http://users.ics.forth.gr/~lourakis/sba/PRCV_colloq.pdf
lecture by Lourakis "Bundle adjustment gone public"

- Bundle Adjustment (BA) is a key ingredient of SaM, almost always used as its last step
  - • It is an optimization problem over the 3D structure and viewing parameters (camera pose, intrinsic calibration, & radial distortion parameters), which are simultaneously refined for minimizing reprojection error
  - very large nonlinear least squares problem, typically solved with the Levenberg-Marquardt (LM) algorithm
  - Std LM involves the repetitive solution of linear systems, each with O(N^3) time and O(N^2) storage complexity, resp.
    - • Example: for 54 cameras and 5207 3D points, N = 15945. ==> N^3 = 1e12
  - Sparse LM is a better solution.
  - Example:
    - M images
    - N features
    - **x_i_j** = projection of feature "i" on image "j"
    - **a_j** = vector of parameters for camera "j"
    - **b_i** = vectors of parameters for point "i"
    - Q(aj , bi) = the predicted projection of point i on image j,
    - d(., .) the Euclidean distance between image points
    - vij = 1 iff point i is visible in image j
    - minimize reproduction error over a_j, b_i: min_aj, bi (summation_i=1_to_N(summation_j=1_to_M((v_i_j * d(Q(aj,bi), x_i_j))^2)))
      - ==> total number of parameters is M*(camera parameters) + N*(point parameters)
    - let **P** = parameter vector of camera then point parameters = [P_C P_P]
    - let **X** = [(x_hat_1_1)^T   x_hat_1_2)^T … x_hat_1_M)^T   x_hat_2_1)^T … x_hat_N_M)^T]
    - where x_hat = Q(a_j, b_i),
    - let error **eps** = [(eps_1_1)^T   eps_1_2)^T … eps_1_M)^T   eps_2_1)^T … eps_N_M)^T]
    - where eps = x_i_j - x_hat_i_j

http://users.ics.forth.gr/~lourakis/sba/PRCV_colloq.pdf

Bundle Adjustment (BA) becomes

min (summation_i=1_to_N(summation_j=1_to_M((eps_i_j)^2))) over P

Jacobian J = d(X_hat) / d(P) which has a block structure because of P being [camera parameters point parameters]

J = [A | B] where A = d(X_hat) / d(a) and B = A = d(X_hat) / d(b)

The LM updating vector delta = [(delta(a))^T (delta(b))^T

The normal equations:

| A^T * A A^T * B | * | delta(a)) | = | A^T * eps |
| B^T * A B^T * B | | delta(b)) | | B^T * eps |

The lhs matrix above is sparse due to A and B being sparse:

∂x̂ij ∂a_k = 0, ∀ j != k and

∂x̂ij ∂b_k = 0, ∀ i != k

http://users.ics.forth.gr/~lourakis/sba/PRCV_colloq.pdf

(example cont.) M images = 3, N features = 4

$\mathbf{J} = \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}}$ has a block structure $[\,\mathbf{A}|\mathbf{B}\,]$,

Let $\mathbf{A}_{ij} = \frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{a}_j}$ and $\mathbf{B}_{ij} = \frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{b}_i}$

- The Jacobian J in block form:

$$
\frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} =
\begin{array}{c}
\\ \mathbf{x}_{11}\\ \mathbf{x}_{12}\\ \mathbf{x}_{13}\\ \mathbf{x}_{21}\\ \mathbf{x}_{22}\\ \mathbf{x}_{23}\\ \mathbf{x}_{31}\\ \mathbf{x}_{32}\\ \mathbf{x}_{33}\\ \mathbf{x}_{41}\\ \mathbf{x}_{42}\\ \mathbf{x}_{43}
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} \mathbf{a}_1{}^T & \mathbf{a}_2{}^T & \mathbf{a}_3{}^T & \mathbf{b}_1{}^T & \mathbf{b}_2{}^T & \mathbf{b}_3{}^T & \mathbf{b}_4{}^T \end{array}\\
\begin{pmatrix}
\mathbf{A}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{A}_{12} & \mathbf{0} & \mathbf{B}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{A}_{13} & \mathbf{B}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{A}_{21} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{21} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{A}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{22} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{A}_{23} & \mathbf{0} & \mathbf{B}_{23} & \mathbf{0} & \mathbf{0}\\
\mathbf{A}_{31} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{31} & \mathbf{0}\\
\mathbf{0} & \mathbf{A}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{32} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{A}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{33} & \mathbf{0}\\
\mathbf{A}_{41} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{41}\\
\mathbf{0} & \mathbf{A}_{42} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{42}\\
\mathbf{0} & \mathbf{0} & \mathbf{A}_{43} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{43}
\end{pmatrix}
\end{array}
\quad (1)
$$

