The Multi-State Constraint Kalman Filter

- A Filtering Approach for Visual-Inertial Navigation -

Mitchell Cohen

Department of Mechanical Engineering, McGill University

January 29th, 2024

Simultaneous Localization and Mapping (SLAM)

What SLAM answers…

Where am I?

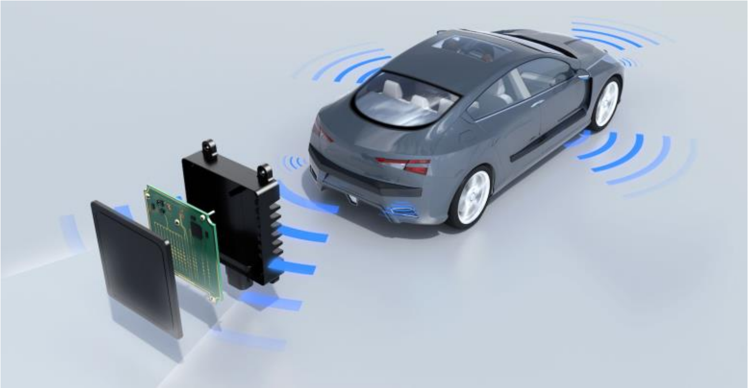
What does the world look like?

What SLAM enables…

Where should I go?

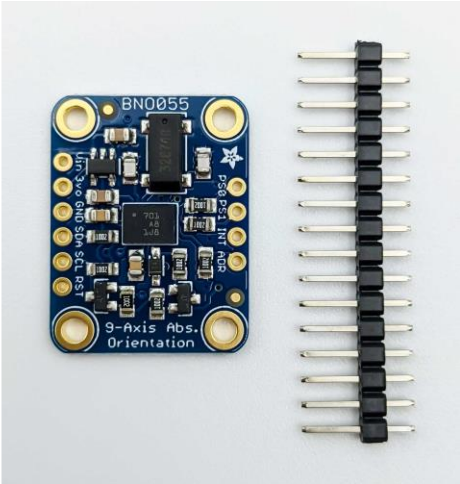
Sensors allow robots to localize and perceive the environment.

LiDAR



Radar

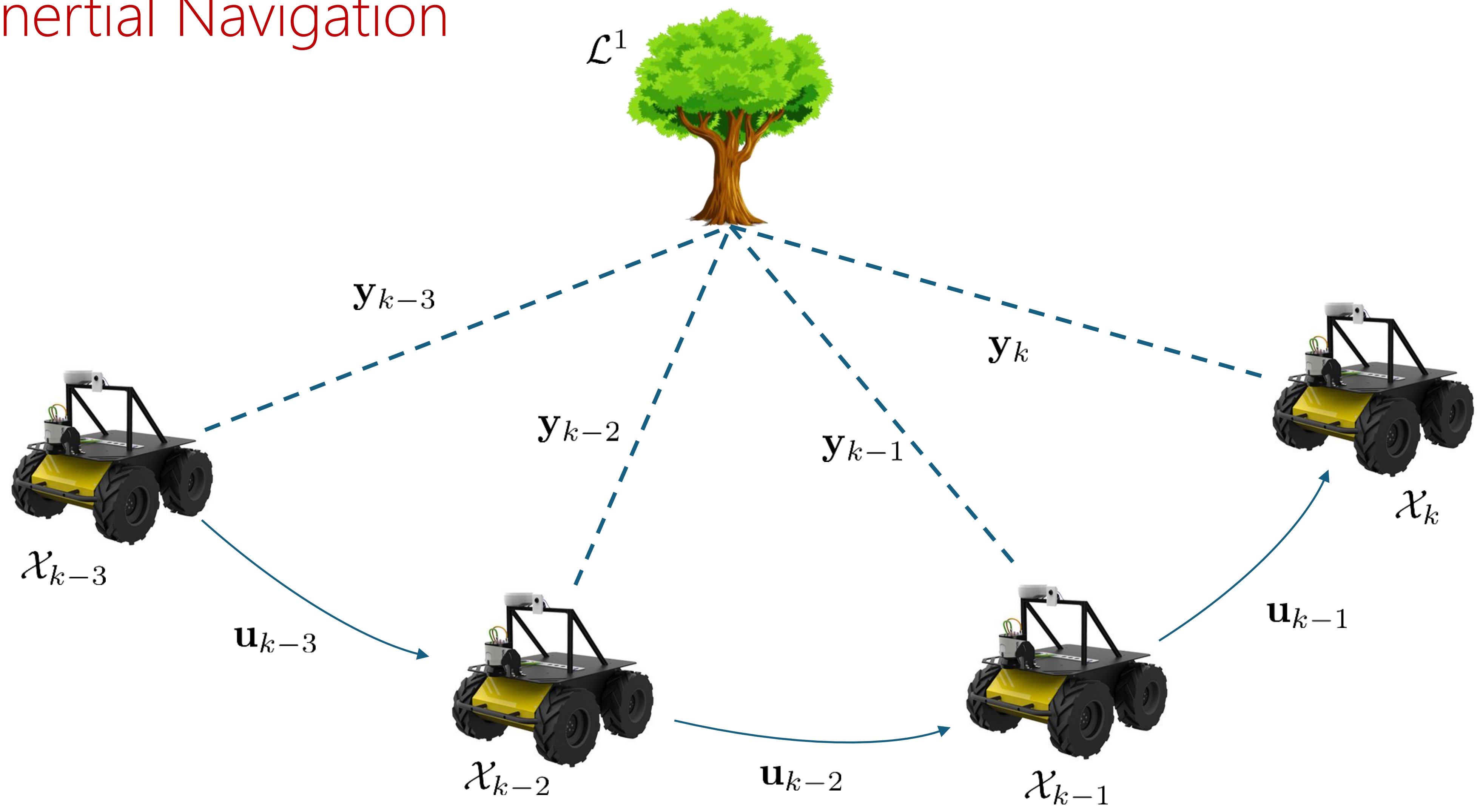
Visual-Inertial



Inertial measurement unit (IMU)

Cameras

Visual-Inertial Navigation



Goal: Infer the robot state $\mathcal{X}_{k-3:k}$ and landmark position $\mathcal{L}^1$ given:

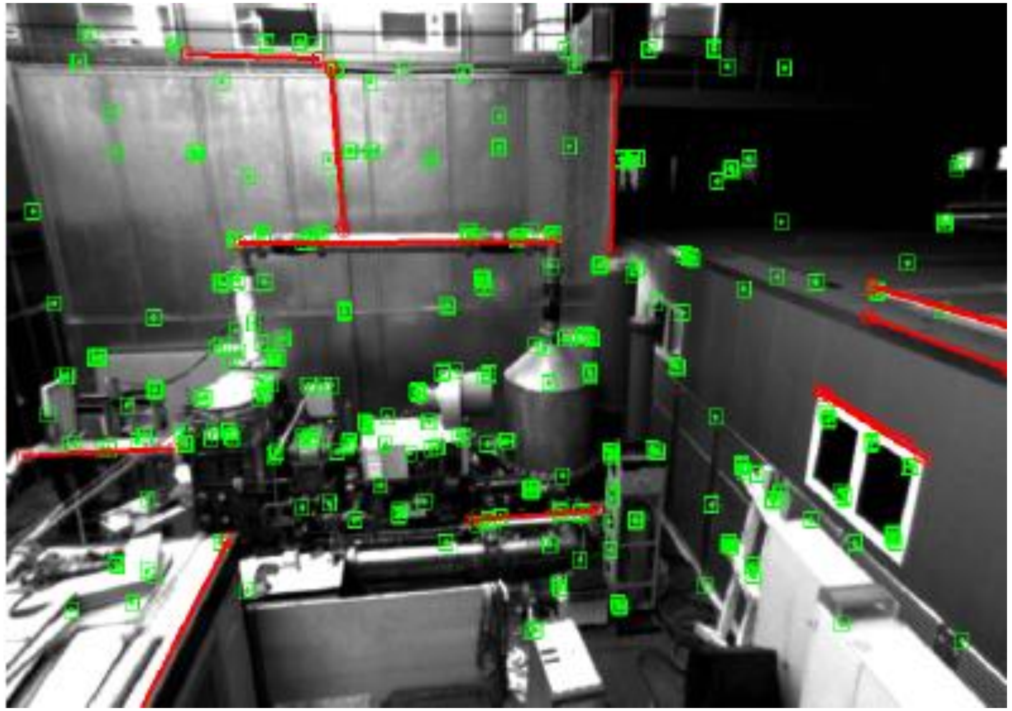
- inertial measurements $\mathbf{u}_{k-3:k-1}$,
- visual feature measurements $\mathbf{y}_{k-3:k}$.

