Bundle Adjustment

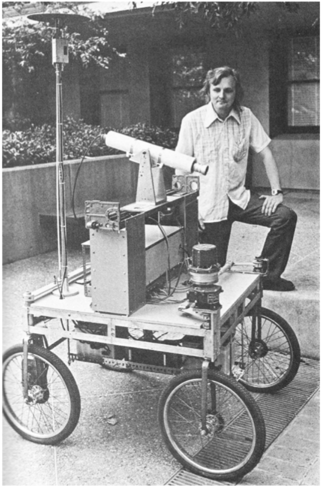
A computationally efficient solution for feature-based batch SLAM

Paul Furgale

University of Toronto Institute for Aerospace Studies

May 29, 2010

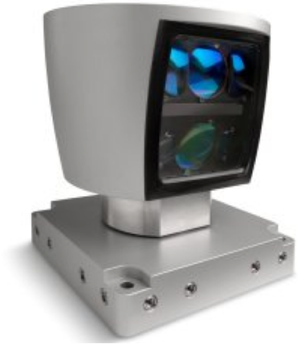
In the beginning...*



*H. Moravec. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. PhD thesis, Stanford University, September 1980. Available as Stanford AIM-340, CS-80-813 and republished as a Carnegie Mellon University Robotics Institue Technical Report to increase availability.

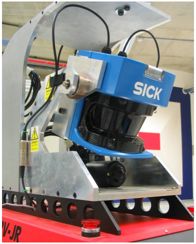
Why cameras?

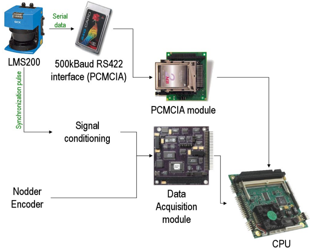
Scaling up to 3D: Expensive



http://www.youtube.com/watch?v=pZwu8_f1Lg0

Scaling up to 3D: Complicated*





http://www.robots.ox.ac.uk/~mobile/wikisite/pmwiki/pmwiki.php?n=Main.3DLaser

---

*A. Harrison and P. Newman. High quality 3d laser ranging under general vehicle motion. In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pages 7 Ð12, Pasadena, CA, May 2008.

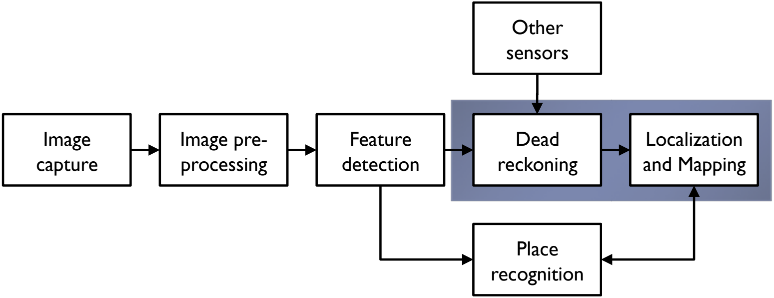
Scaling up to 3D: Cameras*

- Landmarks are easy to detect and track
- Calibrated stereo pair provides metric scale

http://www.youtube.com/watch?v=pZwu8_f1Lg0

*G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment. The International Journal of Robotics Research, 2010

Focus of this talk: Bundle Adjustment

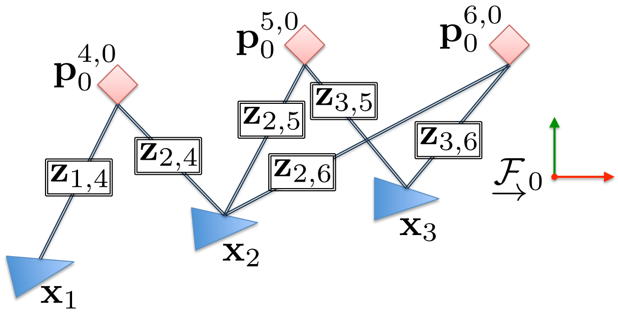


- It is at the core of most state-of-the-art visual slam systems
- The batch maximum likelihood method is the gold standard estimator
- Bundle adjustment seemingly provides the best of both worlds: efficient and accurate!

Bundle Adjustment

- Solution for structure and motion given sparse point feature measurements and data associations
- Resulting equations are highly nonlinear with many design variables
- Factorization of the problem possible to allow an efficient batch solution

Notation



$\underrightarrow{\mathcal{F}}_0$: The inertial frame.

$\underrightarrow{\mathcal{F}}_k$: A frame attached to the robot at time k.

$\mathbf{x}_k$: Parameters representing the pose of the robot at time k.

$\mathbf{p}_0^{j,0}$: The position of feature j in the inertial frame.

$\mathbf{z}_{k,j}$: Measurement of feature j at time k.

