ORB-SLAM: a Versatile and Accurate Monocular SLAM System

Presenter: Sudhanshu Bahety

Raul Mur-Artak, J.M.M. Montiel, and Jaun D. Tardos

What is SLAM?

- Computational problem of constructing and updating map of an environment
- Keeping track of agent's location
- Given a series of observations O_t, estimate the agent's location x_t and a map of the environment m_t.
- In terms of probability, $P(m_t, x_t \mid O_t)$

3 Major SLAM Tasks

- **Tracking:**
  - Estimate the pose of the camera
  - Decide when to insert new keyframe
- **Mapping:**
  - Updating Map points and Keyframes
- **Loop Detection:**
  - Candidate Keyframe selection for loops
  - Loop Fusion
  - Optimize Essential Graph

Pre-requisite

- Bundle Adjustment
- ORB Feature Descriptor
- FAST corner Detector
- Bag of Words Place Recognition

Pre-requisite: Bundle Adjustment

Map point 3D locations $\mathbf{X}_{w,j} \in \mathbb{R}^3$

Keyframe Poses / Camera Poses $\mathbf{T}_{iw} \in \mathrm{SE}(3)$

Matched keypoints $\mathbf{x}_{i,j} \in \mathbb{R}^2$

Error Term for observing map point j in Keyframe i

$$\mathbf{e}_{i,j} = \mathbf{x}_{i,j} - \pi_i(\mathbf{T}_{iw}, \mathbf{X}_{w,j})$$

Cost Function to be minimized

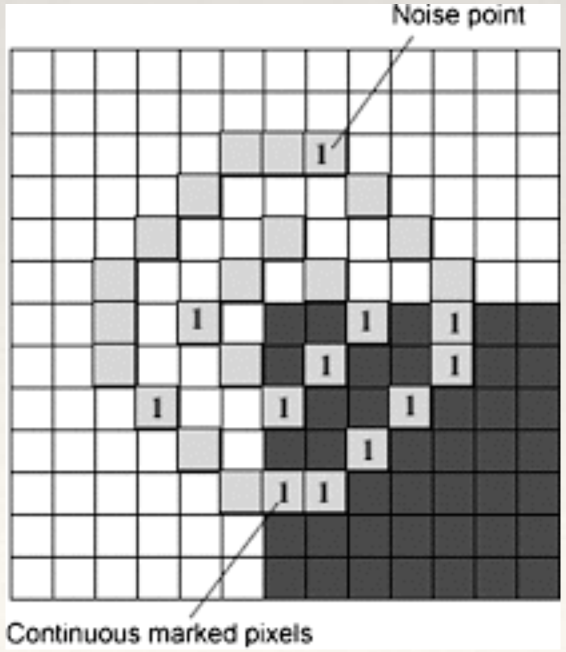
$$C = \sum_{i,j} \rho_h(\mathbf{e}_{i,j}^T \mathbf{\Omega}_{i,j}^{-1} \mathbf{e}_{i,j})$$

Pre-requisite: Bundle Adjustment

- **Full BA:**
  - Optimize for all map points and Keyframes (except first frame)
- **Local BA:**
  - Map points are optimized
  - Camera pose fixed
- **Motion-only BA:**
  - Map points are fixed
  - Camera pose optimized