McGill UNIVERSITY

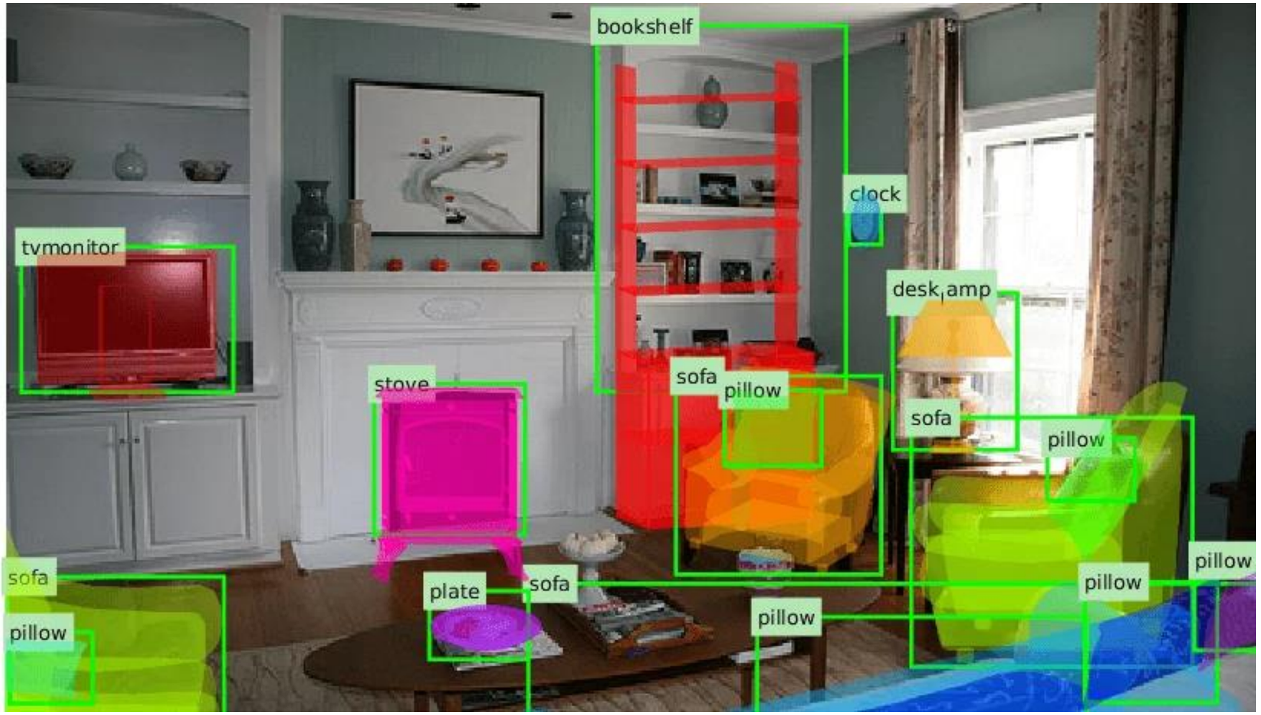
What do we mean by features?

Features: distinctive visual information in the environment.

- Points, lines, planes, objects…



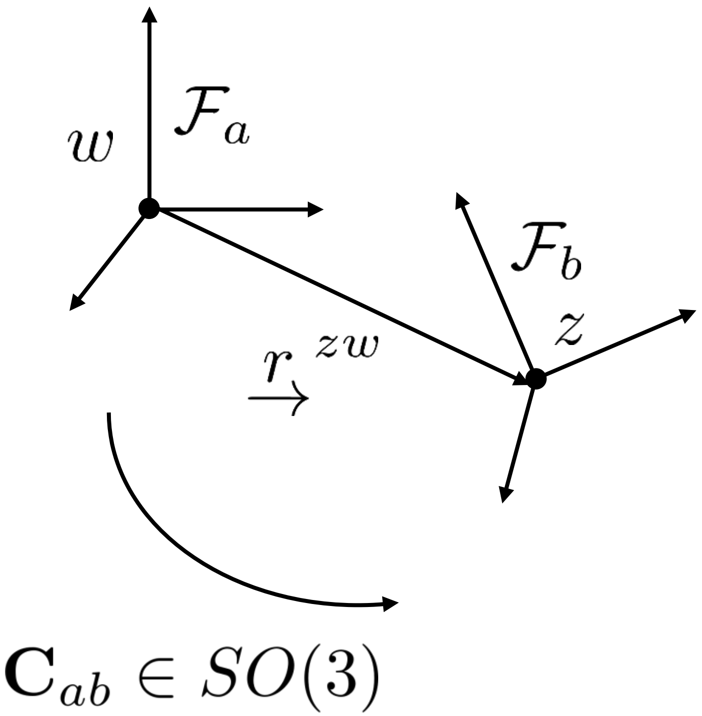
Points and lines extracted from image



Objects extracted from image

IMU Measurements

- IMUs consist of a **rate gyro** and an **accelerometer** and measure the acceleration and angular velocity of a vehicle, with additive bias and noise.



True angular velocity

Gyro: $\mathbf{u}^g_{b_k} = \boldsymbol{\omega}^{b_k a}_{b_k} + \mathbf{b}^g_{b_k} + \mathbf{w}^g_{b_k}$

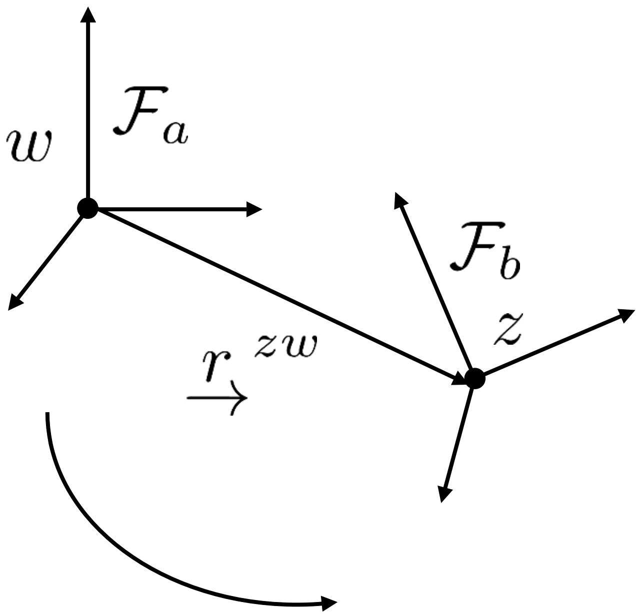
Accelerometer: $\mathbf{u}^a_{b_k} = \mathbf{C}^{\mathsf{T}}_{ab_k}\left(\mathbf{a}^{z_k w/a/a}_a - \mathbf{g}_a\right) + \mathbf{b}^a_{b_k} + \mathbf{w}^a_{b_k}$

True acceleration

The IMU Kinematics

- The IMU state is the **orientation**, **velocity** and **position** of the IMU, and the IMU **biases**.