Observation Model

An observation of feature j at time k is modeled as

$$\mathbf{z}_{k,j} = \mathbf{h}\left(\mathbf{x}_k, \mathbf{p}_0^{j,0}\right) + \mathbf{n}_{k,j},$$

with

$$\mathbf{n}_{k,j} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{k,j}\right).$$

This observation model is specific to 3D point features, but not to the sensor used. It is possible to extend this framework to line, plane, cylinder, or other shape features.

$$J := \sum_{k=1}^{K} \sum_{j=1}^{M} \left(\mathbf{z}_{k,j} - \mathbf{h}\left(\mathbf{x}_k, \mathbf{p}_0^{j,0}\right)\right)^T \mathbf{R}_{k,j}^{-1} \left(\mathbf{z}_{k,j} - \mathbf{h}\left(\mathbf{x}_k, \mathbf{p}_0^{j,0}\right)\right)$$

K : The number of robot poses under consideration.

M : The number of features under consideration.

- This objective function is derived from the negative log likelihood of $p(\mathbf{z}|\mathbf{x}, \mathbf{p})$
- Minimizing this objective finds the camera poses and feature positions that maximize the likelihood of the measurements

The objective function J can be expressed as a big matrix equation. Define some intermediate values:

$$\mathbf{z} := \left(\mathbf{z}_{1,1}, \ldots, \mathbf{z}_{K,1}, \mathbf{z}_{1,2}, \ldots, \mathbf{z}_{K,M}\right)$$

$$\mathbf{x} := \left(\mathbf{x}_1, \ldots, \mathbf{x}_K\right)$$

$$\mathbf{p} := \left(\mathbf{p}_0^{1,0}, \ldots, \mathbf{p}_0^{M,0}\right)$$

$$\mathbf{h}\left(\mathbf{x}, \mathbf{p}\right) := \left(\mathbf{h}\left(\mathbf{x}_1, \mathbf{p}_0^{1,0}\right), \ldots, \mathbf{h}\left(\mathbf{x}_K, \mathbf{p}_0^{1,0}\right), \mathbf{h}\left(\mathbf{x}_1, \mathbf{p}_0^{2,0}\right), \ldots, \mathbf{h}\left(\mathbf{x}_K, \mathbf{p}_0^{M,0}\right)\right)$$

$$\mathbf{R} := \begin{bmatrix} \mathbf{R}_{1,1} & & & & & \\ & \ddots & & & & \\ & & \mathbf{R}_{K,1} & & & \\ & & & \mathbf{R}_{1,2} & & \\ & & & & \ddots & \\ & & & & & \mathbf{R}_{K,M} \end{bmatrix}$$

$$J\left(\mathbf{z}|\mathbf{x},\mathbf{p}\right) = (\mathbf{z}-\mathbf{h}(\mathbf{x},\mathbf{p}))^T\mathbf{R}^{-1}(\mathbf{z}-\mathbf{h}(\mathbf{x},\mathbf{p}))$$

Using the standard Gauss-Newton solution method, we linearize around the current estimates $\bar{\mathbf{x}}$ and $\bar{\mathbf{p}}$:

$$J \approx \left(\mathbf{z}-\mathbf{h}(\bar{\mathbf{x}},\bar{\mathbf{p}}) - \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{p}\end{bmatrix}\right)^T \mathbf{R}^{-1}\left(\mathbf{z}-\mathbf{h}(\bar{\mathbf{x}},\bar{\mathbf{p}}) - \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{p}\end{bmatrix}\right)$$

$$\mathbf{A} := \left.\frac{\partial\mathbf{h}}{\partial\mathbf{x}}\right|_{\bar{\mathbf{x}},\bar{\mathbf{p}}}, \quad \mathbf{B} := \left.\frac{\partial\mathbf{h}}{\partial\mathbf{p}}\right|_{\bar{\mathbf{x}},\bar{\mathbf{p}}}$$

Taking the derivative of J with respect to $(\delta\mathbf{x}, \delta\mathbf{p})$ and setting this to zero, we get

$$\begin{bmatrix}\mathbf{A}^T\\ \mathbf{B}^T\end{bmatrix}\mathbf{R}^{-1}\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{p}\end{bmatrix} = \begin{bmatrix}\mathbf{A}^T\\ \mathbf{B}^T\end{bmatrix}\mathbf{R}^{-1}\left(\mathbf{z}-\mathbf{h}(\bar{\mathbf{x}},\bar{\mathbf{p}})\right) \quad ,$$

which can be rewritten as

$$\begin{bmatrix}\mathbf{A}^T\mathbf{R}^{-1}\mathbf{A} & \mathbf{A}^T\mathbf{R}^{-1}\mathbf{B}\\ \mathbf{B}^T\mathbf{R}^{-1}\mathbf{A} & \mathbf{B}^T\mathbf{R}^{-1}\mathbf{B}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{p}\end{bmatrix} = \begin{bmatrix}\mathbf{A}^T\mathbf{R}^{-1}\left(\mathbf{z}-\mathbf{h}(\bar{\mathbf{x}},\bar{\mathbf{p}})\right)\\ \mathbf{B}^T\mathbf{R}^{-1}\left(\mathbf{z}-\mathbf{h}(\bar{\mathbf{x}},\bar{\mathbf{p}})\right)\end{bmatrix} \quad .$$

We will name these components:

$$\begin{bmatrix}\mathbf{U} & \mathbf{W}\\ \mathbf{W}^T & \mathbf{V}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{p}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\epsilon}_a\\ \boldsymbol{\epsilon}_b\end{bmatrix} \quad .$$

