

[http://users.ics.forth.gr/~lourakis/sba/PRCV\\_colloq.pdf](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,  $N = 15945$ .  $\implies N^3 = 1e12$
  - Sparse LM is a better solution.
  - Example:
    - $M$  images
    - $N$  features
    - $\mathbf{x}_{i_j}$  = measured feature “ $i$ ” on image “ $j$ ”
    - $\mathbf{a}_j$  = vector of parameters for camera “ $j$ ”
    - $\mathbf{b}_i$  = vectors of parameters for point “ $i$ ”
    - $Q(\mathbf{a}_j, \mathbf{b}_i)$  = the predicted projection of point  $i$  on image  $j$ ,
    - $d(., .)$  the Euclidean distance between image points
    - $v_{ij} = 1$  iff point  $i$  is visible in image  $j$
    - minimize reprojection error over  $\mathbf{a}_j, \mathbf{b}_i$ :  $\min_{\mathbf{a}_j, \mathbf{b}_i} (\sum_{i=1}^N (\sum_{j=1}^M (v_{ij} * d(Q(\mathbf{a}_j, \mathbf{b}_i), \mathbf{x}_{i_j}))^2))$ 
      - $\implies$  total number of parameters is  $M * (\text{camera parameters}) + N * (\text{point parameters})$
    - let  $\mathbf{P}$  = parameter vector of camera then point parameters =  $[ \mathbf{P}_C \ \mathbf{P}_P ]$
    - let  $\mathbf{X}_{\text{hat}} = [(\mathbf{x}_{\text{hat}}_{1_1})^T \ \mathbf{x}_{\text{hat}}_{1_2})^T \dots \mathbf{x}_{\text{hat}}_{1_M})^T \ \mathbf{x}_{\text{hat}}_{2_1})^T \dots \mathbf{x}_{\text{hat}}_{N_M})^T ]$ 
      - where  $\mathbf{x}_{\text{hat}}_{i_j} = Q(\mathbf{a}_j, \mathbf{b}_i)$  is the projection onto camera plane
    - let error  $\mathbf{eps} = [(\mathbf{eps}_{1_1})^T \ \mathbf{eps}_{1_2})^T \dots \mathbf{eps}_{1_M})^T \ \mathbf{eps}_{2_1})^T \dots \mathbf{eps}_{N_M})^T ]$ 
      - where  $\mathbf{eps} = \mathbf{x}_{i_j} - \mathbf{x}_{\text{hat}}_{i_j}$

[http://users.ics.forth.gr/~lourakis/sba/PRCV\\_colloq.pdf](http://users.ics.forth.gr/~lourakis/sba/PRCV_colloq.pdf)

**Bundle Adjustment (BA) becomes**

$\min (\text{summation}_{i=1\_to\_N}(\text{summation}_{j=1\_to\_M} ( \text{eps}_{i\_j})^2 )))$  over P

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

$J = [A \mid B]$  where  $A = d(X\_hat) / d(a)$  and  $B = d(X\_hat) / d(b)$

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

The normal equations:

$$\begin{bmatrix} A^T A & A^T B \\ B^T A & B^T B \end{bmatrix} \begin{bmatrix} \delta(a) \\ \delta(b) \end{bmatrix} = \begin{bmatrix} A^T \text{eps} \\ B^T \text{eps} \end{bmatrix}$$

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

$$\partial \hat{x}_{ij} / \partial a_k = 0, \forall j \neq k \text{ and}$$

$$\partial \hat{x}_{ij} / \partial b_k = 0, \forall i \neq k$$

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

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

Let  $A_{ij} = \frac{\partial \hat{x}_{ij}}{\partial a_j}$  and  $B_{ij} = \frac{\partial \hat{x}_{ij}}{\partial b_i}$

• The Jacobian J in block form:

(1)

$$\frac{\partial \hat{X}}{\partial P} = \begin{matrix} & \begin{matrix} a_1^T & a_2^T & a_3^T & b_1^T & b_2^T & b_3^T & b_4^T \end{matrix} \\ \begin{matrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{31} \\ x_{32} \\ x_{33} \\ x_{41} \\ x_{42} \\ x_{43} \end{matrix} & \begin{pmatrix} A_{11} & 0 & 0 & B_{11} & 0 & 0 & 0 \\ 0 & A_{12} & 0 & B_{12} & 0 & 0 & 0 \\ 0 & 0 & A_{13} & B_{13} & 0 & 0 & 0 \\ A_{21} & 0 & 0 & 0 & B_{21} & 0 & 0 \\ 0 & A_{22} & 0 & 0 & B_{22} & 0 & 0 \\ 0 & 0 & A_{23} & 0 & B_{23} & 0 & 0 \\ A_{31} & 0 & 0 & 0 & 0 & B_{31} & 0 \\ 0 & A_{32} & 0 & 0 & 0 & B_{32} & 0 \\ 0 & 0 & A_{33} & 0 & 0 & B_{33} & 0 \\ A_{41} & 0 & 0 & 0 & 0 & 0 & B_{41} \\ 0 & A_{42} & 0 & 0 & 0 & 0 & B_{42} \\ 0 & 0 & A_{43} & 0 & 0 & 0 & B_{43} \end{pmatrix} \end{matrix}$$

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

• Approximate Hessian in block form:

(2)

$$J^T J = \begin{matrix} & \begin{matrix} a_1^T & a_2^T & a_3^T & b_1^T & b_2^T & b_3^T & b_4^T \end{matrix} \\ \begin{matrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{matrix} & \begin{pmatrix} U_1 & 0 & 0 & W_{11} & W_{21} & W_{31} & W_{41} \\ 0 & U_2 & 0 & W_{12} & W_{22} & W_{32} & W_{42} \\ 0 & 0 & U_3 & W_{13} & W_{23} & W_{33} & W_{43} \\ W_{11}^T & W_{12}^T & W_{13}^T & V_1 & 0 & 0 & 0 \\ W_{21}^T & W_{22}^T & W_{23}^T & 0 & V_2 & 0 & 0 \\ W_{31}^T & W_{32}^T & W_{33}^T & 0 & 0 & V_3 & 0 \\ W_{41}^T & W_{42}^T & W_{43}^T & 0 & 0 & 0 & V_4 \end{pmatrix} \end{matrix} \equiv$$

