Bundle Adjustment

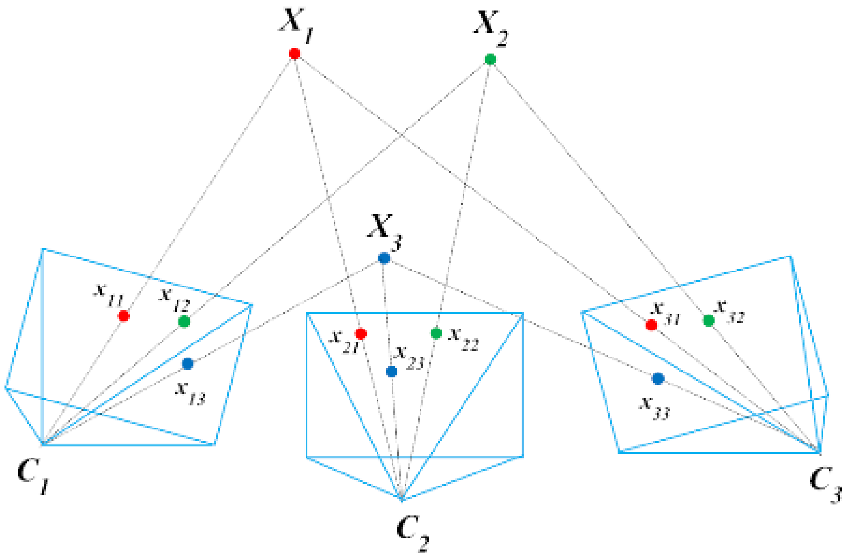
刘浩敏

Baidu 百度

Bundle Adjustment

- Jointly optimize all cameras and points

$$\underset{C_1,\ldots C_{N_c},X_1,\ldots,X_{N_p}}{\arg\min} \sum \left\| \pi(X_i, C_j) - x_{ij} \right\|^2$$



Triggs, B., Mclauchlan, P., Hartley, R., and Fitzgibbon, A. 1999. Bundle adjustment—a modern synthesis. In Proceedings of the International Workshop on Vision Algorithms: Theory and Practice. 298–372.

Nonlinear Least Squares

- Gaussian Newton

$$x^* = \arg\min_x \|\varepsilon(x)\|^2$$

$$\varepsilon(x^*) = \varepsilon(\hat{x} + \delta_x) \approx \varepsilon(\hat{x}) + J\delta_x$$

$$J = \partial\varepsilon/\partial x\big|_{x=\hat{x}}$$ Jacobian matrix

$$\delta_x = \arg\min_{\delta_x} \|\varepsilon + J\delta_x\|^2$$

$$\boxed{J^T J}\,\delta_x = -J^T\varepsilon$$ first order approximation to Hessian

- Levenberg-Marquardt

$$(J^T J + \mu I)\delta x = -J^T\varepsilon$$

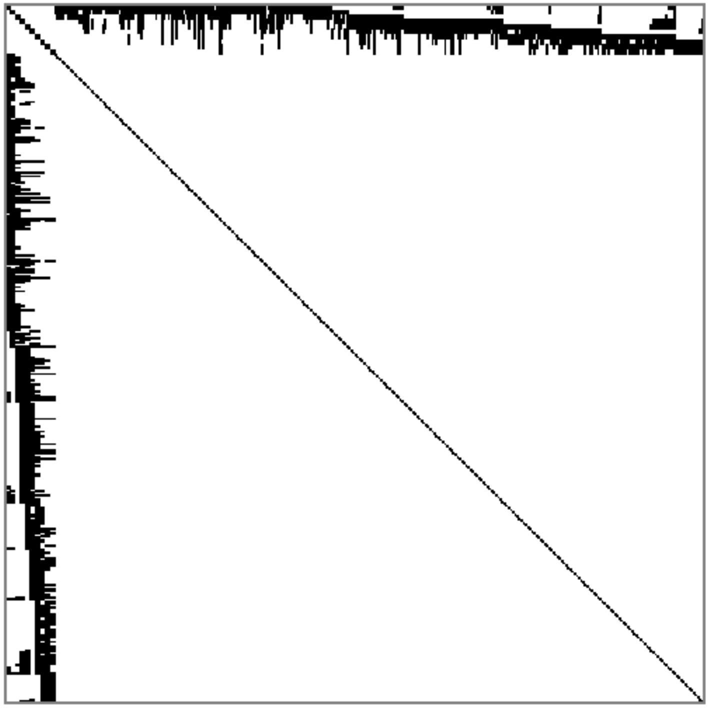
Sparse Bundle Adjustment

$$\underset{C_1,\ldots C_{N_c}, X_1,\ldots,X_{N_p}}{\arg\min} \sum \left\| \pi(X_i, C_j) - x_{ij} \right\|^2$$

1 Point

1 Camera

Sparsity patten of Hessian



Manolis I. A. Lourakis, Antonis A. Argyros: SBA: A software package for generic sparse bundle adjustment. ACM Trans. Math. Softw. 36(1) (2009)

Sparse Bundle Adjustment

- An simple example
  - 4 points
  - 3 cameras
  - all points are visible in all cameras

Sparse Bundle Adjustment

$$
J = \begin{array}{c}
\begin{array}{cc} \overbrace{\qquad\text{3 cameras}\qquad} & \overbrace{\qquad\qquad\text{4 points}\qquad\qquad} \end{array} \\
\begin{pmatrix}
A_{11} & 0 & 0 & B_{11} & 0 & 0 & 0 \\
A_{12} & 0 & 0 & 0 & B_{12} & 0 & 0 \\
A_{13} & 0 & 0 & 0 & 0 & B_{13} & 0 \\
A_{14} & 0 & 0 & 0 & & 0 & B_{14} \\
0 & A_{21} & 0 & B_{21} & 0 & 0 & 0 \\
0 & A_{22} & 0 & 0 & B_{22} & 0 & 0 \\
0 & A_{23} & 0 & 0 & 0 & B_{23} & 0 \\
0 & A_{24} & 0 & 0 & 0 & 0 & B_{34} \\
0 & 0 & A_{31} & B_{31} & 0 & 0 & 0 \\
0 & 0 & A_{32} & 0 & B_{32} & 0 & 0 \\
0 & 0 & A_{33} & 0 & 0 & B_{33} & 0 \\
0 & 0 & A_{34} & 0 & 0 & 0 & B_{34}
\end{pmatrix}
\end{array}
, e = \begin{pmatrix} e_{11} \\ e_{12} \\ e_{13} \\ e_{14} \\ e_{21} \\ e_{22} \\ e_{23} \\ e_{24} \\ e_{31} \\ e_{32} \\ e_{33} \\ e_{34} \end{pmatrix}
$$

Sparse Bundle Adjustment

$$\boxed{J^T J}\delta_x = -J^T \varepsilon$$

$$J^T J = \begin{pmatrix} U & W \\ W^T & V \end{pmatrix} = \begin{pmatrix} U_1 & 0 & 0 & W_{11} & W_{12} & W_{13} & W_{14} \\ 0 & U_2 & 0 & W_{21} & W_{22} & W_{23} & W_{24} \\ 0 & 0 & U_3 & W_{31} & W_{32} & W_{33} & W_{34} \\ W_{11}^T & W_{21}^T & W_{31}^T & V_1 & 0 & 0 & 0 \\ W_{12}^T & W_{22}^T & W_{32}^T & 0 & V_2 & 0 & 0 \\ W_{13}^T & W_{21}^T & W_{33}^T & 0 & 0 & V_3 & 0 \\ W_{14}^T & W_{24}^T & W_{34}^T & 0 & 0 & 0 & V_4 \end{pmatrix}$$

$$U_i = \sum_{j=1}^{4} A_{ij}^T A_{ij}, V_j = \sum_{i=1}^{3} B_{ij}^T B_{ij}, W_{ij} = A_{ij}^T B_{ij}$$

Sparse Bundle Adjustment

$$J^T J \boxed{\delta_x} = -J^T \varepsilon$$

$$d_x = \begin{pmatrix} d_C \\ d_X \end{pmatrix} = \begin{pmatrix} d_{C_1}^T & d_{C_2}^T & d_{C_3}^T & d_{X_1}^T & d_{X_2}^T & d_{X_3}^T & d_{X_4}^T \end{pmatrix}^T$$

Sparse Bundle Adjustment

$$J^T J\delta_x = -\boxed{J^T \varepsilon}$$

$$J^T e = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_1 & u_2 & u_3 & v_1 & v_2 & v_3 & v_4 \end{pmatrix}^T$$

$$u_i = \sum_{j=1}^{4} A_{ij}^T e_{ij}$$

$$v_j = \sum_{i=1}^{3} B_{ij}^T e_{ij}$$

Sparse Bundle Adjustment

- In general, NOT all points are visible in all cameras

$$U_i = \sum_{j=1}^{4} A_{ij}^T A_{ij}, V_j = \sum_{i=1}^{3} B_{ij}^T B_{ij}, W_{ij} = A_{ij}^T B_{ij}$$

  - $A_{ij} = B_{ij} = 0$ if i-th points is invisible (or not matched) in j-th camera
  - More sparse structure, more speed-up

Sparse Bundle Adjustment

$$J^T J \delta_x = -J^T \varepsilon$$

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

$$\begin{pmatrix} U - WV^{-1}W^T & 0 \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u - WV^{-1}v \\ v \end{pmatrix}$$

$S = U - WV^{-1}W^T$ Schur Complement

$Sd_C = -(u - WV^{-1}v)$ Compute cameras first (# cameras << # points)

$Vd_X = -v - W^T d_C$ back substitution for points

Schur Complement for Cameras

$$(U - \boxed{WV^{-1}W^T})d_C = -(u - WV^{-1}v)$$

$$WV^{-1}W^T = \begin{pmatrix} S_{11} & S_{12} & S_{13} \\ S_{12}^T & S_{22} & S_{23} \\ S_{13}^T & S_{23}^T & S_{33} \end{pmatrix}$$

$$S_{i_1 i_2} = \sum_{j=1}^{4} W_{i_1 j} V_j^{-1} W_{i_2 j}^T$$

Schur Complement for Cameras

$$(U - WV^{-1}W^T)d_C = -(u - \boxed{WV^{-1}v})$$

$$WV^{-1}e_X = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix}$$

$$g_i = \sum_{j=1}^{4} W_{ij} V_j^{-1} v_j$$

Schur Complement for Cameras

- Again, in general NOT all points are visible in all cameras

$$S_{i_1 i_2} = \sum_{j=1}^{4} W_{i_1 j} V_j^{-1} W_{i_2 j}^T$$

  - $S_{i_1 i_2} = 0$ if i_1-th camera has no common points with i_2-th camera
  - More sparse structure, more speed-up

Back Substitution for Points

$$V \boxed{d_X} = -v - W^T d_C$$

$$d_{X_j} = -v_j - \sum_{i=1}^{3} W_{ij}^T d_{C_i}$$

- Each point can be solved independently
- Again, $W_{ij} = 0$ if i-th points is invisible in j-th camera

Probability Interpretation

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

joint density $P(d_C, d_X) = P(d_C) P(d_X \mid d_C)$

$$(U - WV^{-1}W^T) d_C = -(u - WV^{-1}v)$$

marginalize out d_X to get $P(d_C)$

$$W^T d_C + V d_X = -v$$

conditional $P(d_X \mid d_C)$

Sparse Bundle Adjustment

- 1. Construct normal equation
  - Compute and store the small non-zero block matrices $\mathbf{U}_i$, $\mathbf{V}_j$, $\mathbf{W}_{ij}$

$$\mathbf{J}^\top \mathbf{J}\delta = \mathbf{J}^\top \mathbf{e}$$

$$\Downarrow$$

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta_\mathbf{C} \\ \delta_\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}$$

$\mathbf{U} = \mathbf{0}$; $\mathbf{V} = \mathbf{0}$; $\mathbf{W} = \mathbf{0}$; $\mathbf{u} = \mathbf{0}$; $\mathbf{v} = \mathbf{0}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
    Construct linearized equation (11)
    $\mathbf{U}_{ii} += \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{C}_{ij}}$
    $\mathbf{V}_{jj} += \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
    $\mathbf{u}_i += \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{e}_{ij}$
    $\mathbf{v}_j += \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{e}_{ij}$
    $\mathbf{W}_{ij} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
end for

Sparse Bundle Adjustment

- 2. Marginalize out points to construct Schur complement
  - $\boldsymbol{S}$ is also sparse, with non-zero block matrix $\mathbf{S}_{i_1 i_2}$ if and only if camera i_1 and i_2 share common points.

$$\mathbf{S}\delta_{\mathbf{C}} = \mathbf{g},$$
$$\mathbf{S} = (\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^\top),$$
$$\mathbf{g} = \mathbf{u} - \mathbf{W}\mathbf{V}^{-1}\mathbf{v}.$$

$\mathbf{S} = \mathbf{U}$
for each point j and each camera pair $(i_1, i_2) \in \mathcal{V}_j \times \mathcal{V}_j$ **do**
    $\mathbf{S}_{i_1 i_2} -= \mathbf{W}_{i_1 j}\mathbf{V}_{jj}^{-1}\mathbf{W}_{i_2 j}^\top$
end for
$\mathbf{g} = \mathbf{u}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
    $\mathbf{g}_i -= \mathbf{W}_{ij}\mathbf{V}_{jj}^{-1}\mathbf{v}_j$
end for

Sparse Bundle Adjustment

- 3. Solve cameras
  - Use sparse solver to solve $\delta_{\mathbf{C}}$
    - Sparse Cholesky factorization
    - Preconditioned Conjugate Gradient (PCG) that naturally leverages the sparseness of $\boldsymbol{S}$
- 4. Update points

