Vision Aided Navigation and SLAM: Overview II

Vadim Indelman

TECHNION Israel Institute of Technology

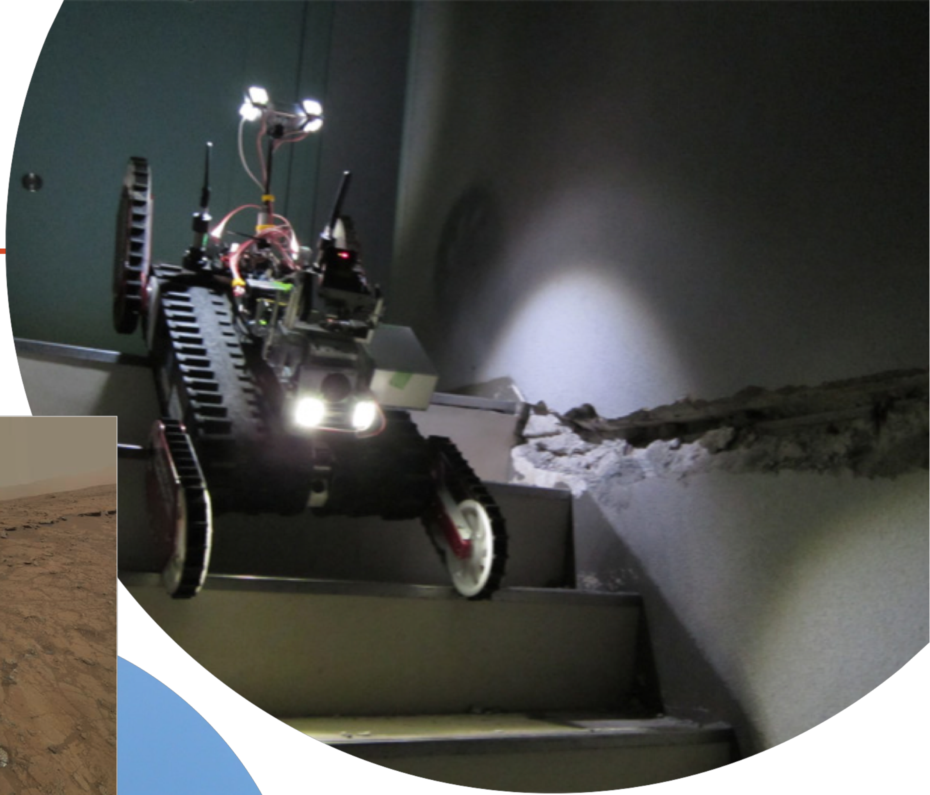
ANPL Autonomous Navigation and Perception Lab

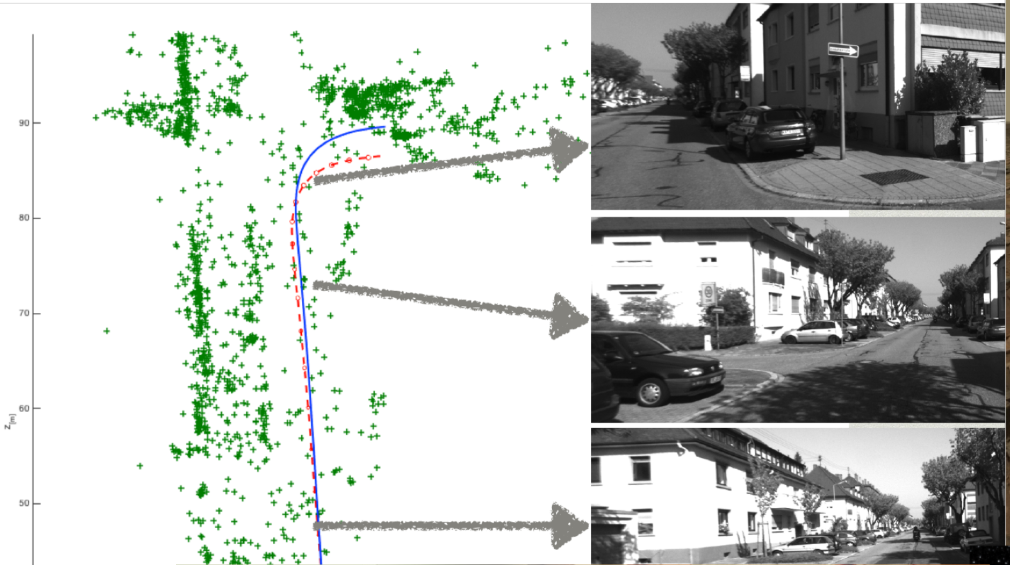
IAAC Workshop on Vision Aided Navigation

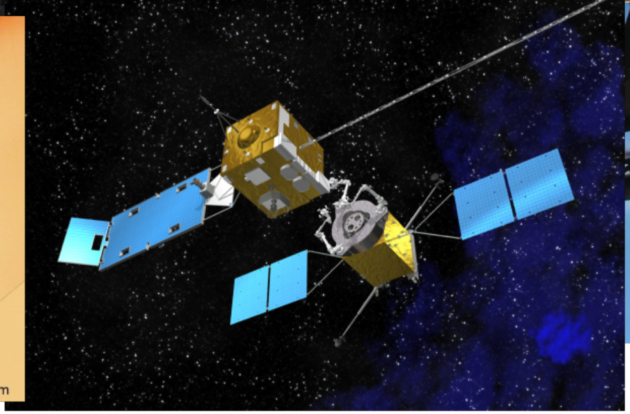
January 15, 2019

Introduction

- Numerous applications



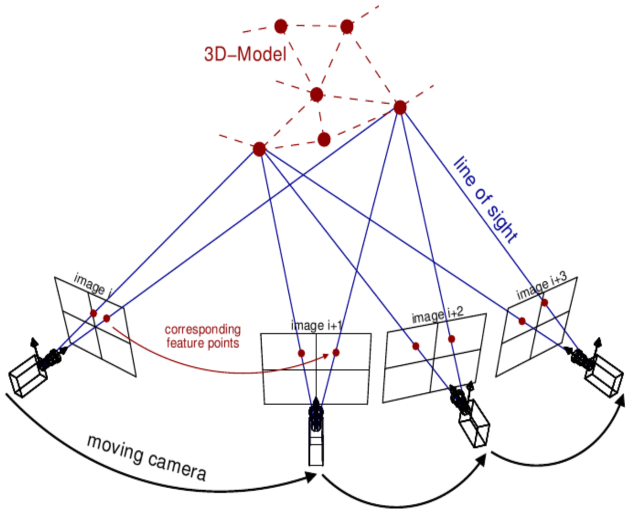


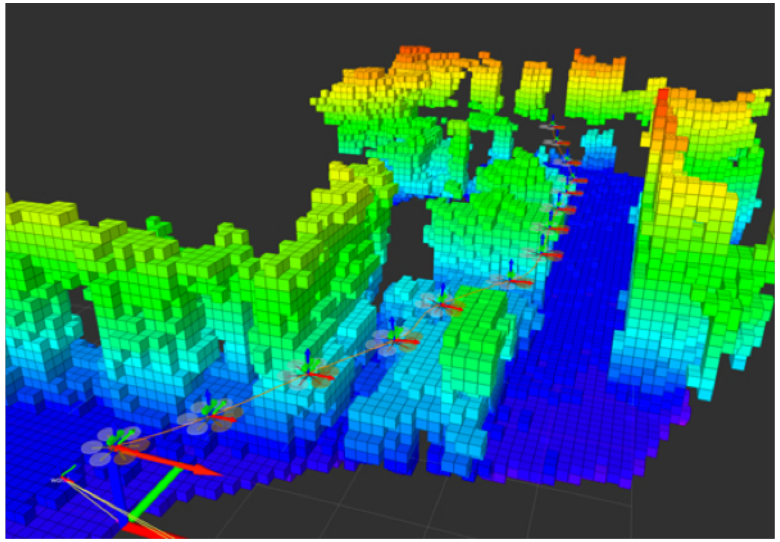


Introduction

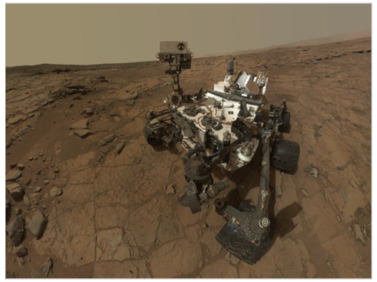
Autonomous navigation and perception in uncertain/unknown environments:

- **Perception and Inference**: Where am I? What is the surrounding environment? **This talk**
- **Planning Under Uncertainty & Active Perception**: Decide next action(s) given partial, noisy data





The "Big" Picture



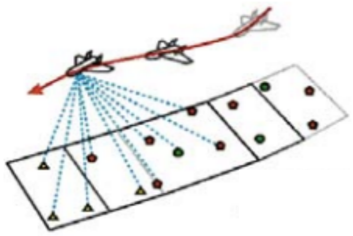
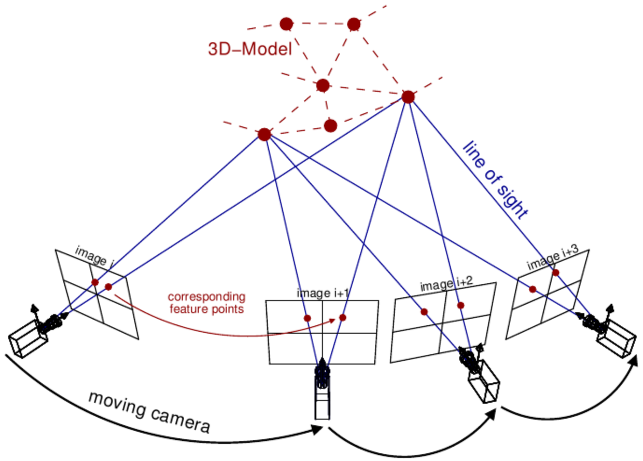
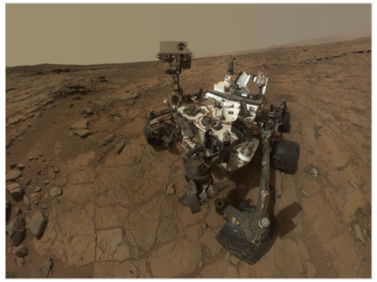


Image from [Kim et al. 2005]

- **Objective**:
  - Estimate platform state and observed environment (e.g. 3D points)
  - Environment is unknown, uncertain or dynamic



The “Big” Picture



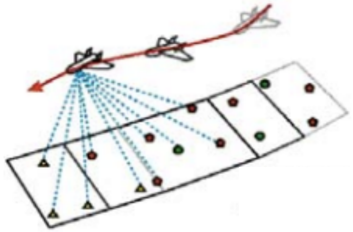
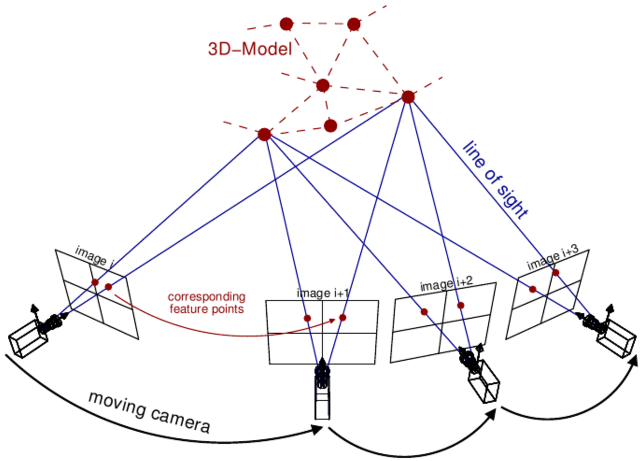


Image from [Kim et al. 2005]

- How?
  - Many images (sensor measurements)
  - Interest points (features) in each image
  - Track features from image to image, data association
  - Probabilistic inference over robot state and environment (e.g. 3D points)
  - Additional sensors
  - Online performance?