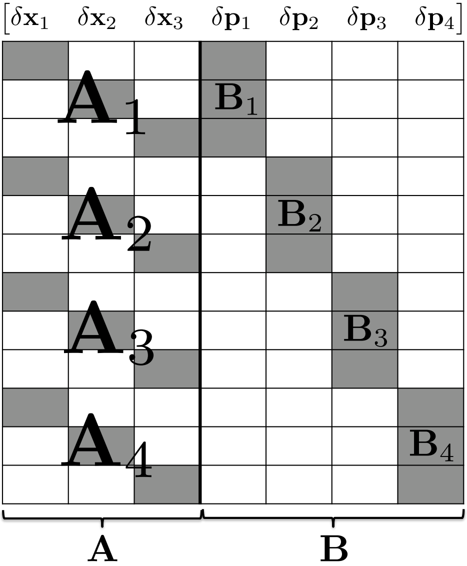
Problem: The naive solution for this equation is computationally demanding for a large number of features, or a large number of poses.

Each equation in the objective function depends on exactly one pose, $\mathbf{x}_k$, and one feature, $\mathbf{p}_0^{j,0}$.

$$J := \sum_{k=1}^{K} \sum_{j=1}^{M} \left(\mathbf{z}_{k,j} - \mathbf{h}\left(\mathbf{x}_k, \mathbf{p}_0^{j,0}\right)\right)^T \mathbf{R}_{k,j}^{-1} \left(\mathbf{z}_{k,j} - \mathbf{h}\left(\mathbf{x}_k, \mathbf{p}_0^{j,0}\right)\right)$$

This means that the Jacobians $\mathbf{A}$ and $\mathbf{B}$ are very sparse. For illustration, let's examine a problem with three cameras and four features.

$$\begin{bmatrix} \mathbf{z}_{1,1} - \mathbf{h}(\mathbf{x}_1, \mathbf{p}_0^{1,0}) \\ \mathbf{z}_{2,1} - \mathbf{h}(\mathbf{x}_2, \mathbf{p}_0^{1,0}) \\ \mathbf{z}_{3,1} - \mathbf{h}(\mathbf{x}_3, \mathbf{p}_0^{1,0}) \\ \mathbf{z}_{1,2} - \mathbf{h}(\mathbf{x}_1, \mathbf{p}_0^{2,0}) \\ \mathbf{z}_{2,2} - \mathbf{h}(\mathbf{x}_2, \mathbf{p}_0^{2,0}) \\ \mathbf{z}_{3,2} - \mathbf{h}(\mathbf{x}_3, \mathbf{p}_0^{2,0}) \\ \mathbf{z}_{1,3} - \mathbf{h}(\mathbf{x}_1, \mathbf{p}_0^{3,0}) \\ \mathbf{z}_{2,3} - \mathbf{h}(\mathbf{x}_2, \mathbf{p}_0^{3,0}) \\ \mathbf{z}_{3,3} - \mathbf{h}(\mathbf{x}_3, \mathbf{p}_0^{3,0}) \\ \mathbf{z}_{1,4} - \mathbf{h}(\mathbf{x}_1, \mathbf{p}_0^{4,0}) \\ \mathbf{z}_{2,4} - \mathbf{h}(\mathbf{x}_2, \mathbf{p}_0^{4,0}) \\ \mathbf{z}_{3,4} - \mathbf{h}(\mathbf{x}_3, \mathbf{p}_0^{4,0}) \end{bmatrix}$$



$$\begin{bmatrix} \mathbf{A}^T\mathbf{R}^{-1}\mathbf{A} & \mathbf{A}^T\mathbf{R}^{-1}\mathbf{B} \\ \mathbf{B}^T\mathbf{R}^{-1}\mathbf{A} & \mathbf{B}^T\mathbf{R}^{-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{p}})) \\ \mathbf{B}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{p}})) \end{bmatrix}$$

$$\mathbf{R}_j := \begin{bmatrix} \mathbf{R}_{1,j} & & \\ & \mathbf{R}_{2,j} & \\ & & \mathbf{R}_{3,j} \end{bmatrix}, \; \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & & \\ & \ddots & \\ & & \mathbf{R}_4 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} \mathbf{A}_1^T & \mathbf{A}_2^T & \mathbf{A}_3^T & \mathbf{A}_4^T \\ \mathbf{B}_1^T & & & \\ & \mathbf{B}_2^T & & \\ & & \mathbf{B}_3^T & \\ & & & \mathbf{B}_4^T \end{bmatrix}}_{\begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix}} \underbrace{\begin{bmatrix} \mathbf{R}_1^{-1} & & & \\ & \mathbf{R}_2^{-1} & & \\ & & \mathbf{R}_3^{-1} & \\ & & & \mathbf{R}_4^{-1} \end{bmatrix}}_{\mathbf{R}^{-1}} \underbrace{\begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 & & & \\ \mathbf{A}_2 & & \mathbf{B}_2 & & \\ \mathbf{A}_3 & & & \mathbf{B}_3 & \\ \mathbf{A}_4 & & & & \mathbf{B}_4 \end{bmatrix}}_{\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}}$$