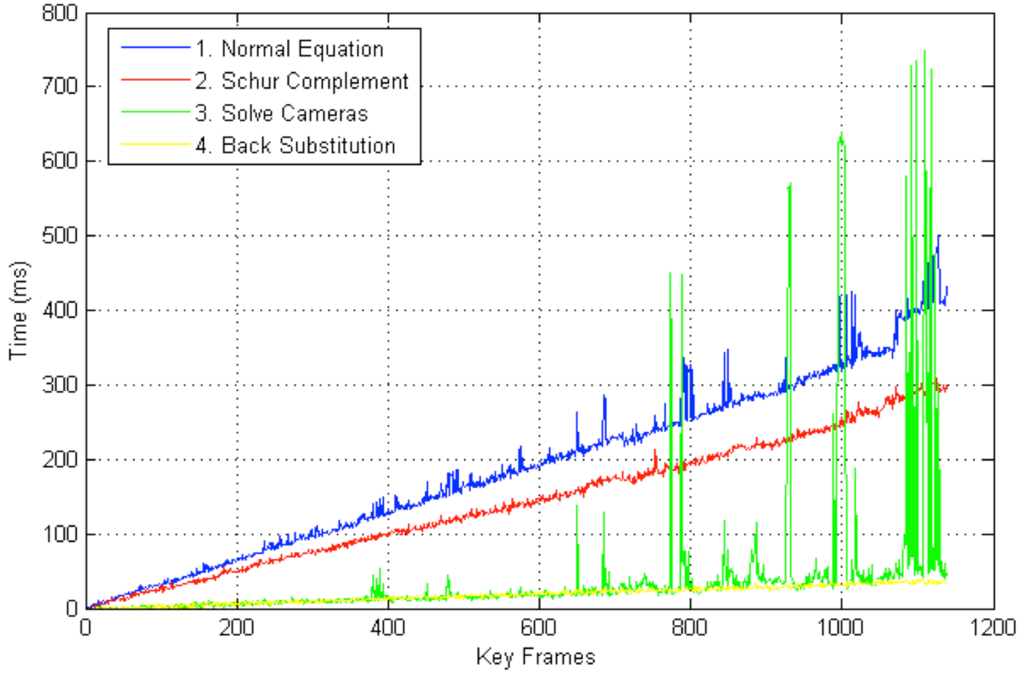
for each point j **do**

$$\delta_{\mathbf{X}_j} = \mathbf{V}_{jj}^{-1}\left(\mathbf{v}_j - \sum_{i \in \mathcal{V}_j} \mathbf{W}_{ij}^{\top}\delta_{\mathbf{C}_i}\right)$$

end for

Sparse Bundle Adjustment

- Runtime increases with the number of cameras



Related Works

- Parallel BA
  - Ni et al. 2007, Wu et al. 2011 (PBA)
- Hierarchical BA
  - Steedly et al. 2003, Snavely et al. 2008, Frahm et al. 2010
- Segment-based BA
  - Zhu et al. 2014, Zhang et al. 2016 (ENFT)
- Incremental BA
  - Kaess et al. 2008 (iSAM), Kaess et al. 2011 (iSAM2), Indelman et al. 2012 (iLBA), Ila et al. 2017 (SLAM++), Liu et al. 2017 (EIBA), Liu et al. 2018 (ICE-BA)

Segment-based Bundle Adjustment

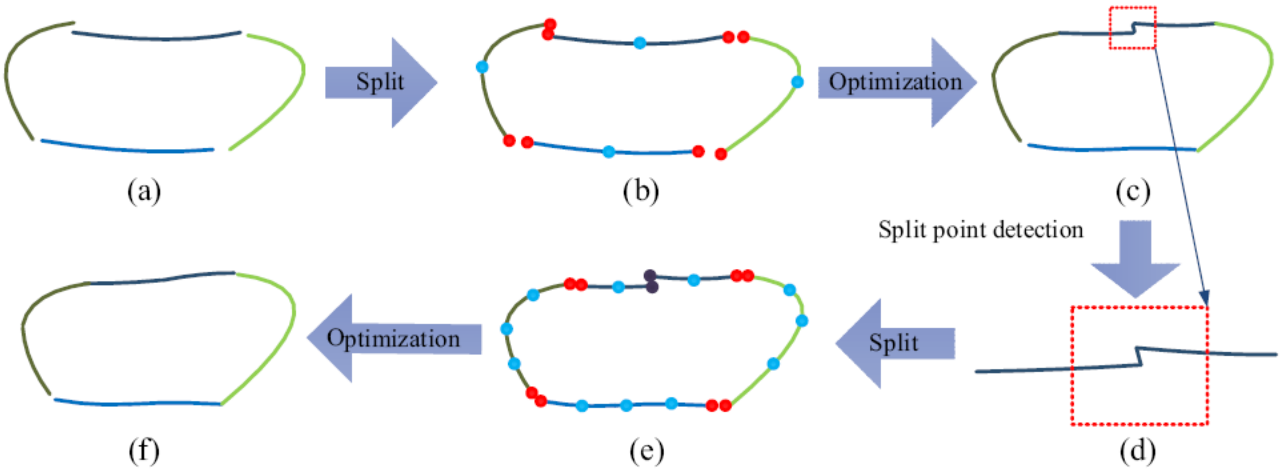
Zhang G, Liu H, Dong Z, et al. Efficient non-consecutive feature tracking for robust structure-from-motion[J]. IEEE Transactions on Image Processing, 2016, 25(12): 5957-5970.

The Difficulties for Large-Scale SfM

- Global Bundle Adjustment
  - Huge variables
  - Memory limit
  - Time-consuming
- Iterative Local Bundle Adjustment
  - Large error is difficult to be propagated to the whole sequence.
  - Easily stuck in a local optimum.
- Pose Graph Optimization
  - May not sufficiently minimize the error.

Segment-based Progressive SfM

- Split a long sequence to multiple short sequences.
- Perform SfM for each sequence and align them together.
- Detect the ``split point'' and further split the sequence if the reprojection error is large.
- The above procedure is repeated until the error is less than a threshold.



Segment-based Progressive SfM

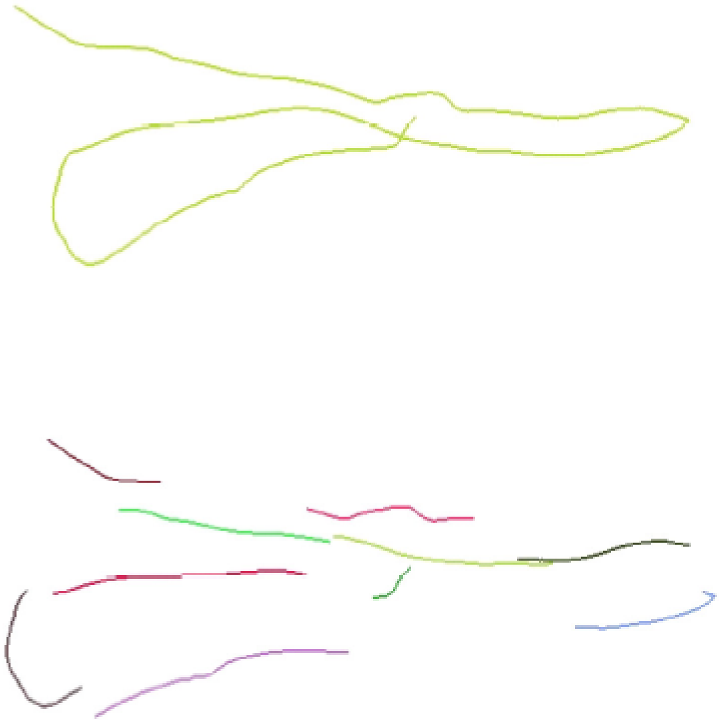
- Split Point Detection
  - Best minimize the reprojection error w.r.t. a, i.e. steepest descent direction

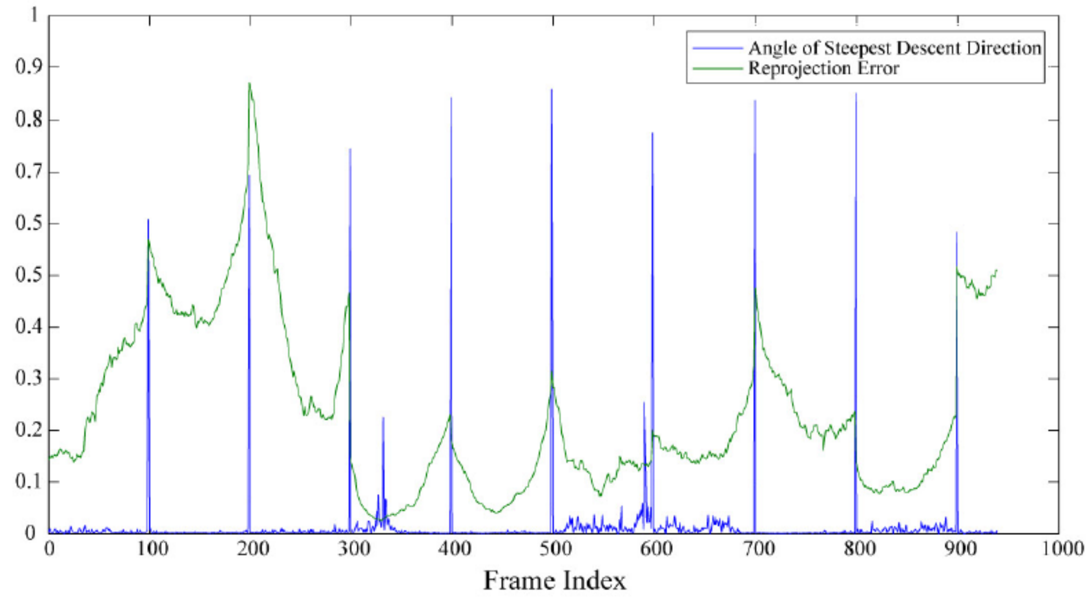
$$g_k = \sum_{i=1\cdots N_k} A_i^T e_i \qquad \begin{aligned} A_i &= \partial\pi(P_k X_i)/\partial a_k \\ e_i &= \mathbf{x}_i - \pi(P_k X_i) \end{aligned}$$

  - The inconsistency between two consecutive frames

$$C(k, k+1) = \arccos \frac{g_k^T \cdot g_{k+1}}{||g_k|| \cdot ||g_{k+1}||}.$$

Split Point Detection





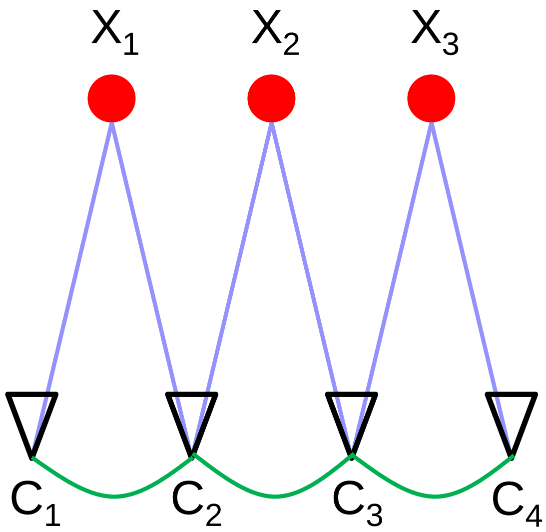
Incremental Bundle Adjustment

In order to benefit from increased accuracy offered by relinearization in batch optimization:

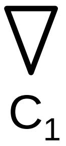
- Fixed-lag / Sliding-window Approaches
- Keyframe-based Approaches
- Incremental Approaches (iSAM & iSAM2, our EIBA & ICE-BA)