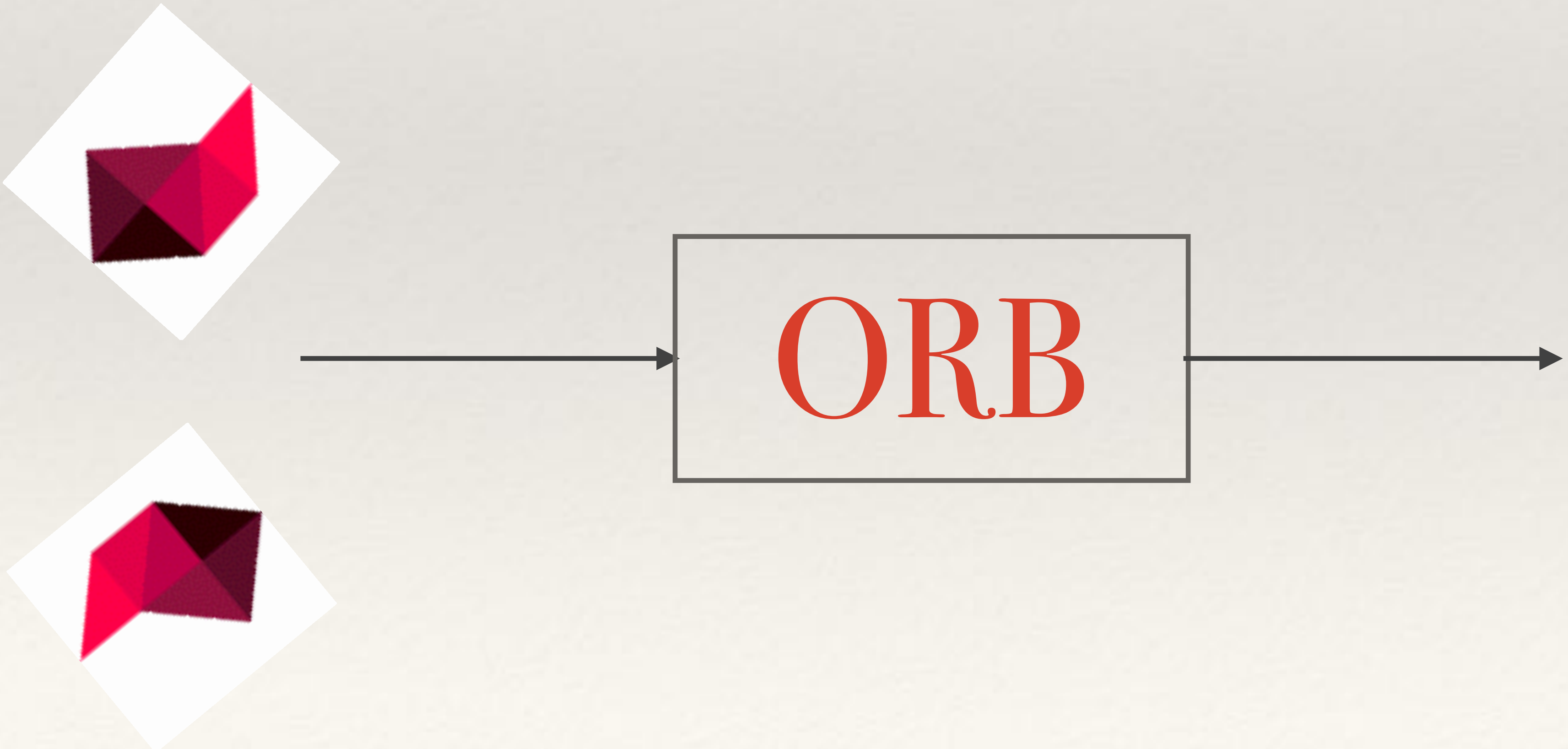
Pre-requisite: FAST Corner Detector

- Uses Circle of 16 pixel
- **Condition:**
    - N Contiguous pixel are brighter than intensity of candidate pixel p
    - N Contiguous pixel are dimmer than intensity of candidate pixel p
- P is classified as a corner



Pre-requisite: ORB Feature Descriptor

- Binary Feature Descriptor
- **Input**: Image patch, **Output**: Binary string
- Orientation Invariance Feature Descriptor