$$\mathcal{X}^{\mathrm{IMU}} = \left(\mathbf{C}_{ab}, \mathbf{v}_a^{zw/a}, \mathbf{r}_a^{zw}, \mathbf{b}_b^g, \mathbf{b}_b^a \right).$$



$$\mathbf{C}_{ab} \in SO(3)$$

Continuous-Time Process Model

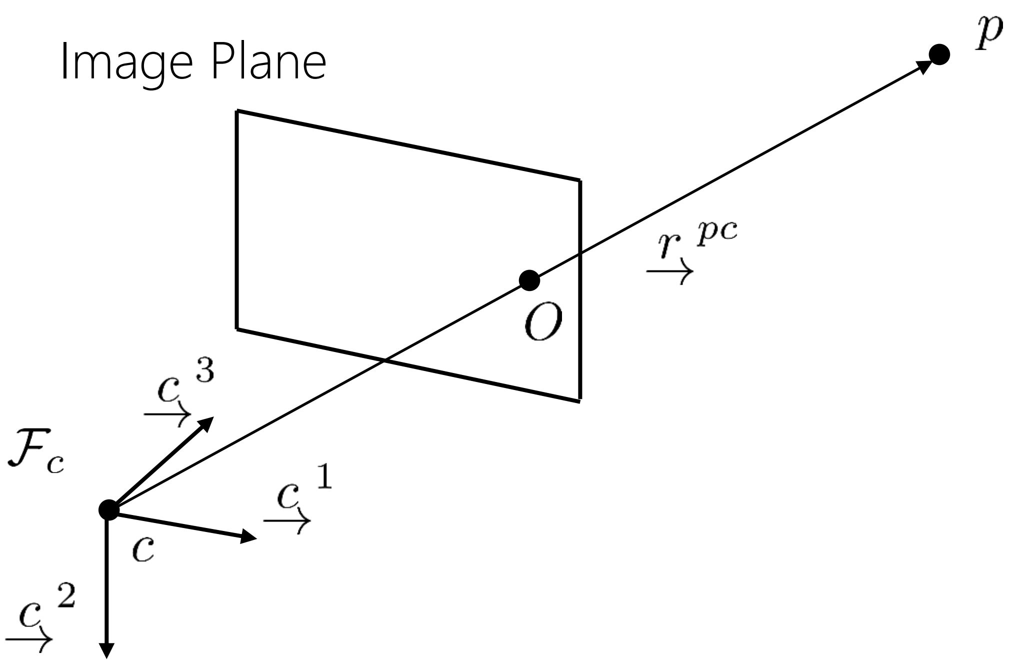
$$\begin{aligned}
\dot{\mathbf{C}}_{ab} &= \mathbf{C}_{ab} \left(\mathbf{u}_b^g - \mathbf{b}_b^g - \mathbf{w}_b^g \right)^{\times}, \\
\dot{\mathbf{v}}_a^{zw/a} &= \mathbf{C}_{ab} \left(\mathbf{u}_b^a - \mathbf{b}_b^a - \mathbf{w}_b^a \right) + \mathbf{g}_a, \\
\dot{\mathbf{r}}_a^{zw} &= \mathbf{v}_a^{zw/a}, \\
\dot{\mathbf{b}}_b^g &= \mathbf{w}_b^{b_g}, \\
\dot{\mathbf{b}}_b^a &= \mathbf{w}_b^{b_a}.
\end{aligned}$$

Discrete-Time Process Model

$$\begin{aligned}
&\mathcal{X}_k^{\mathrm{IMU}} = f_{k-1}\left(\mathcal{X}_{k-1}^{\mathrm{IMU}}, \mathbf{u}_{k-1} \right) \oplus \mathbf{w}_{k-1}, \\
&\mathbf{w}_{k-1} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_{k-1} \right)
\end{aligned}$$

Visual Feature Measurements

- Cameras model a mapping between the 3D world and a 2D image.
- Commonly modelled as **central projection** – project points in 3D space onto a plane called the image plane.



$$\mathbf{r}_c^{pc} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}^\mathsf{T} \in \mathbb{R}^3$$

Pinhole Camera Model

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \mathbf{g}\left(\mathbf{r}_c^{pc}\right) = \begin{bmatrix} r_1/r_3 & r_2/r_3 \end{bmatrix}.$$

Normalized image coordinates of observation

Visual Feature Measurements

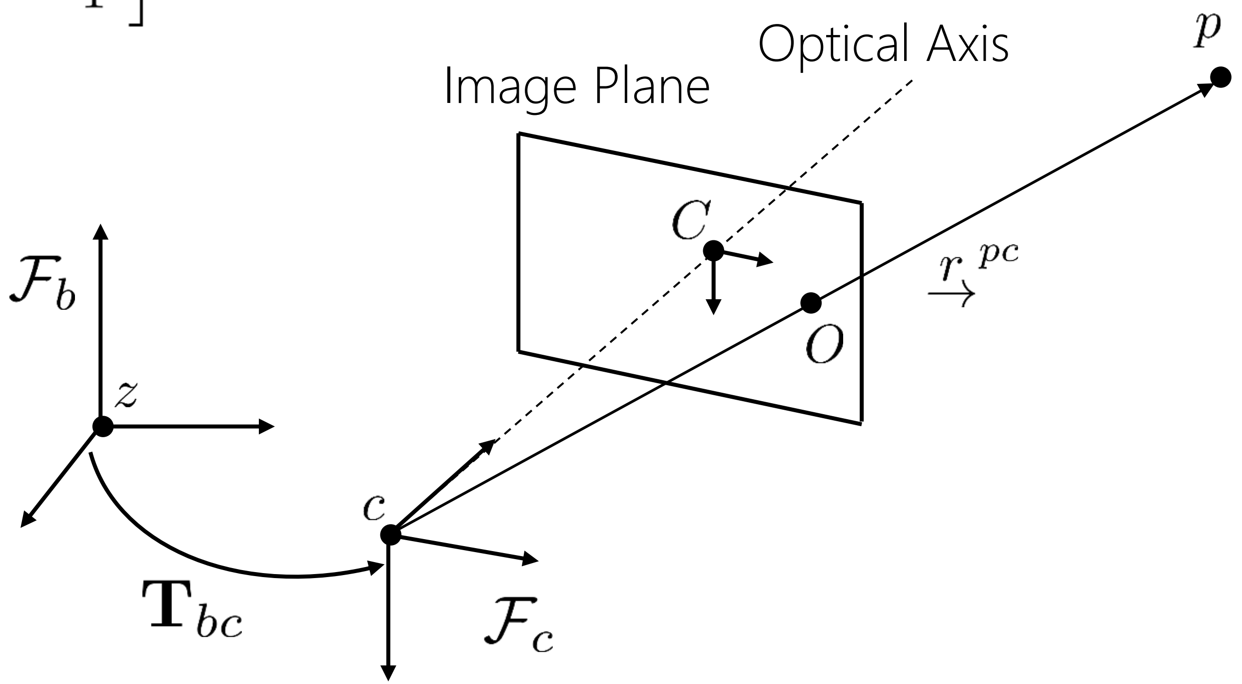
- We need to write our measurement model in terms of the states of interest.
- This can be done using the camera/IMU **extrinsic parameters.**

$$\mathbf{T}_{bc} = \begin{bmatrix} \mathbf{C}_{bc} & \mathbf{r}_b^{cz} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3)$$

Full measurement model

$$\mathbf{r}_c^{pc} = \mathbf{C}_{bc}^{\mathsf{T}} \left(\mathbf{C}_{ab}^{\mathsf{T}} \left(\mathbf{r}_a^{pw} - \mathbf{r}_a^{zw} \right) - \mathbf{r}_b^{cz} \right)$$

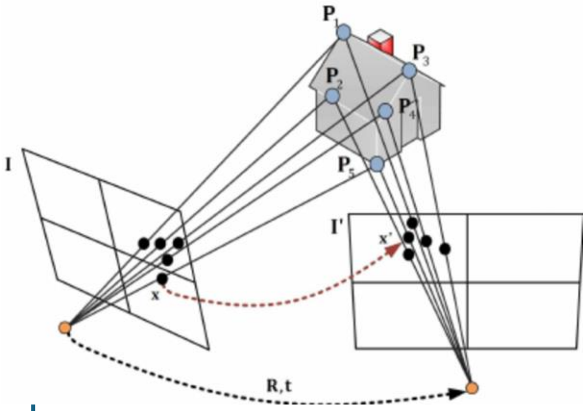
$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \mathbf{g}\left(\mathbf{r}_c^{pc}\right) = \begin{bmatrix} r_1/r_3 & r_2/r_3 \end{bmatrix}$$



Visual-Inertial Navigation Approaches

Loosely-Coupled Fusion

Step 1: Determine relative poses between images using two-view geometry.