Batch VS Incremental BA

- Batch BA

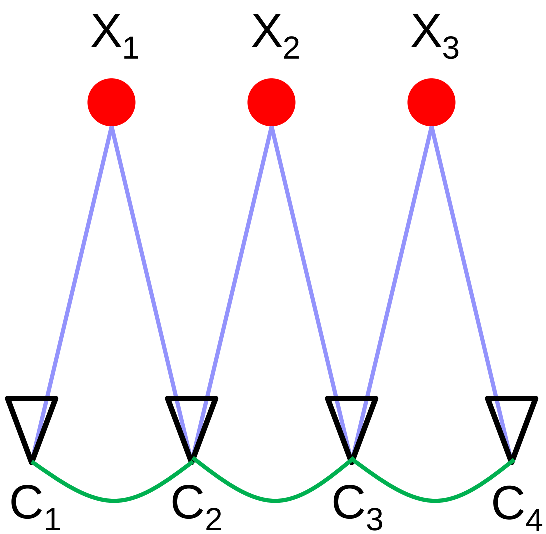


- Incremental BA

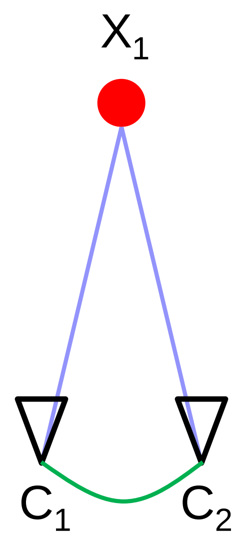


Batch VS Incremental BA

- Batch BA

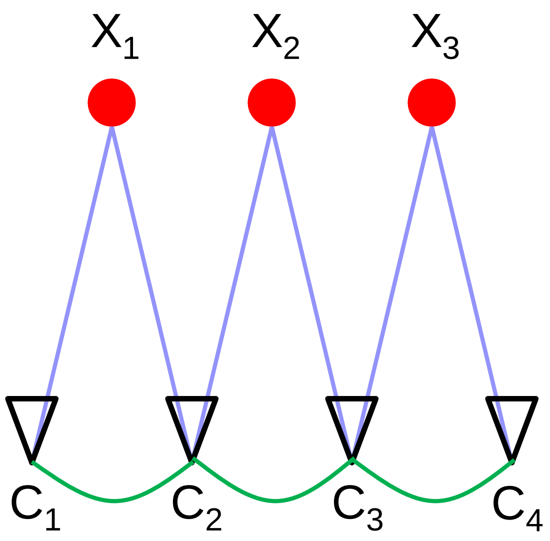


- Incremental BA

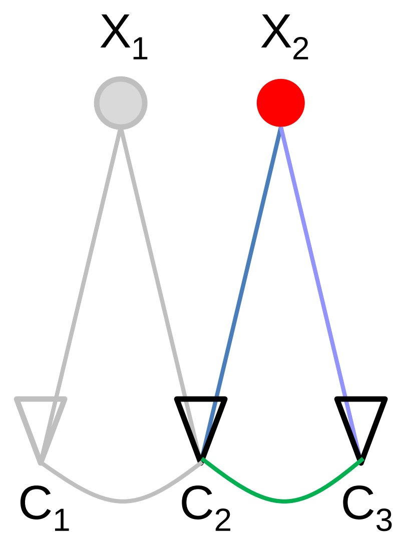


Batch VS Incremental BA

- Batch BA

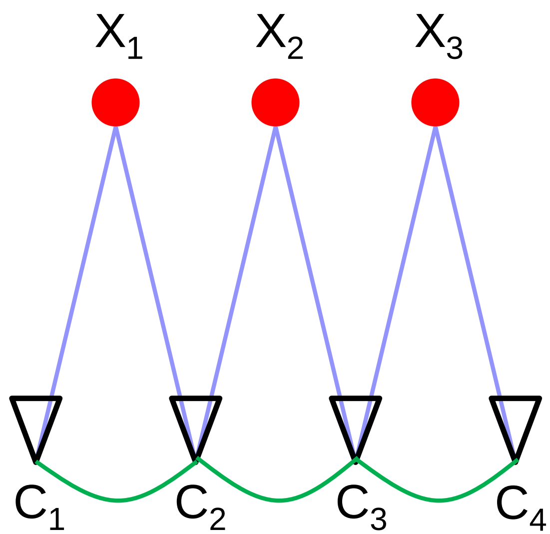


- Incremental BA

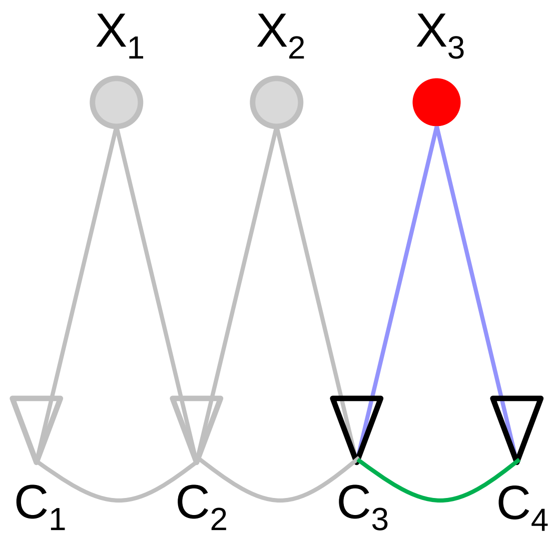


Batch VS Incremental BA

- Batch BA



- Incremental BA



Incremental BA in iSAM2 Based on Bayes Tree

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. The International Journal of Robotics Research, 31(2), 216-235.

Solving Least Square by QR Factorization

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \|A\boldsymbol{\theta} - \mathbf{b}\|^2$$

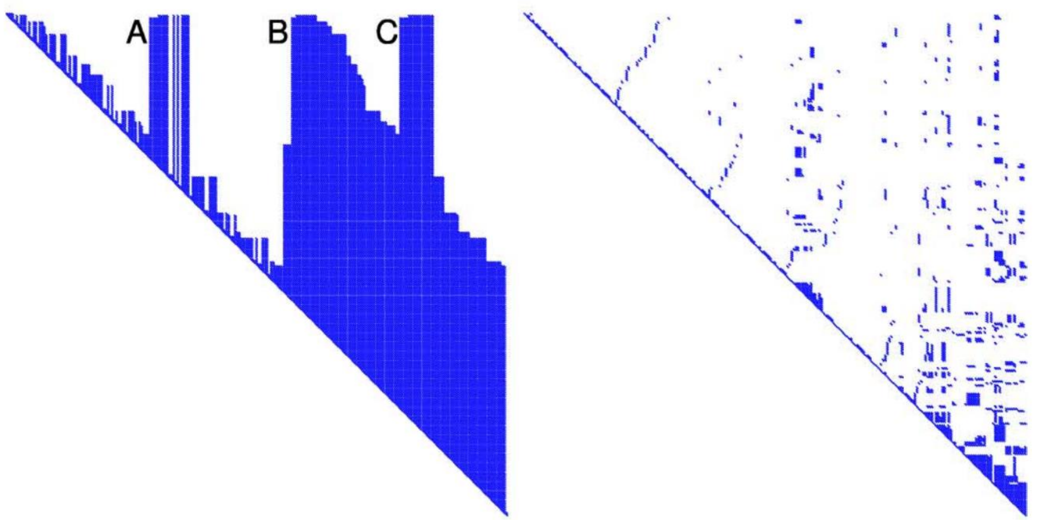
$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$$

$$\begin{aligned}
\|A\boldsymbol{\theta} - \mathbf{b}\|^2 &= \left\| Q \begin{bmatrix} R \\ 0 \end{bmatrix} \boldsymbol{\theta} - \mathbf{b} \right\|^2 \\
&= \left\| Q^T Q \begin{bmatrix} R \\ 0 \end{bmatrix} \boldsymbol{\theta} - Q^T \mathbf{b} \right\|^2 \\
&= \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} \boldsymbol{\theta} - \begin{bmatrix} \mathbf{d} \\ \mathbf{e} \end{bmatrix} \right\|^2 \\
&= \|R\boldsymbol{\theta} - \mathbf{d}\|^2 + \|\mathbf{e}\|^2
\end{aligned}$$

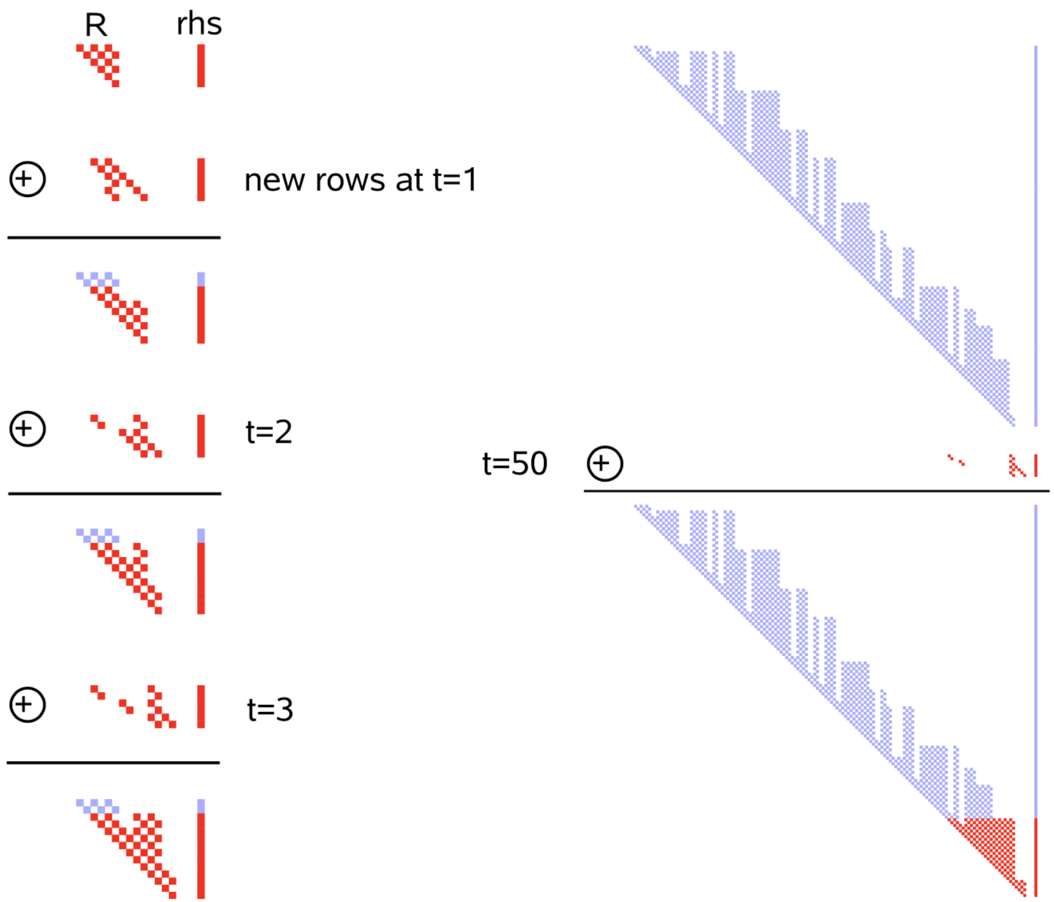
$R\boldsymbol{\theta}^* = \mathbf{d}$ R: upper triangular matrix

QR Factorization VS Schur Complement

- Directly work on Jacobian
  - R can be incrementally updated
  - Numerically more stable
    - $\mathrm{cond}(J) < \mathrm{cond}(J^T J)$
- Efficiency largely depends on variable ordering

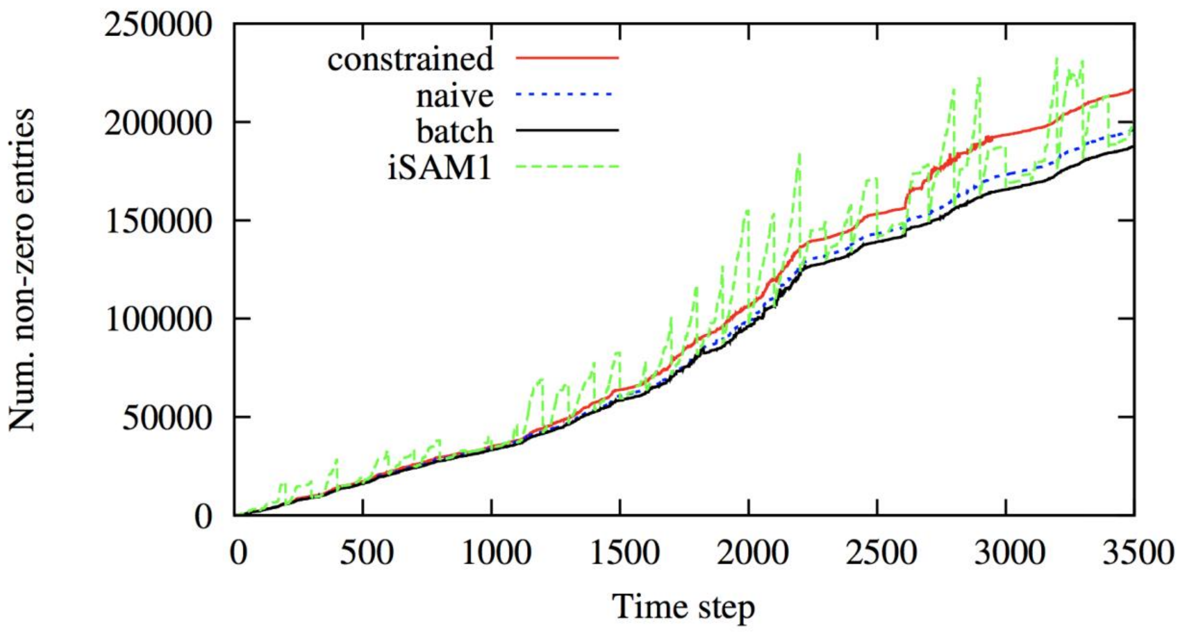


iSAM: Incremental Smoothing and Mapping



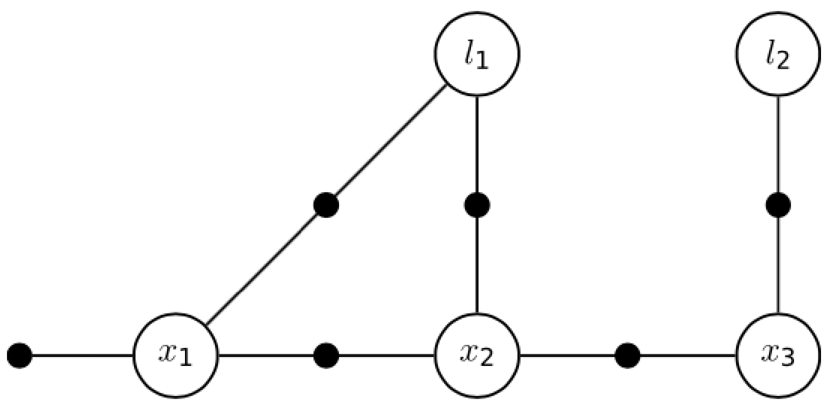
Limitations of iSAM

- Periodically reordering
  - It is difficult to analyze the dependency relationship among variables in a algebraic method



Factor Graph Representation

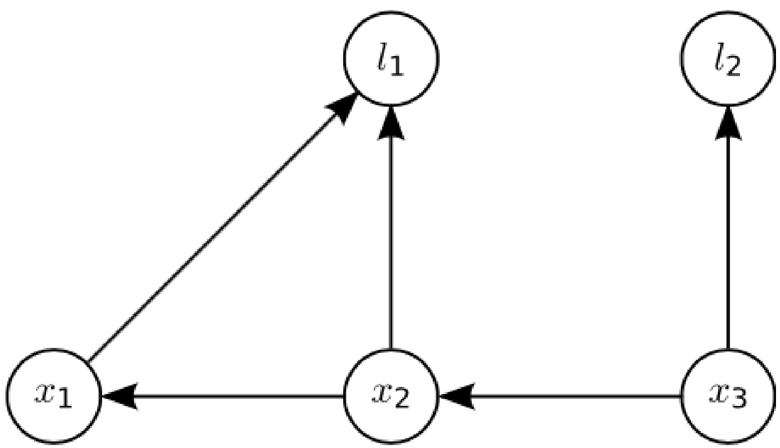
- Variable node (large circle)
- Measurement node (small dot)



$$
A = \begin{array}{c}
\begin{array}{ccccc} l_1 & l_2 & x_1 & x_2 & x_3 \end{array} \\
\begin{bmatrix}
X & & X & & \\
X & & & X & \\
 & X & & & X \\
 & & X & & \\
 & & X & X & \\
 & & & X & X
\end{bmatrix}
\end{array}
$$

Chordal Bayes Net

- Inference and elimination (marginalization) can be understood as converting the factor graph to chordal Bayes net



