y'_k \\ 1 \end{bmatrix} = 0$
- $[x'_{k+1} \;\; y'_{k+1} \;\; 1] \begin{bmatrix} f_{11} f_{12} f_{13} \\ f_{21} f_{22} f_{23} \\ f_{31} f_{32} f_{33} \end{bmatrix} \begin{bmatrix} x'_k \\ y'_k \\ 1 \end{bmatrix} = 0$
- This represents a single linear equation in 9 unknowns for each set of matching Key-point coordinates
- Use RANSAC based least square estimation to solve for $\{f_{11}, f_{12}, f_{12}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}\}$
  - Need to set any one coefficient = 1, and solve for other eight
  - Any scaled version of the resulting coefficients is a valid $F_{k+1,k}$ matrix

Relative Pose Estimation -2

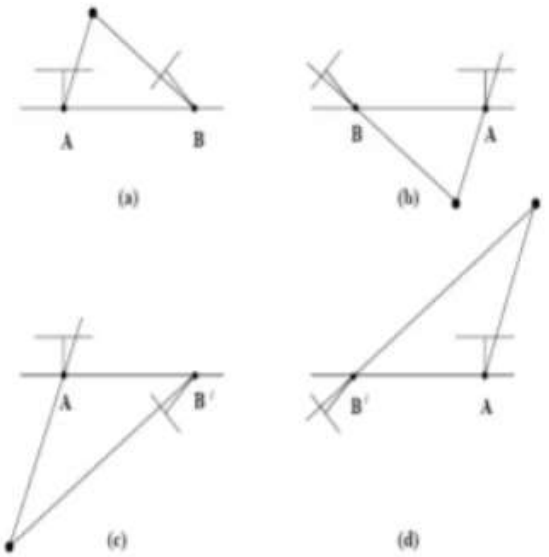
- Fundamental matrix F is a product of two matrices T and R
- $F = T * R = \begin{bmatrix} 0 & -h_{34} & h_{24} \\ h_{34} & 0 & -h_{14} \\ -h_{24} & h_{14} & 0 \end{bmatrix} \begin{bmatrix} r_{11} r_{12} r_{13} \\ r_{21} r_{22} r_{23} \\ r_{31} r_{32} r_{33} \end{bmatrix}$
  - T is a skew symmetric matrix and R is the orthogonal rotation matrix.
- To recover T:
- $(\text{Trace}\ (F * F^T)/2) * I - (F * F^T) = \begin{bmatrix} (h_{14}^2) & (h_{14} * h_{24}) & (h_{14} * h_{34}) \\ (h_{14} * h_{24}) & (h_{24}^2) & (h_{24} * h_{34}) \\ (h_{14} * h_{34}) & (h_{24} * h_{34}) & (h_{34}^2) \end{bmatrix}$
- (a) Compute the above matrix, (b) From square root of first diagonal element, we can recover h_{14} (c) Dividing row-1 entries with h_{14}, we get (h_{14}, h_{24}, h_{34}), (d) Another valid solution is $(-h_{14}, -h_{24}, -h_{34})$

Relative Pose Estimation -3

- r_1, r_2, r_3 are rows of F, or $F = [r_1^T\ r_2^T\ r_3^T]^T$
- $[\text{Cofactor}(F)]^T = [(r_2 \times r_3)^T, (r_3 \times r_1)^T, (r_1 \times r_2)^T]^T$
- Manipulating the matrices and using the fact that R is orthonormal, and T is skew-symmetric we can write:
- $[\text{Cofactor}(F)]^T - (T * F) = (h_{14}^2 + h_{24}^2 + h_{34}^2) * R$
- Since we have recovered two possible values for T, we get two possible values for the rotation matrix R : R_1 and R_2
- The rotation matrices R_1 and R_2 along with (h_{14},h_{24},h_{34}), and $(-h_{14},-h_{24},-h_{34})$ give four possible factorizations of the fundamental matrix F.
- Next we need to choose one of the four possible factorizations