Front-end

Back-end



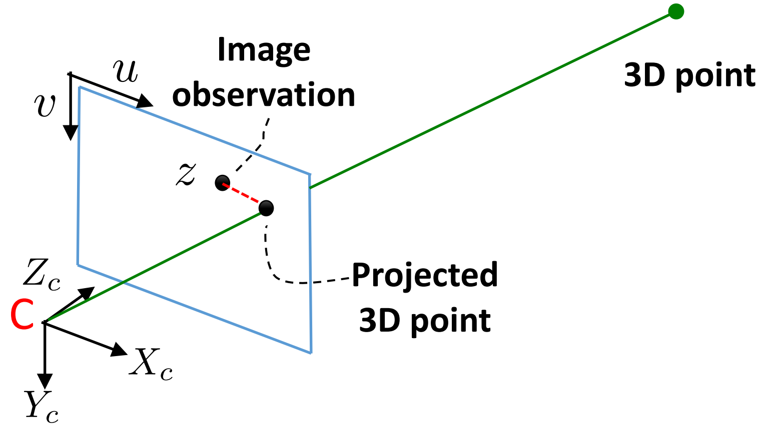
Outline

Projection Matrix & Operator

- Projection of a 3D point $\mathbf{X}^w = (X_w, Y_w, Z_w)$:

$$\begin{pmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{w}\end{pmatrix} = \overbrace{K \underbrace{\left[\ R \ \mid \ t\ \right]}_{\doteq\, x}}^{\text{Projection matrix}} \underbrace{\begin{pmatrix}X_w\\ Y_w\\ Z_w\\ 1\end{pmatrix}}_{\doteq\, l}$$

Camera pose **3D point, landmark**



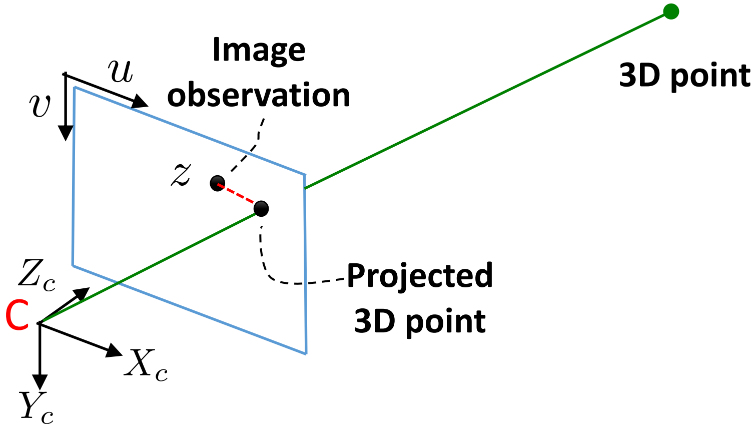
- To recover the pixel (u, v): $\begin{pmatrix}u\\ v\end{pmatrix} = \begin{pmatrix}\frac{\tilde{u}}{\tilde{w}} & \frac{\tilde{v}}{\tilde{w}}\end{pmatrix}^T$

- Projection of a 3D point (notation): $\pi(x, l) \doteq K \left[\ R \quad t\ \right] l$

Known as projection operator

Re-Projection Error

- Ideally: $\underline{(u,v)} = \pi(x,l)$
  $\quad z$
- In practice:
  - Image observations z are noisy
  - Incorrect/imprecise camera pose and 3D point



- Re-projection error: $\underbrace{z}_{\text{Image observation (pixel, feature)}} - \underbrace{\pi(x,l)}_{\text{Predicted measurement}}$

- Assuming Gaussian image noise, measurement likelihood:

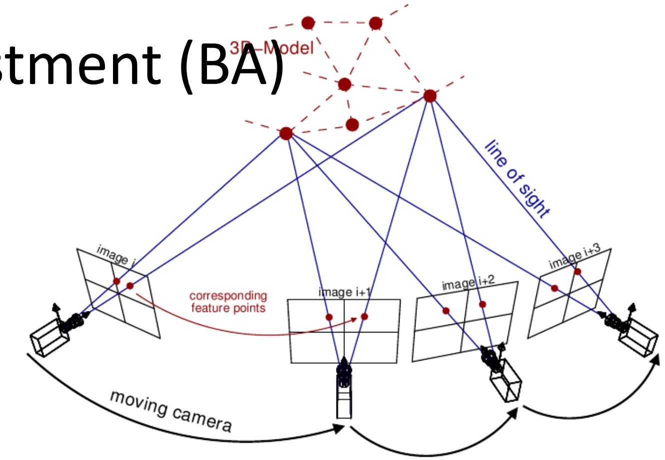
$$p(z|x,l) = \frac{1}{\sqrt{|2\pi\Sigma_v|}} \exp\left(-\frac{1}{2}\|\underbrace{z - \pi(x,l)}_{\text{Re-projection error}}\|^2_{\Sigma_v}\right) \qquad v \sim N(0, \Sigma_v)$$

Outline

- Introduction
- Camera projective geometry
- Bundle Adjustment
- Incremental Smoothing and Mapping (iSAM) algorithms
- Visual-inertial SLAM and IMU pre-integration concept

Visual SLAM and Bundle Adjustment

- Assume we are given a sequence of images
- **Objective**: Would like to infer camera poses and observed 3D points
  - Using **only** images as input (no additional sensors, for now)
  - Assume data association is given (very challenging by itself!)
- Problem known as
  - Computer vision: Structure from motion (SfM), Bundle adjustment (BA)
  - Robotics: Simultaneous localization and mapping (SLAM)

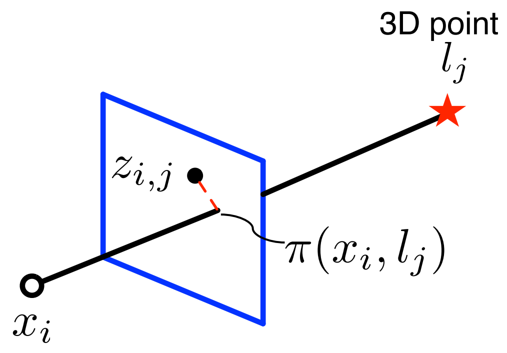


Bundle Adjustment

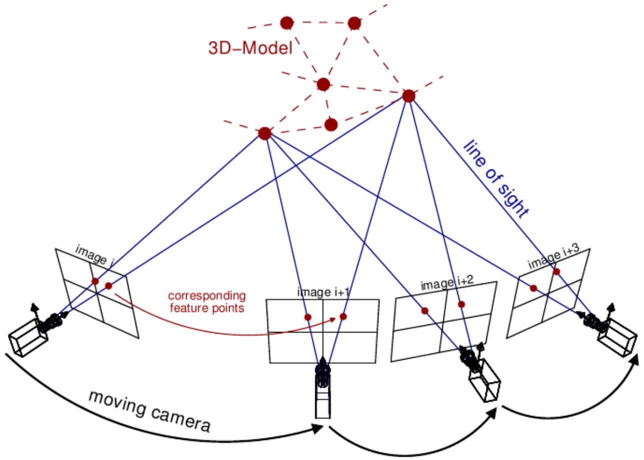
Consider the i-th image:

- A single image observation of a 3D point l_j

$$z_{i,j} = \pi(x_i, l_j) + v \quad \longrightarrow \quad p(z_{i,j} | x_i, l_j)$$



- Notations
  - $\mathcal{M}_i$: indices of 3D points observed in image i
  - Z_i : all image observations from image i



- The joint pdf over camera pose and observed 3D points:

$$p(x_i, \{l_j | j \in \mathcal{M}_i\} | Z_i) = \eta \prod_{j \in \mathcal{M}_i} p(z_{i,j} | x_i, l_j)$$

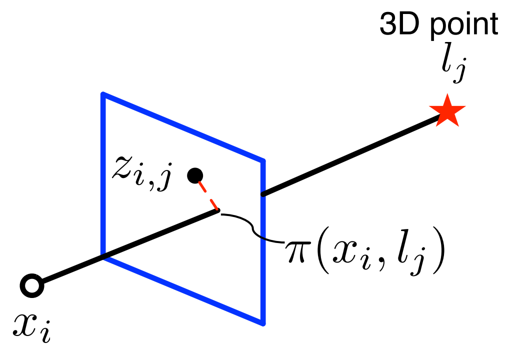
All 3D points observed in image i

Bundle Adjustment

X: all camera poses (or platform states)

L: observed 3D points (in any of the images)

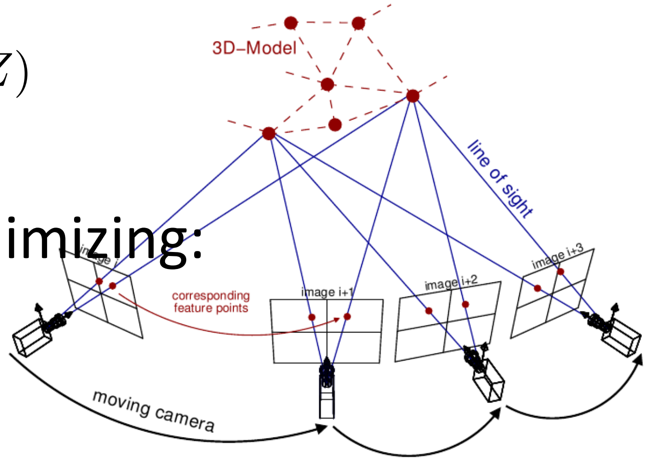
- Consider now N images



- Joint pdf for all image observations, in all camera frames:

$$p\left(X, L|Z\right) \propto \prod_{i}^{N} \prod_{j \in \mathcal{M}_i} p\left(z_{i,j}|x_i, l_j\right)$$

- Maximum a posteriori (MAP) estimate for $X^{\star}, L^{\star} = \arg\max_{X,L} p\left(X, L|Z\right)$



- Assuming Gaussian measurement likelihood – equivalent to minimizing:

$$J_{BA}\left(X, L\right) \doteq \sum_{i=1}^{N} \sum_{j \in \mathcal{M}_i} \left\| z_{i,j} - \pi\left(x_i, l_j\right) \right\|_{\Sigma}^{2}$$

- Approaches: Gauss-Newton, Levenberg-Marquardt, …

Outline

- Introduction
- Camera projective geometry
- Bundle Adjustment
- Incremental Smoothing and Mapping (iSAM) algorithms
- Visual-inertial SLAM and IMU pre-integration concept

Back to visual SLAM & Vision Aided Navigation

- SfM, BA
  - Sensors: camera (monocular/stereo)
  - Images can be unordered (e.g. downloaded from internet)
  - Cameras are sometimes uncalibrated

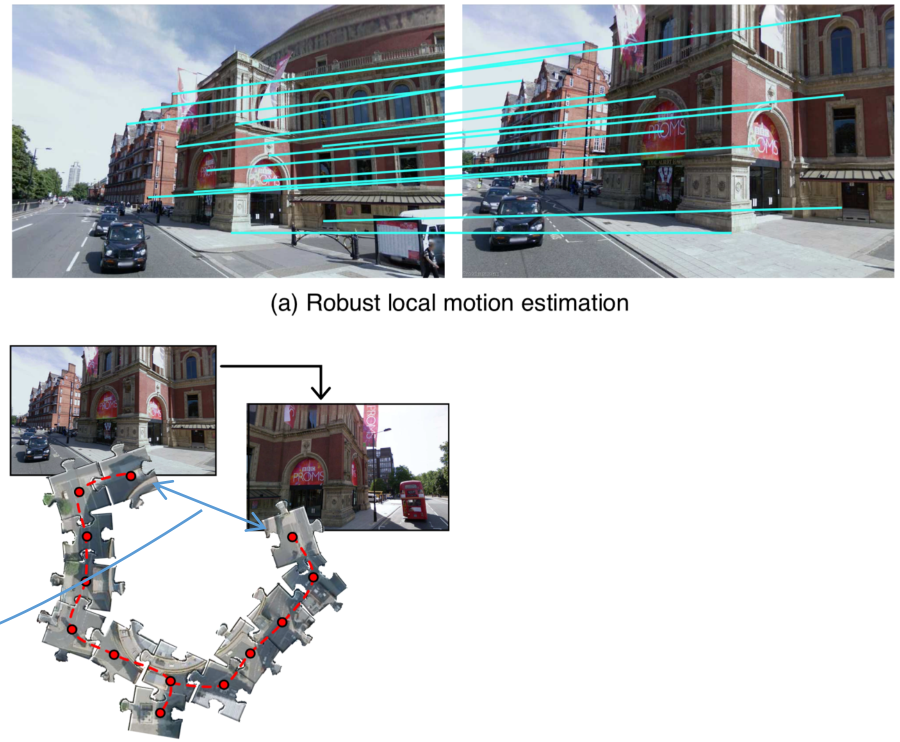
- SLAM & VAN
  - Variety of sensors: camera (monocular/stereo), laser scanner, odometry (IMU, wheel ..)
  - Imagery typically arrives in order (sequential)
  - **Online** operation is required

Loop Closure Observations

- Loop closure observations: Re-observation of a scene
- Essential for reducing drift – resets estimation errors to prior levels
- Challenging to identify

The same scene!!