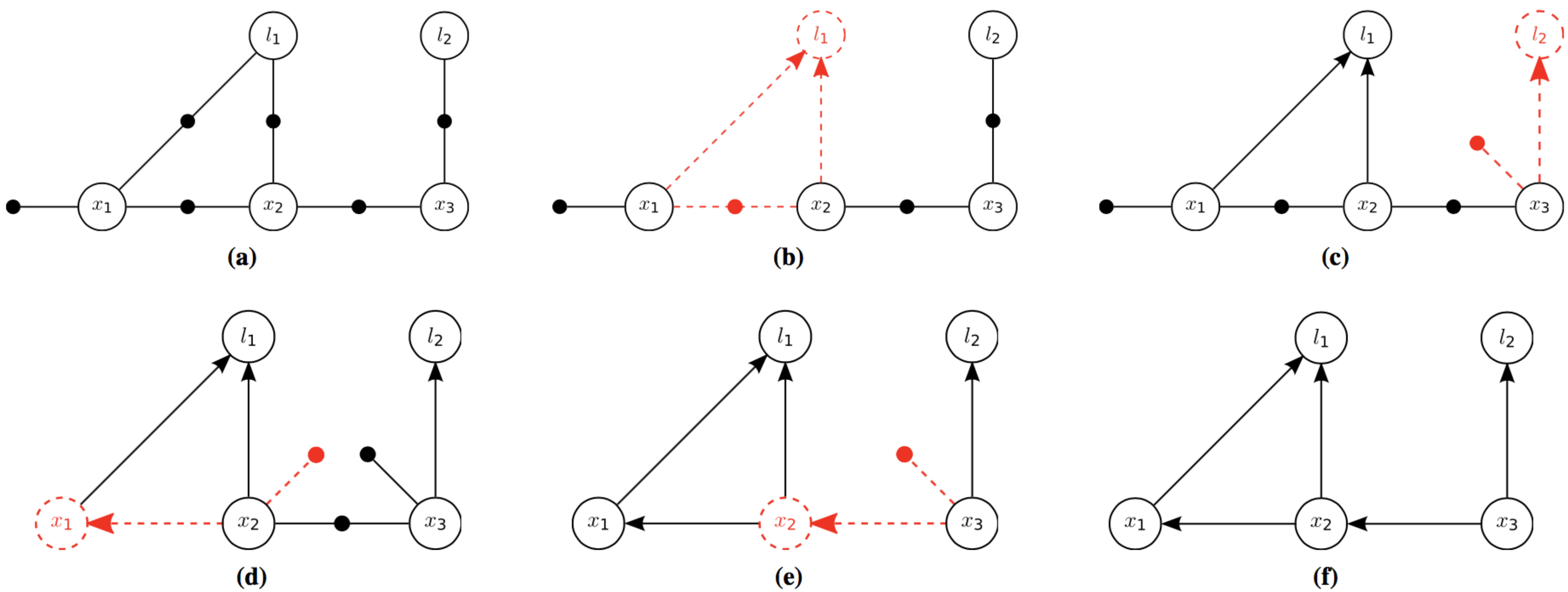
$$R = \begin{array}{c} \begin{matrix} l_1 & l_2 & x_1 & x_2 & x_3 \end{matrix} \\ \begin{bmatrix} X & & X & X & \\ & X & & & X \\ & & X & X & \\ & & & X & X \\ & & & & X \end{bmatrix} \end{array}$$

Chordal Bayes Net

- Inference and elimination (marginalization) can be understood as converting the factor graph to chordal Bayes net

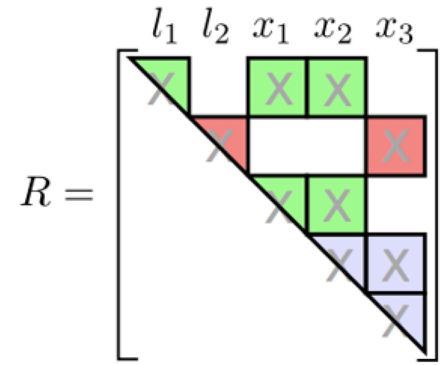
$$P(l_1, x_1, x_2) = P(l_1 \mid x_1, x_2) P(x_1, x_2)$$

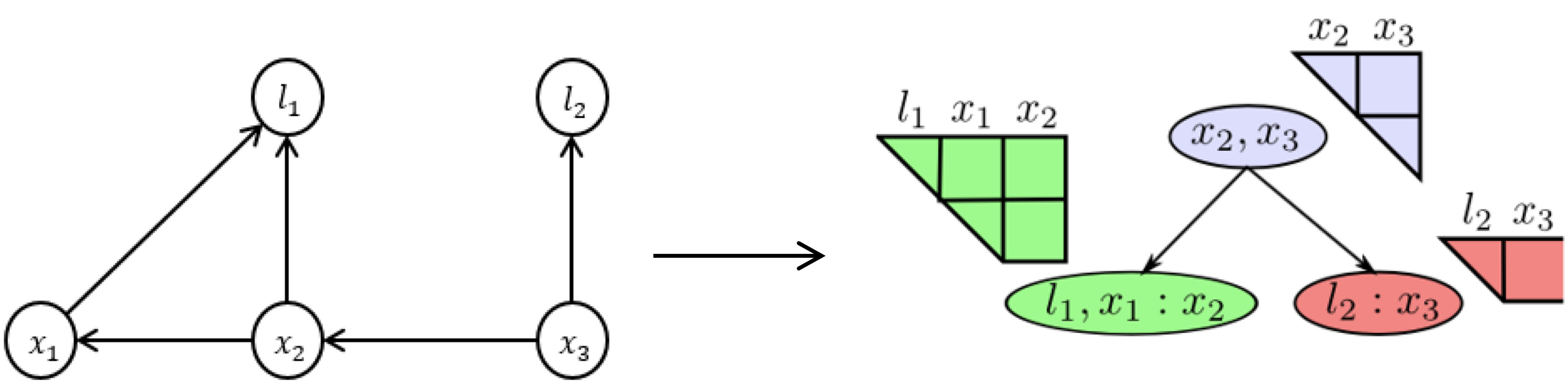
$$P(l_2, x_3) = P(l_2 \mid x_3) P(x_3)$$

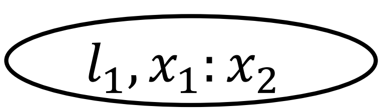


Bayes Tree

- To better reveal the dependence relationship, the chordal Bayes net is converted to a Bayes tree



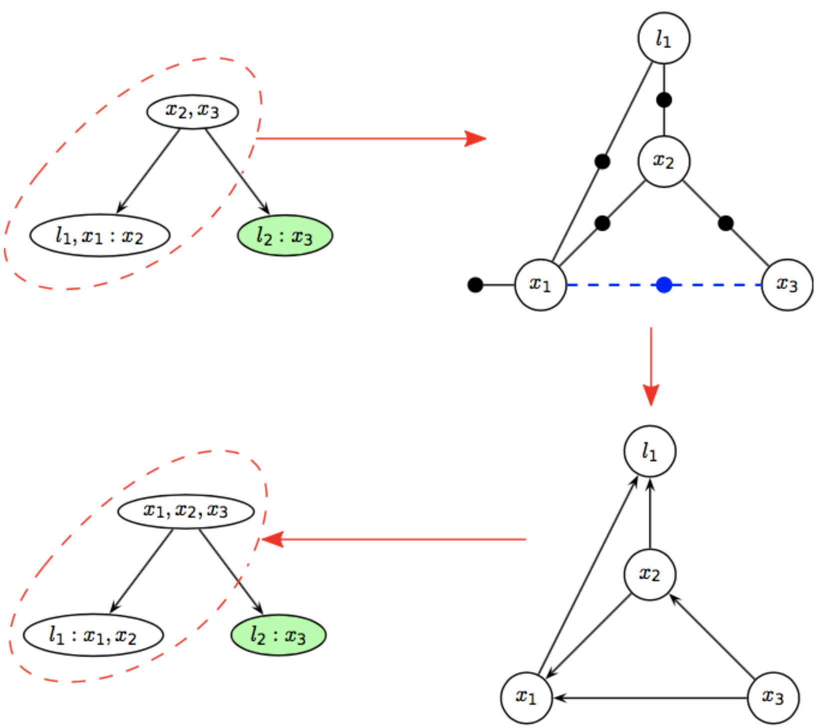




A **clique** encodes the conditional density $P(l_1, x_1 : x_2)$
l_1, x_1 are called the frontal variables
x_2 is called the separator

Incremental Update Bayes Tree

- Update the Bayes tree with a new factor $f(x_1,x_3)$



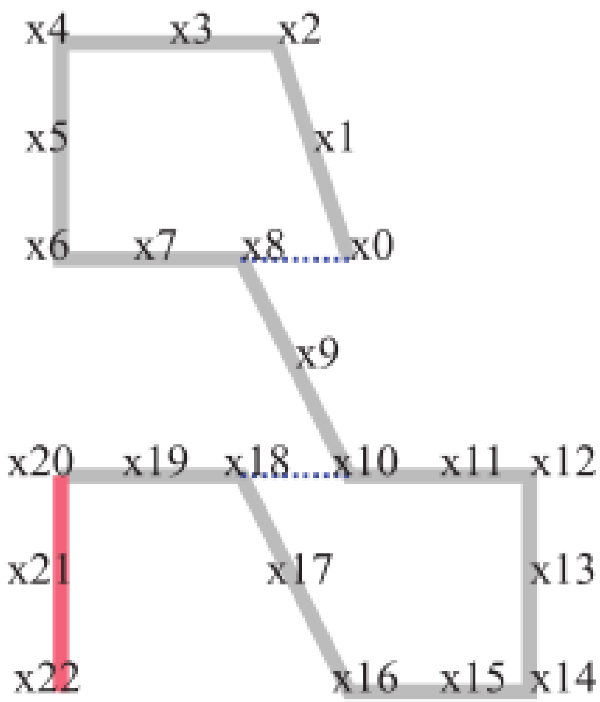
Alg. 4 Updating the Bayes tree with new factors $\mathscr{F}'$.

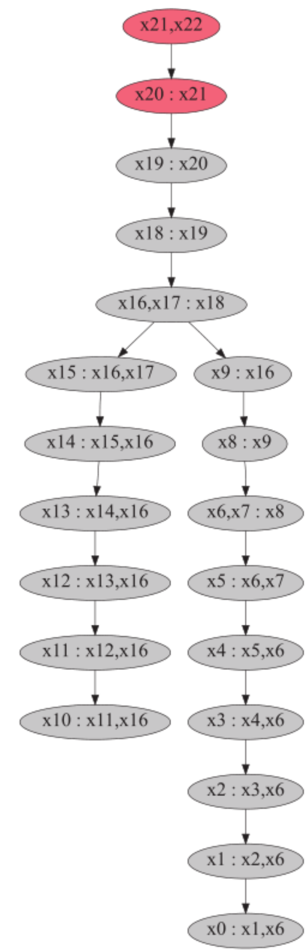
In: Bayes tree $\mathscr{T}$, new linear factors $\mathscr{F}'$
Out: modified Bayes tree $\mathscr{T}$'

1. Remove top of Bayes tree and re-interpret it as a factor graph:
   (a) For each variable affected by new factors, remove the corresponding clique and all parents up to the root.
   (b) Store orphaned sub-trees $\mathscr{T}_{orph}$ of removed cliques.
2. Add the new factors $\mathscr{F}'$ into the resulting factor graph.
3. Re-order variables of factor graph.
4. Eliminate the factor graph (Alg. 2) and create a new Bayes tree (Alg. 3).
5. Insert the orphans $\mathscr{T}_{orph}$ back into the new Bayes tree.

Example of adding new states and factors

Information only propagates upwards.

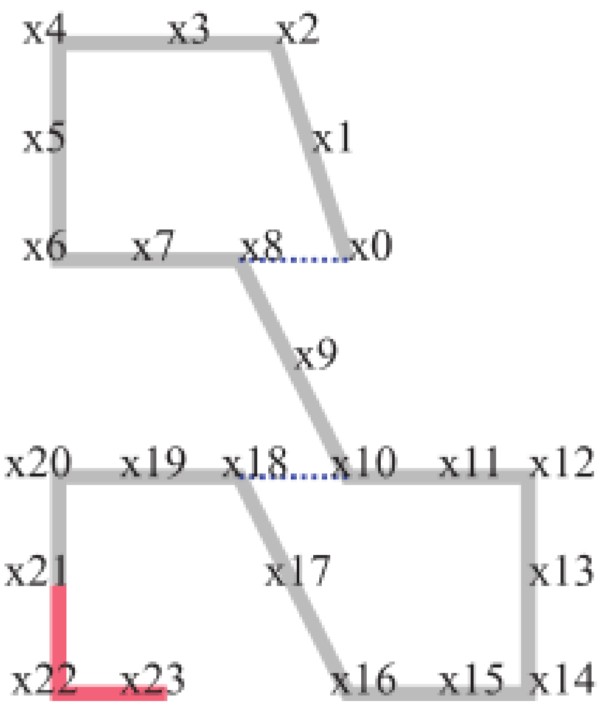


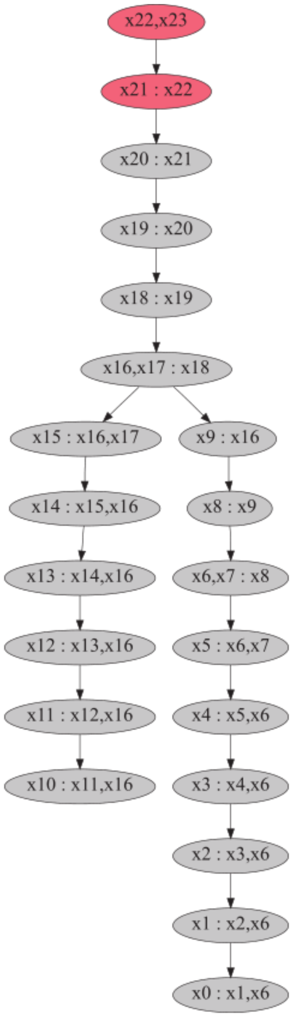


Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. The International Journal of Robotics Research, 31(2), 216-235.

Example of adding new states and factors

Information only propagates upwards.