Relative Pose Estimation -4

- Four possible factorizations of the fundamental matrix represent the four possible scenarios depicted here.
- A and B are the camera centers while the T shaped structure indicates the image plane and the normal to the image plane
- Only one of the 4 poses produce positive Z co-ordinates in both the frames.
- The factorization that gives maximum key points with positive values for (z_k, z_{k+1}) is selected.
- Hence we now have $R_{k+1,k}$ and $\underline{h}_{k+1,k} = [h_{14}, h_{24}, h_{34}]^T$
- Note: We have not recovered the magnitude of $\underline{h}_{k+1,k}$
  - Any scaled version of $\underline{h}_{k+1,k}$ is an equally valid solution
  - We set magnitude of $\underline{h}_{k+1,k} = 1$



Relative Pose Estimation -4 (Local Bundle Adjustment)

- Inputs:
  - Set of 3D points (x_k , y_k , z_k) for key points common to frame Fr_k and Fr_{k+1}
  - Measured 2D image co-ordinates in both frames (x_k', y_k') and (x_{k+1}', y_{k+1}')
  - Pose of Fr_{k+1} w.r.t. Fr_K : $R_{k+1,k}$ and $\underline{h}_{k+1,k}$
- Objective: Tune the 3D points and the Pose of Fr_{K+1} so as to minimize the total reprojection error
- $\sum (x_k/z_k - x_k')^2 + (y_k/z_k - y_k')^2 + \sum (x_{k+1}/z_{k+1} - x_{k+1}')^2 + (y_{k+1}/z_{k+1} - y_{k+1}')^2$
  - $x_{k+1} = r_{11} * x_k + r_{12} * y_k + r_{13} * z_k + h_{14}$
  - $y_{k+1} = r_{21} * x_k + r_{22} * y_k + r_{23} * z_k + h_{24}$
  - $z_{k+1} = r_{31} * x_k + r_{32} * y_k + r_{33} * z_k + h_{34}$
- Can be minimized via steepest descent coefficient tuning

Global Pose Estimation -1

- Problem: Fix the magnitude of $|\underline{h}_{1,0}| = 1$, and Estimate the magnitude of $\underline{h}_{k+1,k}$
- Consider Frames Fr_0, Fr_1 and Fr_2 and key points common to the three frames
- Let (x_0,y_0,z_0) be the 3D co-ordinates of a key point in Frame Fr_0 estimated by triangulation between frames Fr_0, and Fr_1
- Using pose of Fr_1 we can estimate the 3D coordinates of the key point in Fr_1 as:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} r_{11} r_{12} r_{13} \\ r_{21} r_{22} r_{23} \\ r_{31} r_{32} r_{33} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \begin{bmatrix} h_{14} \\ h_{24} \\ h_{34} \end{bmatrix}$$

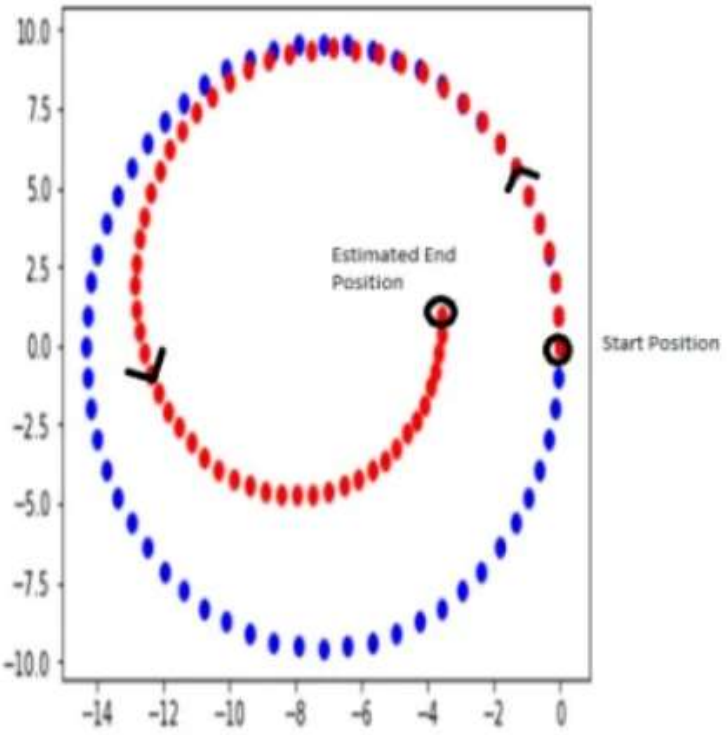
- Similarly we can triangulate frame Fr_1 and Fr_2 and estimate the 3D co-ordinates of the same key point in Fr_1 assuming $|\underline{h}_{2,1}| = 1$, as $[x^*_1, y^*_1, z^*_1]$
- From these the two 3D coordinate values, we get –