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

$U_j \equiv \sum_{i=1}^4 A_{ij}^T A_{ij}$ , for 1 image, summing over all features

$V_i \equiv \sum_{j=1}^3 B_{ij}^T B_{ij}$ , for 1 feature, summing over all images

$$W_{ij} = A_{ij}^T B_{ij}$$

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

## Bundle Adjustment Revisited

Yu Chen<sup>1</sup>, Yisong Chen<sup>1</sup>, Guoping Wang<sup>1</sup>

<sup>1</sup> Peking University, Department of Computer Science and Technology,  
Graphics and Interactive Lab, Beijing, China

For convenience, we use (18) to show how to solve the bundle adjustment problem. set  $\hat{u}_{ij} = \pi(C_j, X_i)$ , and we order the parameter  $x$  into camera block  $c$  and structure block  $p$ :

$$x = [c, p] \quad (20)$$

it's easily to realize that:

$$J_{ij} = \frac{\partial r_{ij}}{\partial x_k} = \frac{\partial \hat{u}_{ij}}{\partial x_k}, \frac{\partial \hat{u}_{ij}}{\partial c_k} = 0, \forall j \neq k, \frac{\partial \hat{u}_{ij}}{\partial p_k} = 0, \forall i \neq k \quad (21)$$

Consider now, that we have  $m = 3$  cameras and  $n = 4$  3D points. Set  $A_{ij} = \frac{\partial \hat{u}_{ij}}{\partial c_j}$ ,  $B_{ij} = \frac{\partial \hat{u}_{ij}}{\partial p_i}$ , we can obtain the Jacobi:

**k dimension is  
row number  
w.r.t. the block  
of i where i is  
the feature  
number**

$$J = \frac{\partial \hat{u}}{\partial x} = \begin{matrix} k = 1 \left\{ \begin{array}{ccccccc} A_{11} & 0 & 0 & B_{11} & 0 & 0 & 0 \\ 0 & A_{12} & 0 & B_{12} & 0 & 0 & 0 \\ 0 & 0 & A_{13} & B_{13} & 0 & 0 & 0 \\ A_{21} & 0 & 0 & 0 & B_{21} & 0 & 0 \\ 0 & A_{22} & 0 & 0 & B_{22} & 0 & 0 \\ 0 & 0 & A_{23} & 0 & B_{23} & 0 & 0 \\ A_{31} & 0 & 0 & 0 & 0 & B_{31} & 0 \\ 0 & A_{32} & 0 & 0 & 0 & B_{32} & 0 \\ 0 & 0 & A_{33} & 0 & 0 & B_{33} & 0 \end{array} \right. \\ k = 4 \left\{ \begin{array}{ccccccc} A_{41} & 0 & 0 & 0 & 0 & 0 & B_{41} \\ 0 & A_{42} & 0 & 0 & 0 & 0 & B_{42} \\ 0 & 0 & A_{43} & 0 & 0 & 0 & B_{43} \end{array} \right. \end{matrix} \quad (22)$$

[http://users.ics.forth.gr/~lourakis/sba/PRCV\\_colloq.pdf](http://users.ics.forth.gr/~lourakis/sba/PRCV_colloq.pdf)

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

NOTE:  $\text{eps}_a = A^T * \text{eps}$ ,  $\text{eps}_b = B^T * \text{eps}$

- The augmented normal equations  $(J^T J + \mu I) \delta_p = J^T \epsilon$  take the form

where  $\mu > 0$

$$(3) \quad \begin{pmatrix} U^* & W \\ W^T & V^* \end{pmatrix} \begin{pmatrix} \delta_a \\ \delta_b \end{pmatrix} = \begin{pmatrix} \epsilon_a \\ \epsilon_b \end{pmatrix}$$

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

$$(4) \quad (U^* - W V^{*-1} W^T) \delta_a = \epsilon_a - W V^{*-1} \epsilon_b$$

Schur complement: multiply 1st matrix on res by

$$\begin{pmatrix} I & 0 \\ (((-V^*)^{-1})W^T & I \end{pmatrix}$$

resulting in:

$$\begin{pmatrix} (U^*) - W(((V^*)^{-1})W^T & W \\ 0 & (V^*) \end{pmatrix} * \begin{pmatrix} \delta_a \\ \delta_b \end{pmatrix} = \begin{pmatrix} \epsilon_a \\ \epsilon_b \end{pmatrix}$$

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

separate  $\delta_b$ :  $0 * \delta_a + (V^*) * \delta_b = \text{eps}_b \implies \delta_b = \text{eps}_b * ((V^*)^{-1})$

solving for  $\delta_a$  (typically M images  $\ll$  N features) after substitute  $\delta_b$ :

$$((U^*) - W(((V^*)^{-1})W^T) * \delta_a + W * \delta_b = \text{eps}_a$$

$$((U^*) - W(((V^*)^{-1})W^T) * \delta_a = \text{eps}_a - W((V^*)^{-1})\text{eps}_b$$

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

$$V^{*-1} = \begin{pmatrix} V_1^{*-1} & 0 & \dots \\ 0 & V_2^{*-1} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

[http://users.ics.forth.gr/~lourakis/sba/PRCV\\_colloq.pdf](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, preconditioners 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

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

$$J_f = \begin{bmatrix} J_P & J_C \end{bmatrix}, \quad (15)$$

$$H = \begin{bmatrix} J_P^\top J_P & J_P^\top J_C \\ J_C^\top J_P & J_C^\top J_C \end{bmatrix}, \quad (16)$$

$$\begin{bmatrix} H_{PP} & H_{PC} \\ H_{PC}^\top & H_{CC} \end{bmatrix} \begin{bmatrix} dP \\ dC \end{bmatrix} = \begin{bmatrix} b_P \\ b_C \end{bmatrix}, \quad (17)$$