Estimation error



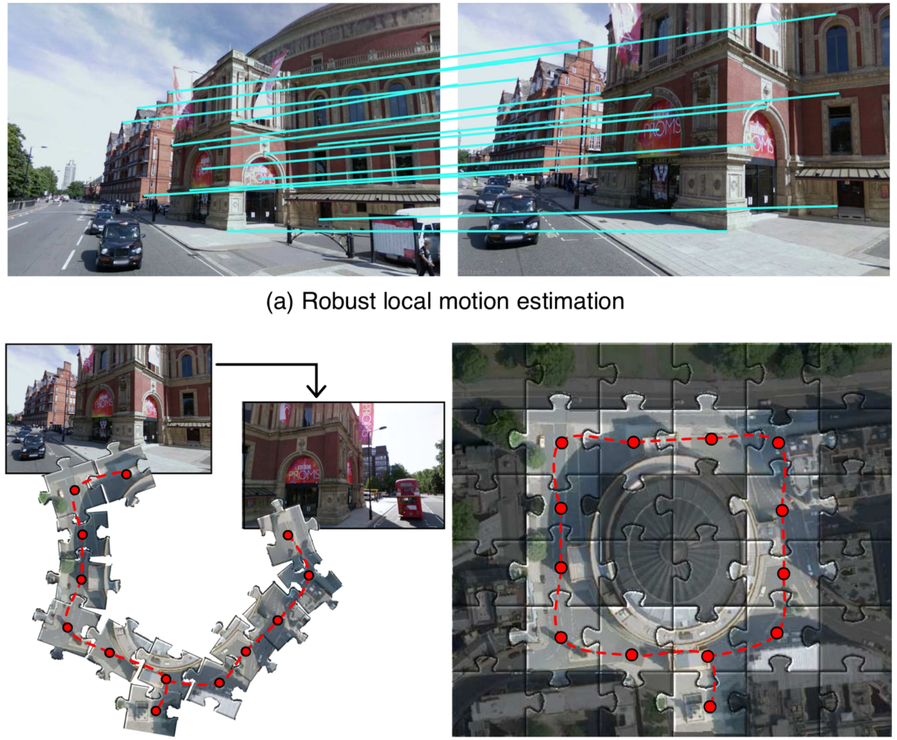
(a) Robust local motion estimation

(b) Mapping and loop-closure detection

Images from: Chli09thesis – "Applying Information Theory to Efficient SLAM", 2009

Loop Closure Observations

- Loop closure observations: Re-observation of a scene
- Essential for reducing drift – resets estimation errors to prior levels
- Challenging to identify



(a) Robust local motion estimation

(b) Mapping and loop-closure detection

(c) Global optimisation

- Measurement equation:

$$z_{k,j} = h\left(x_k, l_j\right) + v$$

3D point that has been observed some time in the past

Images from: Chli09thesis – “Applying Information Theory to Efficient SLAM”, 2009

SLAM – Main Approaches

- Approaches differ in
  - Inference/Filtering techniques
  - Definition of what is estimated (state vector)

Common Inference Approaches

- EKF
- EIF (information form)
- Sparsity-aware optimization
- Particle filters

Latent Variables (State vector)

- Current state (e.g. pose) + landmarks
- Current state + past poses + landmarks
- Current state + past poses

Full SLAM

Pose SLAM

- Deep learning …

Square Root Smoothing and Mapping (SAM)

Dellaert IJRR 2006

Smoothing and Mapping (SAM)

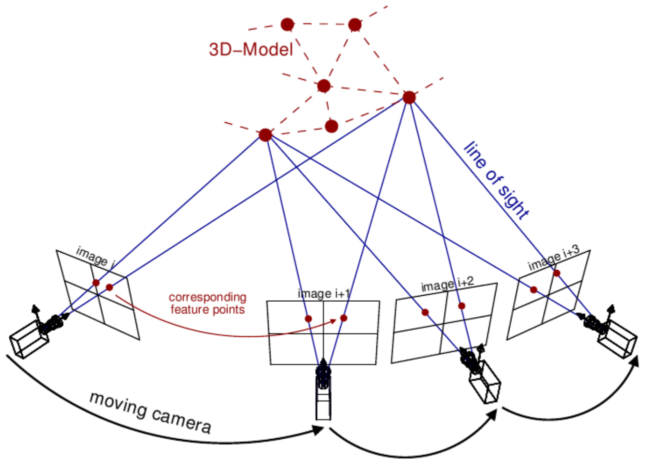
- Camera (or laser) observation model and likelihood

$$z_{k,j} = h\left(x_k, l_j\right) + v \qquad\qquad p\left(z_{k,j} | x_k, l_j\right)$$

- 3D points l_j are random variables (unknown/uncertain environment)
- Need to be estimated, part of the inference process (similarly to SfM)

- Joint pdf:

$$p\left(x_{0:k}, L_k | u_{0:k-1}, z_{0:k}\right)$$



Smoothing and Mapping (SAM)

- Full joint pdf:

$$p\left(x_{0:k}, L_k | u_{0:k-1}, z_{0:k}\right) = \eta p\left(x_0\right) \prod_i \left[p\left(x_i | x_{i-1}, u_{i-1}\right) \prod_{j \in \mathcal{M}_i} p\left(z_{i,j} | x_i, l_j\right) \right]$$

Similarity to BA?

- Maximum a posteriori (MAP) inference:

$$x^\star_{0:k}, L^\star_k = \underset{x_{0:k}, L_k}{\arg\max}\, p\left(x_{0:k}, L_k | u_{0:k-1}, z_{0:k}\right)$$

- For Gaussian distributions, involves solving a nonlinear least-squares problem:

$$x^\star_{0:k}, L^\star_k = \underset{x_{0:k}, L_k}{\arg\min} \left\{ \|x_0 - \hat{x}_0\|^2_{\Sigma_0} + \sum_i \left[\|x_i - f\left(x_{i-1}, u_{i-1}\right)\|^2_{\Sigma_w} + \sum_{j \in \mathcal{M}_i} \|z_{i,j} - h\left(x_i, l_j\right)\|^2_{\Sigma_v} \right] \right\}$$

Smoothing and Mapping (SAM)

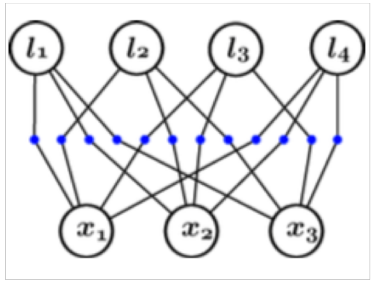
$$x_{0:k}^{\star}, L_k^{\star} = \arg\min_{x_{0:k}, L_k} \left\{ \|x_0 - \hat{x}_0\|_{\Sigma_0}^2 + \sum_i \left[\|x_i - f\left(x_{i-1}, u_{i-1}\right)\|_{\Sigma_w}^2 + \sum_{j \in \mathcal{M}_i} \|z_{i,j} - h\left(x_i, l_j\right)\|_{\Sigma_v}^2 \right] \right\}$$

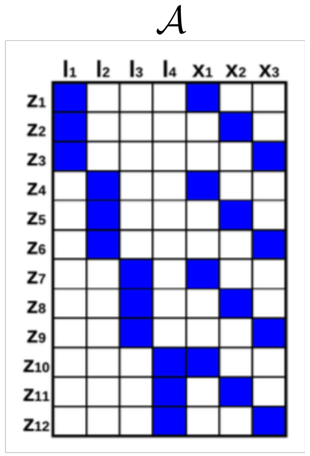
- Define $\Theta \doteq \{x_{0:k}, L_k\}$, linearize and collect terms

$$\Delta\Theta = \arg\min_{\Delta\Theta} \left\| \mathcal{A}\Delta\Theta - \breve{b} \right\|^2$$

- Jacobian $\mathcal{A}$ is a **big** & **sparse** matrix

- Example (camera-only, no motion model):
  - 3 cameras
  - 4 landmarks (3D points)





Smoothing and Mapping (SAM)

$$x_{0:k}^{\star}, L_k^{\star} = \underset{x_{0:k}, L_k}{\arg\min} \left\{ \|x_0 - \hat{x}_0\|_{\Sigma_0}^2 + \sum_i \left[\|x_i - f(x_{i-1}, u_{i-1})\|_{\Sigma_w}^2 + \sum_{j \in \mathcal{M}_i} \|z_{i,j} - h(x_i, l_j)\|_{\Sigma_v}^2 \right] \right\}$$

- Define $\Theta \doteq \{x_{0:k}, L_k\}$, linearize and collect terms

$$\Delta\Theta = \underset{\Delta\Theta}{\arg\min} \left\| \mathcal{A}\Delta\Theta - \breve{b} \right\|^2$$

- Jacobian $\mathcal{A}$ is a **big** & **sparse** matrix
- How to recover MAP estimate efficiently, online?
- Sparsity-aware (incremental) optimization:
  - Solve via factorization (e.g. QR) and back-substitution
  - Update linearization point and repeat process until convergence

Solution via QR factorization

- Least squares problem: $$\Delta\Theta = \underset{\Delta\Theta}{\arg\min} \left\| \mathcal{A}\Delta\Theta - \breve{b} \right\|^2$$

- QR factorization: $$Q^T \mathcal{A} = \begin{bmatrix} R \\ 0 \end{bmatrix} \qquad Q^T \breve{b} \doteq \begin{bmatrix} d \\ e \end{bmatrix}$$

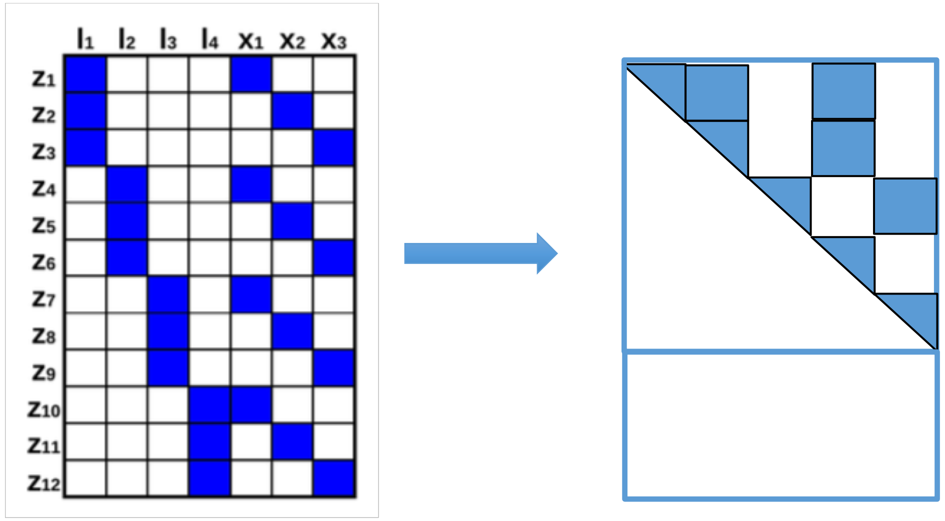
R - Upper triangular (sparse) matrix

Q - Orthogonal matrix

- It can be shown that: $$\left\| \mathcal{A}\Delta\Theta - \breve{b} \right\|_2^2 = \| R\Delta\Theta - d \|_2^2 + \underline{\| e \|_2^2}$$