This is the so-called *primary structure* of BA

- Approximate Hessian in block form:

$$
\mathbf{J}^T\mathbf{J} =
\begin{array}{c}
\\ \mathbf{a}_1\\ \mathbf{a}_2\\ \mathbf{a}_3\\ \mathbf{b}_1\\ \mathbf{b}_2\\ \mathbf{b}_3\\ \mathbf{b}_4
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} \mathbf{a}_1{}^T & \mathbf{a}_2{}^T & \mathbf{a}_3{}^T & \mathbf{b}_1{}^T & \mathbf{b}_2{}^T & \mathbf{b}_3{}^T & \mathbf{b}_4{}^T \end{array}\\
\begin{pmatrix}
\mathbf{U}_1 & 0 & 0 & \mathbf{W}_{11} & \mathbf{W}_{21} & \mathbf{W}_{31} & \mathbf{W}_{41}\\
0 & \mathbf{U}_2 & 0 & \mathbf{W}_{12} & \mathbf{W}_{22} & \mathbf{W}_{32} & \mathbf{W}_{42}\\
0 & 0 & \mathbf{U}_3 & \mathbf{W}_{13} & \mathbf{W}_{23} & \mathbf{W}_{33} & \mathbf{W}_{43}\\
\mathbf{W}_{11}^T & \mathbf{W}_{12}^T & \mathbf{W}_{13}^T & \mathbf{V}_1 & 0 & 0 & 0\\
\mathbf{W}_{21}^T & \mathbf{W}_{22}^T & \mathbf{W}_{23}^T & 0 & \mathbf{V}_2 & 0 & 0\\
\mathbf{W}_{31}^T & \mathbf{W}_{32}^T & \mathbf{W}_{33}^T & 0 & 0 & \mathbf{V}_3 & 0\\
\mathbf{W}_{41}^T & \mathbf{W}_{42}^T & \mathbf{W}_{43}^T & 0 & 0 & 0 & \mathbf{V}_4
\end{pmatrix}
\end{array}
\equiv \quad (2)
$$

$$
\equiv \begin{pmatrix} \mathbf{U} & \mathbf{W}\\ \mathbf{W}^T & \mathbf{V} \end{pmatrix},
$$

$\mathbf{U}_j \equiv \sum_{i=1}^{4} \mathbf{A}_{ij}^T \mathbf{A}_{ij}$, **for 1 image, summing over all features**

$\mathbf{V}_i \equiv \sum_{j=1}^{3} \mathbf{B}_{ij}^T \mathbf{B}_{ij}$, **for 1 feature, summing over all images**

$\mathbf{W}_{ij} = \mathbf{A}_{ij}^T \mathbf{B}_{ij}$

http://users.ics.forth.gr/~lourakis/sba/PRCV_colloq.pdf

(example cont.) M images = 3, N features = 4

- The augmented normal equations $(\mathbf{J}^T\mathbf{J} + \mu\mathbf{I})\delta_\mathbf{p} = \mathbf{J}^T\epsilon$ take the form

where μ > 0

(3) $$\begin{pmatrix} \mathbf{U}^* & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V}^* \end{pmatrix} \begin{pmatrix} \delta_\mathbf{a} \\ \delta_\mathbf{b} \end{pmatrix} = \begin{pmatrix} \epsilon_\mathbf{a} \\ \epsilon_\mathbf{b} \end{pmatrix}$$

- Performing block Gaussian elimination in the lhs matrix, $\delta_\mathbf{a}$ is determined with Cholesky from $\mathbf{V}^*$'s Schur complement:

(4) $$(\mathbf{U}^* - \mathbf{W}\,\mathbf{V}^{*-1}\,\mathbf{W}^T)\,\delta_\mathbf{a} = \epsilon_\mathbf{a} - \mathbf{W}\,\mathbf{V}^{*-1}\,\epsilon_\mathbf{b}$$

Schur complement: multiply 1st matrix on res by
| I 0 |
| (((-V*)^-1)W^T I |
resulting in:
| (U*) - W(((V*)^-1)W^T W | * | delta_a | = | eps_a |
| 0 (V*) | | delta_b | | eps_b |

$$\mathbf{V}^{*-1} = \begin{pmatrix} \mathbf{V}_1^{*-1} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{V}_2^{*-1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

note (V*) is invertible and only the block diagonals are populated, so each V_i is inverted.

separate delta_b: 0*delta_a + (V*) * delta_b = eps_b ==> delta_b = eps_b * ((V*)^-1)
solving for delta_a (typically M images << N features) after substitute delta_b:
((U*) - W(((V*)^-1)W^T) * delta_a + W * delta_b = epa_a
((U*) - W(((V*)^-1)W^T) * delta_a = epa_a - W((V*)^-1)eps_b

NOTE: (U*) - W(((V*)^-1)W^T is called the reduced camera matrix (because delta_a is camera parameters)

http://users.ics.forth.gr/~lourakis/sba/PRCV_colloq.pdf

RCM (reduced camera matrix) is sparse because not all features appear in all cameras.
this is known as ***secondary structure*.**

For very large datasets, RCM tends to be in one of two classes:

(1) ○ visual mapping: extended areas are traversed, limited image overlap (sparse RCM)

(2) ○ centered-object: a large number of overlapping images taken in a small area (dense RCM)

Solving for delta_a in the equation containing RCM. several ways:

(1) Store as dense, decompose with **ordinary linear algebra** ○[M. Lourakis, A. Argyros: SBA: A Software Package For Generic Sparse Bundle Adjustment. ACM Trans. Math. Softw. 36(1): (2009) ○ C. Engels, H. Stewenius, D. Nister: Bundle Adjustment Rules. Photogrammetric Computer Vision (PCV), 2006.