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 simplify 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

$$\begin{bmatrix} I & 0 \\ -H_{PC}^\top & I \end{bmatrix} \quad (20)$$

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

$$\underbrace{(H_{CC} - H_{PC}^\top H_{PP}^{-1} H_{PC})}_A dC = \underbrace{b_C - H_{PC}^\top H_{PP}^{-1} b_P}_B \quad (21)$$

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

**We use straightforward Cholesky factorization.**

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

M images = 3, j  
N features = 4, i

sparse blocks along diagonal: J\_P\_i is [3 X 3]

J\_P is [2\*n\*m X 3\*m]

J\_P\_i is [2\*n\*m X 3]

J\_C is [2n\*m X 9\*n]

J\_C\_i is [2n\*m X 9]

J is [2nm X (3m + 9n)]

J\_C\_i is [9 X 9]

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  $A_{ij} = \frac{\partial \hat{x}_{ij}}{\partial a_j}$  and  $B_{ij} = \frac{\partial \hat{x}_{ij}}{\partial b_i}$   
 $\begin{matrix} \boxed{2 \times 9} & \mathbf{C} & \boxed{2 \times 3} & \mathbf{P} \end{matrix}$

$$\mathbf{J} = \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} = \left[ \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{a}} \quad \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{b}} \right] = \begin{bmatrix} \mathbf{J}_C & \mathbf{J}_P \end{bmatrix}$$

• The Jacobian J in block form:

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

(1)

Engels

Let  $A_{ij} = \frac{\partial \hat{x}_{ij}}{\partial a_j}$  and  $B_{ij} = \frac{\partial \hat{x}_{ij}}{\partial b_i}$   
 $\begin{matrix} \boxed{2 \times 9} & \mathbf{C} & \boxed{2 \times 3} & \mathbf{P} \end{matrix}$

$$\mathbf{J}_f = \begin{bmatrix} \mathbf{J}_P & \mathbf{J}_C \end{bmatrix},$$

• The Jacobian J in block form:

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

(1)



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

M images = 3, j

N features = 4, i

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

J is [2nm X (3n + 9m)]

J<sup>T</sup> \* J is [(3n+9m) X (3n+9m)]

Lourakis

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

$$\mathbf{J} = \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} = \begin{bmatrix} \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{a}} & \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{b}} \end{bmatrix} = \begin{bmatrix} J_C & J_P \end{bmatrix}$$

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

J<sup>T</sup> \* J is [(3\*n + 9\*m)] X (3\*n + 9\*m)]

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

where

$$\mathbf{U}_j \equiv \sum_{i=1}^4 \mathbf{A}_{ij}^T \mathbf{A}_{ij},$$

$$\mathbf{V}_i \equiv \sum_{j=1}^3 \mathbf{B}_{ij}^T \mathbf{B}_{ij},$$

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

for 1 image, sum over features

9X9

for 1 feature, sum over images

3X3

9X3

$$\equiv \sum_{j=1}^3 \left( \frac{\partial \hat{x}_{ij}}{\partial \mathbf{b}_i} \right)^T \frac{\partial \hat{x}_{ij}}{\partial \mathbf{b}_i}$$

Engels

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

$$\mathbf{J} = \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} = \begin{bmatrix} \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{b}} & \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{a}} \end{bmatrix} = \begin{bmatrix} J_P & J_C \end{bmatrix},$$

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

J<sup>T</sup> \* J is [(3\*n + 9\*m)] X (3\*n + 9\*m)]

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

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

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

$\delta_b$  is  $dP$  is  $[3*n\_features \times 1]$   
 $\epsilon_b$  is  $b_P$  is  $[3*n\_features \times 1]$   
 $\delta_a$  is  $dC$  is  $[9*m\_images \times 1]$   
 $\epsilon_a$  is  $b_C$  is  $[9*m\_images \times 1]$

$\mathbf{V}^*$  is  $[3*n \times 3*n]$

$\mathbf{W}$  is  $[9*m \times 3*n]$

$\mathbf{U}^*$  is  $[9*m \times 9*m]$

## Lourakis

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

$$(3) \quad \begin{pmatrix} \mathbf{U}^* & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V}^* \end{pmatrix} \begin{pmatrix} \delta_a \\ \delta_b \end{pmatrix} = \begin{pmatrix} \epsilon_a \\ \epsilon_b \end{pmatrix}$$

$$\begin{bmatrix} \mathbf{U} - \mathbf{W} \mathbf{V}^{*-1} \mathbf{W}^T & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_b \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{W} \mathbf{V}^{*-1} \end{bmatrix} \begin{bmatrix} \epsilon_a \\ \epsilon_b \end{bmatrix}$$

(solve delta a first because typically  $m\_images \ll n\_features$ )

determine  $\delta_a$  with Cholesky (or other method)

$$(\mathbf{U}^* - \mathbf{W} \mathbf{V}^{*-1} \mathbf{W}^T) \delta_a = \epsilon_a - \mathbf{W} \mathbf{V}^{*-1} \epsilon_b$$

$\delta_b$  can be computed by back substitution into

$$\mathbf{V}^* \delta_b = \epsilon_b - \mathbf{W}^T \delta_a$$

$$\delta_b = \mathbf{V}^{*-1} \epsilon_b - \mathbf{V}^{*-1} \mathbf{W}^T \delta_a$$

## Engels

$$\begin{pmatrix} \mathbf{V}^* & \mathbf{W}^T \\ \mathbf{W} & \mathbf{U}^* \end{pmatrix} \begin{pmatrix} \delta_b \\ \delta_a \end{pmatrix} = \begin{pmatrix} \epsilon_b \\ \epsilon_a \end{pmatrix}$$

$$\begin{bmatrix} H_{PP} & H_{PC} \\ H_{PC}^T & H_{CC} \end{bmatrix} \begin{bmatrix} dP \\ dC \end{bmatrix} = \begin{bmatrix} b_P \\ b_C \end{bmatrix}, \quad \begin{aligned} H_{PP} &= J_P^T J_P \\ H_{PC} &= J_P^T J_C, \\ H_{CC} &= J_C^T J_C, \\ b_P &= -J_P^T f, \\ b_C &= -J_C^T f \end{aligned}$$

$$\underbrace{(\mathbf{U}^* - \mathbf{W} \mathbf{V}^{*-1} \mathbf{W}^T)}_A \delta_a = \epsilon_a - \underbrace{\mathbf{W} \mathbf{V}^{*-1} \epsilon_b}_B$$

$$\underbrace{(H_{CC} - H_{PC}^T H_{PP}^{-1} H_{PC})}_A dC = \underbrace{b_C - H_{PC}^T H_{PP}^{-1} b_P}_B$$