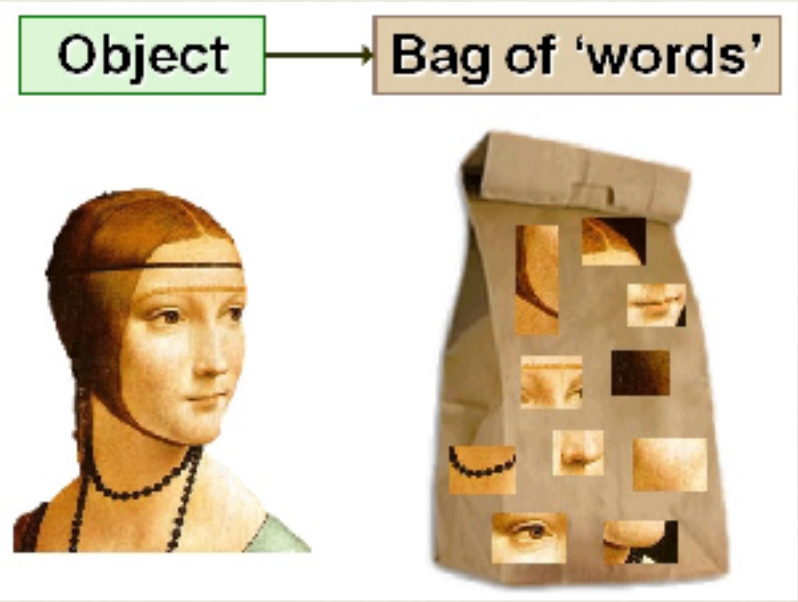


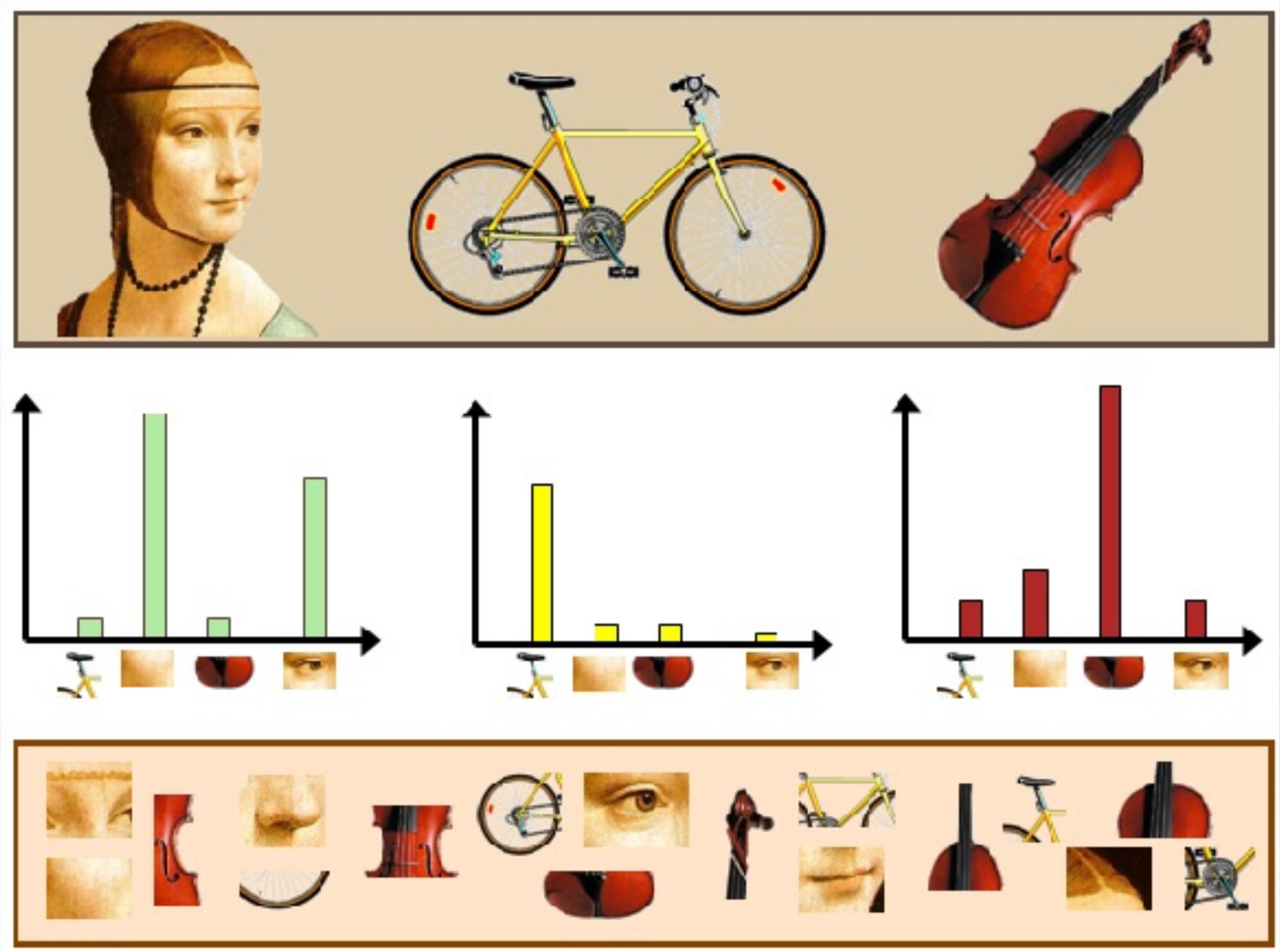
- 010110101
- 010010101

FAST Detector + ORB Descriptor

- Feature generated using this method used for all 3 tasks - Tracking, Mapping, Loop Detection
- Very fast computation of features
- Rotation Invariance