Least-squares residual

$\mathcal{A}$



$R \in \mathbb{R}^{n \times n}$

Solution via QR factorization

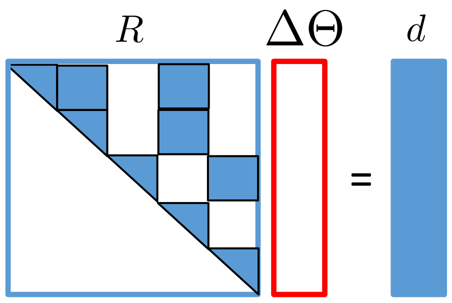
- Least squares problem: $\Delta\Theta = \arg\min_{\Delta\Theta} \left\| \mathcal{A}\Delta\Theta - \breve{b} \right\|^2$

- QR factorization: $Q^T\mathcal{A} = \begin{bmatrix} R \\ 0 \end{bmatrix}$ $\quad Q^T\breve{b} \doteq \begin{bmatrix} d \\ e \end{bmatrix}$

R - Upper triangular (sparse) matrix
Q - Orthogonal matrix

- Least squares (LS) solution $\Delta\Theta^\star$ is obtained via **back-substitution**:

$$R\Delta\Theta = d$$



Graphical Model Perspective

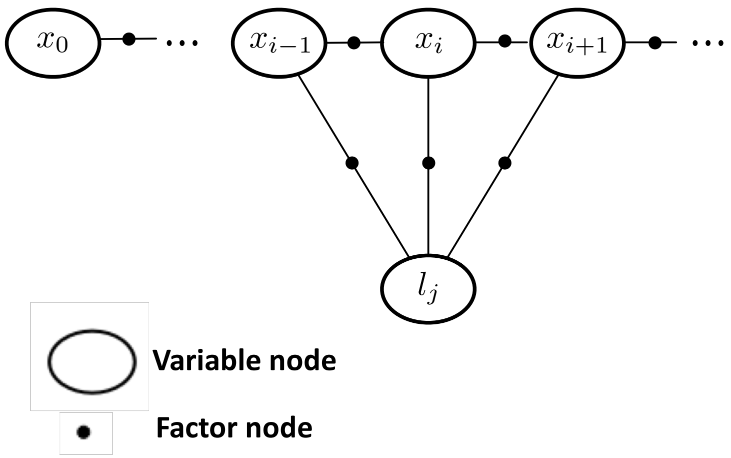
Factor Graph

- Bipartite undirected graph $G(\mathcal{F}, \Theta, \mathcal{E})$ with two node types
  - $\theta_j \in \Theta$: Variable nodes (correspond to states to be inferred)
  - $f_i \in \mathcal{F}$: Factor nodes (associated with process and measurement models)
  - $e_{ij} \in \mathcal{E}$: Edges always connect between variable and factor nodes
- Factor graph describes a factorization of the joint pdf in terms of process and measurement models

$$p(\underbrace{x_{0:k}, L}_{\Theta} | u_{0:k-1}, z_{0:k}) = \eta p(x_0) \prod_i \left[p(x_i | x_{i-1}, u_{i-1}) \prod_{j \in \mathcal{M}_i} p(z_{i,j} | x_i, l_j) \right]$$

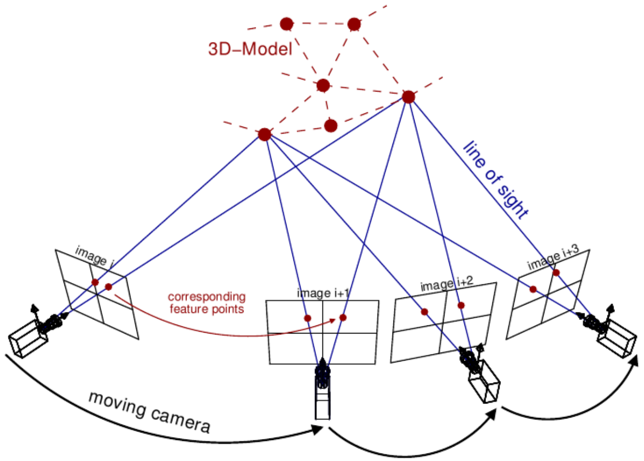
factors $f_i(\Theta_i)$

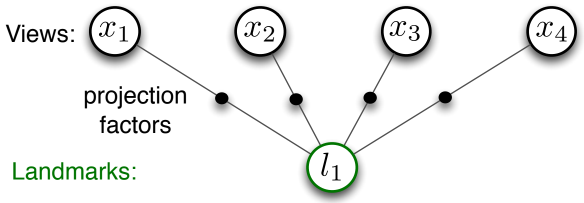
$$\Rightarrow \quad \boxed{p(\Theta) \propto f(\Theta) \propto \prod f_i(\Theta_i)}$$



Factor Graph Representation for BA

$$p\left(X,L|Z\right)\propto\prod_{i}^{N}\prod_{j\in\mathcal{M}_i}\underbrace{p\left(z_{i,j}|x_i,l_j\right)}_{f_{proj}\left(x_i,l_j\right)\doteq\exp\left(-\frac{1}{2}\left\|z_{i,j}-\pi\left(x_i,l_j\right)\right\|_\Sigma^2\right)}$$





Right image from Indelman15ras: “Incremental Light Bundle Adjustment for Structure From Motion and Robotics “

Factor Graph – SLAM Problem

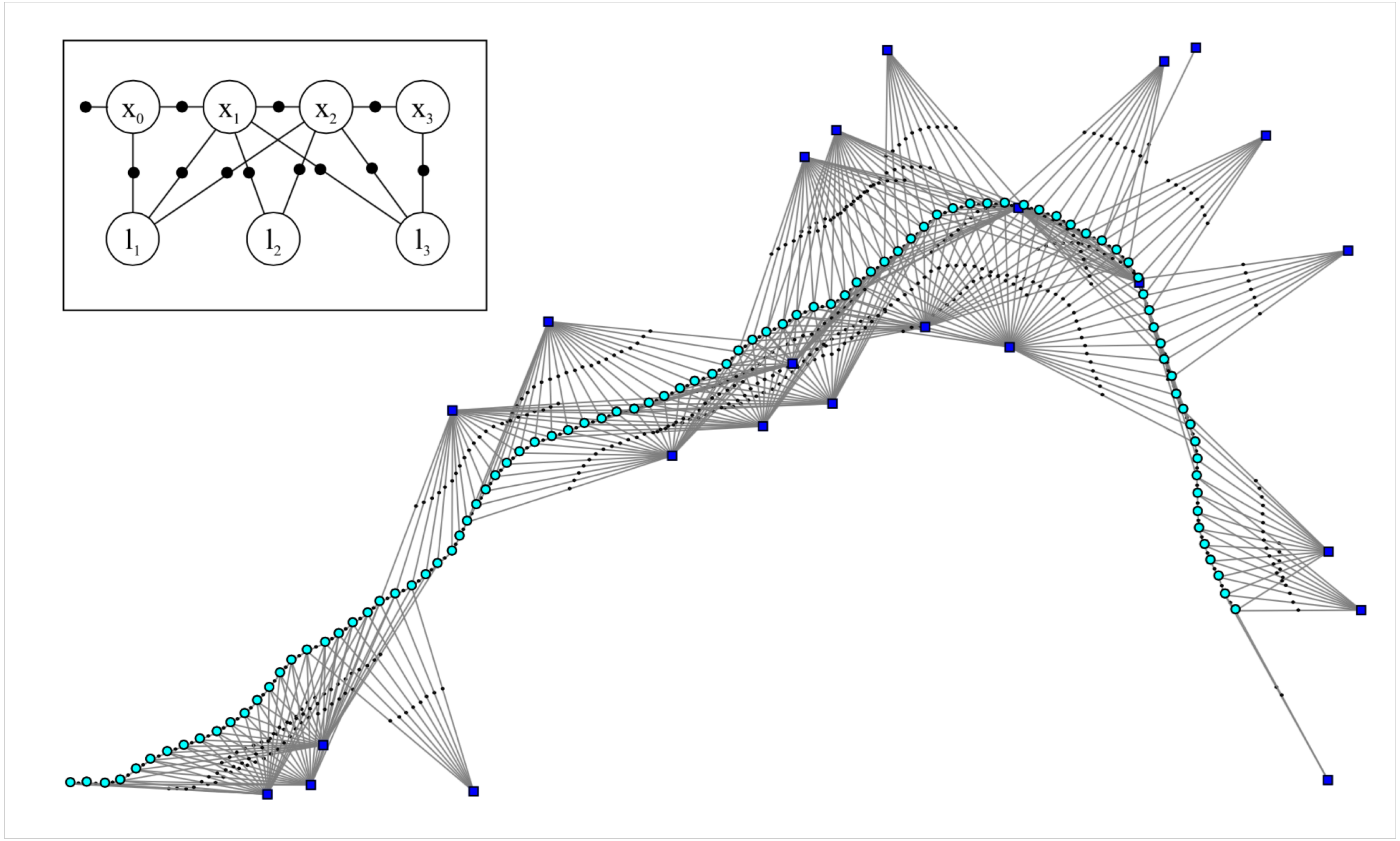
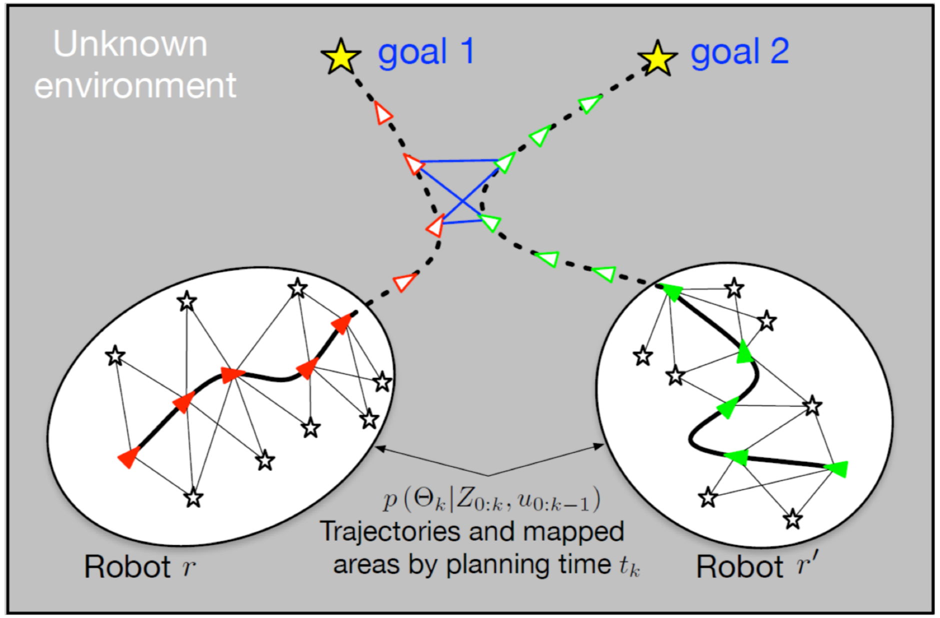
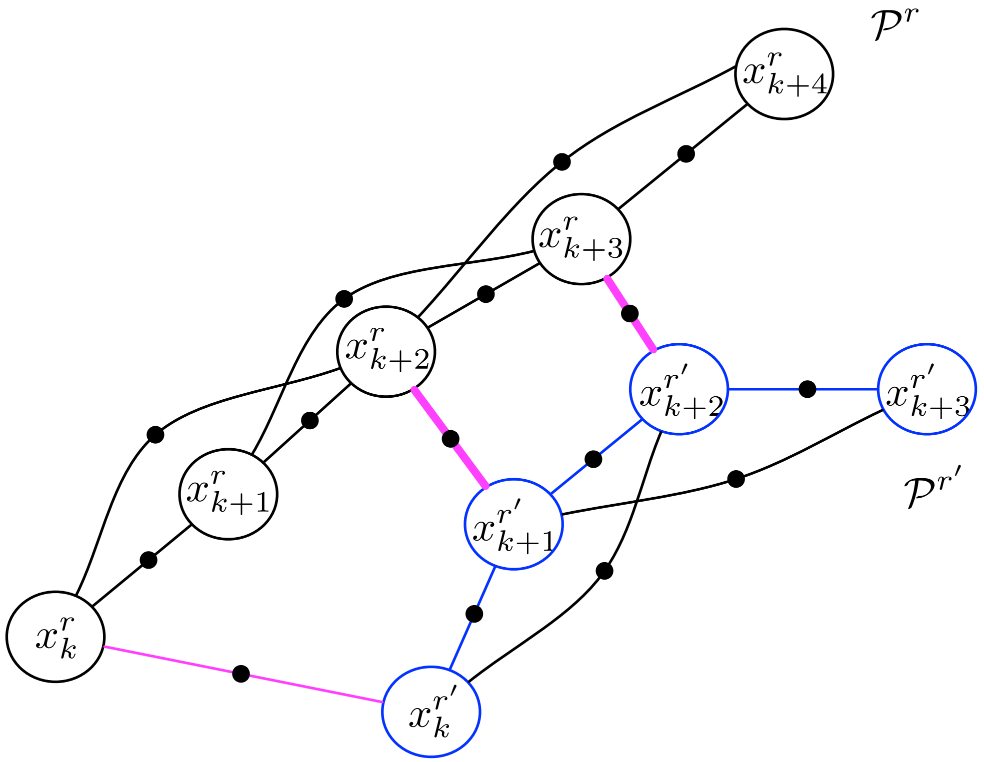