Step 2: Fuse relative pose information with IMU (filters, sliding window filters, etc.)

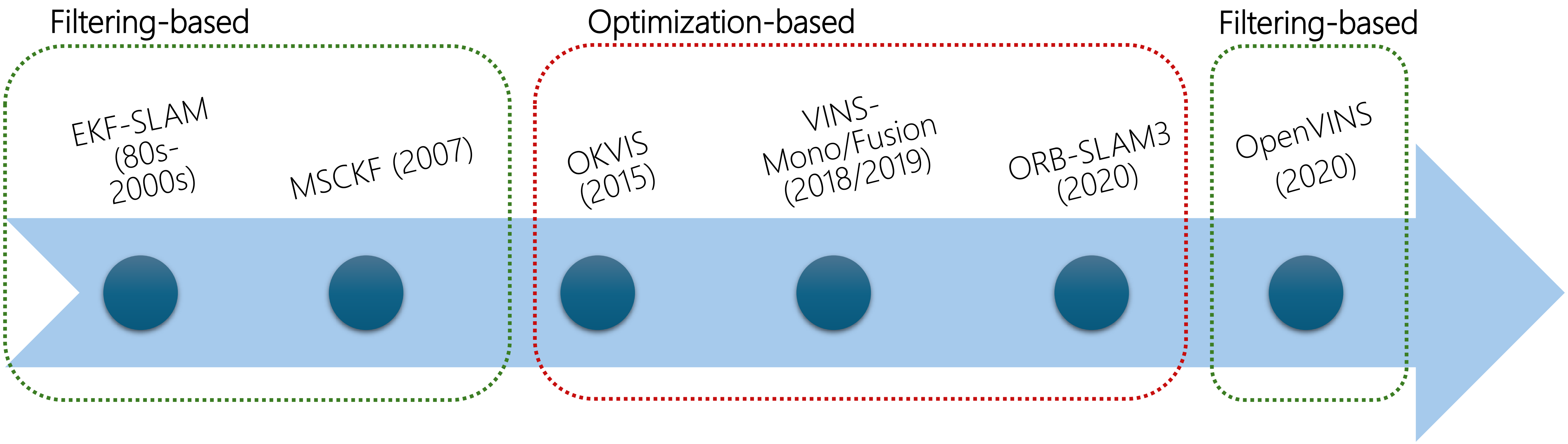
Computationally efficient!

Tightly-Coupled Fusion

Directly fuse camera and IMU data within a single process.

Higher accuracy, higher computational cost.

Tightly-Coupled Visual-Inertial Algorithms



- **Filtering-based** algorithms use measurements a single time to estimate the state.
- **Optimization-based** algorithms perform iterative minimization over a window of states.

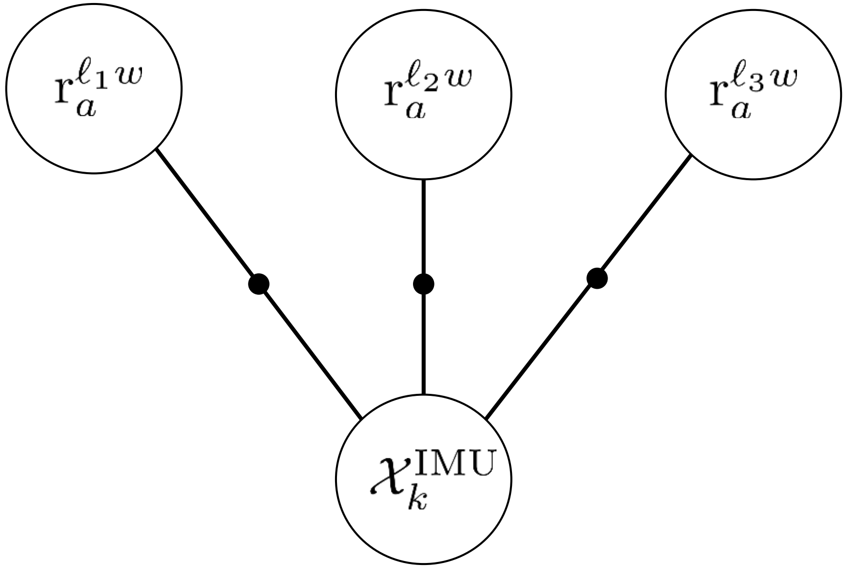
McGill UNIVERSITY

Classic EKF-SLAM

- EKF state includes current IMU state and landmark positions.

$$\mathcal{X}_k = \left(\mathcal{X}_k^{\mathrm{IMU}}, \mathbf{r}_a^{\ell_1 w}, \ldots, \mathbf{r}_a^{\ell_n w}\right)$$

IMU state Landmark position



- State covariance can then be partitioned as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^{\mathcal{X}\mathcal{X}} & \mathbf{P}^{\mathcal{X}\mathcal{L}} \\ \mathbf{P}^{\mathcal{L}\mathcal{X}} & \mathbf{P}^{\mathcal{L}\mathcal{L}} \end{bmatrix}$$

- The IMU state is the orientation, velocity and position of the IMU, and the IMU biases.

$$\mathcal{X}^{\mathrm{IMU}} = \left(\mathbf{C}_{ab}, \mathbf{v}_a^{zw/a}, \mathbf{r}_a^{zw}, \mathbf{b}_b^{g}, \mathbf{b}_b^{a}\right).$$

Classic EKF-SLAM – Prediction Step

- Reduced form of the typical EKF prediction step due to the structure of the problem.

- IMU process model and process model Jacobian are given by

$$\mathcal{X}_k^{\mathrm{IMU}} = f_{k-1}\left(\mathcal{X}_{k-1}, \mathbf{u}_{k-1}\right) \oplus \mathbf{w}_{k-1}, \qquad \mathbf{w}_{k-1} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_{k-1}\right)$$

$$\mathbf{F}_{k-1}^{\mathrm{IMU}} = \left.\frac{Df\left(\mathcal{X}^{\mathrm{IMU}}, \mathbf{u}_{k-1}\right)}{D\mathcal{X}^{\mathrm{IMU}}}\right|_{\hat{\mathcal{X}}_{k-1}}$$

- Landmarks are assumed constant!

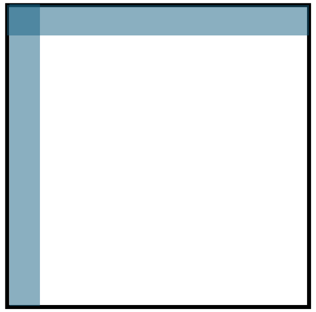
$$\dot{\mathbf{r}}_a^{\ell_i w} = \mathbf{0}, \quad i = 1, \ldots, m$$

- The EKF covariance propagation has the form

Updated parts of the covariance

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1}^{\mathrm{IMU}} \hat{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^{\mathrm{IMU}^{\mathsf{T}}} + \mathbf{Q}_{k-1}$$

$$\check{\mathbf{P}}_k = \begin{bmatrix} \check{\mathbf{P}}_k^{\mathcal{X}\mathcal{X}} & \mathbf{F}_{k-1}^{\mathrm{IMU}} \hat{\mathbf{P}}_{k-1}^{\mathcal{X}\mathcal{L}} \\ \hat{\mathbf{P}}_{k-1}^{\mathcal{L}\mathcal{X}} \mathbf{F}_{k-1}^{\mathrm{IMU}^{\mathsf{T}}} & \hat{\mathbf{P}}_{k-1}^{\mathcal{L}\mathcal{L}} \end{bmatrix}$$



Classic EKF-SLAM – Correction Step

- The EKF correction step can also be modified to exploit sparsity.
- Each measurement links **one** IMU state to **one** landmark state.