$$\begin{bmatrix} \mathbf{A}^T\mathbf{R}^{-1}\mathbf{A} & \mathbf{A}^T\mathbf{R}^{-1}\mathbf{B} \\ \mathbf{B}^T\mathbf{R}^{-1}\mathbf{A} & \mathbf{B}^T\mathbf{R}^{-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{p}})) \\ \mathbf{B}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{p}})) \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{U} & \mathbf{W}_1 & \mathbf{W}_2 & \mathbf{W}_3 & \mathbf{W}_4 \\ \mathbf{W}_1^T & \mathbf{V}_1 & & & \\ \mathbf{W}_2^T & & \mathbf{V}_2 & & \\ \mathbf{W}_3^T & & & \mathbf{V}_3 & \\ \mathbf{W}_4^T & & & & \mathbf{V}_4 \end{bmatrix}$$

$$:= \begin{bmatrix} \sum_{j=1}^4 \mathbf{A}_j^T\mathbf{R}_j^{-1}\mathbf{A}_j & \mathbf{A}_1^T\mathbf{R}_1^{-1}\mathbf{B}_1 & \mathbf{A}_2^T\mathbf{R}_2^{-1}\mathbf{B}_2 & \mathbf{A}_3^T\mathbf{R}_3^{-1}\mathbf{B}_3 & \mathbf{A}_4^T\mathbf{R}_4^{-1}\mathbf{B}_4 \\ \mathbf{B}_1^T\mathbf{R}_1^{-1}\mathbf{A}_1 & \mathbf{B}_1^T\mathbf{R}_1^{-1}\mathbf{B}_1 & & & \\ \mathbf{B}_2^T\mathbf{R}_2^{-1}\mathbf{A}_2 & & \mathbf{B}_2^T\mathbf{R}_2^{-1}\mathbf{B}_2 & & \\ \mathbf{B}_3^T\mathbf{R}_3^{-1}\mathbf{A}_3 & & & \mathbf{B}_3^T\mathbf{R}_3^{-1}\mathbf{B}_3 & \\ \mathbf{B}_4^T\mathbf{R}_4^{-1}\mathbf{A}_4 & & & & \mathbf{B}_4^T\mathbf{R}_4^{-1}\mathbf{B}_4 \end{bmatrix}$$

Note: $\mathbf{U}$ is also block diagonal (Proof is left as an exercise).

Returning to the highest level, we have this equation where both $\mathbf{U}$ and $\mathbf{V}$ are block diagonal:

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x} \\ \delta \mathbf{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_a \\ \boldsymbol{\epsilon}_b \end{bmatrix}$$

We can rearrange this equation using block elementary row operations:

$$\begin{bmatrix} \mathbf{1} & -\mathbf{W}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x} \\ \delta \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & -\mathbf{W}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\epsilon}_a \\ \boldsymbol{\epsilon}_b \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^T & \mathbf{0} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x} \\ \delta \mathbf{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_a - \mathbf{W}\mathbf{V}^{-1}\boldsymbol{\epsilon}_b \\ \boldsymbol{\epsilon}_b \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^T & \mathbf{0} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_a - \mathbf{W}\mathbf{V}^{-1}\boldsymbol{\epsilon}_b \\ \boldsymbol{\epsilon}_b \end{bmatrix}$$

What have we gained?

- $\mathbf{V}$ is block diagonal, inverting it is easy!
- The solution for $\delta\mathbf{x}$ is decoupled from the solution for $\delta\mathbf{p}$.
- $\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^T$ is dense but small.
- After solving for $\delta\mathbf{x}$, $\delta\mathbf{p}$ may be found by back-substitution.
- Using this method, much larger problems may be solved using the same computational resources.

Bundle Adjustment: one iteration *

1. Compute the Jacobian matrices $\mathbf{A}_i$, $\mathbf{B}_i$, and error terms $\boldsymbol{\epsilon}_i$
2. Compute intermediate values:

$$\begin{aligned}
\mathbf{U} &\leftarrow \sum_{i=1}^{M} \mathbf{A}_i^T \mathbf{R}_i^{-1} \mathbf{A}_i \\
\boldsymbol{\epsilon}_a &\leftarrow \sum_{i=1}^{M} \mathbf{A}_i \mathbf{R}_i^{-1} \boldsymbol{\epsilon}_i \\
\mathbf{V}_i^{-1} &\leftarrow (\mathbf{B}_i^T \mathbf{R}_i^{-1} \mathbf{B}_i)^{-1} \\
\mathbf{W}_i &\leftarrow \mathbf{A}_i^T \mathbf{R}_i^{-1} \mathbf{B}_i \\
\boldsymbol{\epsilon}_{b_i} &\leftarrow \mathbf{B}_i^T \mathbf{R}_i^{-1} \boldsymbol{\epsilon}_i \\
\mathbf{Y}_i &\leftarrow \mathbf{W}_i \mathbf{V}_i^{-1}
\end{aligned}$$