(2) • Store as sparse, factorize with **sparse direct solvers** ○ K. Konolige: Sparse Sparse Bundle Adjustment. BMVC 2010: 1-11

(3) Store as sparse, use **conjugate gradient methods** memory efficient, iterative, precoditioners necessary! ○ S. Agarwal, N. Snavely, S.M. Seitz, R. Szeliski: Bundle Adjustment in the Large. ECCV (2) 2010: 29-42 ○ M. Byrod, K. Astrom: Conjugate Gradient Bundle Adjustment. ECCV (2) 2010: 114-127

(4) • Avoid storing altogether ○ C. Wu, S. Agarwal, B. Curless, S.M. Seitz: Multicore Bundle Adjustment. CVPR 2011: 30 57-3064 ○ M. Lourakis: Sparse Non-linear Least Squares Optimization for Geometric Vision. ECCV (2) 2010: 43-56

Engels, Stewenius, Nister 2006, “Bundle Adjustment Rules”

m images (= video frames from same calibrated camera)
? features
each feature **x** has **M** dimensions
n iterations of bundle adjustment over the last m video frames

their “RCM” is formed from a jacobian which places point parameters before camera parameters,
so is different than that in the Lourakis notes.

$$J_f = \left[\begin{array}{cc} J_P & J_C \end{array} \right], \tag{15}$$

$$H = \left[\begin{array}{cc} J_P^\top J_P & J_P^\top J_C \\ J_C^\top J_P & J_C^\top J_C \end{array} \right], \tag{16}$$

$$\left[\begin{array}{cc} H_{PP} & H_{PC} \\ H_{PC}^\top & H_{CC} \end{array} \right] \left[\begin{array}{c} dP \\ dC \end{array} \right] = \left[\begin{array}{c} b_P \\ b_C \end{array} \right], \tag{17}$$

where we have defined $H_{PP} = J_P^\top J_P$, $H_{PC} = J_P^\top J_C$, $H_{CC} = J_C^\top J_C$, $b_P = -J_P^\top f$,$b_C = -J_C^\top f$ to simpify the notation, and dP and dC represent the update of the point parameters and the camera parameters, respectively. Note that the matrices H_{PP} and H_{CC} are block-diagonal, where the blocks correspond to

of as multiplying by

$$\left[\begin{array}{cc} I & 0 \\ -H_{PC}^\top & I \end{array} \right] \tag{20}$$

from the left on both sides, resulting in the smaller equation system (from the lower part)

$$\underbrace{(H_{CC} - H_{PC}^\top H_{PP}^{-1} H_{PC})}_{A}\, dC = \underbrace{b_C - H_{PC}^\top H_{PP}^{-1} b_P}_{B} \tag{21}$$

for the camera parameter update dC. For very large systems,

$\boxed{\text{We use straightforward Cholesky factorization.}}$

Engels, Stewenius, Nister 2006, “Bundle Adjustment Rules”

their “RCM” is formed from a jacobian which places point parameters before camera parameters, so is different than that in the Lourakis notes.

Lourakis

Let $\mathbf{A}_{ij} = \underset{\textbf{C}}{\dfrac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{a}_j}}$ and $\mathbf{B}_{ij} = \underset{\textbf{P}}{\dfrac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{b}_i}}$

$$\mathbf{J} = \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} = \left[\frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{a}} \; \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{b}} \right] = \left[\; J_C \quad J_P \; \right]$$

- The Jacobian J in block form:

$$\frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} = \begin{array}{c} \\ \mathbf{x}_{11} \\ \mathbf{x}_{12} \\ \mathbf{x}_{13} \\ \mathbf{x}_{21} \\ \mathbf{x}_{22} \\ \mathbf{x}_{23} \\ \mathbf{x}_{31} \\ \mathbf{x}_{32} \\ \mathbf{x}_{33} \\ \mathbf{x}_{41} \\ \mathbf{x}_{42} \\ \mathbf{x}_{43} \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathbf{a}_1{}^T & \mathbf{a}_2{}^T & \mathbf{a}_3{}^T & \mathbf{b}_1{}^T & \mathbf{b}_2{}^T & \mathbf{b}_3{}^T & \mathbf{b}_4{}^T \end{array} \\ \begin{pmatrix} \mathbf{A}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{12} & \mathbf{0} & \mathbf{B}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{13} & \mathbf{B}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{21} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{22} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{23} & \mathbf{0} & \mathbf{B}_{23} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{31} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{31} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{32} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{33} & \mathbf{0} \\ \mathbf{A}_{41} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{41} \\ \mathbf{0} & \mathbf{A}_{42} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{42} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{43} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{43} \end{pmatrix} \end{array} \quad (1)$$