Image from Dellaert06ijrr: “Square Root SAM”

Factor Graph – Multi-Robot SLAM





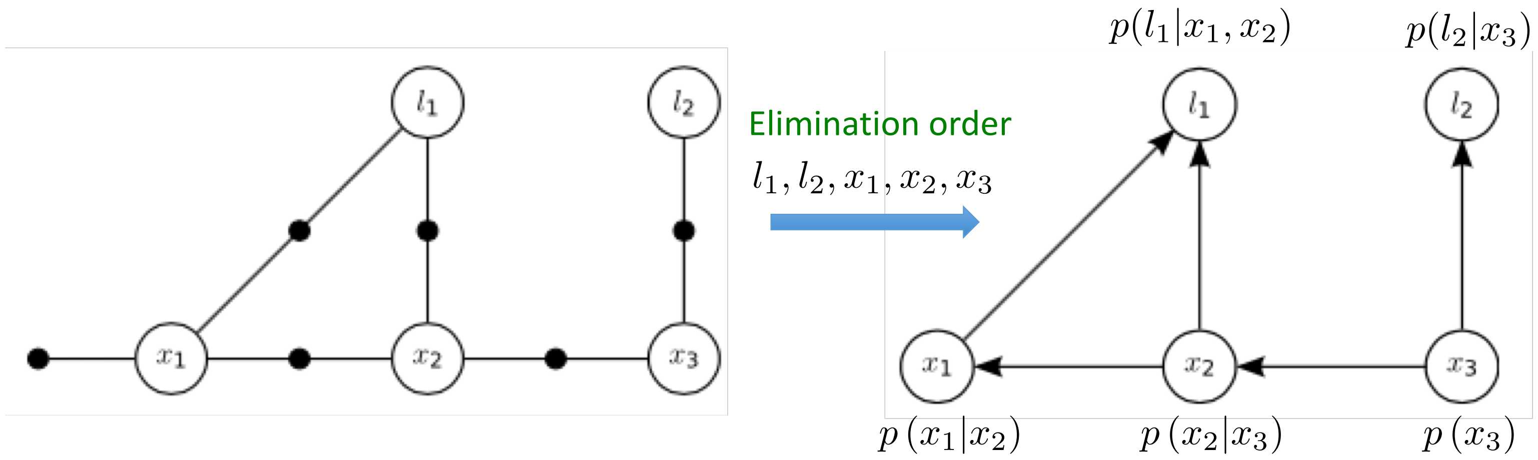
Images from Indelman17arj and Regev17arj

Inference and Variable Elimination

- **Key insight**: Inference == Converting a factor graph to a Bayes net using the elimination alg.

- Factor graph: $f(\Theta) = \prod_j f_j(\theta_j) = f(x_1) f(x_1, x_2) f(x_2, x_3) f(l_1, x_1) f(l_1, x_2) f(l_2, x_3)$
- Represents the joint pdf (e.g.) $p(x_{1:3}, l_{1:2} | u_{1:2}, z_{1:3}) = \eta p(x_1) p(x_2|x_1, u_1) p(x_3|x_2, u_2) p(z_1|x_1, l_1) p(z_2|x_2, l_1) p(z_3|x_3, l_2)$

- Final result – Bayes net, corresponds to the factorization: $p(l_1|x_1, x_2) p(l_2|x_3) p(x_1|x_2) p(x_2|x_3) p(x_3)$



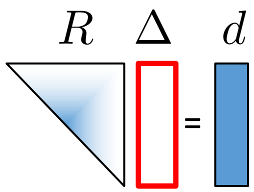
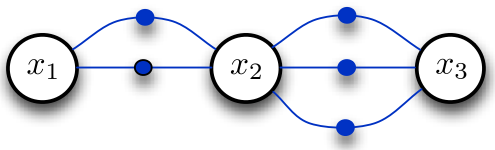


Image from Kaess12ijrr: “iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree”

Example

Factor graph



Linearization

Jacobian matrix

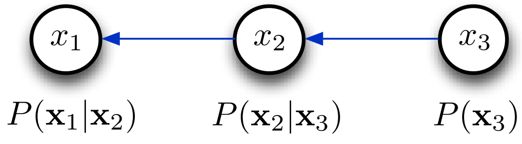
$$A = \begin{array}{c} \begin{bmatrix} \times & \times & \\ \times & \times & \\ & \times & \times \\ & \times & \times \\ & \times & \times \end{bmatrix} \\ \begin{array}{ccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \end{array} \end{array}$$

Linearization and elimination

Elimination order x_1, x_2, x_3

Factorization

Bayes net



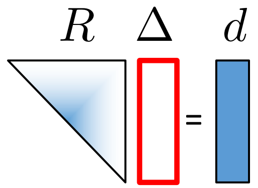
Factorized Jacobian matrix

$$R = \begin{array}{c} \begin{bmatrix} \times & \times & \\ & \times & \times \\ & & \times \end{bmatrix} \\ \begin{array}{ccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \end{array} \end{array}$$

$$\Delta^* = \arg\min_{\Delta} (A\Delta - b)$$

$$A = QR \qquad d \doteq Q^T b$$

$$\Delta^* = \arg\min_{\Delta} (R\Delta - d)$$



Thus Far: Smoothing and Mapping (SAM)

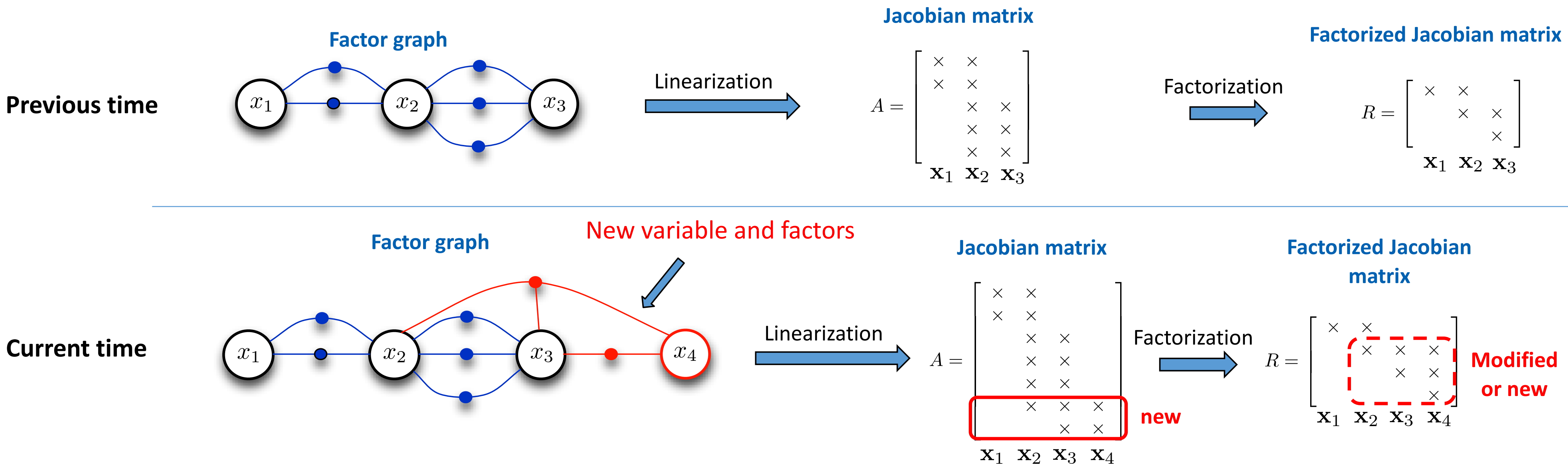
- Efficient, sparisity-aware nonlinear optimization
- Inference in graphical models
  - Joint pdf can be represented by a **factor graph**
  - Factorization (calculating the R matrix) is equivalent to variable elimination, represented by a **Bayes net**

- Still, **batch** algorithm:
  - Each time, solves the entire NLS problem
  - Online performance?

Incremental Smoothing & Mapping (iSAM)

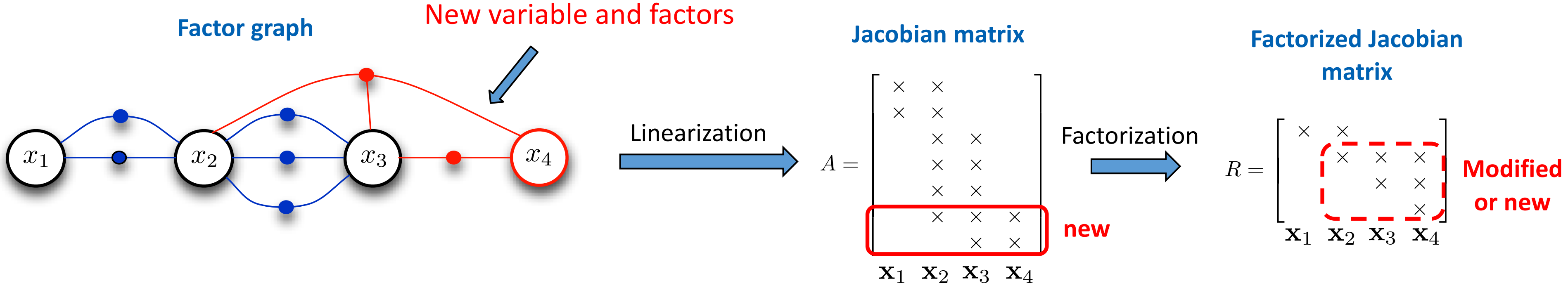
Each time new data is obtained (e.g. measurement):

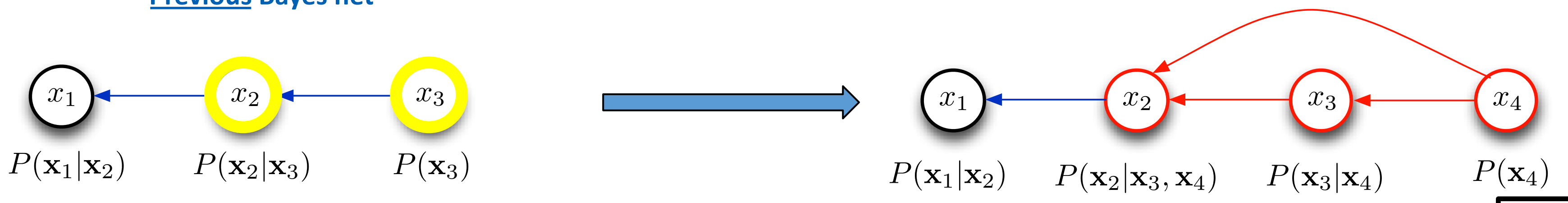
- Previous approach (SAM): performs factorization from scratch
- Is this really required?
  - Typically, only a **very small** subset of entries is updated
  - Key idea – identify and update only these entries, i.e. **update** factorization (e.g. QR update)



Incremental Smoothing & Mapping (iSAM)

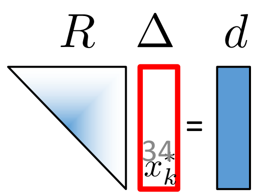
- Adding new variable (camera pose or navigation state) and factors