$$|\underline{h}_{2,1}| = (x_1 / x^*_1) = (y_1 / y^*_1) = (z_1 / z^*_1)$$

- We can proceed in a recursive manner and estimate magnitude of all $\underline{h}_{k+1,k}$

Global Pose Estimation -2

- We can also recursively estimate the camera rotation for each Frame:
  - $R_{k+1} = R_k * R_{k+1,k}$
- Relative translation vector between frame Fr_{k+1} and Fr_k is:
- $(\underline{t}_{k+1} - \underline{t}_k) = - R_{k+1}^{-1} * \underline{h}_{k+1,k}$
- Adding the relative translation vectors, we can estimate the 3D trajectory of the camera path
- An example of the estimated camera trajectory looks as shown here.
- The blue curve is the true trajectory while red curve is the estimated trajectory.

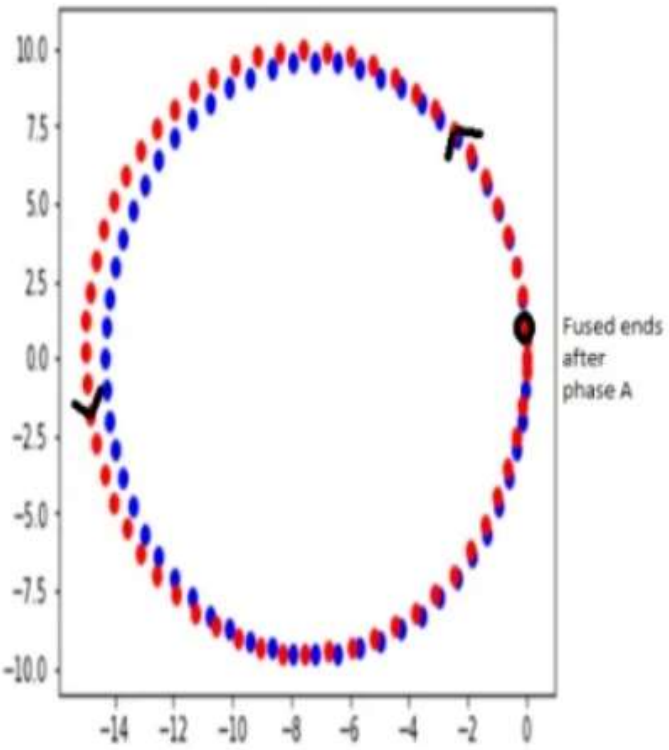


Loop closure

- As we continue to analyze key points in frames, we start encountering cases wherein the keypoints seen in the current keyframe are almost identical to the one seen in the starting keyframes.
- At this stage we say that loop closure is detected and start the loop closure tuning.
- The estimated trajectory and the true trajectory are significantly different due to the accumulated errors as evident in the picture on the previous slide.
- The objective of loop closure is thus to fuse the last keyframe position with the starting keyframe position by correcting the errors in the estimated pose for all the keyframes.