$\delta_b$  can be computed by back substitution into

$$\mathbf{V}^* \delta_b = \epsilon_b - \mathbf{W}^T \delta_a$$

$$\begin{aligned} H_{PP} dP &= b_P - H_{PC} dC \\ dP &= H_{PP}^{-1} b_P - H_{PP}^{-1} H_{PC} dC. \end{aligned}$$

## Qu

(3.61)

$$\begin{pmatrix} \mathbf{B} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{C} \end{pmatrix} \begin{pmatrix} p_c \\ p_p \end{pmatrix} = - \begin{pmatrix} g_c \\ g_p \end{pmatrix}$$

see eqn (3.70) too

$$-g^k = -\mathbf{J}^{kT} \mathbf{F}^k$$

where k is a feature block summed over all images?

$$(\mathbf{B} - \mathbf{E} \mathbf{C}^{-1} \mathbf{E}^T) p_c = -g_c + \mathbf{E} \mathbf{C}^{-1} g_p$$

$$p_p = \mathbf{C}^{-1} (-g_p - \mathbf{E}^T p_c)$$

**Engels, et al 2006**

**M images = 3, j**  
**N features = 4, i**

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

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

$2 \times 9$       **C**       $2 \times 3$       **P**

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

$$\begin{aligned} \mathbf{U}_j &\equiv \sum_{i=1}^4 \mathbf{A}_{ij}^T \mathbf{A}_{ij}, & [9 \times 9], \text{ for 1 image, sum over features} \\ \mathbf{V}_i &\equiv \sum_{j=1}^3 \mathbf{B}_{ij}^T \mathbf{B}_{ij}, & [3 \times 3], \text{ for 1 feature, sum over images} \\ \mathbf{W}_{ij} &= \mathbf{A}_{ij}^T \mathbf{B}_{ij} & [9 \times 3] \end{aligned}$$

$\mathbf{V}^*$  is  $[3^*n \times 3^*n]$

$\mathbf{W}$  is  $[9^*m \times 3^*n]$

$\mathbf{U}^*$  is  $[9^*m \times 9^*m]$

$$\begin{pmatrix} \mathbf{V}^* & \mathbf{W}^T \\ \mathbf{W} & \mathbf{U}^* \end{pmatrix} \begin{pmatrix} \delta_b \\ \delta_a \end{pmatrix} = \begin{pmatrix} \epsilon_b \\ \epsilon_a \end{pmatrix}$$

$\mathbf{JP}^{\wedge T} = [3n \times 2mn]$

$\mathbf{B}_{11}^{\wedge T}$	$\mathbf{B}_{12}^{\wedge T}$	$\mathbf{B}_{13}^{\wedge T}$	0	0	0	0	0	0	0	0	0	0
0	0	0	$\mathbf{B}_{21}^{\wedge T}$	$\mathbf{B}_{22}^{\wedge T}$	$\mathbf{B}_{23}^{\wedge T}$	0	0	0	0	0	0	0
0	0	0	0	0	0	$\mathbf{B}_{31}^{\wedge T}$	$\mathbf{B}_{32}^{\wedge T}$	$\mathbf{B}_{33}^{\wedge T}$	0	0	0	0
0	0	0	0	0	0	0	0	0	$\mathbf{B}_{41}^{\wedge T}$	$\mathbf{B}_{42}^{\wedge T}$	$\mathbf{B}_{43}^{\wedge T}$	0

$\mathbf{bP} = -\mathbf{JP}^{\wedge T} * \mathbf{F} = [3n \times 1]$

$$\begin{aligned} &-\mathbf{B}_{11}^{\wedge T} * \mathbf{F}_{11} - \mathbf{B}_{12}^{\wedge T} * \mathbf{F}_{12} - \mathbf{B}_{13}^{\wedge T} * \mathbf{F}_{13} \\ &-\mathbf{B}_{21}^{\wedge T} * \mathbf{F}_{21} - \mathbf{B}_{22}^{\wedge T} * \mathbf{F}_{22} - \mathbf{B}_{23}^{\wedge T} * \mathbf{F}_{23} \\ &-\mathbf{B}_{31}^{\wedge T} * \mathbf{F}_{31} - \mathbf{B}_{32}^{\wedge T} * \mathbf{F}_{32} - \mathbf{B}_{33}^{\wedge T} * \mathbf{F}_{33} \\ &-\mathbf{B}_{41}^{\wedge T} * \mathbf{F}_{41} - \mathbf{B}_{42}^{\wedge T} * \mathbf{F}_{42} - \mathbf{B}_{43}^{\wedge T} * \mathbf{F}_{43} \end{aligned}$$

where each row is  $[3 \times 2] * [2 \times 1] = [3 \times 1]$

F11  
F12  
F13  
F21  
F22  
F23  
F31  
F32  
F33  
F41  
F42  
F43

$\mathbf{JC}^{\wedge T} = [9m \times 2mn]$

$\mathbf{A}_{11}^{\wedge T}$	0	0	$\mathbf{A}_{21}^{\wedge T}$	0	0	$\mathbf{A}_{31}^{\wedge T}$	0	0	$\mathbf{A}_{41}^{\wedge T}$	0	0
0	$\mathbf{A}_{12}^{\wedge T}$	0	0	$\mathbf{A}_{22}^{\wedge T}$	0	0	$\mathbf{A}_{32}^{\wedge T}$	0	0	$\mathbf{A}_{42}^{\wedge T}$	0
0	0	$\mathbf{A}_{13}^{\wedge T}$	0	0	$\mathbf{A}_{23}^{\wedge T}$	0	0	$\mathbf{A}_{33}^{\wedge T}$	0	0	$\mathbf{A}_{43}^{\wedge T}$