Engels

Let $\mathbf{A}_{ij} = \underset{\textbf{C}}{\dfrac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{a}_j}}$ and $\mathbf{B}_{ij} = \underset{\textbf{P}}{\dfrac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{b}_i}}$

$$J_f = \left[\; J_P \quad J_C \; \right],$$

- The Jacobian J in block form:

$$\frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} = \begin{array}{c} \\ \mathbf{x}_{11} \\ \mathbf{x}_{12} \\ \mathbf{x}_{13} \\ \mathbf{x}_{21} \\ \mathbf{x}_{22} \\ \mathbf{x}_{23} \\ \mathbf{x}_{31} \\ \mathbf{x}_{32} \\ \mathbf{x}_{33} \\ \mathbf{x}_{41} \\ \mathbf{x}_{42} \\ \mathbf{x}_{43} \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathbf{b}_1{}^T & \mathbf{b}_2{}^T & \mathbf{b}_3{}^T & \mathbf{b}_4{}^T & \mathbf{a}_1{}^T & \mathbf{a}_2{}^T & \mathbf{a}_3{}^T \end{array} \\ \begin{pmatrix} \mathbf{B}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{12} & \mathbf{0} \\ \mathbf{B}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{13} \\ \mathbf{0} & \mathbf{B}_{21} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{21} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{31} & \mathbf{0} & \mathbf{A}_{31} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{32} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{33} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{41} & \mathbf{A}_{41} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{42} & \mathbf{0} & \mathbf{A}_{42} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{43} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{43} \end{pmatrix} \end{array} \quad (1)$$

Engels, Stewenius, Nister 2006, "Bundle Adjustment Rules"

their "RCM" is formed from a jacobian which places point parameters before camera parameters, so is different than that in the Lourakis notes.

Lourakis

Let $\mathbf{A}_{ij} = \underbrace{\frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{a}_j}}_{\mathbf{C}}$ and $\mathbf{B}_{ij} = \underbrace{\frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{b}_i}}_{\mathbf{P}}$

$$\mathbf{J} = \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} = \left[\frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{a}} \; \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{b}} \right] = [\; J_C \quad J_P \;]$$

$$\mathbf{J}^T\mathbf{J} =$$

$$\begin{array}{c} \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} {\mathbf{a}_1}^T & {\mathbf{a}_2}^T & {\mathbf{a}_3}^T & {\mathbf{b}_1}^T & {\mathbf{b}_2}^T & {\mathbf{b}_3}^T & {\mathbf{b}_4}^T \end{array} \\
\begin{pmatrix}
\mathbf{U}_1 & 0 & 0 & \mathbf{W}_{11} & \mathbf{W}_{21} & \mathbf{W}_{31} & \mathbf{W}_{41} \\
0 & \mathbf{U}_2 & 0 & \mathbf{W}_{12} & \mathbf{W}_{22} & \mathbf{W}_{32} & \mathbf{W}_{42} \\
0 & 0 & \mathbf{U}_3 & \mathbf{W}_{13} & \mathbf{W}_{23} & \mathbf{W}_{33} & \mathbf{W}_{43} \\
\mathbf{W}_{11}^T & \mathbf{W}_{12}^T & \mathbf{W}_{13}^T & \mathbf{V}_1 & 0 & 0 & 0 \\
\mathbf{W}_{21}^T & \mathbf{W}_{22}^T & \mathbf{W}_{23}^T & 0 & \mathbf{V}_2 & 0 & 0 \\
\mathbf{W}_{31}^T & \mathbf{W}_{32}^T & \mathbf{W}_{33}^T & 0 & 0 & \mathbf{V}_3 & 0 \\
\mathbf{W}_{41}^T & \mathbf{W}_{42}^T & \mathbf{W}_{43}^T & 0 & 0 & 0 & \mathbf{V}_4
\end{pmatrix}
\end{array}$$

Engels

Let $\mathbf{A}_{ij} = \underbrace{\frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{a}_j}}_{\mathbf{C}}$ and $\mathbf{B}_{ij} = \underbrace{\frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{b}_i}}_{\mathbf{P}}$