Only red parts are re-calculated

$$\Delta^* = \arg\min_{\Delta} (A\Delta - b)$$
$$A = QR \qquad d \doteq Q^T b$$
$$\Delta^* = \arg\min_{\Delta} (R\Delta - d)$$



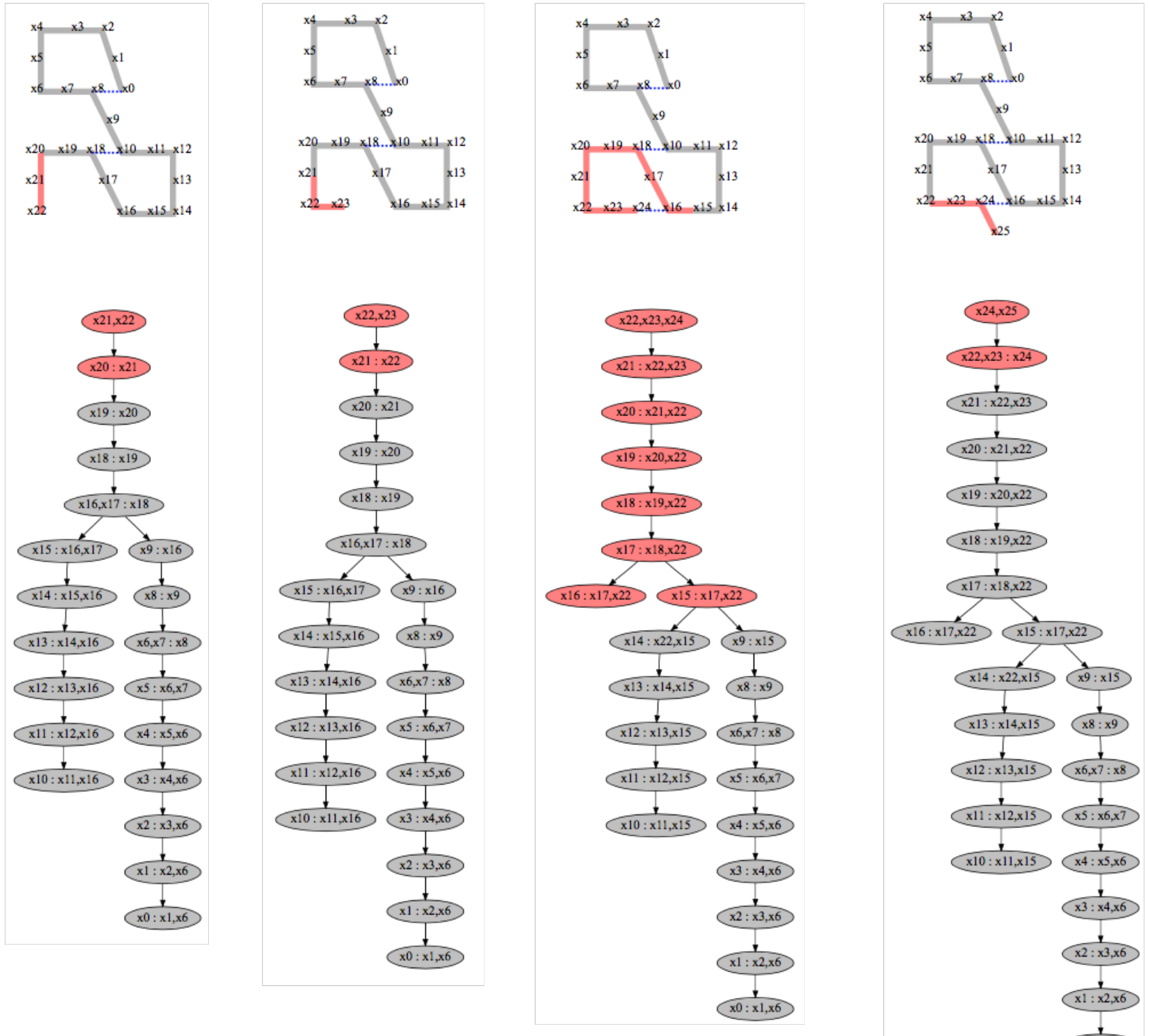
- What should be re-calculated?
- Nodes in all paths that lead from the last-eliminated node to nodes involved in new factors

Incremental Smoothing & Mapping (iSAM)

time

Trajectory:

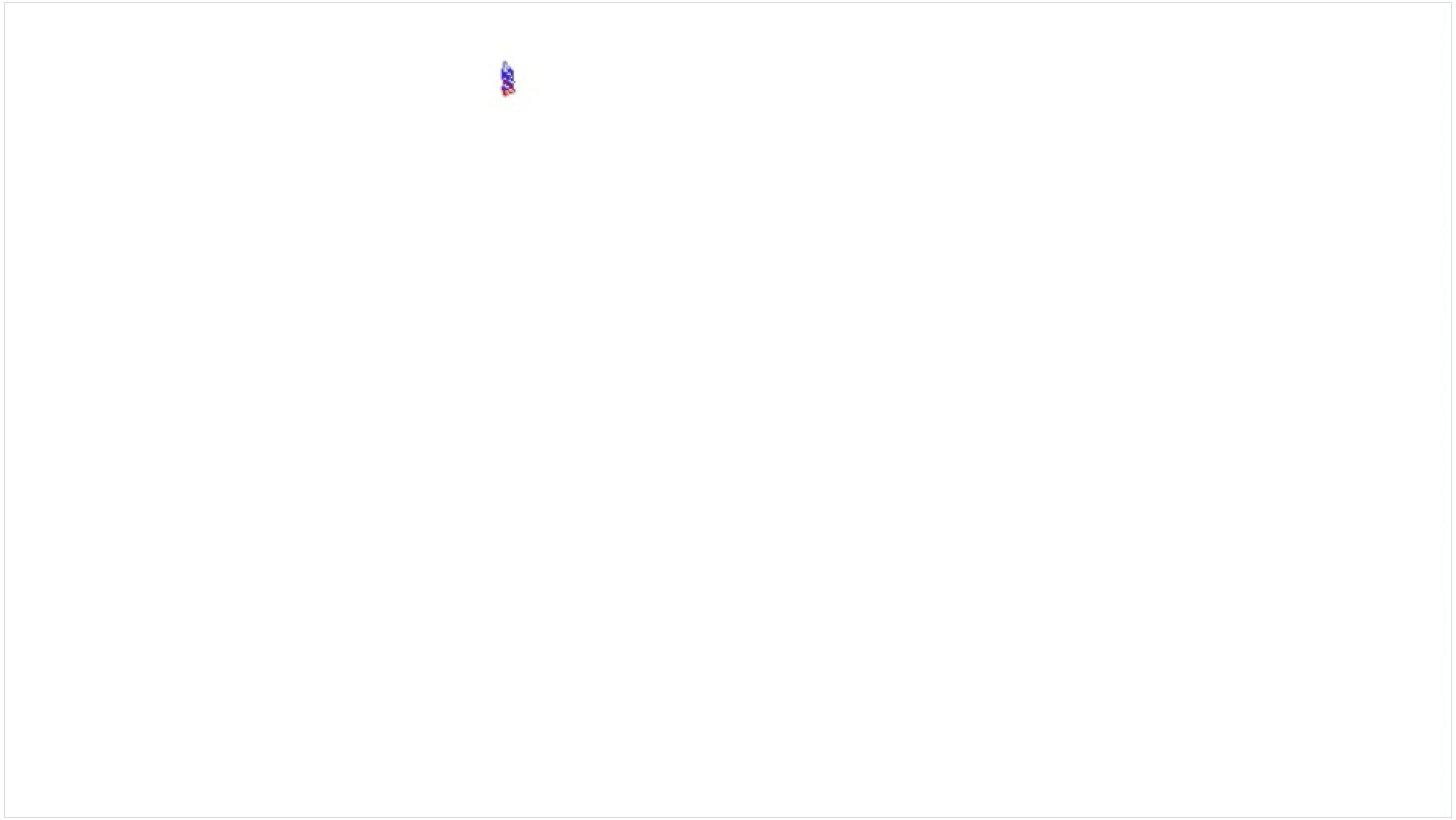
Bayes tree:
(calculated from Bayes net)



Only red parts are re-calculated

Image from Kaess12ijrr: "iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree"

Example: Incremental Light Bundle Adjustment



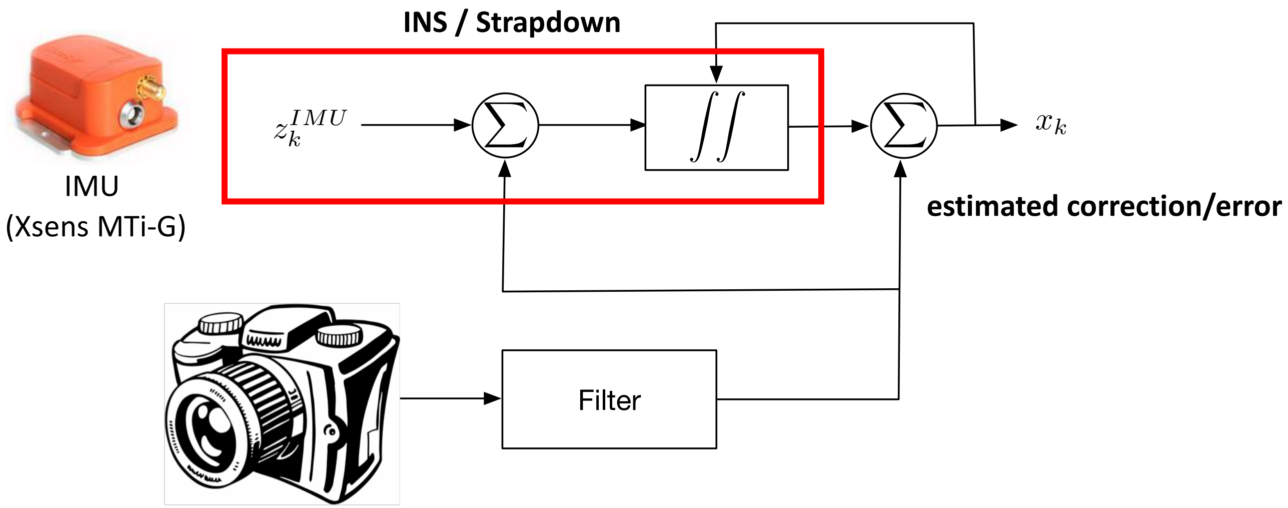
Indelman12bmvc: “Incremental Light Bundle Adjustment”, 2012

Outline

- Introduction
- Camera projective geometry
- Bundle Adjustment
- Incremental Smoothing and Mapping (iSAM) algorithms
- Visual-inertial SLAM and IMU pre-integration concept

Vision-Aided Navigation (VAN)

- Common approach:
  - Integrate IMU measurements **outside** the filter (in real time)
  - Use external sensors to correct solution
  - Does not support re-linearization (of past IMU measurements)



Vision-Aided Navigation (VAN)

- Common approach:
  - Integrate IMU measurements **outside** the filter (in real time)
  - Use external sensors to correct solution
  - Does not support re-linearization (of past IMU measurements)

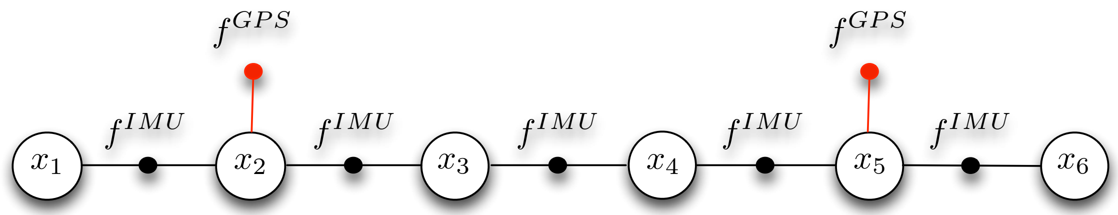
- An alternative – **visual-inertial bundle adjustment**
  - Concept: one big optimization
  - Better accuracy, improved estimation consistency
  - Real time performance?
    - Incremental factorization
    - IMU Pre-Integration

Information Fusion via Incremental Smoothing (iSAM algorithms)