$\mathbf{bC} = -\mathbf{JC}^{\wedge T} * \mathbf{F} = [9m \times 1]$

$$\begin{aligned} &-\mathbf{A}_{11}^{\wedge T} * \mathbf{F}_{11} - \mathbf{A}_{21}^{\wedge T} * \mathbf{F}_{21} - \mathbf{A}_{31}^{\wedge T} * \mathbf{F}_{31} - \mathbf{A}_{41}^{\wedge T} * \mathbf{F}_{41} \\ &-\mathbf{A}_{12}^{\wedge T} * \mathbf{F}_{12} - \mathbf{A}_{22}^{\wedge T} * \mathbf{F}_{22} - \mathbf{A}_{32}^{\wedge T} * \mathbf{F}_{32} - \mathbf{A}_{42}^{\wedge T} * \mathbf{F}_{42} \\ &-\mathbf{A}_{13}^{\wedge T} * \mathbf{F}_{13} - \mathbf{A}_{23}^{\wedge T} * \mathbf{F}_{23} - \mathbf{A}_{33}^{\wedge T} * \mathbf{F}_{33} - \mathbf{A}_{43}^{\wedge T} * \mathbf{F}_{43} \end{aligned}$$

where each row is  $[9 \times 2] * [2 \times 1] = [9 \times 1]$

[2mn X 1]

F11  
F12  
F13  
F21  
F22  
F23  
F31  
F32  
F33  
F41  
F42  
F43

$\mathbf{V}^*$        $\delta_b$

n blocks

$$\begin{Bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ 0 \\ 0 \end{Bmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}_3 \\ \mathbf{V}_4 \end{pmatrix}$$

$[3 \times 3]$        $[3 \times 3]$        $[3 \times 3]$        $[3 \times 3]$

$[x_0, y_0, z_0]^T$   
 $[x_1, y_1, z_1]^T$   
 $[x_2, y_2, z_2]^T$   
 $[x_3, y_3, z_3]^T$

$[3^*n \times 1]$

n blocks

$\mathbf{U}^*$        $\delta_a$

m blocks

$$\begin{Bmatrix} \mathbf{U}_1 \\ 0 \\ 0 \end{Bmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U}_2 \\ \mathbf{U}_3 \end{pmatrix}$$

$[9 \times 9]$        $[9 \times 9]$        $[9 \times 9]$

$[9 \times 1]^T$   
 $[9 \times 1]^T$   
 $[9 \times 1]^T$

$[9^*m \times 1]$

m blocks

Engels, et al 2006

M images = 3, j  
N features = 4, i

$$\begin{aligned} H_{PP} &= J_P^T J_P && \equiv V^* \\ H_{PC} &= J_P^T J_C, && \equiv W^T \\ H_{CC} &= J_C^T J_C, && \equiv U^* \\ b_P &= -J_P^T f, && \equiv \epsilon_b \\ b_C &= -J_C^T f, && \equiv \epsilon_a \end{aligned}$$

$$\underbrace{(U^* - W V^{*-1} W^T)}_A \delta_a = \underbrace{\epsilon_a - W V^{*-1} \epsilon_b}_B$$

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

$W^* V^{-1} =$  [9mX3n] the V's are inverses  
 $W_{11}^* V_1 \quad W_{21}^* V_2 \quad W_{31}^* V_3 \quad W_{41}^* V_4$   
 $W_{12}^* V_1 \quad W_{22}^* V_2 \quad W_{32}^* V_3 \quad W_{42}^* V_4$   
 $W_{13}^* V_1 \quad W_{23}^* V_2 \quad W_{33}^* V_3 \quad W_{43}^* V_4$   
 where each block is [9X3]\*[3X3] = [9X3]

$U - W^* V^{-1} W^T =$  [9mX9m] the V's are inverses

$$\begin{bmatrix} U_1 & 0 & 0 & | & - \\ 0 & U_2 & 0 & | & - \\ 0 & 0 & U_3 & | & - \end{bmatrix} \begin{bmatrix} W_{11}^* V_1^* W_{11}^T + W_{21}^* V_2^* W_{21}^T + W_{31}^* V_3^* W_{31}^T + W_{41}^* V_4^* W_{41}^T \\ W_{12}^* V_1^* W_{12}^T + W_{22}^* V_2^* W_{22}^T + W_{32}^* V_3^* W_{32}^T + W_{42}^* V_4^* W_{42}^T \\ W_{13}^* V_1^* W_{13}^T + W_{23}^* V_2^* W_{23}^T + W_{33}^* V_3^* W_{33}^T + W_{43}^* V_4^* W_{43}^T \end{bmatrix}$$

A is [9\*mImages X 9\*mImages]

$$bC = JC^T * F = [9m \times 1]$$

$$\begin{aligned} & -A_{11}^T F_{11} - A_{21}^T F_{21} - A_{31}^T F_{31} - A_{41}^T F_{41} \\ & -A_{12}^T F_{12} - A_{22}^T F_{22} - A_{32}^T F_{32} - A_{42}^T F_{42} \\ & -A_{13}^T F_{13} - A_{23}^T F_{23} - A_{33}^T F_{33} - A_{43}^T F_{43} \end{aligned}$$

where each row is [9X2][2X1] = [9X1]

$W^* V^{-1} =$  [9mX3n] the V's are inverses

$$\begin{aligned} & W_{11}^* V_1 \quad W_{21}^* V_2 \quad W_{31}^* V_3 \quad W_{41}^* V_4 \\ & W_{12}^* V_1 \quad W_{22}^* V_2 \quad W_{32}^* V_3 \quad W_{42}^* V_4 \\ & W_{13}^* V_1 \quad W_{23}^* V_2 \quad W_{33}^* V_3 \quad W_{43}^* V_4 \end{aligned}$$

where each block is [9X3]\*[3X3] = [9X3]

$$bP = JP^T * F [3n \times 1]$$

$$\begin{aligned} & -B_{11}^T F_{11} - B_{12}^T F_{12} - B_{13}^T F_{13} \\ & -B_{21}^T F_{21} - B_{22}^T F_{22} - B_{23}^T F_{23} \\ & -B_{31}^T F_{31} - B_{32}^T F_{32} - B_{33}^T F_{33} \\ & -B_{41}^T F_{41} - B_{42}^T F_{42} - B_{43}^T F_{43} \end{aligned}$$

where each row is [3X2]\*[2X1] = [3X1]

$$A_{ij} = \frac{\partial \hat{x}_{ij}}{\partial a_j} \text{ and } B_{ij} = \frac{\partial \hat{x}_{ij}}{\partial b_i}$$