Measurement Model: $\mathbf{y}_{jk} = \mathbf{g}\left(\mathcal{X}_k^{\text{IMU}}, \mathbf{r}_a^{\ell_j w}\right) + \mathbf{v}_{jk}$

Measurement Model Jacobian: $\mathbf{H}_{jk} = \begin{bmatrix} \mathbf{H}_k & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_j & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$

$$\mathbf{H}_k = \left.\frac{D\mathbf{g}(\mathcal{X}, \mathbf{r}_a^{\ell w})}{D\mathcal{X}}\right|_{\hat{\mathcal{X}}_k^{\text{IMU}}}, \qquad \mathbf{H}_j = \left.\frac{D\mathbf{g}(\mathcal{X}, \mathbf{r}_a^{\ell w})}{D\mathbf{r}_a^{\ell w}}\right|_{\hat{\mathbf{r}}_a^{\ell w}}$$

- The computation of the innovation is sparse, but the Kalman gain is **dense**.

$$\hat{\mathcal{X}} = \check{\mathcal{X}} \oplus \mathbf{K}\mathbf{z}$$
$$\mathbf{P}_k = \left(\mathbf{1} - \mathbf{K}\mathbf{H}_{jk}\right)\check{\mathbf{P}}_k$$

$\longrightarrow$ Entire state and covariance must be updated for each landmark observation!

Complexity $\mathcal{O}\left(n^2\right)$ per measurement.

McGill UNIVERSITY

The Problem with EKF-SLAM

- The computational complexity of the EKF scales worst-case **cubically** with the number of landmarks.
- The EKF quickly becomes computationally intractable for large maps.
- Possible remedies:
  - decouple the estimation problem into a series of smaller submaps [1],
  - utilize the Extended Information Filter [2].

- **Is there a way to remove the landmarks from the state vector?**

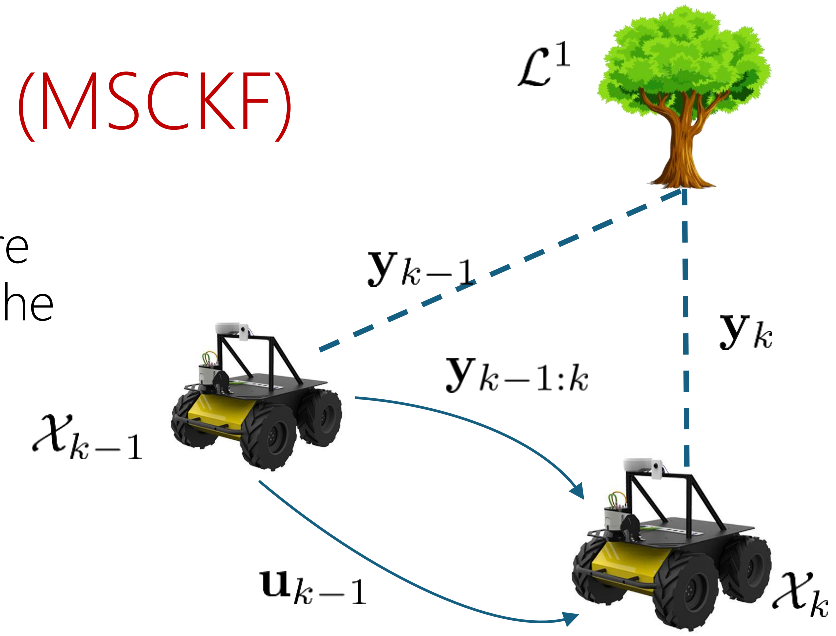
[1] J. Leonard and H. Feder, "Decoupled Stochastic Mapping," IEEE Journal of Oceanic Engineering, vol. 26, no. 4, pp. 561–571, 2001

[2] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," International Journal of Robotics Research, 2004

The Multi-State Constraint Kalman Filter (MSCKF)

- Key idea: Express the constraint imposed by a static feature observed from multiple camera poses, **without** including the feature position in the state vector.

- The MSCKF state includes m IMU poses.



MSCKF State

$$\mathcal{X}_k = (\mathcal{X}_k^{\mathrm{IMU}}, \mathcal{X}_{k-1}^p, \mathcal{X}_{k-2}^p, \cdots, \mathcal{X}_{k-m}^p)$$

$$\mathcal{X}_k^p = (\mathbf{C}_{ab_k}, \mathbf{r}_a^{z_k w})$$

EKF State

$$\mathcal{X}_k = (\mathcal{X}_k^{\mathrm{IMU}}, \mathbf{r}_a^{\ell_1 w}, \cdots, \mathbf{r}_a^{\ell_m w})$$

- The MSCKF is still an EKF-based estimator.
  - Follows a predict-correct structure!

MSCKF Overview

- **Step 1 – Prediction:** Propagate the state forward using the IMU measurements.
- **Step 2 – State Augmentation:** At each new image, augment the state and covariance with a copy of the current IMU pose.
- **Step 3 – Image Processing:** Extract and match features on the image.
- **Step 4 – Correction:** when an update step is triggered, utilize *all* measurements of a given landmark to perform the EKF correction step and update the state.

MSCKF Overview – Prediction Step

- Similar propagation step to EKF-SLAM due to covariance partitioning!

$$\mathcal{X}_k = \left(\mathcal{X}_k^{\text{IMU}}, \mathcal{X}_{k-1}^p, \mathcal{X}_{k-2}^p, \cdots, \mathcal{X}_{k-m}^p\right)$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^{\mathcal{X}\mathcal{X}} & \mathbf{P}^{\mathcal{X}\mathcal{P}} \\ \mathbf{P}^{\mathcal{P}\mathcal{X}} & \mathbf{P}^{\mathcal{P}\mathcal{P}} \end{bmatrix}$$

IMU State Covariance: $\mathbf{P}^{\mathcal{X}\mathcal{X}} \in \mathbb{R}^{15\times 15}$

IMU Clone Covariances: $\mathbf{P}^{\mathcal{P}\mathcal{P}} \in \mathbb{R}^{6m\times 6m}$

- Current IMU state is propagated forward using process model,

$$\mathcal{X}_k^{\text{IMU}} = f_{k-1}\left(\mathcal{X}_{k-1}, \mathbf{u}_{k-1}\right).$$

- Covariance is propagated forwards as

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1}^{\text{IMU}} \hat{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^{\text{IMU}^\mathsf{T}} + \mathbf{Q}_{k-1}$$

$$\check{\mathbf{P}}_k = \begin{bmatrix} \check{\mathbf{P}}_k^{\mathcal{X}\mathcal{X}} & \mathbf{F}_{k-1}^{\text{IMU}} \hat{\mathbf{P}}_{k-1}^{\mathcal{X}\mathcal{L}} \\ \hat{\mathbf{P}}_{k-1}^{\mathcal{L}\mathcal{X}} \mathbf{F}_{k-1}^{\text{IMU}^\mathsf{T}} & \hat{\mathbf{P}}_{k-1}^{\mathcal{L}\mathcal{L}} \end{bmatrix}$$

MSCKF Overview – State Augmentation

- When an image is received, augment the state and covariance with a copy of the current IMU pose.

$$\hat{\mathcal{X}} = \left(\hat{\mathcal{X}}_k^{\mathrm{IMU}}\right) \longrightarrow \hat{\mathcal{X}}_k^{+} = \left(\hat{\mathcal{X}}_k^{\mathrm{IMU}}, \hat{\mathcal{X}}_k^{p}\right)$$