Indelman et al. RAS 2013

Information Fusion via iSAM2.0

- The same concept applies also to multiple sensors, possibly operating at different rates
- The joint pdf and the factor graph should be accordingly
- Information fusion from different sensors:
  - Sensors introduce appropriate factors to the graph
  - Calculate MAP estimate via incremental smoothing (iSAM)
- Conceptually, factor graph representing IMU and GPS measurements:

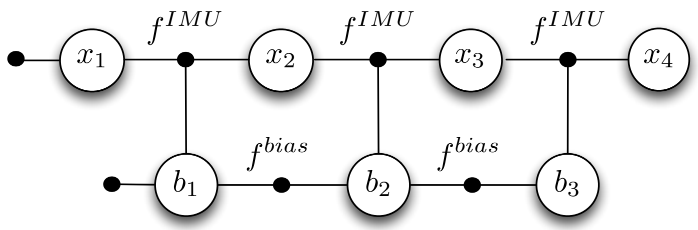


IMU Only – Inertial Navigation

- Joint pdf
  - Formulated in terms of discrete inertial navigation equations (numerical integration)
  - Considers a basic model for IMU calibration parameters (for simplicity)

$$p\left(X_k, B_k | Z_k\right) \propto \prod_i^k \underbrace{p\left(x_{k+1} | x_k, b_k, z_k^{IMU}\right)}_{\doteq f^{IMU}} \underbrace{p\left(b_{k+1} | b_k\right)}_{\doteq f^{bias}}$$

Factor graph



IMU Only – Inertial Navigation

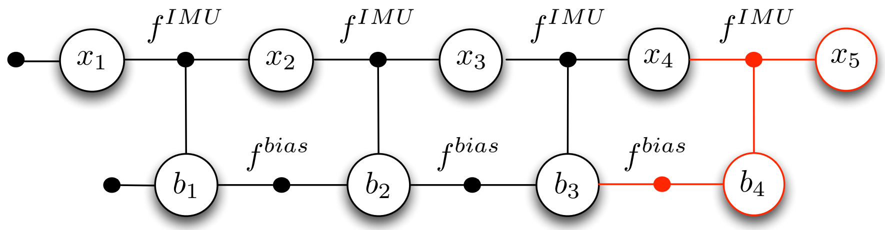
- Joint pdf
  - Formulated in terms of discrete inertial navigation equations (numerical integration)
  - Considers a basic model for IMU calibration parameters (for simplicity)

$$p\left(X_k, B_k | Z_k\right) \propto \prod_i^k \underbrace{p\left(x_{k+1} | x_k, b_k, z_k^{IMU}\right)}_{\doteq f^{IMU}} \underbrace{p\left(b_{k+1} | b_k\right)}_{\doteq f^{bias}}$$

– Assume MAP estimate at time t_k has been calculated

– What involves recovering MAP estimate at time t_{k+1} ?

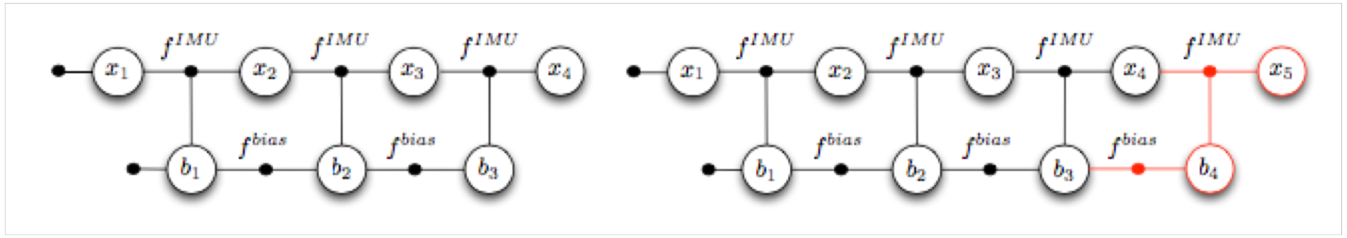
Factor graph

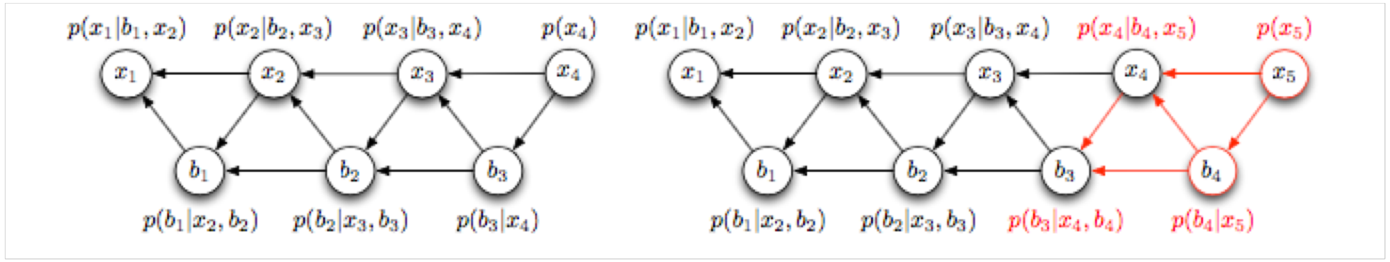


IMU Only – Inertial Navigation

- Corresponding factor graph and Bayes net:

Only 2 last variables should be re-eliminated





… but, what happens in presence of additional sensors?

Images from Indelman15chapter: “Incremental Light Bundle Adjustment: Probabilistic Analysis and Application to Robotic Navigation”, 2015

Visual-Inertial Bundle Adjustment (SLAM, VAN)

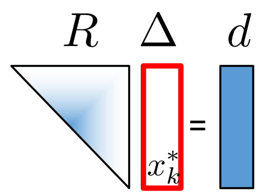
- IMU + single camera:

$$p\left(X_k, B_k, L_k | Z_k\right) \propto \prod_i^k \underbrace{p\left(x_{k+1} | x_k, b_k, z_k^{IMU}\right)}_{\doteq f^{IMU}} \underbrace{p\left(b_{k+1} | b_k\right)}_{\doteq f^{bias}} \prod_{j \in \mathcal{M}_i} \underbrace{p\left(z_{i,j} | x_i, l_j\right)}_{\doteq f^{proj}}$$

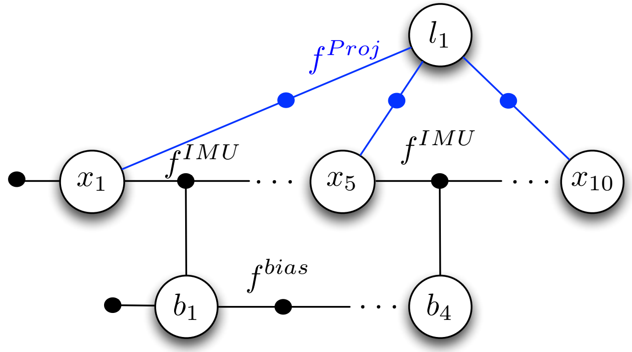
$$\Delta^* = \arg\min_{\Delta} \left(A\Delta - b\right)$$

$$A = QR \qquad d \doteq Q^T b$$

$$\Delta^* = \arg\min_{\Delta} \left(R\Delta - d\right)$$

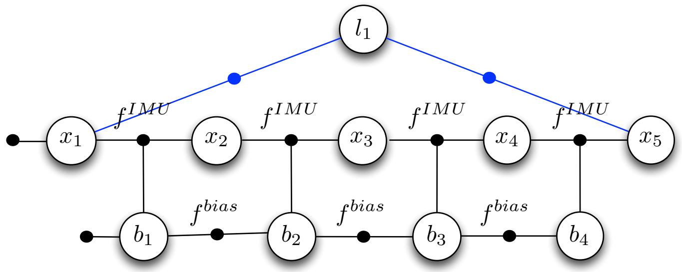


Factor graph



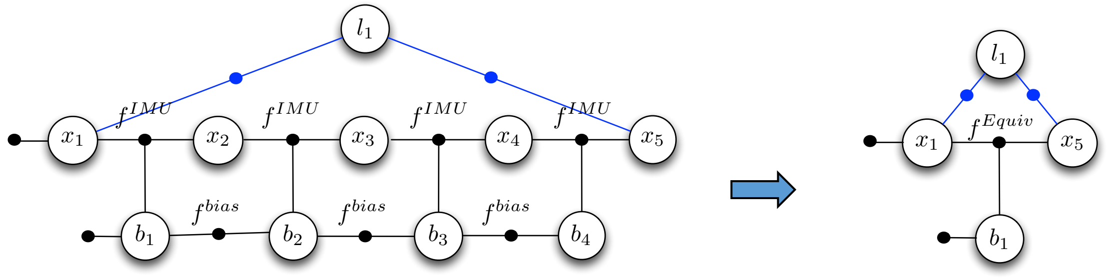
Incorporating High Rate Sensors

- Challenges
  - Many variables to re-eliminate each time new measurements from other sensors (e.g. camera) come in
  - Recover MAP estimate at IMU rate?
  - How to avoid adding state variables to the optimization at IMU rate?



IMU Pre-Integration

- How to avoid adding state variables to the optimization at IMU rate?
  - **Pre-integrate** IMU observations in body frame of last keyframe [Lupton et al. 2012]
  - Navigation states can be added at camera rate



IMU Pre-Integration

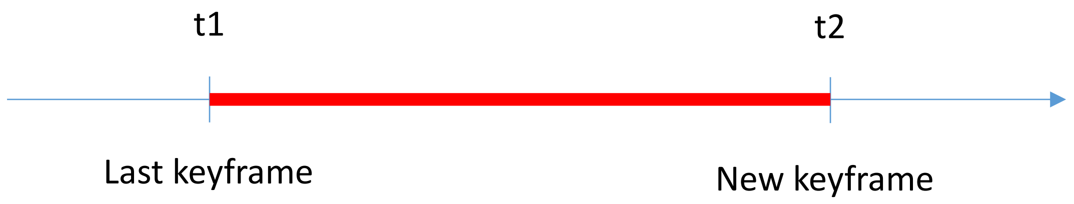
- In navigation (global) frame:

$$v^n_{t2} = v^n_{t1} + \int_{t1}^{t2} (C^n_{bt}(f^b_t - bias^{obs}_f) + g^n)dt$$

Similar concept for position & orientation

- Instead - **in body frame of the last pose** within optimization:

$$v^n_{t2} = v^n_{t1} + \int_{t1}^{t2} g^n \, dt + C^n_{bt1} \int_{t1}^{t2} \left(C^{bt1}_{bt} \left(f^b_t - bias^{obs}_f\right)\right) dt$$



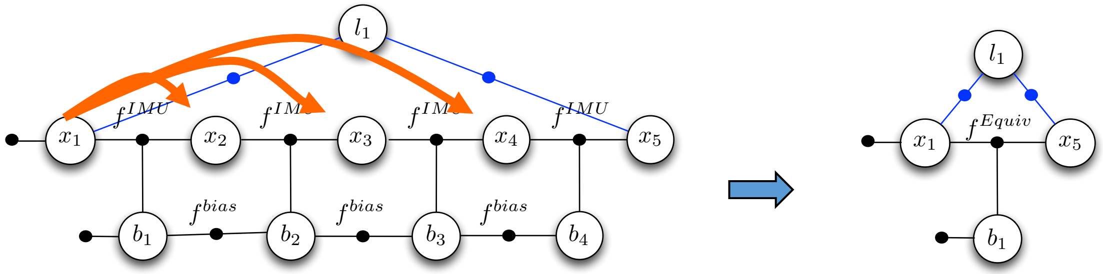
Images from Lupton12tro: “Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions ”

IMU Pre-Integration

- How to avoid adding state variables to the optimization at IMU rate?
  - **<u>Pre-integrate</u>** IMU observations in body frame of last keyframe [Lupton et al. 2012]
  - Navigation states can be added at camera rate

- Real time performance – predict solution using pre-integrated IMU information and current MAP estimate
  – Without adding new variables to optimization

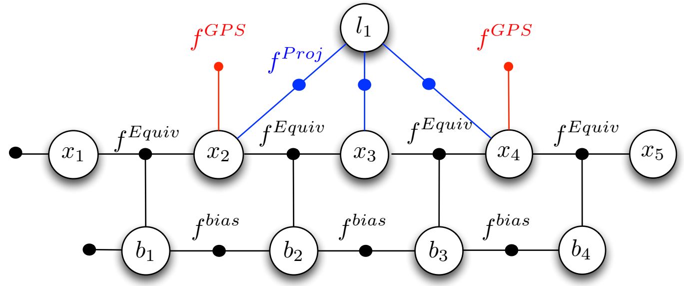
- Expressing in relative frame (i.e. body frame) allows to avoid re-playing all observations when re-linearizing!