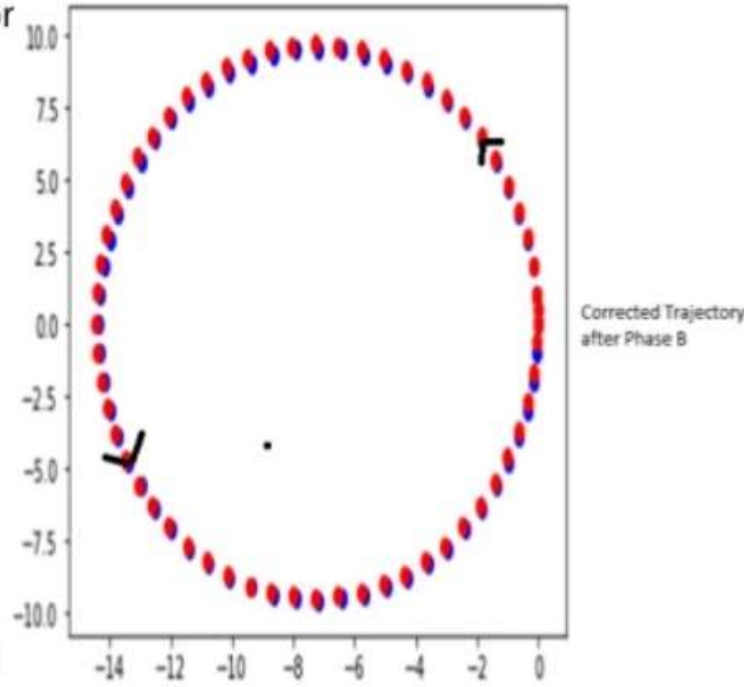
Loop closure – Phase A

- Fr_N and Fr_0 are the two frames which have almost same key points. The absolute pose of Fr_N with respect to Fr_0 is $H_{N,0}$
- Estimated fundamental matrix between frames Fr_N and Fr_0 is
- $$F_{N,0} = \begin{bmatrix} 0 & -h_{34} & h_{24} \\ h_{34} & 0 & -h_{14} \\ -h_{24} & h_{14} & 0 \end{bmatrix} \begin{bmatrix} r_{11} r_{12} r_{13} \\ r_{21} r_{22} r_{23} \\ r_{31} r_{32} r_{33} \end{bmatrix}$$
- The estimated fundamental matrix is derived from an erroneous pose and will thus show significant deviation from the fundamental matrix equation. Thus the error function is -
- $$E_{N,0} = \sum \left(\begin{bmatrix} x'_N & y'_N & 1 \end{bmatrix} * F_{N,0} * \begin{bmatrix} x'_0 \\ y'_0 \\ 1 \end{bmatrix} \right)^2$$
- Where (x'_N, y'_N) and (x'_0, y'_0) are the matching point 2D co-ordinates of key points common for Fr_0, and Fr_N.
- We minimize $E_{N,0}$ by tuning the estimated translation magnitudes as well as the rotation and translation for the pose of individual keyframes.



Loop closure – Phase B

- The cost function in phase A only corrected the relative pose between frames Fr_N and Fr_0 - without accounting for any discrepancies in the intermediate frames.
- Here we add some more components to the error function.
- Consider frames Fr_k and Fr_{k+2} and common keypoints between these two frames. We then define the error component
- $$E_{K,K+2} = \sum \left(\begin{bmatrix} x'_{K+2} & y'_{K+2} & 1 \end{bmatrix} * F_{K+2,K} * \begin{bmatrix} x'_K \\ y'_K \\ 1 \end{bmatrix} \right)^2$$
- The total cost function $C = \sum E_{K,K+2} + E_{N-1,0} + E_{N,1} + E_{N,0}$
- The objective is thus to minimize the cost function C by tuning the translation magnitudes $| \underline{h}_{k,k+1} |$ and also the translation and rotation coefficients.
- The pose tuning in Loop Closure – Phase A, and Phase B is also known to as Global Pose Optimization



Conclusion

Summary and Implementation Considerations:

- ORB based key point detection and matching
    - Typically fixed point implementation in a mix of 16b and 32b precision
    - Tends to be significant contributor to computational load
- All subsequent pose estimation, tuning, loop closure, 3D key point location estimation, bundle adjustment need floating point arithmetic
    - Single precision implementation possible

References

[1] Andrew Zisserman Richard Hartley. *Multiple View Geometry*. 2003.

[2] Raul Mur-Artal, J. M. M. Montiel and Juan D. Tardos , *"ORB-SLAM: A Versatile and Accurate Monocular SLAM System"*

[3] Raul Mur-Artal, Juan D. Tardos , *"ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras "*

[4] Berthold K.P. Horn , "Recovering Baseline and Orientation from 'Essential' Matrix "

Acknowledgement: **"Fundamentals of Monocular SLAM," a Presentation from Cadence**

To continue…