Pre-requisite: Bag of Words Place Recognition





Bag of words – representing object as histograms of words occurrences

Pre-requisite: Bag of Words Place Recognition

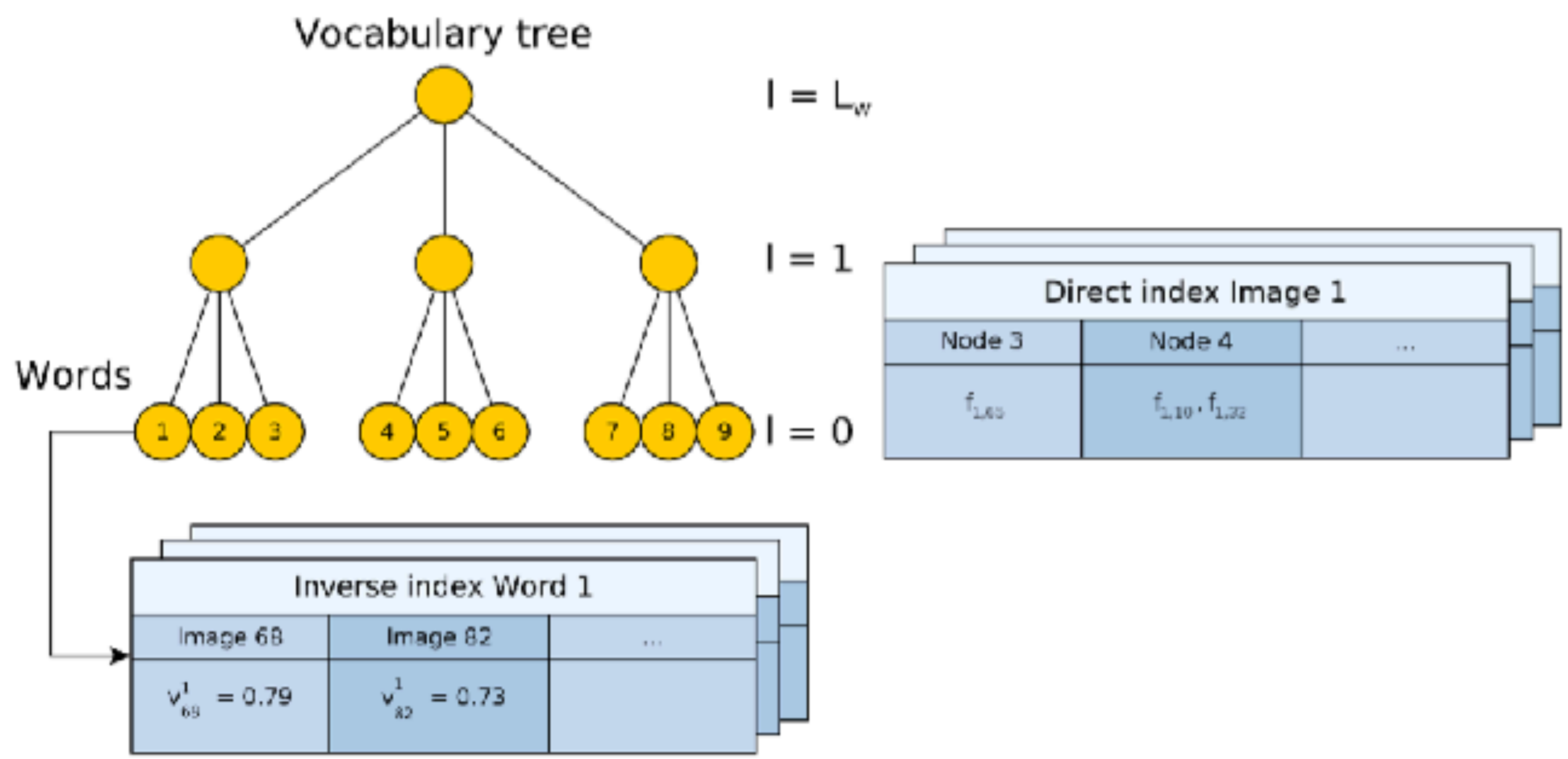


Fig. 1. Example of vocabulary tree and direct and inverse indexes that compose the image database. The vocabulary words are the leaf nodes of the tree. The inverse index stores the weight of the words in the images in which they appear. The direct index stores the features of the images and their associated nodes at a certain level of the vocabulary tree.