2X9 C 2X3 P

Note that  $V^{*-1} = \begin{pmatrix} V_1^{*-1} & 0 & \dots \\ 0 & V_2^{*-1} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$

$$J^T J = \begin{matrix} & b_1^T & b_2^T & b_3^T & b_4^T & a_1^T & a_2^T & a_3^T \\ \begin{matrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ a_1 \\ a_2 \\ a_3 \end{matrix} & \begin{pmatrix} V_1 & 0 & 0 & 0 & W_{11}^T & W_{12}^T & W_{13}^T \\ 0 & V_2 & 0 & 0 & W_{21}^T & W_{22}^T & W_{23}^T \\ 0 & 0 & V_3 & 0 & W_{31}^T & W_{32}^T & W_{33}^T \\ 0 & 0 & 0 & V_4 & W_{41}^T & W_{42}^T & W_{43}^T \\ W_{11} & W_{21} & W_{31} & W_{41} & U_1 & 0 & 0 \\ W_{12} & W_{22} & W_{32} & W_{42} & 0 & U_2 & 0 \\ W_{13} & W_{23} & W_{33} & W_{43} & 0 & 0 & U_3 \end{pmatrix} \end{matrix}$$

$$U_j \equiv \sum_{i=1}^4 A_{ij}^T A_{ij}, \quad [9X9], \text{ for 1 image, sum over features}$$

$$V_i \equiv \sum_{j=1}^3 B_{ij}^T B_{ij}, \quad [3X3], \text{ for 1 feature, sum over images}$$

$$W_{ij} = A_{ij}^T B_{ij} \quad [9X3]$$

$V^*$  is [3\*n X 3\*n]

$W$  is [9\*m X 3\*n]

$U^*$  is [9\*m X 9\*m]

$$\begin{pmatrix} V^* & W^T \\ W & U^* \end{pmatrix} \begin{pmatrix} \delta_b \\ \delta_a \end{pmatrix} = \begin{pmatrix} \epsilon_b \\ \epsilon_a \end{pmatrix}$$

$$\begin{matrix} & V^* & & \delta_b \\ \left\{ \begin{matrix} V_1 \\ V_2 \\ 0 \\ 0 \end{matrix} \right\}_{[3X3]} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & V_2 & 0 & 0 \\ 0 & 0 & V_3 & 0 \\ 0 & 0 & 0 & V_4 \end{pmatrix}_{[3X3]} & \begin{pmatrix} [x0, y0, z0]^T \\ [x1, y1, z1]^T \\ [x2, y2, z2]^T \\ [x3, y3, z3]^T \end{pmatrix}_{[3 \times n \times 1]} \end{matrix}$$

n blocks

$$\begin{matrix} & U^* & & \delta_a \\ \left\{ \begin{matrix} U_1 \\ 0 \\ 0 \end{matrix} \right\}_{[9X9]} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{pmatrix}_{[9X9]} & \begin{pmatrix} [9X1]^T \\ [9X1]^T \\ [9X1]^T \end{pmatrix}_{[9 \times m \times 1]} \end{matrix}$$

m blocks

Engels, et al 2006

M images = 3, j  
N features = 4, i

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

$$\underbrace{(\mathbf{U}^* - \mathbf{W} \mathbf{V}^{*-1} \mathbf{W}^T)}_A \delta_a = \epsilon_a - \underbrace{\mathbf{W} \mathbf{V}^{*-1} \epsilon_b}_B$$

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

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

2X9      C      2X3      P

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

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

$$\mathbf{U}_j \equiv \sum_{i=1}^4 \mathbf{A}_{ij}^T \mathbf{A}_{ij}, \quad \text{[9X9], for 1 image, sum over features}$$

$$\mathbf{V}_i \equiv \sum_{j=1}^3 \mathbf{B}_{ij}^T \mathbf{B}_{ij}, \quad \text{[3X3], for 1 feature, sum over images}$$

$$\mathbf{W}_{ij} = \mathbf{A}_{ij}^T \mathbf{B}_{ij} \quad \text{[9X3]}$$

$\mathbf{V}^*$  is [3\*n X 3\*n]

$\mathbf{W}$  is [9\*m X 3\*n]

$\mathbf{U}^*$  is [9\*m X 9\*m]

$$\begin{pmatrix} \mathbf{V}^* & \mathbf{W}^T \\ \mathbf{W} & \mathbf{U}^* \end{pmatrix} \begin{pmatrix} \delta_b \\ \delta_a \end{pmatrix} = \begin{pmatrix} \epsilon_b \\ \epsilon_a \end{pmatrix}$$

$$\mathbf{tP} = \mathbf{hPPI}^{-1} * \mathbf{bPI} \text{ is } \mathbf{V}^{-1} * \mathbf{bPI} // [\mathbf{3X3}][\mathbf{3X1}] = [\mathbf{3X1}]$$

$$\mathbf{V}^{-1} = [\mathbf{3nX3n}]$$

$$\mathbf{V}^{-1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \mathbf{V}_2^{-1} & 0 & 0 \\ 0 & 0 & \mathbf{V}_3^{-1} & 0 \\ 0 & 0 & 0 & \mathbf{V}_4^{-1} \end{pmatrix}$$

where each block is [3X3]

$$\mathbf{bP} = \mathbf{JP}^T * \mathbf{F} \quad [\mathbf{3n} \times \mathbf{1}]$$