$$J_f = [\; J_P \quad J_C \;],$$

$$\mathbf{J}^T\mathbf{J} =$$

$$\begin{array}{c} \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} {\mathbf{b}_1}^T & {\mathbf{b}_2}^T & {\mathbf{b}_3}^T & {\mathbf{b}_4}^T & {\mathbf{a}_1}^T & {\mathbf{a}_2}^T & {\mathbf{a}_3}^T \end{array} \\
\begin{pmatrix}
\mathbf{V}_1 & 0 & 0 & 0 & \mathbf{W}_{11}^T & \mathbf{W}_{12}^T & \mathbf{W}_{13}^T \\
0 & \mathbf{V}_2 & 0 & 0 & \mathbf{W}_{21}^T & \mathbf{W}_{22}^T & \mathbf{W}_{23}^T \\
0 & 0 & \mathbf{V}_3 & 0 & \mathbf{W}_{31}^T & \mathbf{W}_{32}^T & \mathbf{W}_{33}^T \\
0 & 0 & 0 & \mathbf{V}_4 & \mathbf{W}_{41}^T & \mathbf{W}_{42}^T & \mathbf{W}_{43}^T \\
\mathbf{W}_{11} & \mathbf{W}_{21} & \mathbf{W}_{31} & \mathbf{W}_{41} & \mathbf{U}_1 & 0 & 0 \\
\mathbf{W}_{12} & \mathbf{W}_{22} & \mathbf{W}_{32} & \mathbf{W}_{42} & 0 & \mathbf{U}_2 & 0 \\
\mathbf{W}_{13} & \mathbf{W}_{23} & \mathbf{W}_{33} & \mathbf{W}_{43} & 0 & 0 & \mathbf{U}_3
\end{pmatrix}
\end{array}$$

where

$\mathbf{U}_j \equiv \sum_{i=1}^{4} \mathbf{A}_{ij}^T \mathbf{A}_{ij}$, **for 1 image, sum over features**

$\mathbf{V}_i \equiv \sum_{j=1}^{3} \mathbf{B}_{ij}^T \mathbf{B}_{ij}$, **for 1 feature, sum over images**

$\mathbf{W}_{ij} = \mathbf{A}_{ij}^T \mathbf{B}_{ij}$

Engels, Stewenius, Nister 2006, “Bundle Adjustment Rules”

Note that $\mathbf{V}^{*-1} = \begin{pmatrix} \mathbf{V}_1^{*-1} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{V}_2^{*-1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$

Lourakis

The augmented normal equations $(\mathbf{J}^T\mathbf{J} + \mu\mathbf{I})\delta_{\mathbf{p}} = \mathbf{J}^T\epsilon$ take the form

$$\begin{pmatrix} \mathbf{U}^* & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V}^* \end{pmatrix} \begin{pmatrix} \delta_{\mathbf{a}} \\ \delta_{\mathbf{b}} \end{pmatrix} = \begin{pmatrix} \epsilon_{\mathbf{a}} \\ \epsilon_{\mathbf{b}} \end{pmatrix} \tag{3}$$

$$\begin{bmatrix} U - WV^{-1}W^T & 0 \end{bmatrix} \begin{bmatrix} \delta_{\mathbf{a}} \\ \delta_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} I & -WV^{-1} \end{bmatrix} \begin{bmatrix} \epsilon_{\mathbf{a}} \\ \epsilon_{\mathbf{b}} \end{bmatrix}$$

(solve delta a first because typically m_images << n_features)

determine $\delta_{\mathbf{a}}$ with Cholesky **(or other method)**

$$(\mathbf{U}^* - \mathbf{W}\ \mathbf{V}^{*-1}\ \mathbf{W}^T)\ \delta_{\mathbf{a}} = \epsilon_{\mathbf{a}} - \mathbf{W}\ \mathbf{V}^{*-1}\ \epsilon_{\mathbf{b}}$$

$\delta_{\mathbf{b}}$ can be computed by back substitution into

$$\mathbf{V}^*\ \delta_{\mathbf{b}} = \epsilon_{\mathbf{b}} - \mathbf{W}^T\ \delta_{\mathbf{a}}$$

$$\delta_{\mathbf{b}} = \mathbf{V}^{*-1}\ \epsilon_{\mathbf{b}} - \mathbf{V}^{*-1}\ \mathbf{W}^T\ \delta_{\mathbf{a}}$$

Engels

$$\begin{pmatrix} \mathbf{V}^* & \mathbf{W}^T \\ \mathbf{W} & \mathbf{U}^* \end{pmatrix} \begin{pmatrix} \delta_{\mathbf{b}} \\ \delta_{\mathbf{a}} \end{pmatrix} = \begin{pmatrix} \epsilon_{\mathbf{b}} \\ \epsilon_{\mathbf{a}} \end{pmatrix}$$

$$\begin{bmatrix} H_{PP} & H_{PC} \\ H_{PC}^\top & H_{CC} \end{bmatrix} \begin{bmatrix} dP \\ dC \end{bmatrix} = \begin{bmatrix} b_P \\ b_C \end{bmatrix},$$

$$\begin{aligned} H_{PP} &= J_P^\top J_P \\ H_{PC} &= J_P^\top J_C, \\ H_{CC} &= J_C^\top J_C, \\ b_P &= -J_P^\top f, \\ b_C &= -J_C^\top f \end{aligned}$$

$$(\mathbf{U}^* - \mathbf{W}\ \mathbf{V}^{*-1}\ \mathbf{W}^T)\ \delta_{\mathbf{a}} = \epsilon_{\mathbf{a}} - \mathbf{W}\ \mathbf{V}^{*-1}\ \epsilon_{\mathbf{b}}$$

$$\underbrace{(H_{CC} - H_{PC}^\top H_{PP}^{-1} H_{PC})}_{A}\, dC = \underbrace{b_C - H_{PC}^\top H_{PP}^{-1} b_P}_{B}$$