Some Definitions

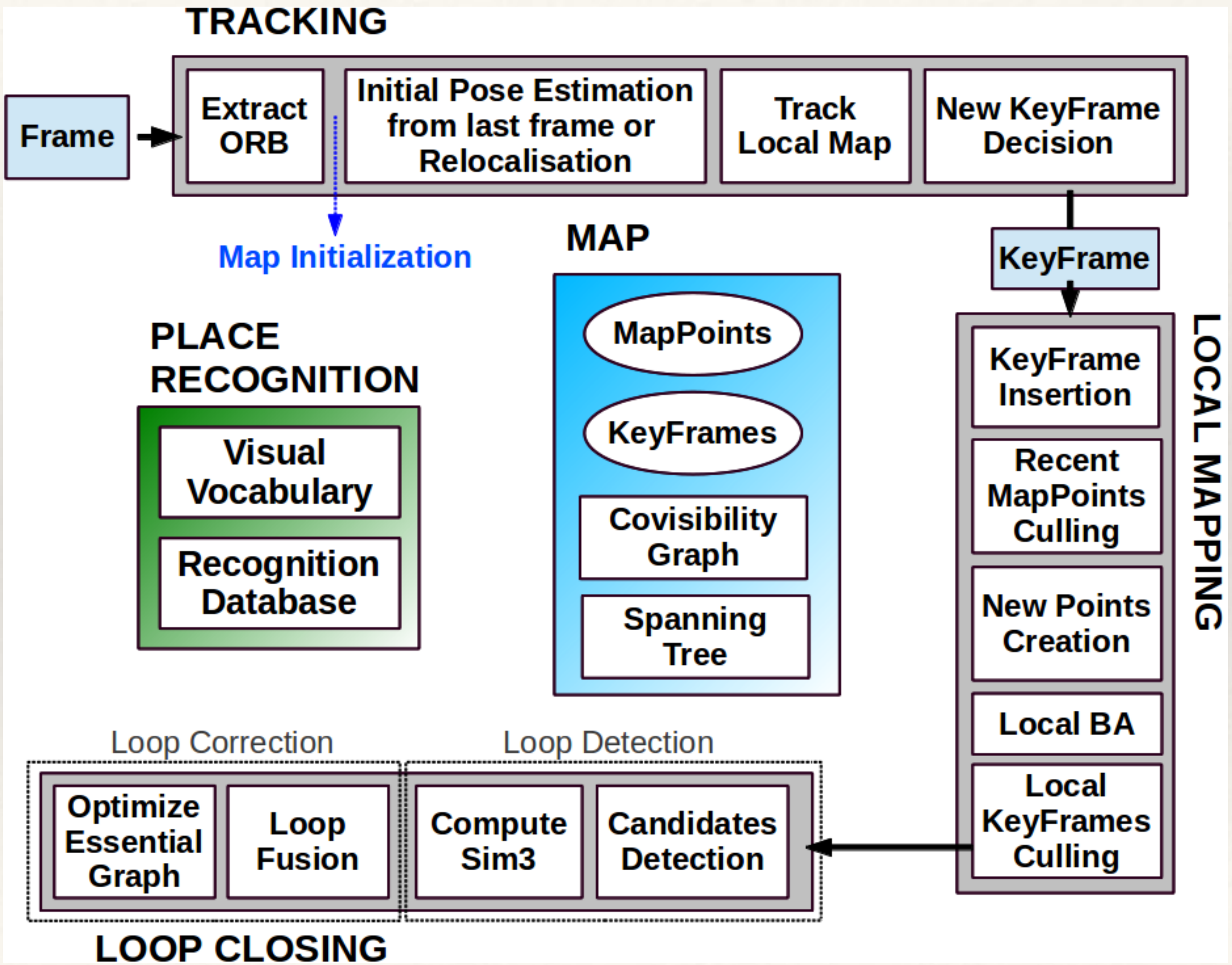
- **Keyframes:** An image stored within the system that contains informational cues for localization and tracking
- **Map points:** A point in 3D space that is associated with 1 or more keyframes
- **Covisibility Graph:** A graph consisting of a Keyframe as a node and edge between Keyframe exists if they share at least 15 common map points
- **Essential Graph:** A subgraph of covisibility graph(contains all the nodes) that has at least 100 common map points.

Some Definitions

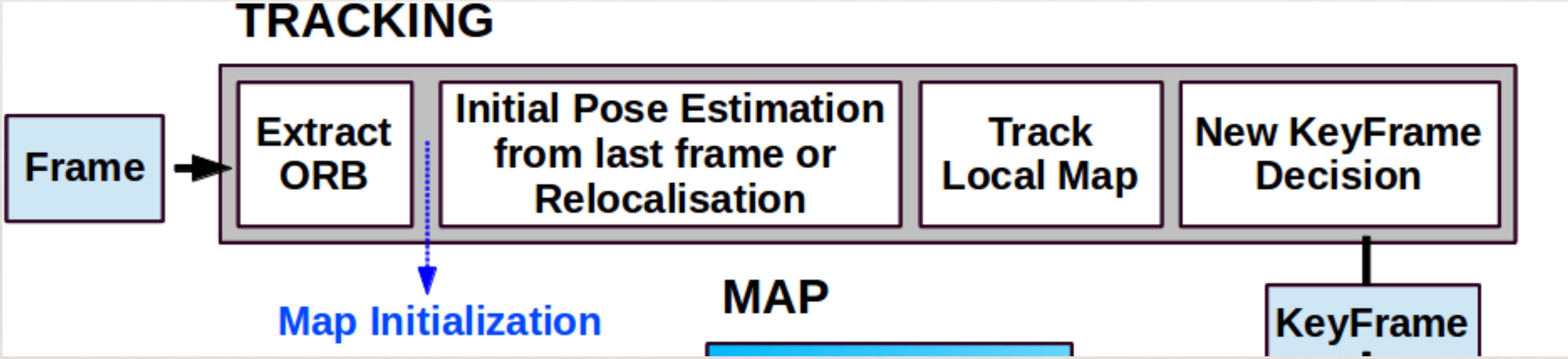
- **Keyframe K_i** stores the following information:
  - $\mathbf{T}_{iw}$ —> transforms point from WC to CC system
  - The camera intrinsics, including focal length and principal point.
  - All the ORB features extracted in the frame
- **Map Point p_i** stores the following information:
  - $\mathbf{X}_{w,i}$ —>3D position in WC system
  - $\mathbf{n}_i$ —> Viewing direction
  - $\mathbf{D}_i$ —> ORB descriptor
  - $d_{\min}$, $d_{\max}$ —> max and min distance where observed