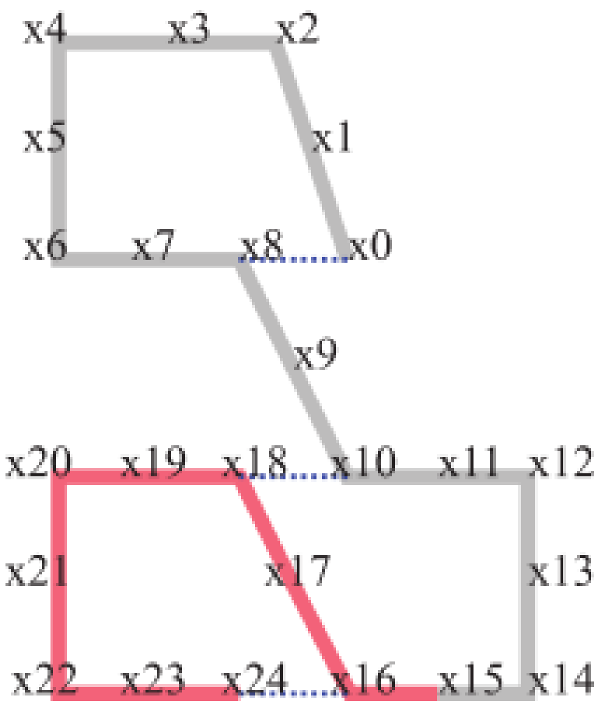


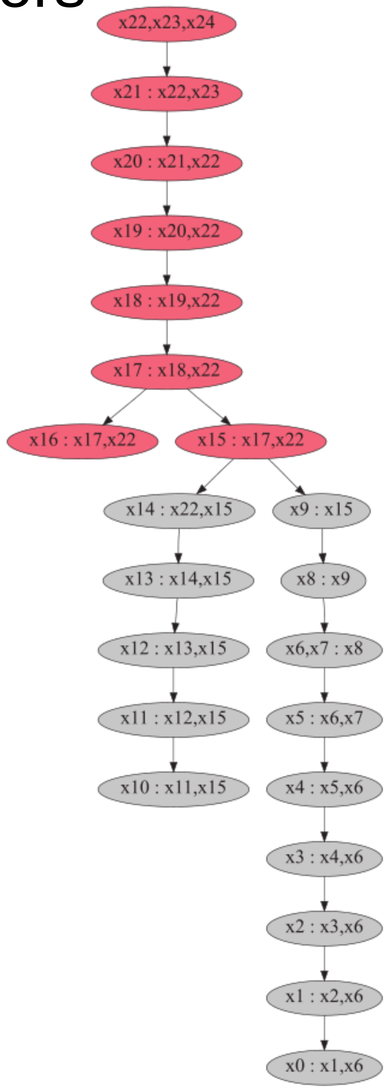


Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. The International Journal of Robotics Research, 31(2), 216-235.

Example of adding new states and factors

Loop detected.

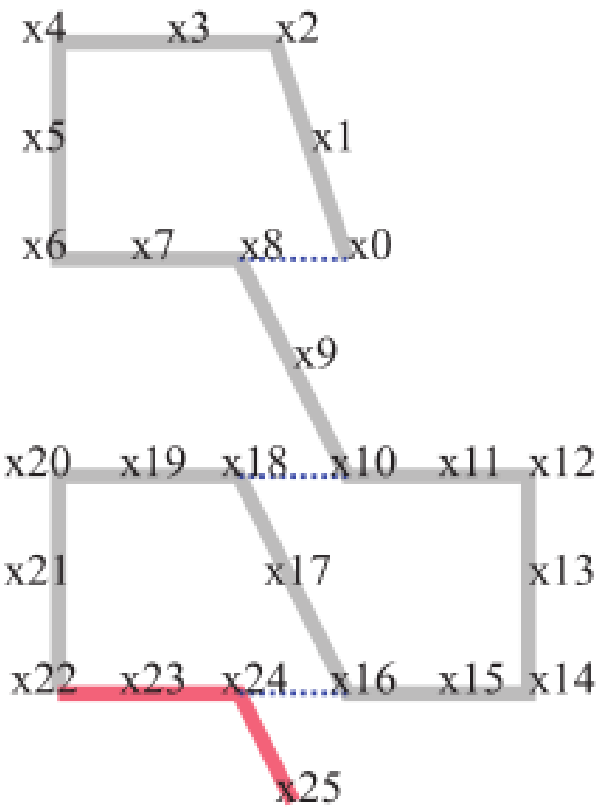


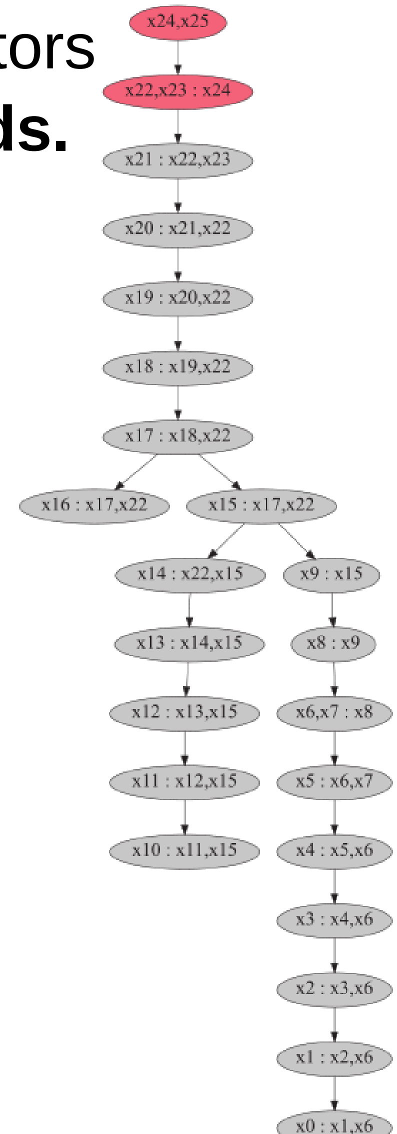


Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. The International Journal of Robotics Research, 31(2), 216-235.

Example of adding new states and factors

Information only propagates upwards.

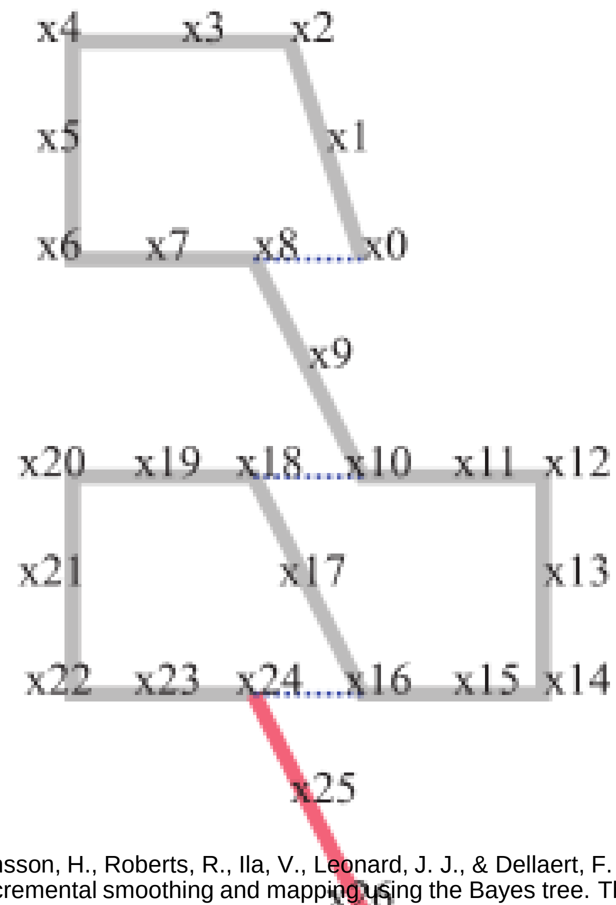


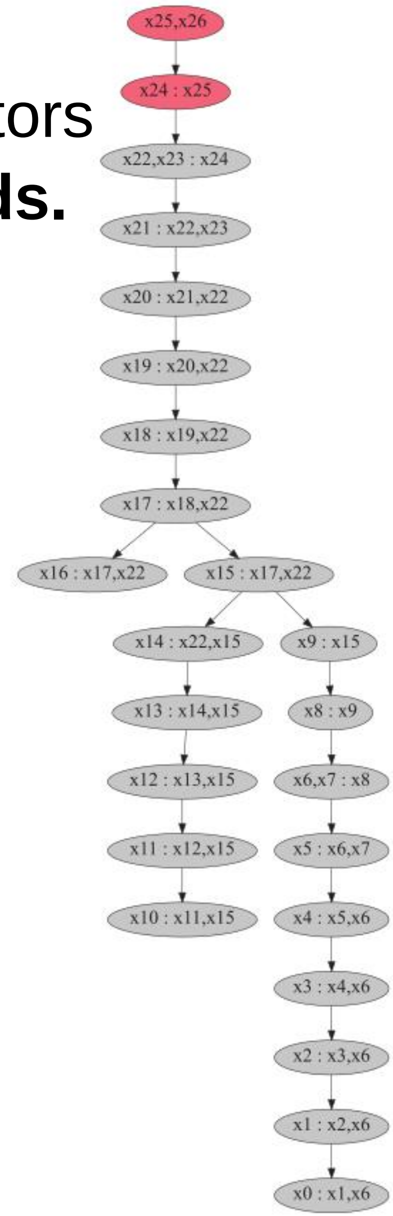


Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. The International Journal of Robotics Research, 31(2), 216-235.

Example of adding new states and factors

Information only propagates upwards.

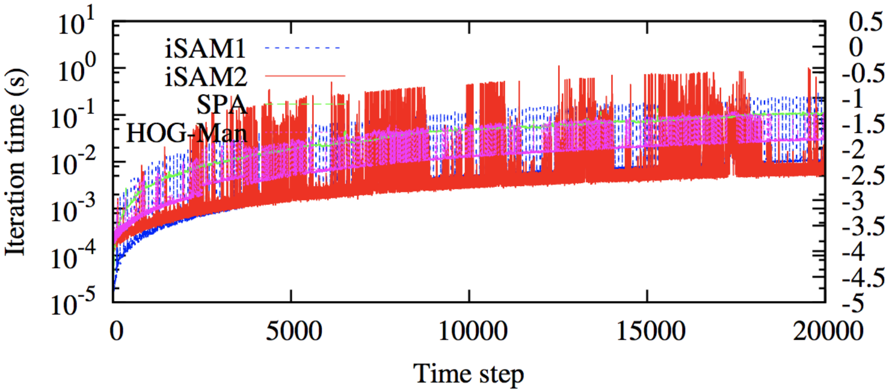


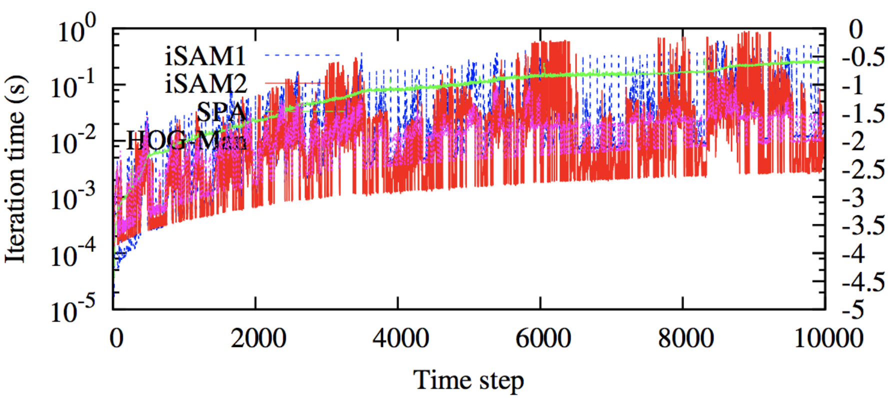


Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. The International Journal of Robotics Research, 31(2), 216-235.

Efficiency Comparison

- Many spikes
  - keep forward ✓
  - to and fro ✗



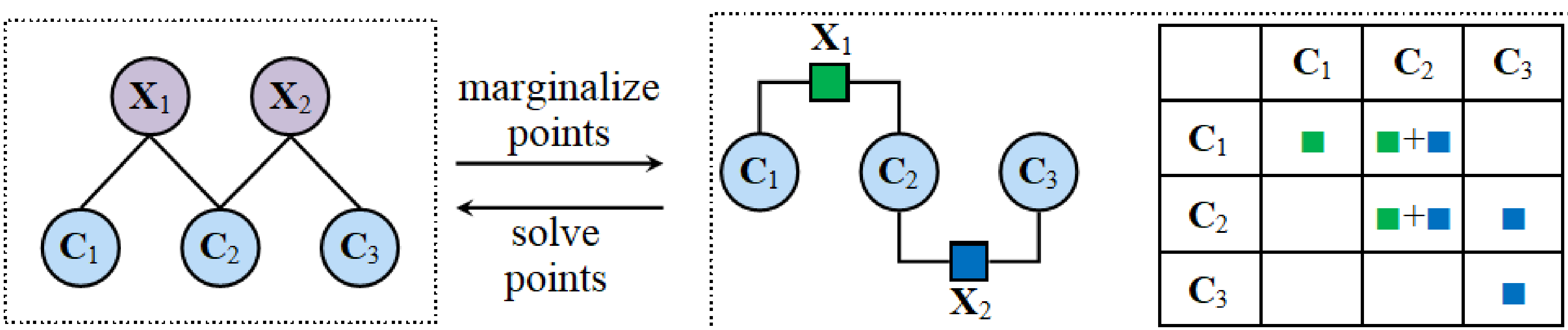


Efficient Incremental BA

Liu H, Li C, Zhang G, et al. Robust Keyframe-based Dense SLAM with an RGB-D Camera[J]. arXiv preprint arXiv:1711.05166, 2017.