Available as open source!
(part of gtsam)

Information Fusion via iSAM2.0

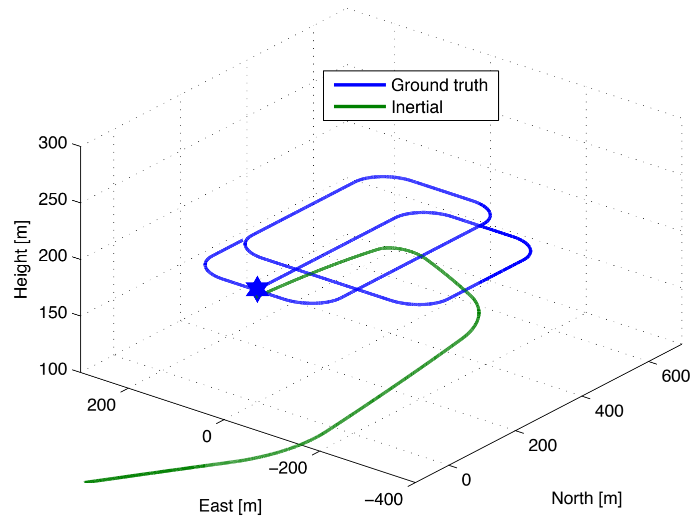
- Any other sensors can be similarly incorporated
  For example:
  - IMU
  - GPS
  - Camera + explicit estimation of 3D points



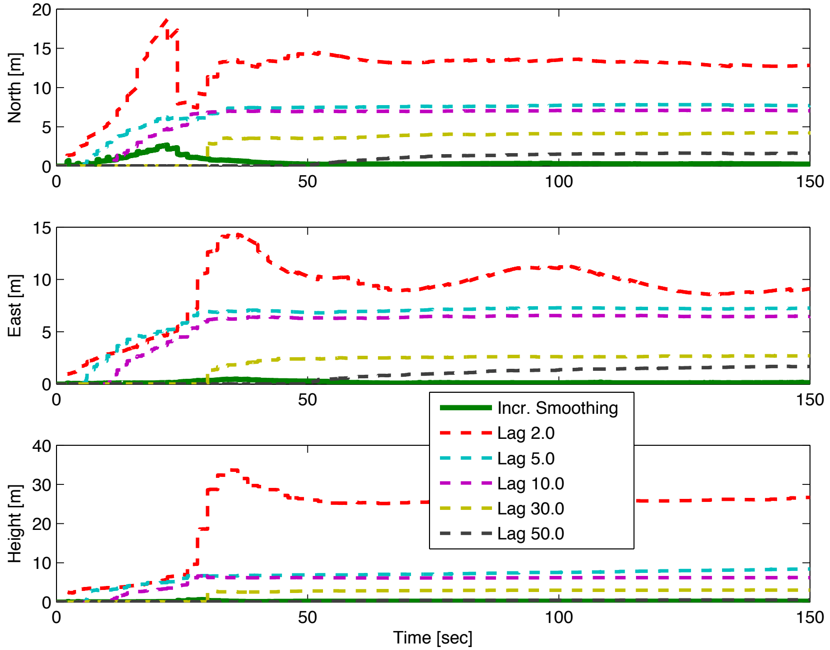
Information Fusion via iSAM2.0

- Aerial scenario (simulation, Monte-Carlo study)
  - IMU, Monocular camera, Magnetometer
  - <u>Short-track</u> features only
  - Initial navigation errors

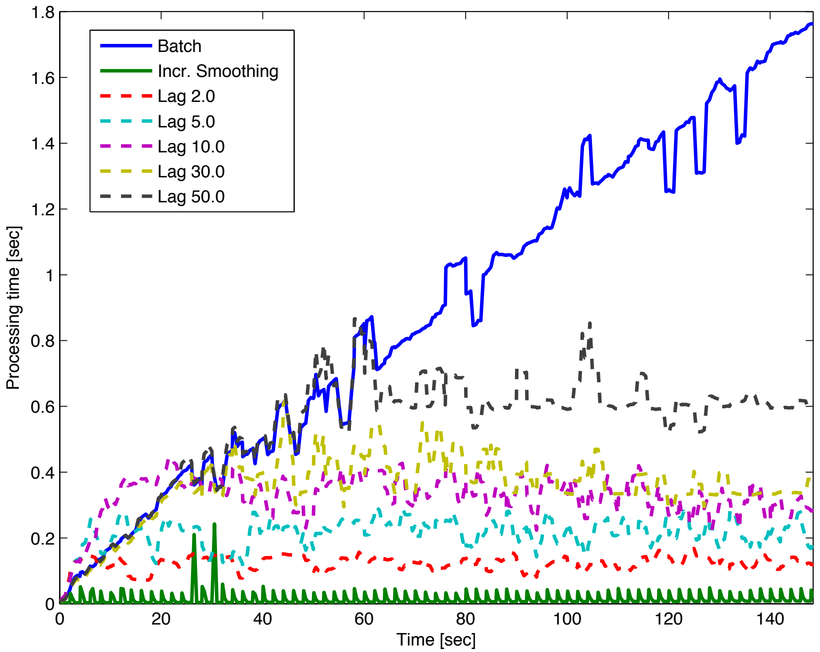
$$X_k^*, B_k^*, L_k^* = \underset{X_k, B_k, L_k}{\arg\min} \; J\left(X_k, B_k, L_k\right)$$



Position – RMS difference w.r.t. Batch



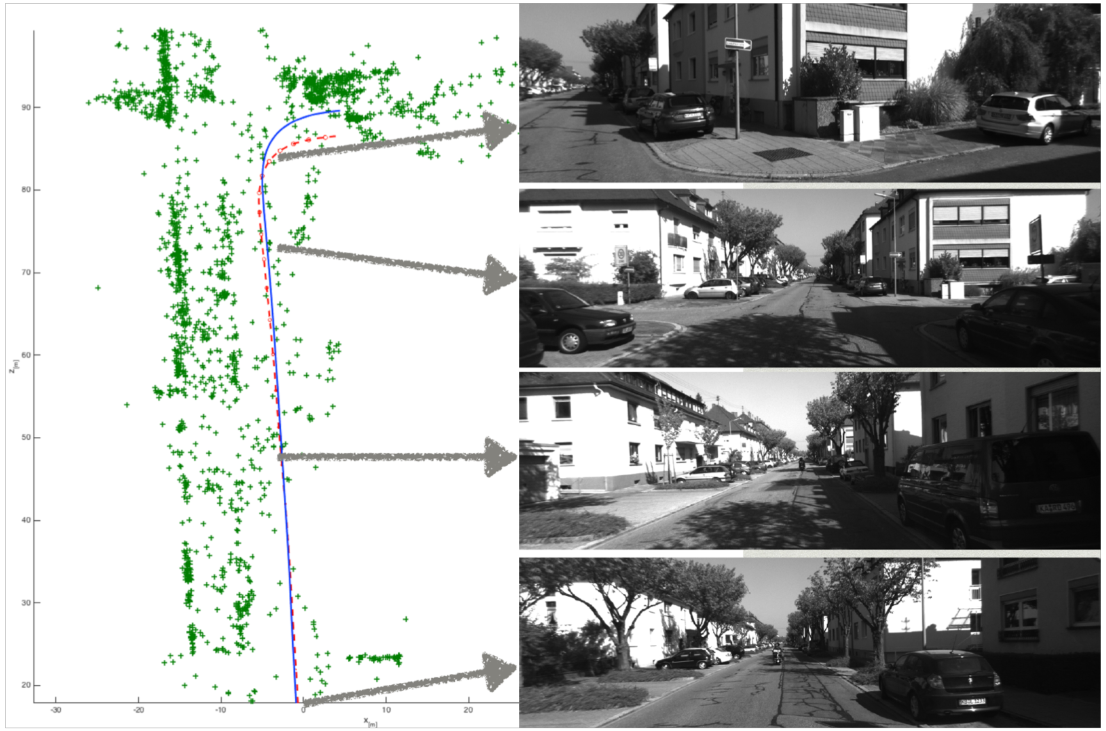
Processing time



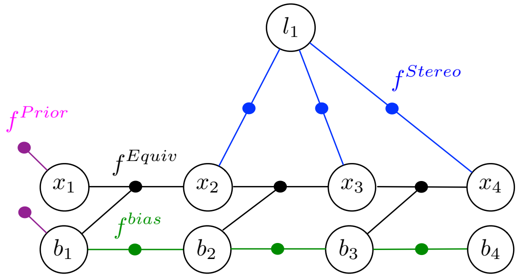
Images from Indelman13ras: "Information Fusion in Navigation Systems via Factor Graph Based Incremental Smoothing"

Information Fusion via iSAM2.0

- KITTI Vision Benchmark
  - IMU
  - Stereo Camera (no loop closures)



(Simplified) Factor Graph

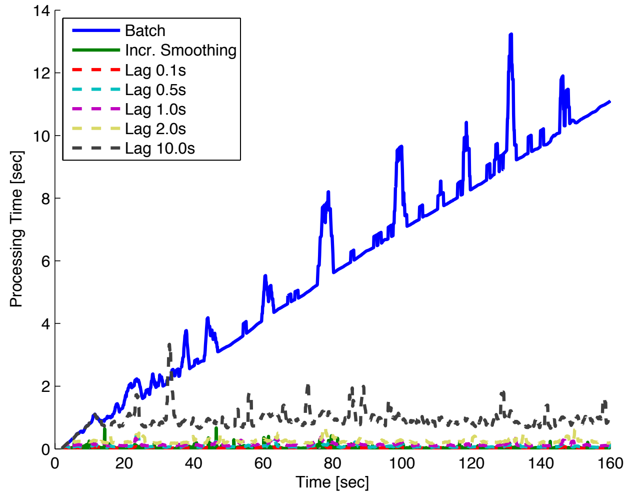


Information Fusion via iSAM2.0

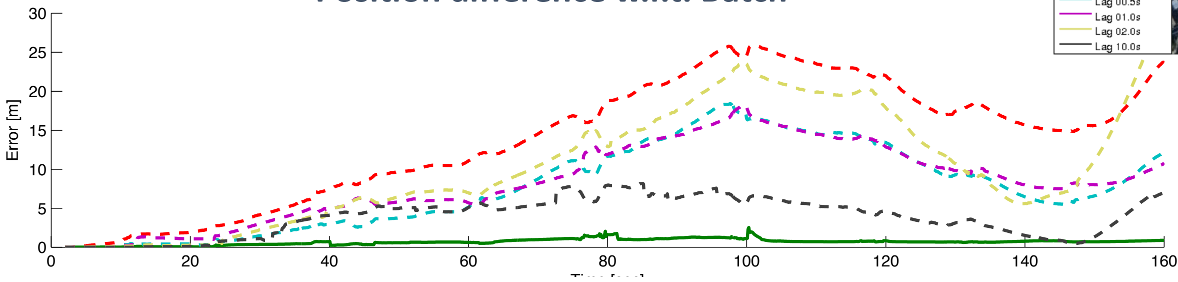
- KITTI Vision Benchmark
  - IMU
  - Stereo Camera (no loop closures)

$$X_k^*, B_k^*, L_k^* = \underset{X_k, B_k, L_k}{\arg\max}\, p\left(X_k, B_k, L_k | Z_k\right)$$

Processing time



Position difference w.r.t. Batch



Images from Indelman13ras: "Information Fusion in Navigation Systems via Factor Graph Based Incremental Smoothing"

TECHNION Israel Institute of Technology

Summary

SLAM and VAN Overview:

- Front-end & Back-end
- Projection operator, re-projection error
- Bundle adjustment
- Smoothing and Mapping (SAM)
- Incremental SAM (iSAM)
- Information fusion with iSAM
- IMU Pre-Integration