$\delta_{\mathbf{b}}$ can be computed by back substitution into

$$\mathbf{V}^*\ \delta_{\mathbf{b}} = \epsilon_{\mathbf{b}} - \mathbf{W}^T\ \delta_{\mathbf{a}}$$

$$H_{PP}\ dP = b_P - H_{PC}\ dC$$

$$dP = H_{PP}^{-1} b_P - H_{PP}^{-1} H_{PC} dC.$$

Engels, et al 2006

Note that $\mathbf{V}^{*-1} = \begin{pmatrix} \mathbf{V}_1^{*-1} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{V}_2^{*-1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$

$$\begin{aligned} H_{PP} &= J_P^\top J_P &\equiv \mathbf{V}^* \\ H_{PC} &= J_P^\top J_C, &\equiv \mathbf{W}^T \\ H_{CC} &= J_C^\top J_C, &\equiv \mathbf{U}^* \\ b_P &= -J_P^\top f, &\equiv \epsilon_\mathbf{b} \\ b_C &= -J_C^\top f &\equiv \epsilon_\mathbf{a} \end{aligned}$$

$\mathbf{V}_i \equiv \sum_{j=1}^3 \mathbf{B}_{ij}^T \mathbf{B}_{ij}$, <u>**for 1 feature, sum over images**</u>

$\mathbf{U}_j \equiv \sum_{i=1}^4 \mathbf{A}_{ij}^T \mathbf{A}_{ij}$, <u>**for 1 image, sum over features**</u>

$$(\mathbf{U}^* - \mathbf{W}\ \mathbf{V}^{*-1}\ \mathbf{W}^T)\ \delta_\mathbf{a} = \epsilon_\mathbf{a} - \mathbf{W}\ \mathbf{V}^{*-1}\ \epsilon_\mathbf{b}$$

Engels

$$\underbrace{(H_{CC} - H_{PC}^\top H_{PP}^{-1} H_{PC})}_{A}\, dC = \underbrace{b_C - H_{PC}^\top H_{PP}^{-1} b_P}_{B}$$

1 Initialize λ.

2 **Compute cost function** at initial camera and point configuration.

3 Clear the left hand side matrix A and right hand side vector B.

4 For each track p **(p is feature i of N)**

{

$\mathbf{V}^*$ $\mathbf{V}^*$

Clear a variable H_{pp} to represent block p of H_{PP} (in our case a symmetric 3×3 matrix) and a variable b_p to represent part p of b_P (in our case a 3-vector). $\epsilon_\mathbf{b}$ $\epsilon_\mathbf{b}$

(Compute derivatives) For each camera c on track p **(c is image j of M)**

{ Compute error vector f of reprojection in camera c of point p and its Jacobians J_p and J_c with respect to the point parameters (in our case a 2×3 matrix) and the camera parameters (in our case a 2×6 matrix), respectively.

$\mathbf{B}$ $\mathbf{A}$

$\mathbf{B}^T\mathbf{B}$ $\mathbf{V}^*$

Add $J_p^\top J_p$ to the upper triangular part of H_{pp}.

Subtract $J_p^\top f$ from b_p.

$\mathbf{B}^T$ $\epsilon_\mathbf{b}$

If camera c is free

{ $\mathbf{A}^T\mathbf{A}$

Add $J_c^\top J_c$ (optionally with an augmented diagonal) to upper triangular part of block (c, c) of left hand side matrix A (in our case a 6×6 matrix).

Compute block (p, c) of H_{PC} as $H_{pc} = J_p^\top J_c$ (in our case a 3×6 matrix) and store it until track is done.

Subtract $J_c^\top f$ from part c of right hand side vector B (related to b_C).

} }

Augment diagonal of H_{pp}, which is now accumulated and ready. Invert H_{pp}, taking advantage of the fact that it is a symmetric matrix.

Compute $H_{pp}^{-1} b_p$ and store it in a variable t_p.

(Outer product of track) For each free camera c on track p

{

Subtract $H_{pc}^\top t_p = H_{pc}^\top H_{pp}^{-1} b_p$ from part c of right hand side vector B.

Compute the matrix $H_{pc}^\top H_{pp}^{-1}$ and store it in a variable T_{pc}

For each free camera $c2 \geq c$ on track p

{

Subtract $T_{pc} H_{pc2} = H_{pc}^\top H_{pp}^{-1} H_{pc2}$ from block $(c, c2)$ of left hand side matrix A.