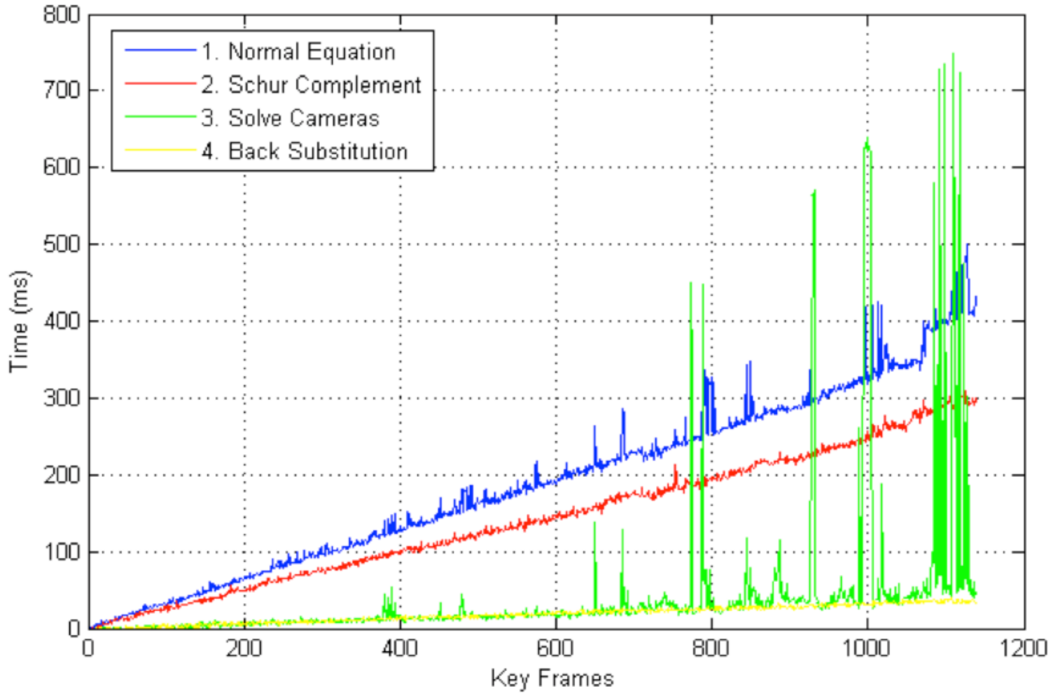
Revisit Standard BA

- Steps in one iteration
  - 1. Normal equation
  - 2. Schur compelment
  - 3. Solve cameras
  - 4. Solve points
- Factor graph representation



Observations in Standard BA

- Runtime for steps 1,2 >> 3,4
  - #projection functions >> #cameras

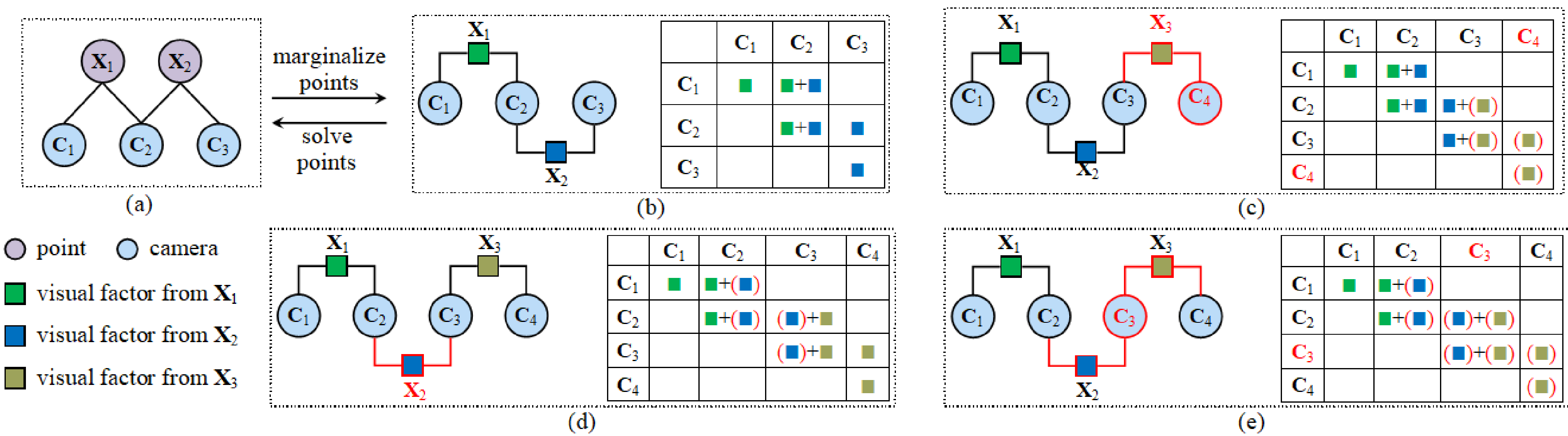


Observations in Standard BA

- Most cameras and points are nearly unchanged
  - Contribution of most projection functions nearly remains the same
  - No need to re-compute at each iteration

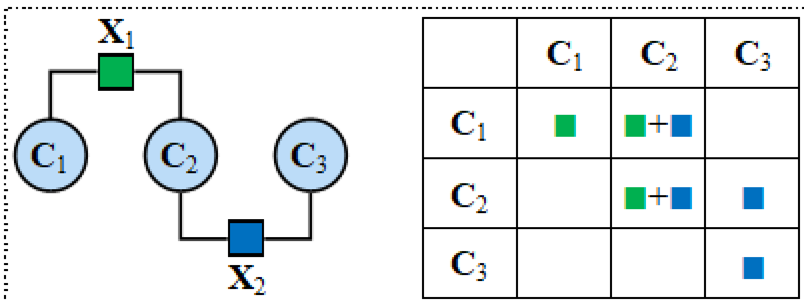
Efficient Incremental BA (EIBA)

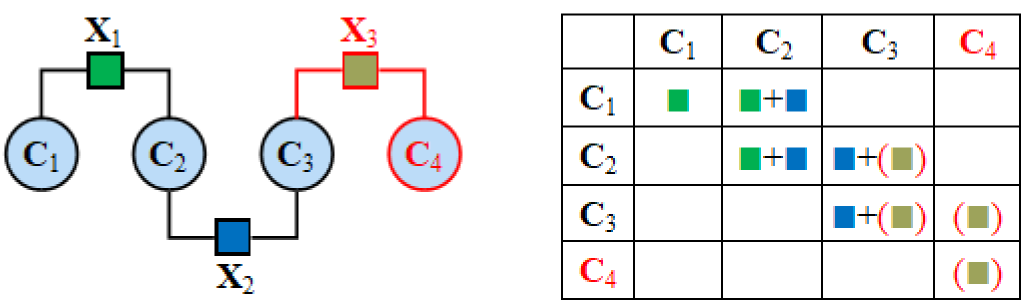
- Factor graph representation



Efficient Incremental BA (EIBA)

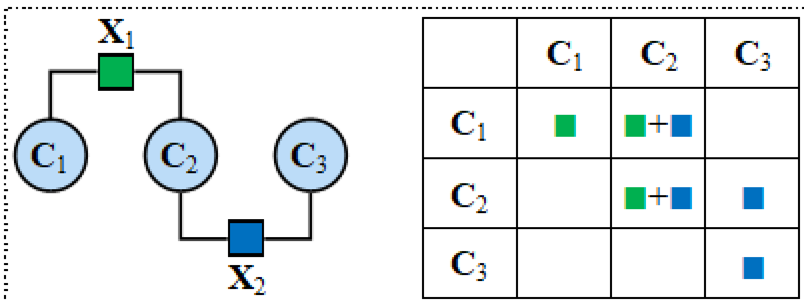
- New cameras or points come

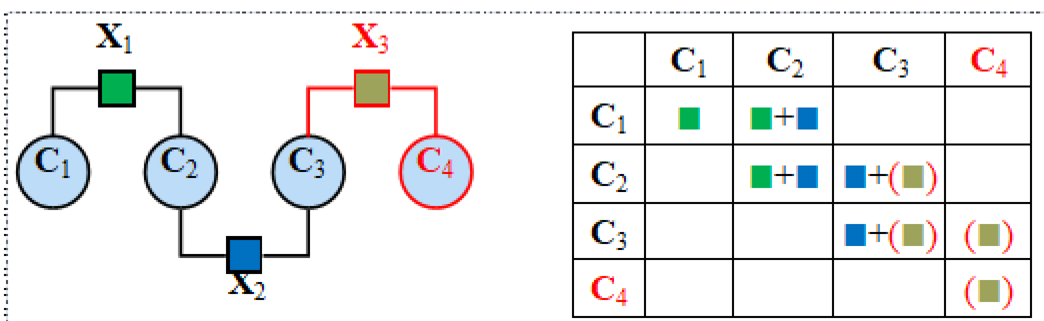




Efficient Incremental BA (EIBA)

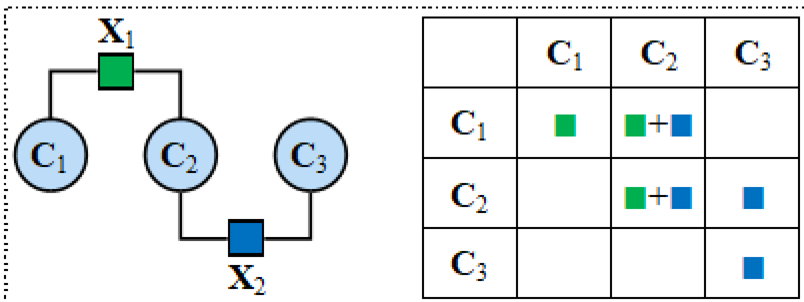
- New cameras or points come

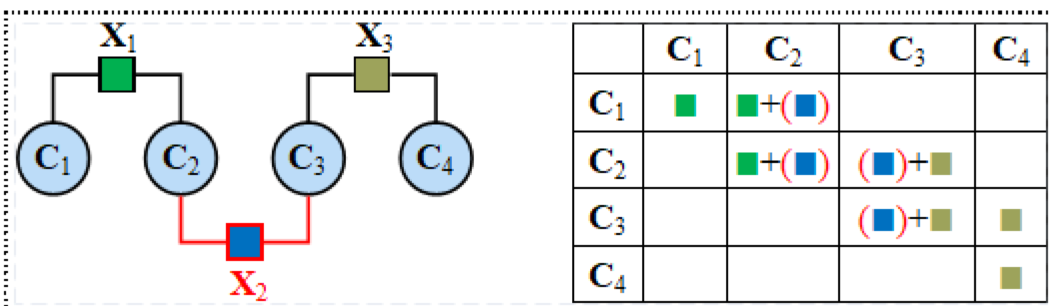




Efficient Incremental BA (EIBA)

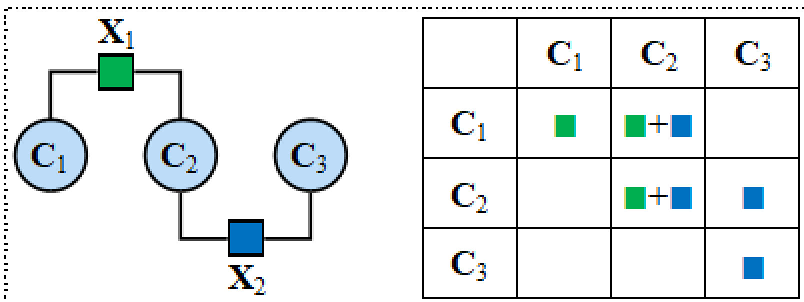
- Points change after iteration

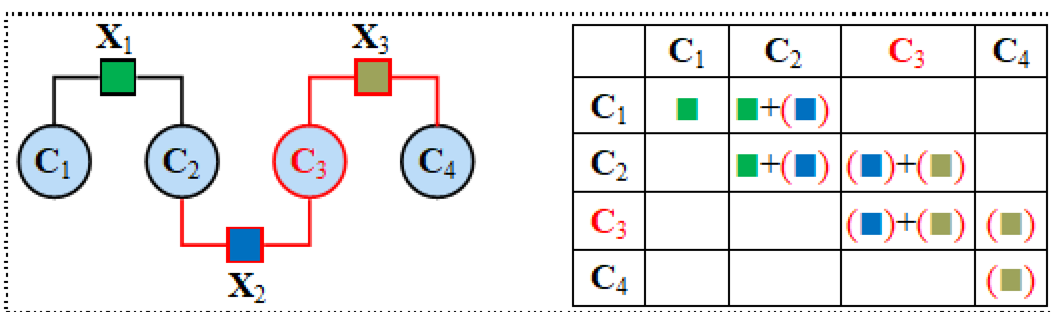




Efficient Incremental BA (EIBA)

- Cameras change after iteration





Step 1: Normal Equation

- Batch BA

$\mathbf{U} = \mathbf{0};\ \mathbf{V} = \mathbf{0};\ \mathbf{W} = \mathbf{0};\ \mathbf{u} = \mathbf{0};\ \mathbf{v} = \mathbf{0}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
  Construct linearized equation (11)
  $\mathbf{U}_{ii} += \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{C}_{ij}}$
  $\mathbf{V}_{jj} += \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
  $\mathbf{u}_i += \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{e}_{ij}$
  $\mathbf{v}_j += \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{e}_{ij}$
  $\mathbf{W}_{ij} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
end for

- Incremental BA

for each point j and each camera $i \in \mathcal{V}_j$ that $\mathbf{C}_i$ or $\mathbf{X}_j$ is changed **do**
  Construct linearized equation (11)
  $\mathbf{S}_{ii} -= \mathbf{A}_{ij}^{\mathbf{U}};\ \mathbf{A}_{ij}^{\mathbf{U}} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{C}_{ij}};\ \mathbf{S}_{ii} += \mathbf{A}_{ij}^{\mathbf{U}}$
  $\mathbf{V}_{jj} -= \mathbf{A}_{ij}^{\mathbf{V}};\ \mathbf{A}_{ij}^{\mathbf{V}} = \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}};\ \mathbf{V}_{jj} += \mathbf{A}_{ij}^{\mathbf{V}}$
  $\mathbf{g}_i -= \mathbf{b}_{ij}^{\mathbf{u}};\ \mathbf{b}_{ij}^{\mathbf{u}} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{e}_{ij};\ \mathbf{g}_i += \mathbf{b}_{ij}^{\mathbf{u}}$
  $\mathbf{v}_j -= \mathbf{b}_{ij}^{\mathbf{v}};\ \mathbf{b}_{ij}^{\mathbf{v}} = \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{e}_{ij};\ \mathbf{v}_j += \mathbf{b}_{ij}^{\mathbf{v}}$
  $\mathbf{W}_{ij} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
  Mark $\mathbf{V}_{jj}$ updated
end for

Step 2: Schur Complement

- Batch BA

$\mathbf{S} = \mathbf{U}$
for each point j and each camera pair $(i_1, i_2) \in \mathcal{V}_j \times \mathcal{V}_j$ **do**
$\quad \mathbf{S}_{i_1 i_2} -= \mathbf{W}_{i_1 j} \mathbf{V}_{jj}^{-1} \mathbf{W}_{i_2 j}^{\top}$
end for
$\mathbf{g} = \mathbf{u}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
$\quad \mathbf{g}_i -= \mathbf{W}_{ij} \mathbf{V}_{jj}^{-1} \mathbf{v}_j$
end for