$$\begin{aligned} & -(\mathbf{V}_1^{-1}) * (\mathbf{B}_{11}^T * \mathbf{F}_{11} - \mathbf{B}_{12}^T * \mathbf{F}_{12} - \mathbf{B}_{13}^T * \mathbf{F}_{13}) \\ & -(\mathbf{V}_2^{-1}) * (\mathbf{B}_{21}^T * \mathbf{F}_{21} - \mathbf{B}_{22}^T * \mathbf{F}_{22} - \mathbf{B}_{23}^T * \mathbf{F}_{23}) \\ & -(\mathbf{V}_3^{-1}) * (\mathbf{B}_{31}^T * \mathbf{F}_{31} - \mathbf{B}_{32}^T * \mathbf{F}_{32} - \mathbf{B}_{33}^T * \mathbf{F}_{33}) \\ & -(\mathbf{V}_4^{-1}) * (\mathbf{B}_{41}^T * \mathbf{F}_{41} - \mathbf{B}_{42}^T * \mathbf{F}_{42} - \mathbf{B}_{43}^T * \mathbf{F}_{43}) \end{aligned}$$

=

$$\mathbf{H}_{PP}^{-1} \mathbf{bP} \quad [\mathbf{3n} \times \mathbf{1}]$$

$$\begin{aligned} & -(\mathbf{V}_1^{-1}) * (\mathbf{B}_{11}^T * \mathbf{F}_{11} - \mathbf{B}_{12}^T * \mathbf{F}_{12} - \mathbf{B}_{13}^T * \mathbf{F}_{13}) \\ & -(\mathbf{V}_2^{-1}) * (\mathbf{B}_{21}^T * \mathbf{F}_{21} - \mathbf{B}_{22}^T * \mathbf{F}_{22} - \mathbf{B}_{23}^T * \mathbf{F}_{23}) \\ & -(\mathbf{V}_3^{-1}) * (\mathbf{B}_{31}^T * \mathbf{F}_{31} - \mathbf{B}_{32}^T * \mathbf{F}_{32} - \mathbf{B}_{33}^T * \mathbf{F}_{33}) \\ & -(\mathbf{V}_4^{-1}) * (\mathbf{B}_{41}^T * \mathbf{F}_{41} - \mathbf{B}_{42}^T * \mathbf{F}_{42} - \mathbf{B}_{43}^T * \mathbf{F}_{43}) \end{aligned}$$

where each row is [3X1]

where each row is [3X2]\*[2X1]=[3X1]

$$\mathbf{tPC} = \mathbf{hPCIJT} * \mathbf{invHPPI} \text{ is } \mathbf{W} * \mathbf{V}^{-1} // [\mathbf{9X3}][\mathbf{3X3}] = [\mathbf{9X3}]$$

$$\mathbf{H}_{PC}^T \mathbf{H}_{PP}^{-1} \quad [\mathbf{9mX3n}]$$

$$\mathbf{W} * \mathbf{V}^{-1} = [\mathbf{9mX3n}]$$

$$\begin{aligned} & \mathbf{W}_{11} * \mathbf{V}_1^{-1} \quad \mathbf{W}_{21} * \mathbf{V}_2^{-1} \quad \mathbf{W}_{31} * \mathbf{V}_3^{-1} \quad \mathbf{W}_{41} * \mathbf{V}_4^{-1} \\ & \mathbf{W}_{12} * \mathbf{V}_1^{-1} \quad \mathbf{W}_{22} * \mathbf{V}_2^{-1} \quad \mathbf{W}_{32} * \mathbf{V}_3^{-1} \quad \mathbf{W}_{42} * \mathbf{V}_4^{-1} \\ & \mathbf{W}_{13} * \mathbf{V}_1^{-1} \quad \mathbf{W}_{23} * \mathbf{V}_2^{-1} \quad \mathbf{W}_{33} * \mathbf{V}_3^{-1} \quad \mathbf{W}_{43} * \mathbf{V}_4^{-1} \end{aligned}$$

where each block is [9X3]\*[3X3] = [9X3]

$$\begin{matrix} \mathbf{V}^* & \delta_b \\ \left\{ \begin{matrix} \mathbf{V}_1 & 0 & 0 & 0 \\ 0 & \mathbf{V}_2 & 0 & 0 \\ 0 & 0 & \mathbf{V}_3 & 0 \\ 0 & 0 & 0 & \mathbf{V}_4 \end{matrix} \right\}_{\substack{\text{n blocks} \\ \text{[3X3]}}} & \begin{matrix} [\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0]^T \\ [\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1]^T \\ [\mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2]^T \\ [\mathbf{x}_3, \mathbf{y}_3, \mathbf{z}_3]^T \end{matrix} \end{matrix}$$

[3\*nX1]

$$\begin{matrix} \mathbf{U}^* & \delta_a \\ \left\{ \begin{matrix} \mathbf{U}_1 & 0 & 0 \\ 0 & \mathbf{U}_2 & 0 \\ 0 & 0 & \mathbf{U}_3 \end{matrix} \right\}_{\substack{\text{m blocks} \\ \text{[9X9]}}} & \begin{matrix} [\mathbf{9X1}]^T \\ [\mathbf{9X1}]^T \\ [\mathbf{9X1}]^T \end{matrix} \end{matrix}$$

[9\*mX1]

Engels, et al 2006

M images = 3, j  
N features = 4, i

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

$$\underbrace{(\mathbf{U}^* - \mathbf{W} \mathbf{V}^{*-1} \mathbf{W}^T)}_A \delta_a = \epsilon_a - \mathbf{W} \mathbf{V}^{*-1} \epsilon_b$$

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

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

$\mathbf{C}$   $\mathbf{P}$

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

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

$$\mathbf{U}_j \equiv \sum_{i=1}^4 \mathbf{A}_{ij}^T \mathbf{A}_{ij}, \quad \text{[9X9], for 1 image, sum over features}$$

$$\mathbf{V}_i \equiv \sum_{j=1}^3 \mathbf{B}_{ij}^T \mathbf{B}_{ij}, \quad \text{[3X3], for 1 feature, sum over images}$$

$$\mathbf{W}_{ij} = \mathbf{A}_{ij}^T \mathbf{B}_{ij} \quad \text{[9X3]}$$

$\mathbf{V}^*$  is [3\*n X 3\*n]

$\mathbf{W}$  is [9\*m X 3\*n]

$\mathbf{U}^*$  is [9\*m X 9\*m]

$$\begin{pmatrix} \mathbf{V}^* & \mathbf{W}^T \\ \mathbf{W} & \mathbf{U}^* \end{pmatrix} \begin{pmatrix} \delta_b \\ \delta_a \end{pmatrix} = \begin{pmatrix} \epsilon_b \\ \epsilon_a \end{pmatrix}$$

- 1 Initialize  $\lambda$ .
- 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)

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

(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 x 3 matrix) and the camera parameters (in our case a 2 x 6 matrix), respectively.