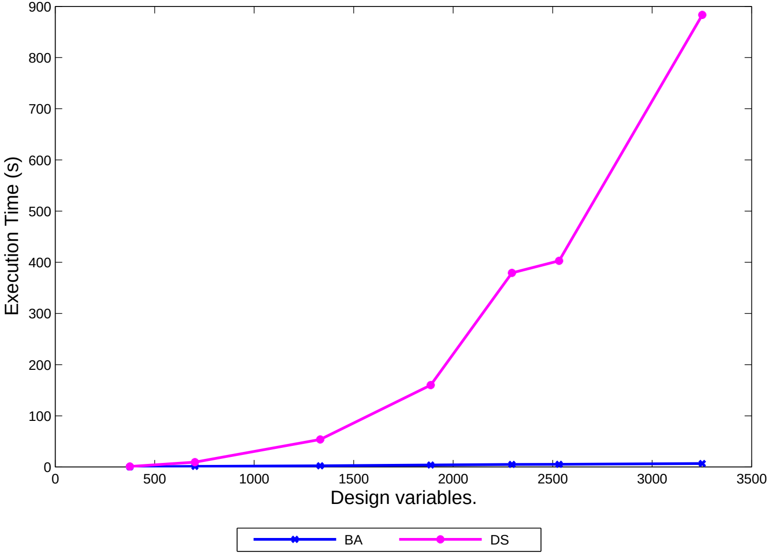
3. Solve the reduced bundle system

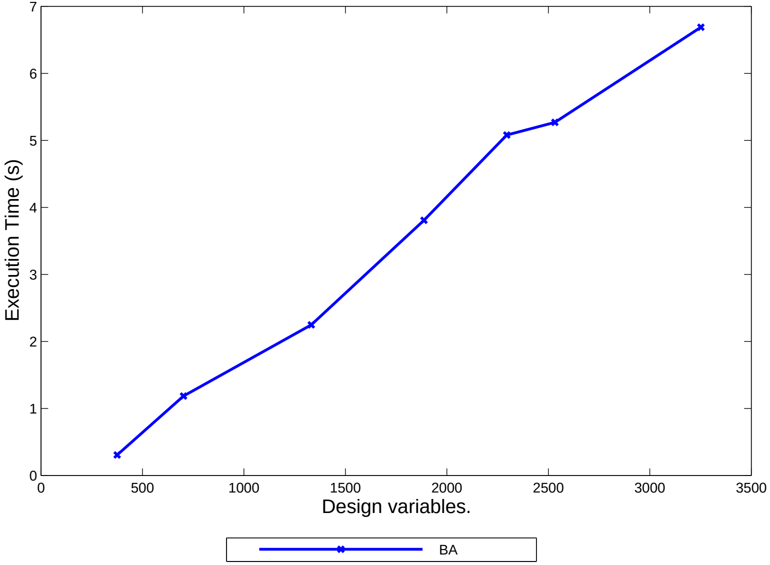
$$\left(\mathbf{U} - \sum_{i=1}^{M} \mathbf{Y}_i \mathbf{W}_i^T\right) \delta\mathbf{x} = \boldsymbol{\epsilon}_a - \sum_{i=1}^{M} \mathbf{Y}_i \boldsymbol{\epsilon}_{b_i}$$

4. Back substitute to recover the feature perturbations

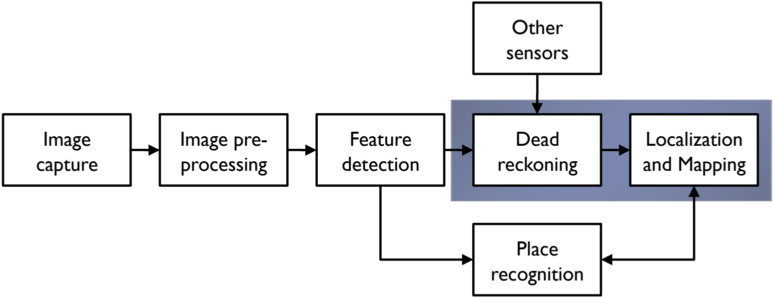
$$\delta\mathbf{p}_i = \mathbf{V}_i^{-1} \left(\boldsymbol{\epsilon}_{b_i} - \mathbf{W}_i^T \delta\mathbf{x}\right)$$

* Adapted from R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521623049, 2000.


900
800
700
600
500
400
300
200
100
0
Execution Time (s)
0
500
1000
1500
2000
2500
3000
3500
Design variables.
BA
DS


7
6
5
4
3
2
1
0
Execution Time (s)
0
500
1000
1500
2000
2500
3000
3500
Design variables.
BA

Bundle adjustment is a central component of the emerging visual SLAM architecture



Now we consider several upgrades and modifications used in state-of-the-art SLAM systems

1. Including a motion model
2. Using a sliding window
3. Keyframe selection
4. Relative coordinates

Upgrade 1: Motion Model

Include dead-reckoning data from other sensors: *GraphSLAM*[*].