} }

}

Engels, et al 2006

Note that $\mathbf{V}^{*-1} = \begin{pmatrix} \mathbf{V}_1^{*-1} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{V}_2^{*-1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$

$$
\begin{aligned}
H_{PP} &= J_P^\top J_P & \equiv \mathbf{V}^* \\
H_{PC} &= J_P^\top J_C, & \equiv \mathbf{W}^T \\
H_{CC} &= J_C^\top J_C, & \equiv \mathbf{U}^* \\
b_P &= -J_P^\top f, & \equiv \epsilon_{\mathbf{b}} \\
b_C &= -J_C^\top f & \equiv \epsilon_{\mathbf{a}}
\end{aligned}
$$

$\mathbf{V}_i \equiv \sum_{j=1}^{3} \mathbf{B}_{ij}^T \mathbf{B}_{ij},$ <u>for 1 feature, sum over images</u>

$\mathbf{U}_j \equiv \sum_{i=1}^{4} \mathbf{A}_{ij}^T \mathbf{A}_{ij},$ <u>for 1 image, sum over features</u>

$$(\mathbf{U}^* - \mathbf{W}\ \mathbf{V}^{*-1}\ \mathbf{W}^T)\ \delta_{\mathbf{a}} = \epsilon_{\mathbf{a}} - \mathbf{W}\ \mathbf{V}^{*-1}\ \epsilon_{\mathbf{b}}$$

$$\underbrace{(H_{CC} - H_{PC}^\top H_{PP}^{-1} H_{PC})}_{A}\, dC = \underbrace{b_C - H_{PC}^\top H_{PP}^{-1} b_P}_{B}$$

Engels

5 (Optional) Fix gauge by freezing appropriate coordinates and thereby reducing the linear system with a few dimensions.

6 **(Linear Solving)** Cholesky factor the left hand side matrix B and solve for dC. Add frozen coordinates back in.

7 **(Back-substitution)** For each track p

{

Start with point update for this track $dp = t_p$.

For each camera c on track p

{

Subtract $T_{pc}^\top dc$ from dp (where dc is the update for camera c).

}

Compute updated point.

}

8 **Compute the cost function** for the updated camera and point configuration.

9 If cost function has improved, accept the update step, decrease λ and go to Step 3 (unless converged, in which case quit).

10 Otherwise, increase λ and go to Step 3 (unless exceeded the maximum number of iterations, in which case quit).

every real-valued symmetric positive-definite matrix has a unique Cholesky decomposition.

A in the RCM is the Schur complement of HPP (HPP is called V* by Lourakis).

V* is a symmetric positive definite matrix (spdm). There exists proof the that Schur complement of a spdm is a symmetric positive definite matrix.

So A can be solved by Cholesky decomposition.

Bill **Triggs**, Philip Mclauchlan, Richard Hartley, Andrew Fitzgibbon.
Bundle Adjustment – A Modern Synthesis.
International Workshop on Vision Algorithms,
Sep **2000**, Corfu, Greece. pp.298–372,
10.1007/3-540-44480-7_21 . inria-00548290 **see Appendix B, and page 23…**

$$(\mathbf{U}^* - \mathbf{W}\boxed{\mathbf{V}^{*-1}}\mathbf{W}^T)\,\delta_\mathbf{a} = \epsilon_\mathbf{a} - \mathbf{W}\,\mathbf{V}^{*-1}\,\epsilon_\mathbf{b}$$

$$\underbrace{\underbrace{(H_{CC} - H_{PC}^\top \boxed{H_{PP}^{-1}} H_{PC})}_{A}}_{\overline{\mathsf{D}}} dC = \underbrace{b_C - H_{PC}^\top H_{PP}^{-1} b_P}_{B}$$

6.1 The Schur Complement and the Reduced Bundle System

Schur complement: Consider the following block triangular matrix factorization:

$$\mathsf{M} = \begin{pmatrix} \mathsf{A} & \mathsf{B} \\ \mathsf{C} & \mathsf{D} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \mathsf{C}\,\mathsf{A}^{-1} & 1 \end{pmatrix} \begin{pmatrix} \mathsf{A} & 0 \\ 0 & \overline{\mathsf{D}} \end{pmatrix} \begin{pmatrix} 1 & \mathsf{A}^{-1}\mathsf{B} \\ 0 & 1 \end{pmatrix}, \qquad \overline{\mathsf{D}} \equiv \mathsf{D} - \mathsf{C}\,\mathsf{A}^{-1}\mathsf{B} \tag{16}$$

$$\begin{pmatrix} \mathsf{A} & \mathsf{B} \\ \mathsf{C} & \mathsf{D} \end{pmatrix}^{-1} = \begin{pmatrix} 1 & -\mathsf{A}^{-1}\mathsf{B} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathsf{A}^{-1} & 0 \\ 0 & \overline{\mathsf{D}}^{-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\mathsf{C}\,\mathsf{A}^{-1} & 1 \end{pmatrix} = \begin{pmatrix} \mathsf{A}^{-1} + \mathsf{A}^{-1}\mathsf{B}\,\overline{\mathsf{D}}^{-1}\mathsf{C}\,\mathsf{A}^{-1} & -\mathsf{A}^{-1}\mathsf{B}\,\overline{\mathsf{D}}^{-1} \\ -\overline{\mathsf{D}}^{-1}\mathsf{C}\,\mathsf{A}^{-1} & \overline{\mathsf{D}}^{-1} \end{pmatrix} \tag{17}$$