- This operation is called **stochastic cloning.**

Current state

$$\mathcal{V} = c\left(\mathcal{X}\right)$$

New variable

Cloning function

- Augment state and covariance as

$$\hat{\mathcal{X}}^{+} = \left(\hat{\mathcal{X}}, \quad c\left(\hat{\mathcal{X}}\right)\right),$$

$$\hat{\mathbf{P}}^{+} = \begin{bmatrix} \hat{\mathbf{P}} & \hat{\mathbf{P}}\mathbf{C}^{\mathsf{T}} \\ \mathbf{C}\hat{\mathbf{P}} & \mathbf{C}\hat{\mathbf{P}}\mathbf{C}^{\mathsf{T}} \end{bmatrix},$$

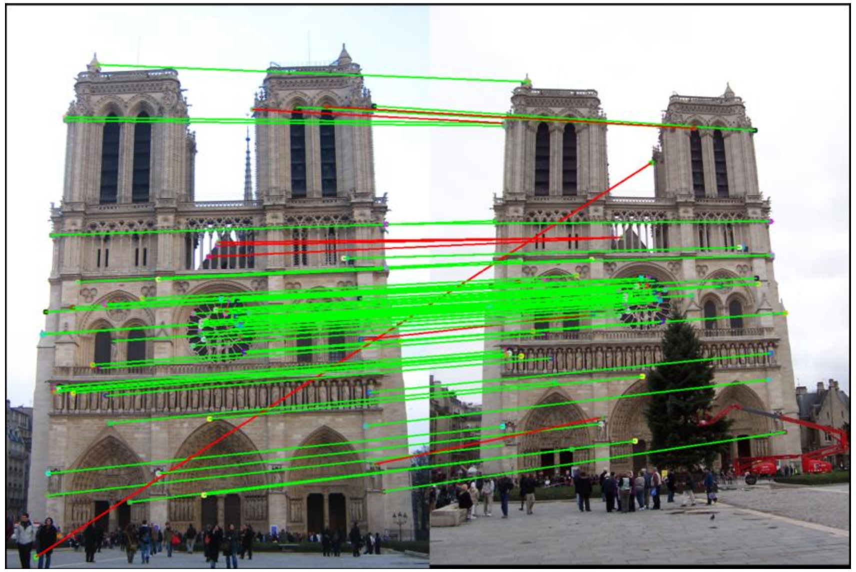
$$\mathbf{C} = \left.\frac{Dc\left(\mathcal{X}\right)}{D\mathcal{X}}\right|_{\hat{\mathcal{X}}}.$$

McGill UNIVERSITY

MSCKF Overview – Image Processing

- Extract and match features on the image.
- Common techniques include utilizing **optical flow**, or **descriptor matching**.

Optical flow finds apparent motion between images.



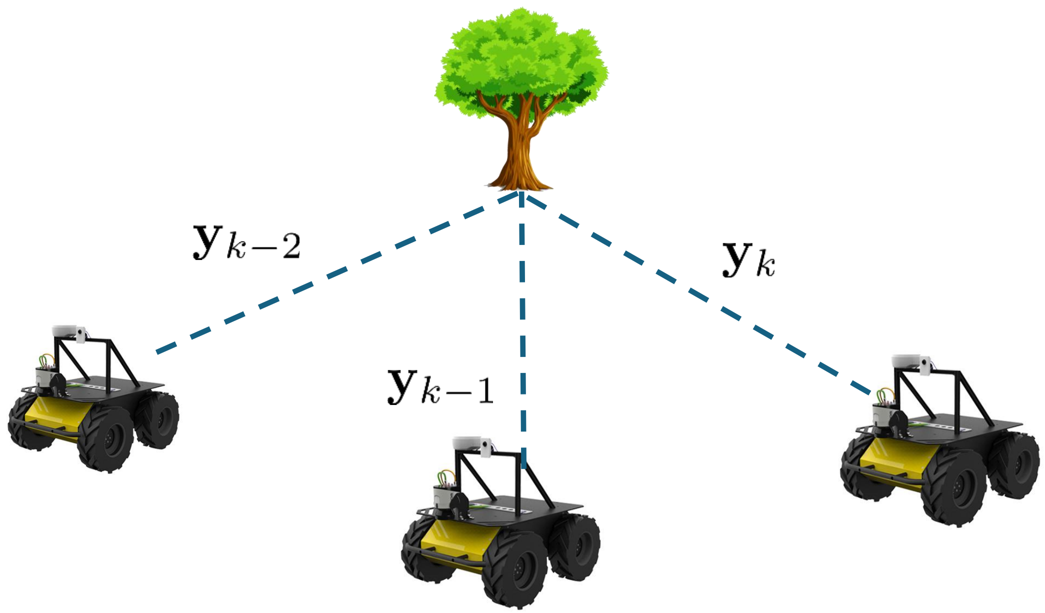
Descriptor matching tries to find similar features in "descriptor" space.

McGill UNIVERSITY

[Images taken from](http://jevois.org/tutorials/ProgrammerInvFlowLK.html)
http://jevois.org/tutorials/ProgrammerInvFlowLK.html
https://faculty.cc.gatech.edu/~hays/compvision2018/proj2/

MSCKF Overview – Correction Step

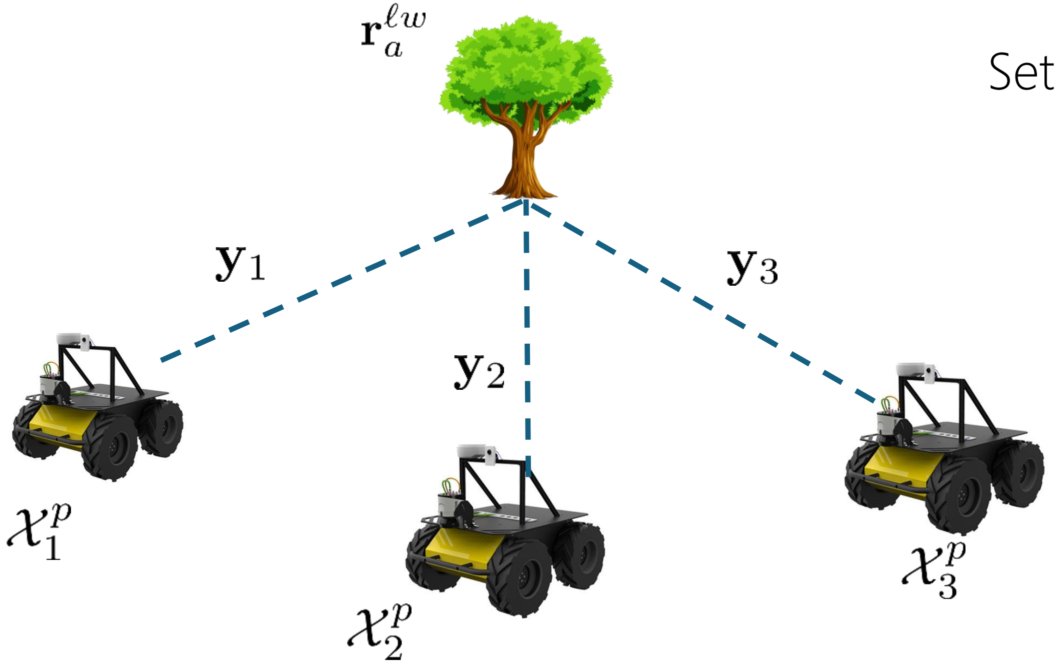
- **Step 4 – Correction:** when an update step is triggered, utilize *all* measurements of a given feature to perform the EKF correction step and update the state.



- **First question:** How do we utilize all measurements of a given feature to correct the MSCKF state?
- **Second question:** When do we trigger the correction step?

MSCKF Overview – The Measurement Model

- Consider the measurements of a single feature, $\mathbf{r}_a^{\ell w}$, observed from a set of M_j robot poses.