Overall Flow

- All phases are done in 3 different thread



Overall Flow



Tracking

- The tracking is in charge of **localizing the camera** with every frame and deciding when to **insert a new keyframe**.
- **Part 1. ORB Extraction:**
  - Extract FAST corners at 8 scale levels with a scale factor of 1.2.
  - 512 × 384 to 752 × 480 pixels —> 1000 corners
  - 1241 × 376 —> 2000 corners
  - Each scale level divided in a grid
  - 5 corners per cell (adapting value of N, threshold to obtain corners)

Tracking

- **Part 2. Automatic Map Initialization (If not done):**
  - **Why?** Depth cannot be recovered from a single image, define global co-ordinates.
  - Scene independent
  - Two Models proposed:
    - Homography - Assuming planar scene
    - Fundamental Matrix - Assuming non-planar scene

Tracking (2. Map initialization)

- **Step 1: Find Initial Correspondence**
  - ORB features extracted in current frame only.
  - Find Correspondences.
  - Not Enough Correspondences? Reset Reference frame.

Tracking (2. Map initialization)

- **Step 2: Compute Homography and Fundamental Matrix**

$$\mathbf{x}_c = \mathbf{H}_{cr}\,\mathbf{x}_r \qquad\qquad \mathbf{x}_c^T\,\mathbf{F}_{cr}\,\mathbf{x}_r = 0$$

  - DLT —> Homography Matrix
  - 8 point algorithm —> Fundamental Matrix
  - RANSAC scheme used
  - Fundamental
  - Measure symmetric transfer error

$$S_M = \sum_i \left(\rho_M\left(d_{cr}^2(\mathbf{x}_c^i, \mathbf{x}_r^i, M)\right) + \rho_M\left(d_{rc}^2(\mathbf{x}_c^i, \mathbf{x}_r^i, M)\right)\right)$$

$$\rho_M(d^2) = \begin{cases} \Gamma - d^2 & \text{if} \quad d^2 < T_M \\ 0 & \text{if} \quad d^2 \geq T_M \end{cases}$$

Tracking (2. Map initialization)

- **Step 2: Compute Homography and Fundamental Matrix(Contd)**
  - T_M is outlier rejection threshold (based on chi-square at 95%)
  - T_H = 5.99 (Homography Matrix)
  - T_F = 3.84 (Fundamental Matrix)
  - Gamma = T_H so that both models score equally
  - Keep F, H with highest score, discard all others.
  - If none found, restart **Step1**.

$$S_M = \sum_i \left(\rho_M \left(d_{cr}^2 (\mathbf{x}_c^i, \mathbf{x}_r^i, M) \right) + \rho_M \left(d_{rc}^2 (\mathbf{x}_c^i, \mathbf{x}_r^i, M) \right) \right)$$

$$\rho_M(d^2) = \begin{cases} \Gamma - d^2 & \text{if} \quad d^2 < T_M \\ 0 & \text{if} \quad d^2 \geq T_M \end{cases}$$

Tracking (2. Map initialization)

- **Step 3: Model selection**
  - Planar scene —> Homography should be used
  - Non-planar scene —> Fundamental should be used

$$R_H = \frac{S_H}{S_H + S_F}$$

  - Select Homography if $R_H > 0.45$

Tracking (2. Map initialization)

- **Step 4: Motion and Structure from Motion recovery**
  - Camera Pose computed from Homography using 8 motion hypothesis
  - Triangulate all 8 solutions and select if:
    - most points seen with parallax
    - In front of both cameras
    - Low reprojection error
    - ELSE go back to **Step 1**
  - For Fundamental matrix, convert it to Essential matrix and use SVD to retrieve 4 motion hypothesis

$$\mathbf{E}_{rc} = \mathbf{K}^T \mathbf{F}_{rc} \mathbf{K}$$

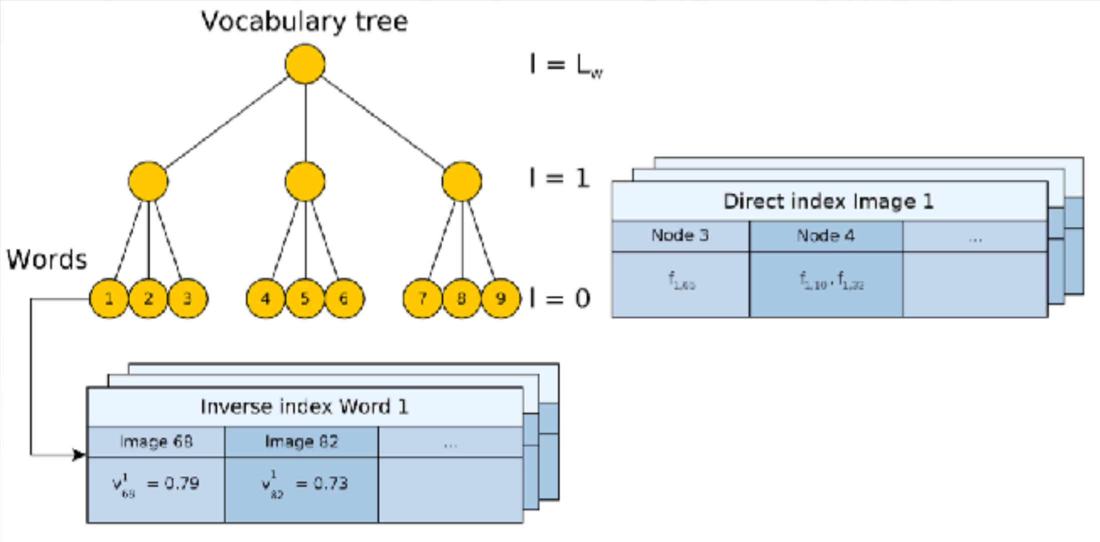
Tracking (3.Camera Pose Estimation)