- Incremental BA

for each point j that $\mathbf{V}_{jj}$ is updated and each camera pair $(i_1, i_2) \in \mathcal{V}_j \times \mathcal{V}_j$ **do**
$\quad \mathbf{S}_{i_1 i_2} += \mathbf{A}^{\mathbf{S}}_{i_1 i_2 j}$
$\quad \mathbf{A}^{\mathbf{S}}_{i_1 i_2 j} = \mathbf{W}_{i_1 j} \mathbf{V}_{jj}^{-1} \mathbf{W}_{i_2 j}^{\top}$
$\quad \mathbf{S}_{i_1 i_2} -= \mathbf{A}^{\mathbf{S}}_{i_1 i_2 j}$
end for
for each point j that $\mathbf{V}_{jj}$ is updated and each camera $i \in \mathcal{V}_j$ **do**
$\quad \mathbf{g}_i += \mathbf{b}^{\mathbf{g}}_{ij}$; $\mathbf{b}^{\mathbf{g}}_{ij} = \mathbf{W}_{ij} \mathbf{V}_{jj}^{-1} \mathbf{v}_j$; $\mathbf{g}_i -= \mathbf{b}^{\mathbf{g}}_{ij}$
end for

Performance of EIBA

- Computation time

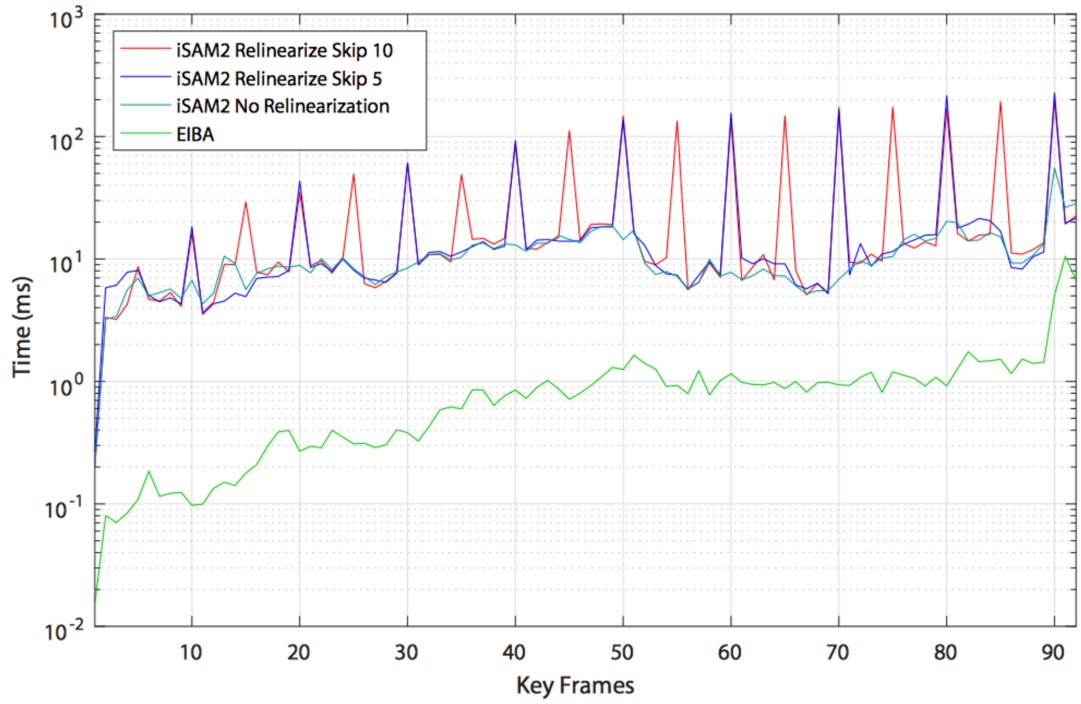


Fig. 4. The computation time of our EIBA and iSAM2 while incrementally adding each new keyframe on “fr3_long_office” sequence.

Performance of EIBA

- Computation time
  - Our EIBA is faster by an order of one magnitude than iSAM2.

Sequence	Num. of Camera / Points	Num. of Observations	EIBA	iSAM2		
				No relinearization	relinearizeSkip = 10	relinearizeSkip = 5
fr3_long_office	92 / 4322	12027	88.9ms	983.9ms	1968.2ms	2670.9ms
fr2_desk	63 / 2780	6897	34.8ms	507.8ms	850.4ms	1152.0ms

Performance of EIBA

- Optimized reprojection error

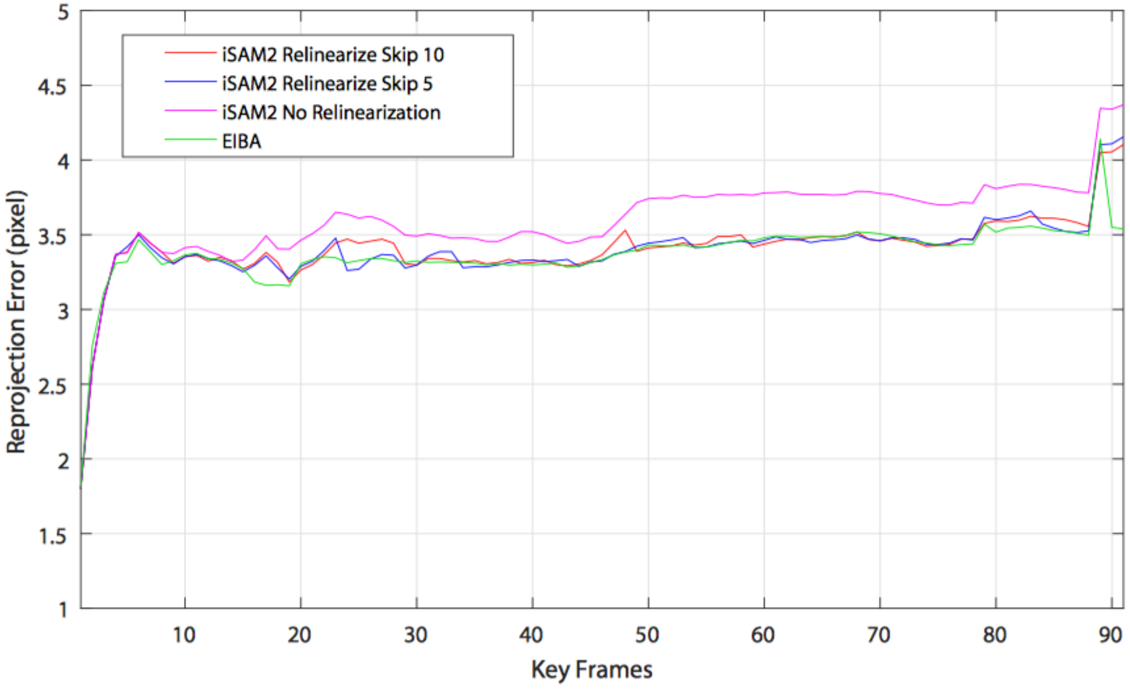


Fig. 5. The optimized reprojection error (RMSE) for our EIBA and iSAM2 while incrementally adding each new keyframe on “fr3_long_office” sequence.

ICE-BA for Visual-Inertial SLAM

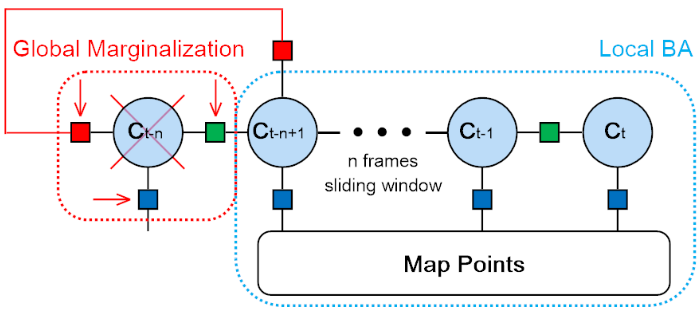
Liu H, Chen M, Zhang G, et al. ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM. CVPR 2018.

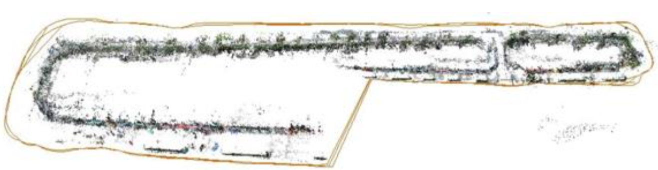
Requirements for Visual-Inertial SLAM

- Camera state includes velocity and bias
- Optimize each frame rather than just keyframes
- Information must be remained as much as possible

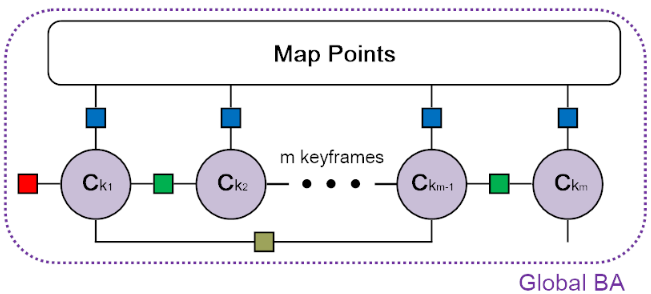
Existing Optimization Framework for Visual-Inertial SLAM

- Local BA (LBA)





- Global BA (GBA)



	Optimized frame	Non-optimized frame	Accuracy	Global Consistency	Latency	IMU Utilization
LBA	Sliding window	Marginalization	✗	✗	✓	✓
GBA	Keyframe	Discard	✓	✓	✗	✗

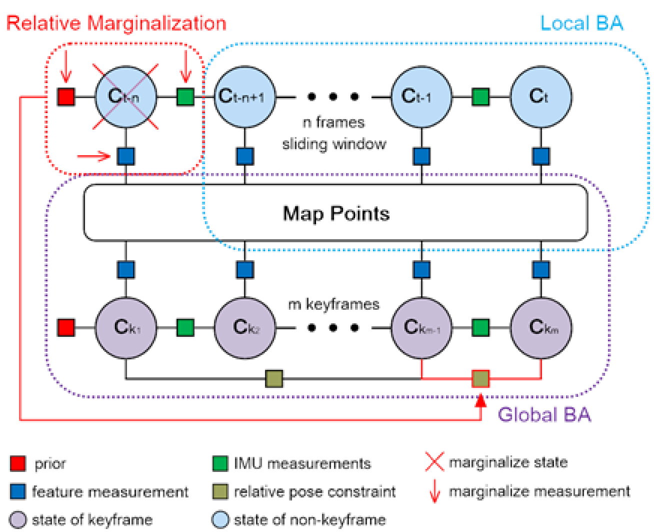
ICE-BA: Incremental, Consistent and Efficient BA

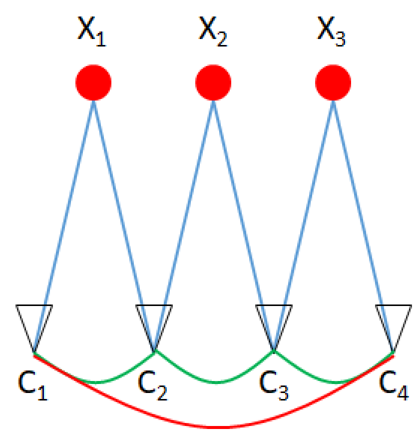
- Combine LBA and GBA
- Modified EIBA for efficient optimization
- Relative marginalization for global consistency

	Optimized frame	Non-optimized frame	Accuracy	Global Consistency	Latency	IMU Utilization
LBA	Sliding window	Marginalization	✗	✗	✓	✓
GBA	Keyframe	Discard	✓	✓	✗	✗
ICE-BA	Sliding window + keyframe	Relative marginalization	✓	✓	✓	✓

Framework

- Combine LBA and GBA

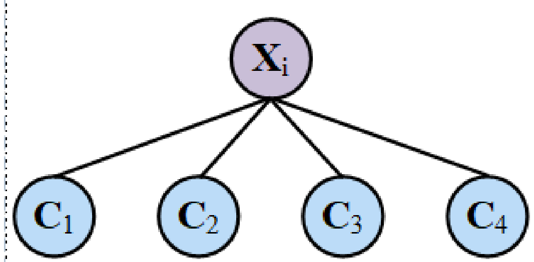




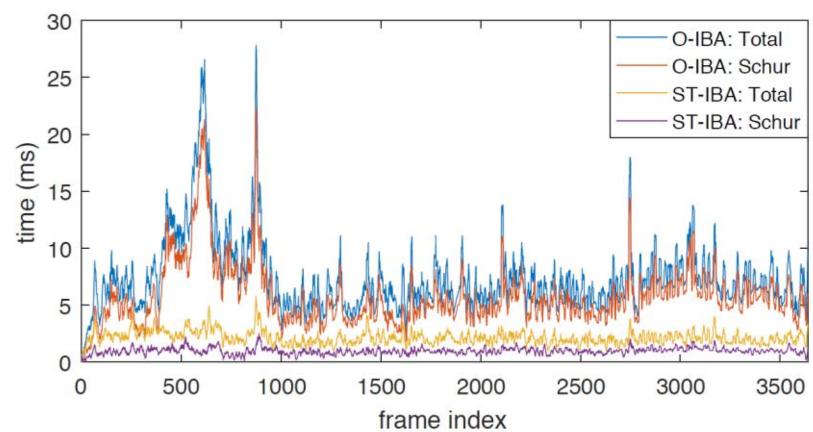
$$\underset{C_{1\cdots4}X_{1\cdots4}}{\operatorname{argmin}} \underbrace{\boxed{\sum_{ij}\left\|f(C_i,X_j)\right\|_2^2}}_{\text{visual term}} + \underbrace{\boxed{\sum_{i}\left\|h(C_{i-1},C_i)\right\|_2^2}}_{\text{inertial term}} + \underbrace{\boxed{\sum_{ij}\left\|g(C_i,C_j)\right\|_2^2}}_{\text{relative constraint}}$$

Limitation of EIBA for LBA

- In LBA, most points may be observed by most frames in the sliding window
  - Dense Schur complement
  - A large portion need to be re-evaluated