$$\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)$$

This equation has no dependence on the features so the normal equations become

$$\begin{bmatrix} \mathbf{M}^T & \mathbf{A}^T \\ & \mathbf{B}^T \end{bmatrix} \begin{bmatrix} \mathbf{Q}^{-1} & \\ & \mathbf{R}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{M} & \\ \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{M}^T\mathbf{Q}^{-1} & \mathbf{A}^T\mathbf{R}^{-1} \\ & \mathbf{B}^T\mathbf{R}^{-1} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}} - \mathbf{g}(\bar{\mathbf{x}}, \mathbf{u}, \mathbf{0}) \\ \mathbf{z} - \mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) \end{bmatrix},$$

which is just

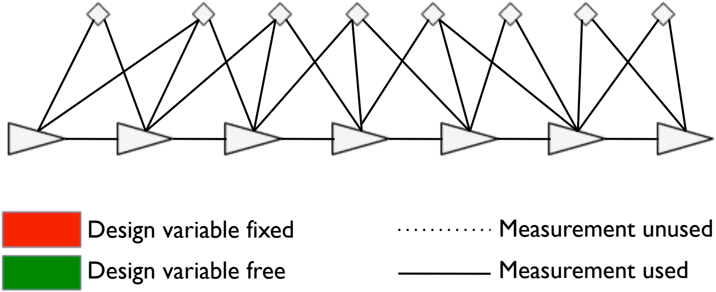
$$\begin{bmatrix} \mathbf{M}^T\mathbf{Q}^{-1}\mathbf{M} + \mathbf{U} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{M}^T\mathbf{Q}^{-1}\left(\bar{\mathbf{x}} - \mathbf{g}(\bar{\mathbf{x}}, \mathbf{u}, \mathbf{0})\right) + \boldsymbol{\epsilon}_a \\ \boldsymbol{\epsilon}_b \end{bmatrix}.$$

Now the top left block is not block diagonal but the marginalization presented in the previous slides (map onto path) will still work.

[*] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press, 2001.

Upgrade 2: Sliding Window

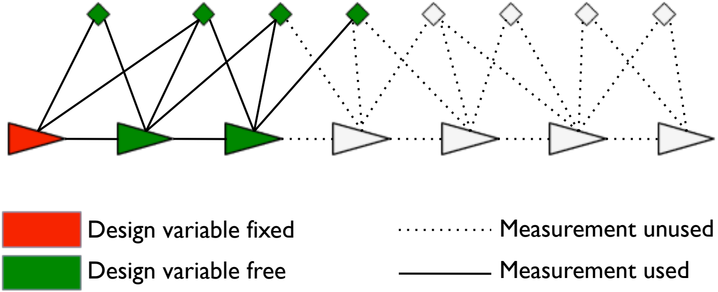
- Perform bundle adjustment over a sliding window of poses.
- Variation 1: Hold some poses fixed, leave some free*



*K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In Proceedings of the International Symposium on Research in Robotics (ISRR), November 2007.

Upgrade 2: Sliding Window

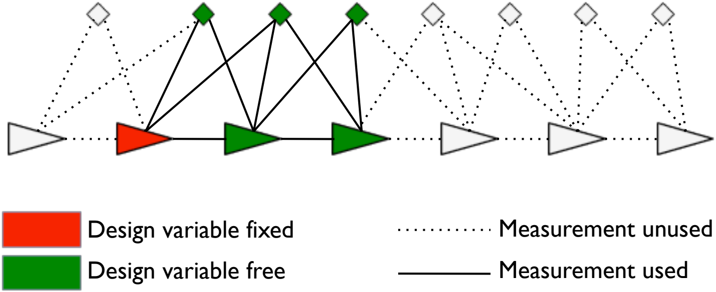
- Perform bundle adjustment over a sliding window of poses.
- Variation 1: Hold some poses fixed, leave some free*



*K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In Proceedings of the International Symposium on Research in Robotics (ISRR), November 2007.

Upgrade 2: Sliding Window

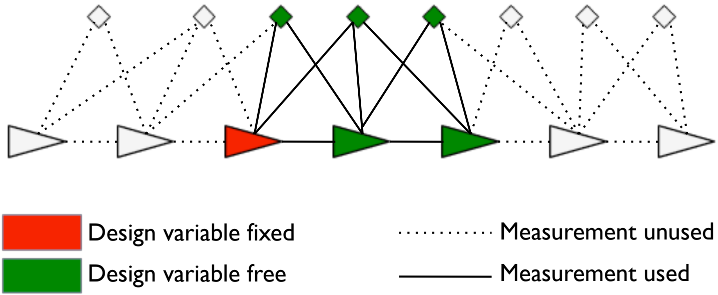
- Perform bundle adjustment over a sliding window of poses.
- Variation 1: Hold some poses fixed, leave some free*



*K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In Proceedings of the International Symposium on Research in Robotics (ISRR), November 2007.

Upgrade 2: Sliding Window

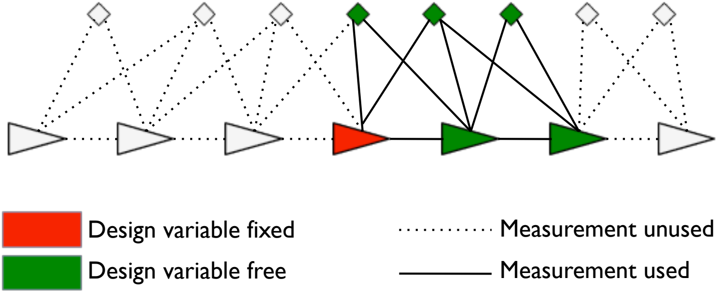
- Perform bundle adjustment over a sliding window of poses.
- Variation 1: Hold some poses fixed, leave some free*



*K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In Proceedings of the International Symposium on Research in Robotics (ISRR), November 2007.