Set of robot poses: $\{\mathcal{X}_i^p\}, i \in \mathcal{S}_j$

Observations: $\mathbf{y}_i = \mathbf{g}\left(\mathbf{r}_{c_i}^{\ell c_i}\right) + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_k\right).$

Feature resolved in camera frame:

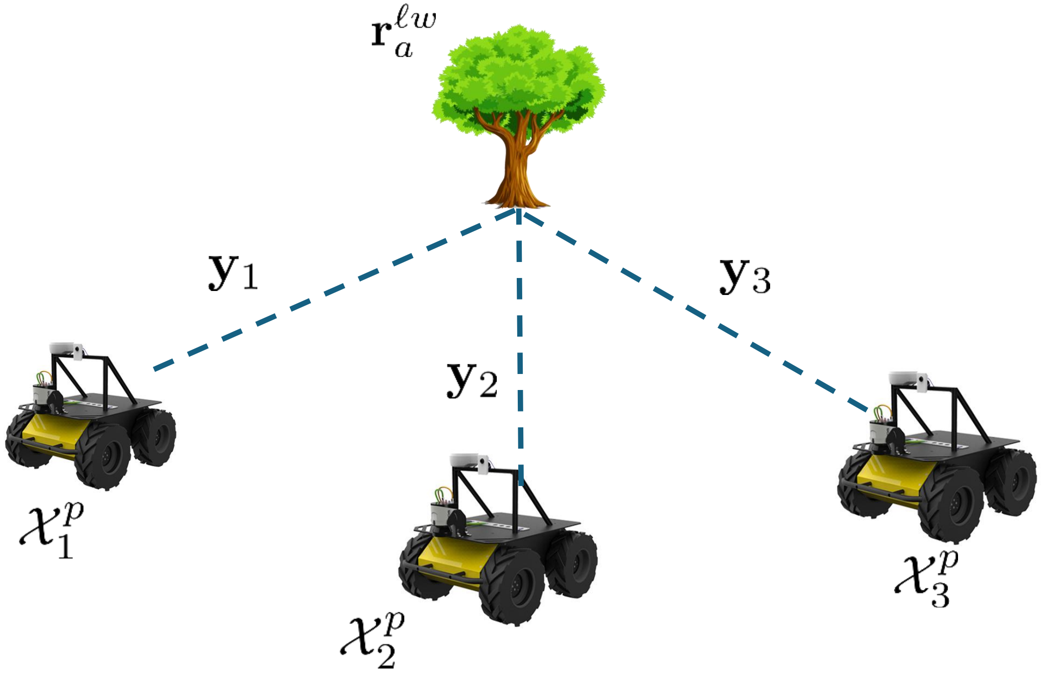
$$\mathbf{r}_{c_i}^{\ell c_i} = \mathbf{C}_{ac_i}^{\mathsf{T}}\left(\boxed{\mathbf{r}_a^{\ell w}} - \mathbf{r}_a^{c_i w}\right)$$

Inertial landmark position

- We want to be able to predict the measurements from the state estimates.

MSCKF Overview – Solving for the Landmark Position

- We need an we need an estimate of $\mathbf{r}_a^{\ell w}$!
- Conduct batch estimation to solve for an estimate of the landmark position as



$$\hat{\mathbf{r}}_a^{\ell w} = \arg\max_{\mathbf{r}_a^{\ell w}} p\left(\mathbf{r}_a^{\ell w} | \mathbf{y}_i, \hat{\mathcal{X}}_i^p\right), \quad i \in \mathcal{S}_j$$

Predict measurement

$$\hat{\mathbf{r}}_{c_i}^{\ell c_i} = \hat{\mathbf{C}}_{ac_i}^{\mathsf{T}} \left(\hat{\mathbf{r}}_a^{\ell w} - \hat{\mathbf{r}}_a^{c_i w}\right)$$

$$\hat{\mathbf{y}}_i = \mathbf{g}\left(\hat{\mathbf{r}}_{c_i}^{\ell c_i}\right)$$

- IMU poses are treated as known constants and are held **fixed** in the batch problem.

McGill UNIVERSITY

MSCKF Overview – The Measurement Model

- Measurement residual can now be computed using the estimated feature position as

$$\mathbf{z}_i = \mathbf{y}_i - \mathbf{g}\left(\hat{\mathbf{r}}_{c_i}^{\ell c_i}\right)$$

- Linearize residual:

$$\delta \mathbf{r}_a^{\ell w} = \mathbf{r}_a^{\ell w} - \hat{\mathbf{r}}_a^{\ell w}$$

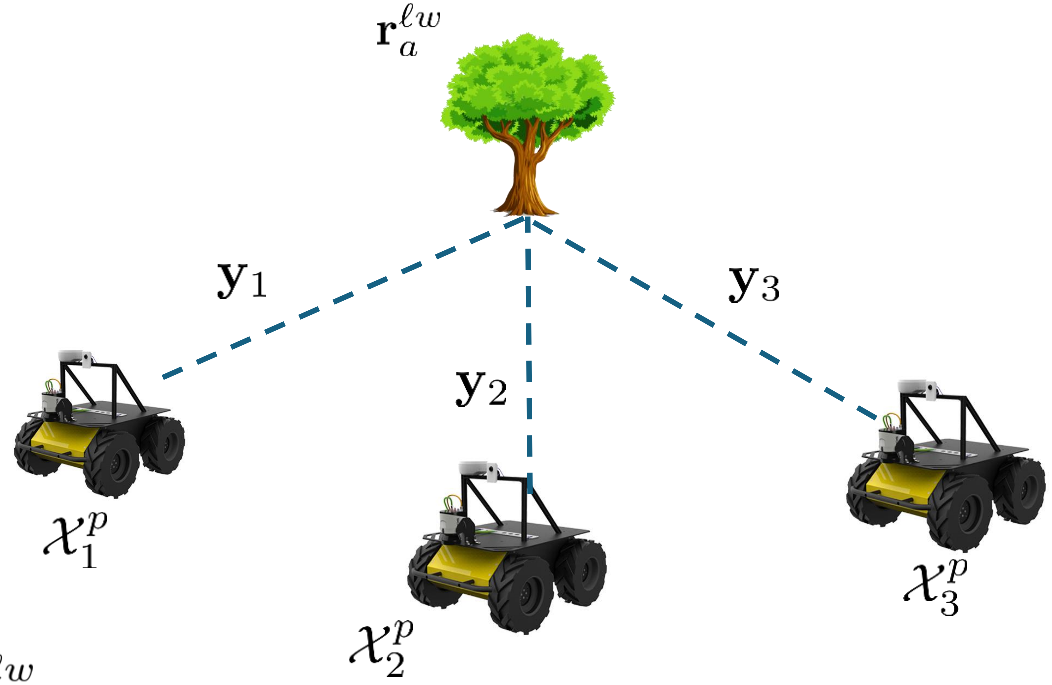
$$\delta \mathbf{z}_i = \mathbf{H}^{\mathcal{X}_i^p} \delta \boldsymbol{\xi}_i^p + \mathbf{H}^{\ell} \delta \mathbf{r}_a^{\ell w}$$

Pose Jacobian

Landmark Jacobian

- Stack all residuals of a single feature:

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_{M_j} \end{bmatrix} \xrightarrow{\text{Stack linearizations}} \delta \mathbf{z} = \mathbf{H}^{\mathcal{X}^p} \delta \boldsymbol{\xi}^p + \mathbf{H}^{\ell} \delta \mathbf{r}_a^{\ell w}$$



- Cannot directly use this in the EKF correction step…

McGill UNIVERSITY

MSCKF Overview – MSCKF Nullspace Projection

- The MSCKF "secret sauce": project original residual onto the left **nullspace** of the feature Jacobian.