- **Part 3a: Initial Pose Estimation from Previous Frame**
- Tracking is not lost
- Use “constant velocity motion model” to estimate camera pose
- If model is violated, use wider search of map points.

Tracking (3. Camera Pose Estimation)

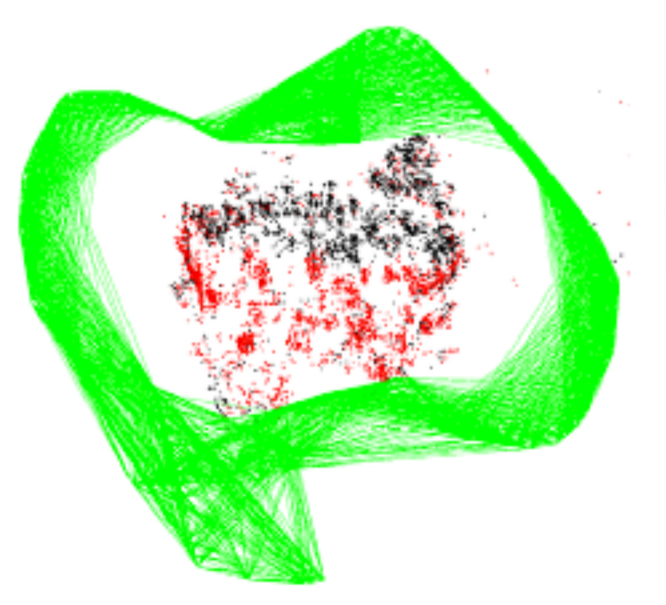
- **Part 3b: Initial Pose Estimation via Global Relocalization**

- Tracking is lost
- Keyframe —> Bag of Words
- Query database for candidate Keyframe
- Compute correspondences to map points in Keyframe
- Perform RANSAC to compute the camera pose
- Optimization is done if enough inliers are found



Tracking (4. Track Local Map)

- Search for more map point correspondences
- K1 —> Set of Keyframes that share map points with the current frame **(indirect indexing)**.
- K2 —> Set of Keyframes that are neighbors to K1 in covisibility graph.



(b) Covisibility Graph

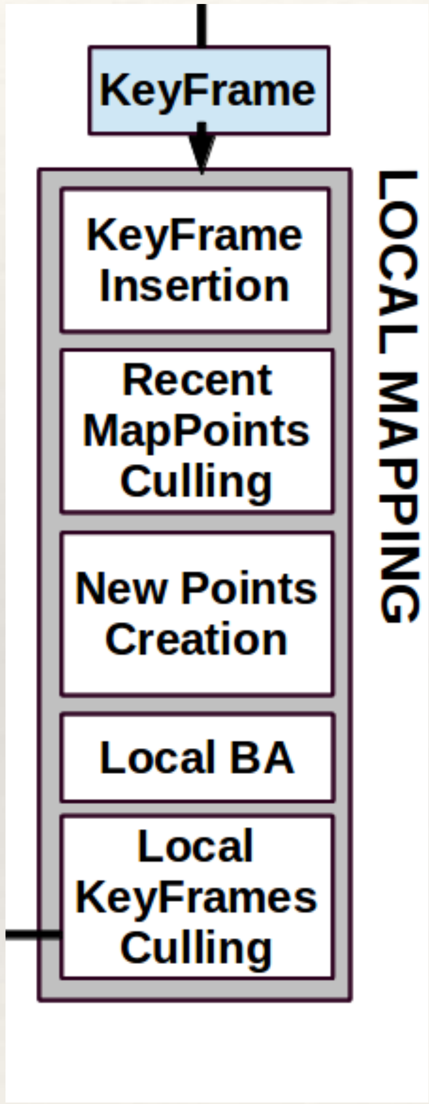
Tracking (4. Track Local Map Contd..)