Upgrade 2: Sliding Window

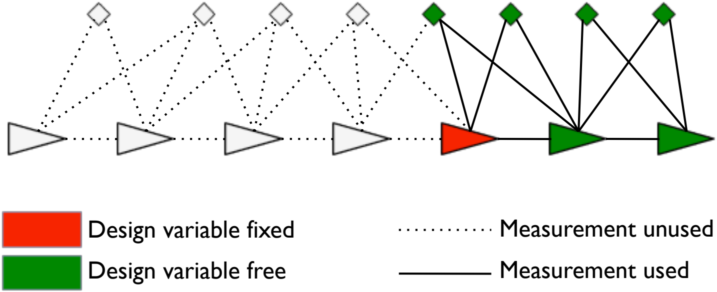
- Perform bundle adjustment over a sliding window of poses.
- Variation 1: Hold some poses fixed, leave some free*



*K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In Proceedings of the International Symposium on Research in Robotics (ISRR), November 2007.

Upgrade 2: Sliding Window

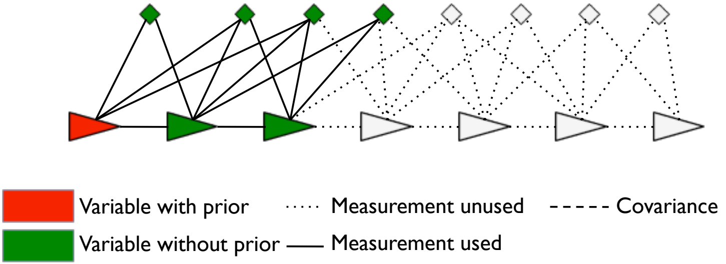
- Perform bundle adjustment over a sliding window of poses.
- Variation 1: Hold some poses fixed, leave some free*



*K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In Proceedings of the International Symposium on Research in Robotics (ISRR), November 2007.

Upgrade 2: Sliding Window

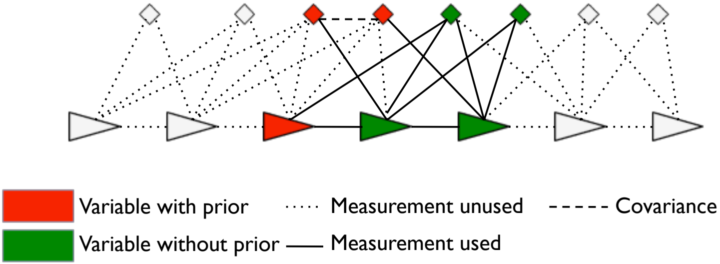
- Perform bundle adjustment over a sliding window of poses.
- Variation 2: Marginalize out variables as the window slides along*



* G. Sibley, L. Matthies, and G. Sukhatme. A sliding window filter for incremental slam. In Unifying Perspectives in Computational and Robot Vision, volume 8 of Lecture Notes in Electrical Engineering, pages 103Ð112. Springer US, 2008.
P. F. Mclauchlan. The variable state dimension filter applied to surface-based structure from motion. Technical Report VSSP-TR-4/99, University of Surrey, Guildford, UK, 1999.

Upgrade 2: Sliding Window

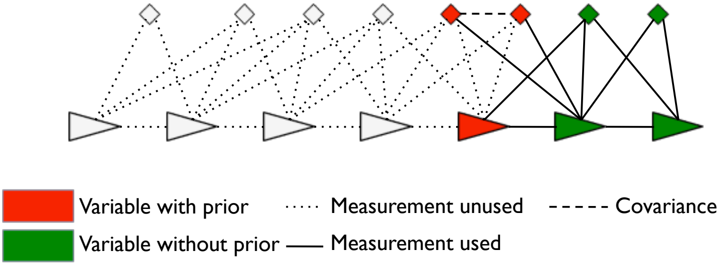
- Perform bundle adjustment over a sliding window of poses.
- Variation 2: Marginalize out variables as the window slides along*



*G. Sibley, L. Matthies, and G. Sukhatme. A sliding window filter for incremental slam. In Unifying Perspectives in Computational and Robot Vision, volume 8 of Lecture Notes in Electrical Engineering, pages 103Ð112. Springer US, 2008.
P. F. Mclauchlan. The variable state dimension filter applied to surface-based structure from motion. Technical Report VSSP-TR-4/99, University of Surrey, Guildford, UK, 1999.

Upgrade 2: Sliding Window

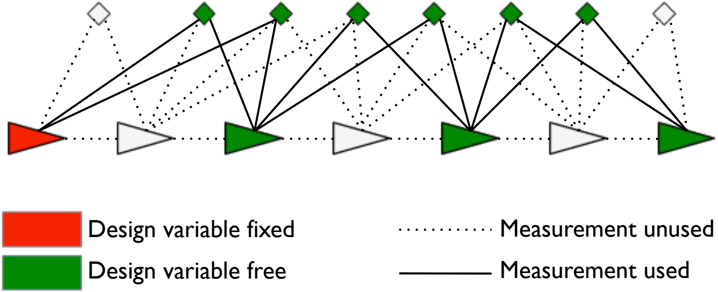
- Perform bundle adjustment over a sliding window of poses.
- Variation 2: Marginalize out variables as the window slides along*



*G. Sibley, L. Matthies, and G. Sukhatme. A sliding window filter for incremental slam. In Unifying Perspectives in Computational and Robot Vision, volume 8 of Lecture Notes in Electrical Engineering, pages 103Ð112. Springer US, 2008.
P. F. Mclauchlan. The variable state dimension filter applied to surface-based structure from motion. Technical Report VSSP-TR-4/99, University of Surrey, Guildford, UK, 1999.