Here A must be square and invertible, and for (17), the whole matrix must also be square and invertible. $\overline{\mathsf{D}}$ is called the **Schur complement** of A in M. If both A and D are invertible, complementing on D rather than A gives **$\overline{\mathsf{D}}$, the Schur complement, is HPP is V*.**

$$\begin{pmatrix} \mathsf{A} & \mathsf{B} \\ \mathsf{C} & \mathsf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \overline{\mathsf{A}}^{-1} & -\overline{\mathsf{A}}^{-1}\mathsf{B}\,\mathsf{D}^{-1} \\ -\mathsf{D}\,\mathsf{C}\,\overline{\mathsf{A}}^{-1} & \mathsf{D}^{-1} + \mathsf{D}^{-1}\mathsf{C}\,\overline{\mathsf{A}}^{-1}\mathsf{B}\,\mathsf{D}^{-1} \end{pmatrix}, \qquad \overline{\mathsf{A}} = \mathsf{A} - \mathsf{B}\boxed{\mathsf{D}^{-1}}\mathsf{C}$$

Equating upper left blocks gives the **Woodbury formula**:

$$(\mathsf{A} \pm \mathsf{B}\,\mathsf{D}^{-1}\mathsf{C})^{-1} = \mathsf{A}^{-1} \mp \mathsf{A}^{-1}\mathsf{B}\,(\mathsf{D} \pm \mathsf{C}\,\mathsf{A}^{-1}\mathsf{B})^{-1}\,\mathsf{C}\,\mathsf{A}^{-1} \tag{18}$$

This is the usual method of updating the inverse of a nonsingular matrix A after an update (especially a low rank one) $\mathsf{A} \to \mathsf{A} \pm \mathsf{B}\,\mathsf{D}^{-1}\mathsf{C}$. (See §8.1).

Bill **Triggs**, Philip Mclauchlan, Richard Hartley, Andrew Fitzgibbon.
Bundle Adjustment – A Modern Synthesis.
International Workshop on Vision Algorithms,
Sep **2000**, Corfu, Greece. pp.298–372,
10.1007/3-540-44480-7_21 . inria-00548290

see Appendix B, and page 23…

$$
\begin{array}{l}
\mathsf{L} \;=\; \textbf{profile\_cholesky\_decomp}(\mathsf{A}) \\
\quad \textbf{for } i = 1 \textbf{ to } n \textbf{ do} \\
\quad\quad \textbf{for } j = \mathrm{first}(i) \textbf{ to } i \textbf{ do} \\
\quad\quad\quad a \;=\; \mathsf{A}_{ij} - \displaystyle\sum_{k=\max(\mathrm{first}(i),\mathrm{first}(j))}^{j-1} \mathsf{L}_{ik}\,\mathsf{L}_{jk} \\
\quad\quad\quad \mathsf{L}_{ij} \;=\; (j<i) \;?\; a\,/\,\mathsf{L}_{jj} \;:\; \sqrt{a}
\end{array}
$$

$$
\begin{array}{l}
\mathsf{x} \;=\; \textbf{profile\_cholesky\_forward\_subs}(\mathsf{A},\mathsf{b}) \\
\quad \textbf{for } i = \mathrm{first}(\mathsf{b}) \textbf{ to } n \textbf{ do} \\
\quad\quad \mathsf{x}_i \;=\; \left(\mathsf{b}_i - \displaystyle\sum_{k=\max(\mathrm{first}(i),\mathrm{first}(\mathsf{b}))}^{i-1} \mathsf{L}_{ik}\,\mathsf{x}_k\right) / \,\mathsf{L}ii
\end{array}
$$

$$
\begin{array}{l}
\mathsf{y} \;=\; \textbf{profile\_cholesky\_back\_subs}(\mathsf{A},\mathsf{x}) \\
\quad \mathsf{y} \;=\; \mathsf{x} \\
\quad \textbf{for } i = \mathrm{last}(\mathsf{b}) \textbf{ to } 1 \textbf{ step } -1 \textbf{ do} \\
\quad\quad \textbf{for } k = \max(\mathrm{first}(i),\mathrm{first}(\mathsf{y})) \textbf{ to } i \textbf{ do} \\
\quad\quad\quad \mathsf{y}_k \;=\; \mathsf{y}_k - \mathsf{y}_i\,\mathsf{L}_{ik} \\
\quad\quad \mathsf{y}_i \;=\; \mathsf{y}_i\,/\,\mathsf{L}_{ii}
\end{array}
$$

Figure 10: A complete implementation of profile Cholesky decomposition.

cholesky:

_A_ = L ∗ D ∗ L^T
= L ∗ √D ∗ √D **∗ L^T**
let C = L ∗ √D
then **_A_** = **C ∗ C^T**
aside: L is invertible if none of its diagonal elements are 0.

for _A_=L ∗ L*
(_A_)^-1 = (L*) ∗ (L ∗ L*)^-1

but usually, for A ∗ x= b:
(1) A=L ∗ L*
(2) L ∗ y = b ==> y via forward subst
(3) L* ∗ x = y ==> x via backward subst