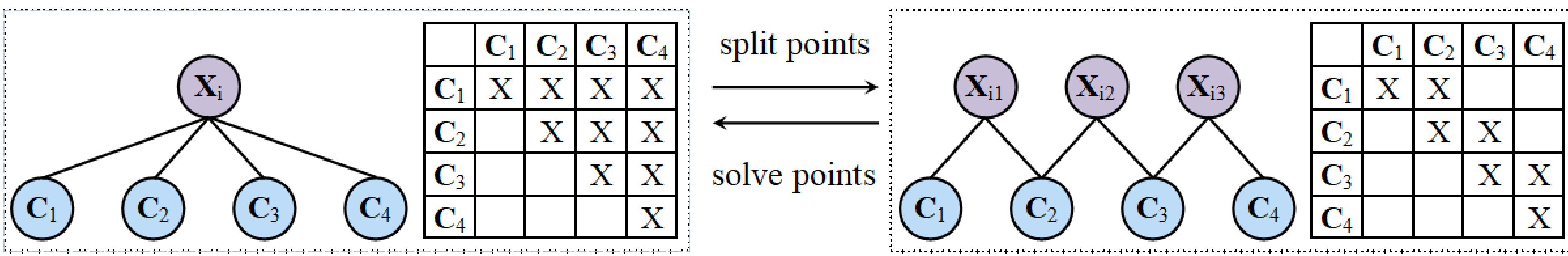


	C_1	C_2	C_3	C_4
C_1	X	X	X	X
C_2		X	X	X
C_3			X	X
C_4				X



Modified EIBA for LBA

- split the original long feature track $\mathbf{X}_i$ into several short overlapping sub-tracks $\mathbf{X}_{i1}$, $\mathbf{X}_{i2}$, …,



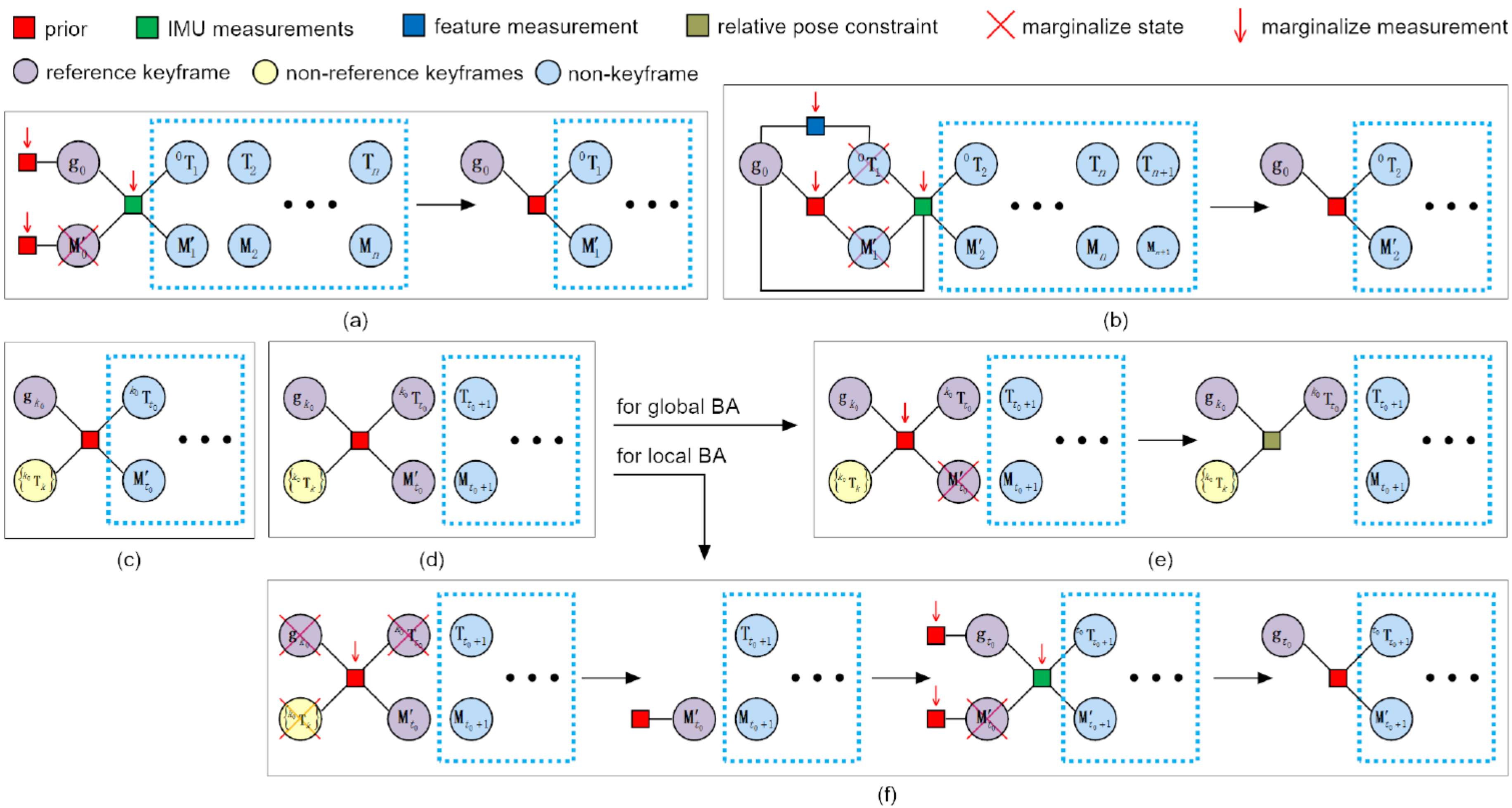
	C_1	C_2	C_3	C_4
C_1	X	X	X	X
C_2		X	X	X
C_3			X	X
C_4				X

	C_1	C_2	C_3	C_4
C_1	X	X		
C_2		X	X	
C_3			X	X
C_4				X

Relative Marginalization

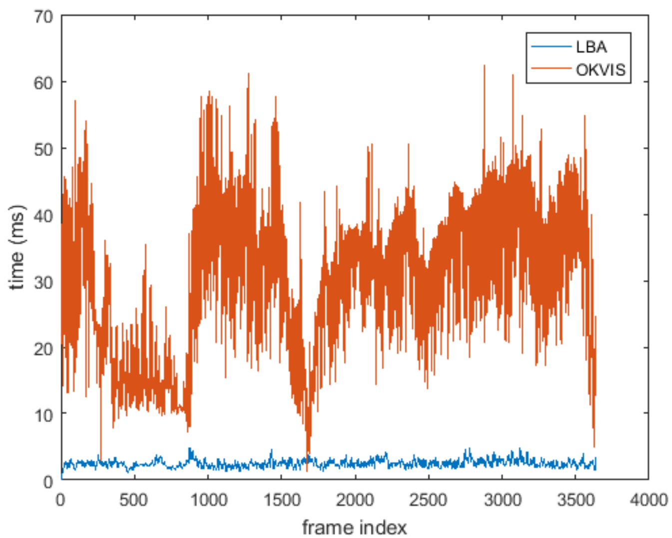
- Standard marginalization produces linear prior on camera pose $\mathbf{T}_t$ and IMU state $\mathbf{M}_t$ (velocity and bias) represented in global reference
- We present the prior camera pose ${}^{k0}\mathbf{T}_{t0}$ of the marginalized frame t_0 in the reference of its nearest keyframe k_0, and the prior IMU state $\mathbf{M}'_{t0}$ in its own reference

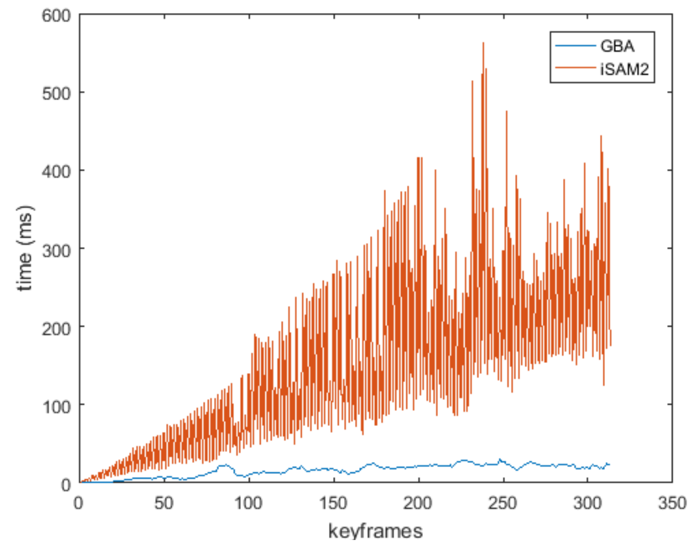
Relative Marginalization



Efficiency Comparison

- Local BA (LBA)
  - ICE-BA (50 frames)
  - OKVIS (8 frames)
  - 10x speedup
- Global BA (GBA)
  - ICE-BA: steady and smooth
  - iSAM: steep and peaks
  - 20x speedup



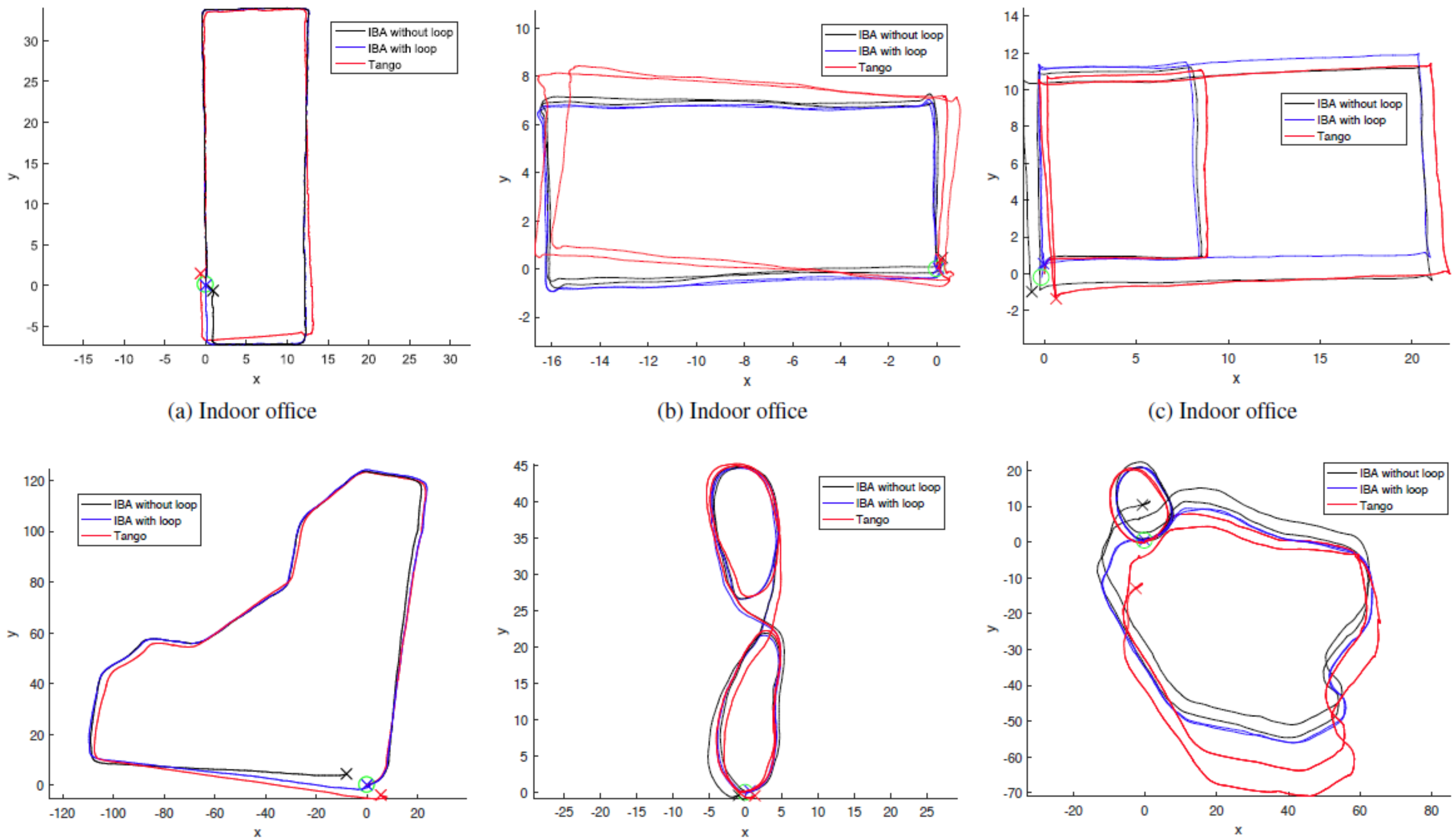


	Ours w/o loop	Ours w/ loop	OKVIS	iSAM2 (SVO)
local BA	2.45	2.45	26.83	-
global BA	12.90	24.67	-	225.87

Accuracy Comparison

Seq.	Ours w/ loop	Ours w/o loop	OKVIS	SVO	iSAM2
MH_01	0.11	0.09	0.22	**0.06**	0.07
MH_02	0.08	**0.07**	0.16	0.08	0.11
MH_03	**0.05**	0.11	0.12	0.16	0.12
MH_04	**0.13**	0.16	0.18	-	0.16
MH_05	**0.11**	0.27	0.29	0.63	0.25
V1_01	0.07	0.05	**0.03**	0.06	0.07
V1_02	0.08	**0.05**	0.06	0.12	0.08
V1_03	**0.06**	0.11	0.12	0.21	0.12
V2_01	0.06	0.12	**0.05**	0.22	0.10
V2_02	**0.04**	0.09	0.07	0.16	0.13
V2_03	**0.11**	0.17	0.14	-	0.20
Avg	**0.08**	0.12	0.14	0.20	0.13

Global Consistency Comparison with Google Tango



(a) Indoor office (b) Indoor office (c) Indoor office

(d) Outdoor road (e) Outdoor road (f) Outdoor road

Open-source Solver & BA

- Bundler: http://www.cs.cornell.edu/~snavely/bundler/
- g2o: https://github.com/RainerKuemmerle/g2o
- Ceres Solver: http://ceres-solver.org/
- PBA: https://grail.cs.washington.edu/projects/mcba/
- GTSAM& iSAM: https://bitbucket.org/gtborg/gtsam/
- EIBA: https://github.com/ZJUCVG/EIBA
- ICE-BA: https://github.com/baidu/ICE-BA