Upgrade 3: Keyframe Selection

- Variation 1: Minimize the complete system for a subset of frames*



*G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR07), Nara, Japan, November 2007.

Upgrade 3: Keyframe Selection

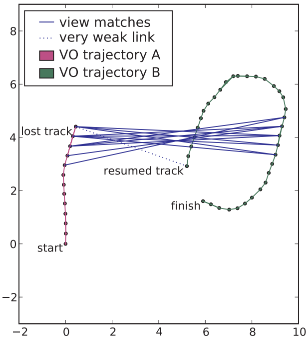
- Variation 1: Minimize the complete system for a subset of frames*

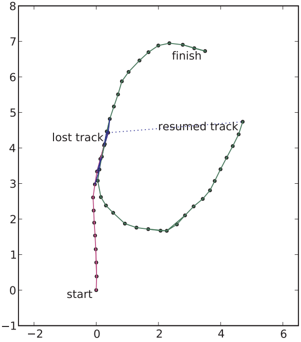
http://www.youtube.com/watch?v=Y9HMn6bd-v8

*G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR07), Nara, Japan, November 2007.

Upgrade 3: Keyframe Selection

- Variation 2: Build a skeleton graph of relative pose constraints*

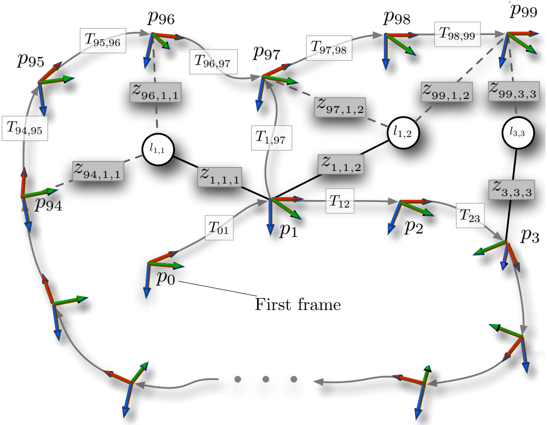




*K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based Maps. The International Journal of Robotics Research (2010)

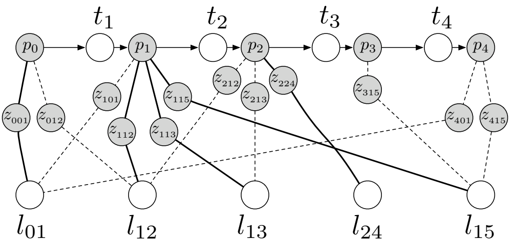
Upgrade 4: Relative Formulation

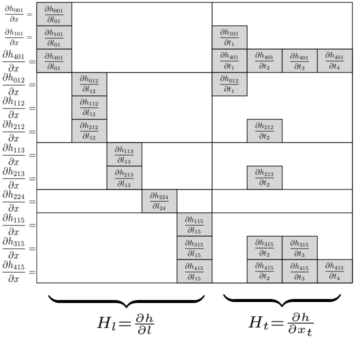
- Rather than estimating everything in a global frame, estimate the relative transformations between poses.*



* G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment. The International Journal of Robotics Research, 2010

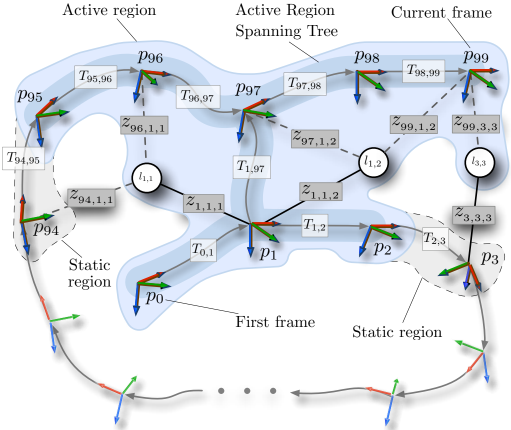
- Slightly more complicated Jacobian evaluations. This ruins the sparsity of $\mathbf{U}$ but not of $\mathbf{V}$.*





* G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment. The International Journal of Robotics Research, 2010

- Adaptive window approximates the full batch solution.*

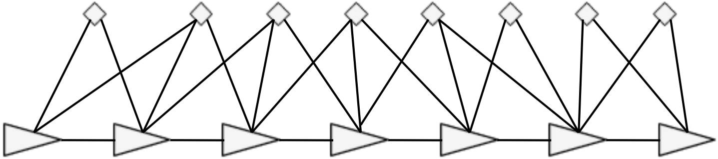


* G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment. The International Journal of Robotics Research, 2010

Summary

- Bundle adjustment is the gold-standard, maximum-likelihood, batch SLAM method. We should know it.
- The sparse factorization method enables the application of this method to large problems
- Modern visual SLAM systems use bundle adjustment as one of the fundamental algorithms enabling fast, accurate mapping over large scales

Questions?



paul.furgale@utoronto.ca