- For each map point in K_1 and K_2:
  - Compute projection **x** in the current frame and discard if lays out of image bounds.
  - Discard if v.n < cos (60)   v—> current viewing ray
  - Discard if d < d_min and d > d_max **d—>** distance from camera center
  - Compare descriptor with ORB features near **x,** and associate map point with the best match
  - Optimize the camera pose after all the matches are found

Tracking (5. New Keyframe Decision)

- To insert new Keyframe:
  - > 20 frames must've past from last global relocalization
  - > 20 frames have past since last keyframe insertion
  - At least 50 keypoints are tracked by the current frame
  - < 90 % of the points present in reference frame are tracked

Overall Flow



Local Mapping (1. Keyframe Insertion)

- On inserting new Keyframe:
  - Update covisibility graph
  - Update spanning tree
  - Update Bag of words representation of the keyframe

Local Mapping (2. Recent Map Points Culling)

- In order to retain recent map points:
  - Present >25% of the frames on which it is predicted to be visible
  - Must be observed in at least 3 consecutive keyframes when it was created first
- If above tests are passed, it can be removed only when 2nd point is violated.

Local Mapping (3. New Map point creation)

- Find: K_c —> set of Keyframes connected to K_i
- For each unmatched ORB in K_i:
  - Search for other unmatched point in other keyframe
  - Discard if do not satisfy epipolar constraint.
  - ORB pairs are triangulated, and to accept the new points, positive depth in both cameras, parallax, reprojection error and scale consistency are checked.

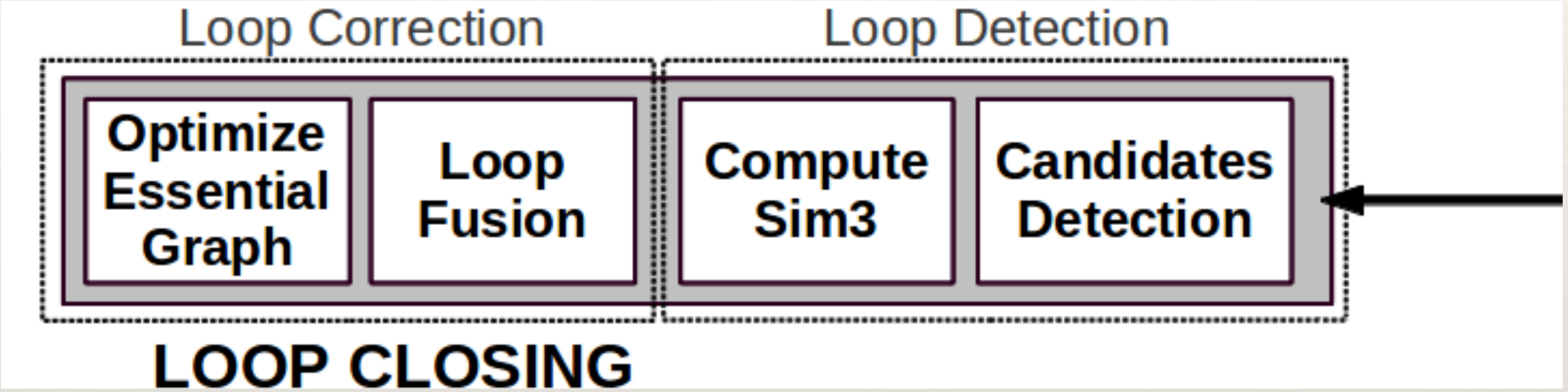
Local Mapping (3. Local BA)

- K_i, K_c are fixed.
- The new points obtained from ORB triangulation are optimized
- Outliers are removed in the middle and the end of the optimization

Local Mapping (3. Keyframe Culling)

- Detect redundant Keyframes and delete them
- Useful during BA
- Discards all Keyframes in K_c whose 90% map points are visible in 3 other keyframes

Overall Flow



Loop Closing (1. Loop Candidate Detection)

- Compute the similarity between K_i and all its neighbors in the covisibility graph (θ_min = 30)
- retain the lowest score s_min
- Query place recognition database and discard all those keyframes with scores less than s_min
- K_c are discarded from the results
- Consecutively 3 loop candidates that are consistent (keyframes connected in the covisibility graph)

Loop Closing (2. Compute Similarity Transformation)

- 3 Rotation, 3 Transformation, 1 Scale —> 7 DOF
- Compute a similarity transformation from current keyframe K_i to the loop keyframe K_l
- Compute ORB correspondences
- Perform RANSAC with each candidate K_l and find similarity matrix using method of Horn[42].
- If S_il is found with enough outliers, optimize it further
- Accept the loop with K_l with enough inliers are found even after optimization

Loop Closing (3. Loop Fusion)

- Fuse duplicated map points
- Insert new edges in the covisibility graph
- K_i pose T_iw is corrected with the similarity transformation S_il.
- Correction is propagated to all the neighbors of K_i
- Map points in K_l and K_c are projected into K_i and matches are searched.
- Map points are fused if they were inlier in S_il computaion.
- All keyframes involved in the fusion will update their edges in the covisibility graph.

Loop Closing (4. Essential Graph Optimization)

- Perform pose graph optimization over the Essential graph
- Distributes the loop closing error along the graph
- Optimization performed over similarity transformation to correct the scale drift