Original linearized residual: $\delta \mathbf{z} = \mathbf{H}^{\mathcal{X}^p} \delta \boldsymbol{\xi}^p + \mathbf{H}^{\ell} \delta \mathbf{r}_a^{\ell w}, \quad \mathbf{H}^{\ell} \in \mathbb{R}^{2M_j \times 3}$

- Nullspace of the feature Jacobian is spanned by the columns of $\mathbf{N}$,

$$\mathbf{N} = \begin{bmatrix} \mathbf{n}_1 & \cdots & \mathbf{n}_{2M_j-3} \end{bmatrix} \in \mathbb{R}^{2M_j \times (2M_j-3)}, \qquad \mathbf{H}^{\ell^{\mathsf{T}}} \mathbf{n}_i = \mathbf{0}.$$

$$\mathbf{N}^{\mathsf{T}} \mathbf{H}^{\ell} = \mathbf{0}$$

$$\downarrow$$

$$\mathbf{N}^{\mathsf{T}} \delta \mathbf{z} = \mathbf{N}^{\mathsf{T}} \mathbf{H}^{\mathcal{X}^p} \delta \boldsymbol{\xi}^p + \cancelto{0}{\mathbf{N}^{\mathsf{T}} \mathbf{H}^{\ell} \delta \mathbf{r}_a^{\ell w}}$$

New residual definition: $\mathbf{z}' = \mathbf{N}^{\mathsf{T}} \mathbf{z} \in \mathbb{R}^{2M_j-3}$

- SVD can be used to compute the left nullspace.

$$\mathbf{H}^{f} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^{\mathsf{T}}$$

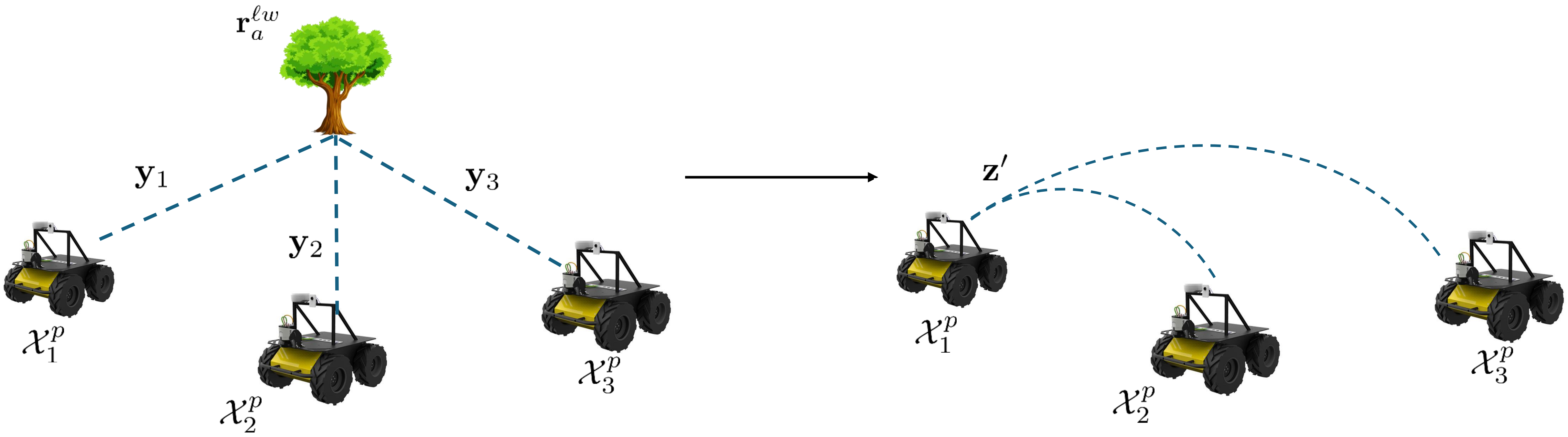
Take columns corresponding to 0 singular values

McGill UNIVERSITY

MSCKF Overview – MSCKF Nullspace Projection

- New linearized residual is **independent** of the errors in the feature positions.
  - We can now use this residual and linearization in an EKF update step!

- A constraint is defined between all the camera poses from which a given feature was observed.



MSCKF Overview – EKF Update Trigger

- Now we've showed how to express the constraint imposed by observing a landmark from multiple camera poses.

- An EKF updated using this constraint is triggered by one of two conditions:
    1. Once a feature has lost tracking, all measurements of that feature are used in an EKF update.

    2. Once a maximum allowable number of IMU poses ($N_{\max}$) in the state vector has been reached, remove a number of these camera poses and utilize all measurements from these selected poses.

Example: if $N_{\max} = 5$

$$\mathcal{X}_k = (\mathcal{X}_k^{\mathrm{IMU}}, \mathcal{X}_{k-1}^p, \mathcal{X}_{k-2}^p, \mathcal{X}_{k-3}^p, \mathcal{X}_{k-4}^p, \mathcal{X}_{k-5}^p)$$

Remove from state vector and utilize all measurements.

McGill UNIVERSITY

MSCKF Overview – EKF Update Equations

- Consider a situation in which the constraints from L features, selected from the two previous criteria, must be processed.

- Compute the (projected) residual and linearization for each feature

$$\mathbf{z}'_j = \mathbf{N}_j^{\mathsf{T}} \mathbf{z}_j \in \mathbb{R}^{2M_j - 3}, \quad j = 1, \dots, L.$$

$$\delta \mathbf{z}'_j = \mathbf{N}_j^{\mathsf{T}} \delta \mathbf{z}_j = \mathbf{N}_j^{\mathsf{T}} \mathbf{H}^{\mathcal{X}^p} \delta \boldsymbol{\xi}$$

- Stack all residuals and linearizations as

$$\begin{aligned} \mathbf{z}' &= \begin{bmatrix} \mathbf{z}_1^{\mathsf{T}} & \cdots & \mathbf{z}_L^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \\ \delta \mathbf{z}' &= \mathbf{H} \delta \boldsymbol{\xi} \end{aligned}$$

- Residual dimension can now be quite large – size of residual is given by

$$d = \sum_{j=1}^{L} (2M_j - 3)$$

McGill UNIVERSITY

MSCKF Overview – EKF Update Equations

- QR decomposition saves the day (as usual).
- If the dimension of the residual is larger than the state dimension, decompose the full Jacobian as

$$\begin{aligned}\mathbf{H} &= \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{H}^r \\ \mathbf{0}\end{bmatrix} \\ &= \mathbf{Q}_1\mathbf{H}^r\end{aligned}$$

- Project the residual onto this basis vectors of the range of $\mathbf{H}$ as

$$\mathbf{z}^r = \mathbf{Q}_1^\mathsf{T}\mathbf{z}$$

$$\downarrow$$

$$\begin{bmatrix}\mathbf{Q}_1^\mathsf{T}\delta\mathbf{z} \\ \mathbf{Q}_2^\mathsf{T}\delta\mathbf{z}\end{bmatrix} = \begin{bmatrix}\mathbf{H}^r \\ \mathbf{0}\end{bmatrix}\delta\boldsymbol{\xi}$$

$$\delta\mathbf{z}^r = \mathbf{H}^r\delta\boldsymbol{\xi}$$

- This reduced residual and Jacobian can now be used in the EKF update!

McGill UNIVERSITY

MSCKF Summary

- The computational complexity of the MSCKF is **linear** in the number of features, and at worst **cubic** in the number of poses that are included in the state vector.

- The MSCKF can be thought of as a hybrid between a sliding window filter and the standard EKF.
  - Both maintain historical poses in the state vector.

$$\mathcal{X}_k = \left(\mathcal{X}_k^{\mathrm{IMU}}, \mathcal{X}_{k-1}^{p}, \mathcal{X}_{k-2}^{p}, \cdots, \mathcal{X}_{k-m}^{p}\right)$$

  - The MSCKF still only linearizes the measurement model a single time!

MSCKF Nullspace Projection Dimentions

- Matrix dimensions of involved quantities in nullspace projection:

$$\mathbf{H}^{\ell} \in \mathbb{R}^{2n \times 3}$$

$$\operatorname{rank}(\mathbf{H}^{\ell}) \leq \min(2n, 3) = 3$$

$$\operatorname{nullity}\left(\mathbf{H}^{\ell^{\mathsf{T}}}\right) = 2n - 3$$

$$\mathbf{N} \in \mathbb{R}^{2n \times (2n-3)}$$

$$\mathbf{z}' = \mathbf{N}^{\mathsf{T}} \mathbf{H}^{\ell} \in \mathbb{R}^{2n \times 3}$$