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

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

Subtract  $J_p^T f$  from  $b_p$ .

$$\mathbf{B}^T \quad \epsilon_b$$

If camera c is free

- {  
Add  $J_c^T J_c$  (optionally with an augmented diagonal) to upper triangular part of block (c, c) of left hand side matrix A (in our case a 6 x 6 matrix).  
Compute block (p, c) of  $H_{PC}$  as  $H_{pc} = J_p^T J_c$  (in our case a 3 x 6 matrix) and store it until track is done.  
Subtract  $J_c^T f$  from part c of right hand side (vector B) (related to  $b_C$ ).

} // end c loop  
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  $H_{pc}^T t_p = H_{pc}^T H_{pp}^{-1} b_p$  from part c of right hand side (vector B).  
Compute the matrix  $H_{pc}^T H_{pp}^{-1}$  and store it in a variable  $T_{pc}$ .  
For each free camera  $c2 \geq c$  on track p  
{  
Subtract  $T_{pc} H_{pc2} = H_{pc}^T H_{pp}^{-1} H_{pc2}$  from block (c, c2) of left hand side (matrix A).  
}  
}  
} // end p loop

$$\begin{matrix} \mathbf{V}^* & \delta_b \\ \left\{ \begin{matrix} \mathbf{V}_1 & 0 & 0 & 0 \\ 0 & \mathbf{V}_2 & 0 & 0 \\ 0 & 0 & \mathbf{V}_3 & 0 \\ 0 & 0 & 0 & \mathbf{V}_4 \end{matrix} \right\} & \begin{matrix} [x0, y0, z0]^T \\ [x1, y1, z1]^T \\ [x2, y2, z2]^T \\ [x3, y3, z3]^T \end{matrix} \end{matrix}$$

n blocks

$$\begin{matrix} \mathbf{U}^* & \delta_a \\ \left\{ \begin{matrix} \mathbf{U}_1 & 0 & 0 \\ 0 & \mathbf{U}_2 & 0 \\ 0 & 0 & \mathbf{U}_3 \end{matrix} \right\} & \begin{matrix} [9X1]^T \\ [9X1]^T \\ [9X1]^T \end{matrix} \end{matrix}$$

m blocks



Engels, et al 2006

Note that  $V^{*-1} = \begin{pmatrix} V_1^{*-1} & 0 & \dots \\ 0 & V_2^{*-1} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$

$$\begin{aligned} H_{PP} &= J_P^T J_P && \equiv V^* \\ H_{PC} &= J_P^T J_C, && \equiv W^T \\ H_{CC} &= J_C^T J_C, && \equiv U^* \\ b_P &= -J_P^T f, && \equiv \epsilon_b \\ b_C &= -J_C^T f && \equiv \epsilon_a \end{aligned}$$

$$V_i \equiv \sum_{j=1}^3 B_{ij}^T B_{ij}, \text{ for 1 feature, sum over images}$$

$$U_j \equiv \sum_{i=1}^4 A_{ij}^T A_{ij}, \text{ for 1 image, sum over features}$$

$\delta_b$  is  $dP$  is  $[3*n\_features \times 1]$   
 $\epsilon_b$  is  $b_P$  is  $[3*n\_features \times 1]$   
 $\delta_a$  is  $dC$  is  $[9*m\_images \times 1]$   
 $\epsilon_a$  is  $b_C$  is  $[9*m\_images \times 1]$

$$\begin{aligned} (U^* - W V^{*-1} W^T) \delta_a &= \epsilon_a - W V^{*-1} \epsilon_b \\ \underbrace{(H_{CC} - H_{PC}^T H_{PP}^{-1} H_{PC})}_{A} dC &= \underbrace{b_C - H_{PC}^T H_{PP}^{-1} b_P}_B \end{aligned}$$

$$V^* \delta_b = \epsilon_b - W^T \delta_a$$

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

$$\begin{aligned} &= \begin{matrix} tP \\ [3n \times 1] \end{matrix} - \begin{matrix} tPC^T & * & dC \\ [3n \times 9m] & & [9m \times 1] \end{matrix} \end{aligned}$$

## 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  $T_{pc}^T dc$  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  $\lambda$  and go to Step 3 (unless converged, in which case quit).
- 10 Otherwise, increase  $\lambda$  and go to Step 3 (unless exceeded the maximum number of iterations, in which case quit).

$$tP = hPPI^{-1} * bPI \text{ is } V^{-1} * bPI \quad [3n \times 1]$$

$$\begin{aligned} &-(V1^{-1}) * (B11T * F11 - B12T * F12 - B13T * F13) \\ &-(V2^{-1}) * (B21T * F21 - B22T * F22 - B23T * F23) \\ &-(V3^{-1}) * (B31T * F31 - B32T * F32 - B33T * F33) \\ &-(V4^{-1}) * (B41T * F41 - B42T * F42 - B43T * F43) \end{aligned}$$

where each row is  $[3 \times 1]$

$$tPC^T \quad [3n \times 9m]$$

$$\begin{aligned} &V1^{-1} * W11T \quad V1^{-1} * W12T \quad V1^{-1} * W13T \\ &V2^{-1} * W21T \quad V2^{-1} * W22T \quad V2^{-1} * W23T \\ &V3^{-1} * W31T \quad V3^{-1} * W32T \quad V3^{-1} * W33T \\ &V4^{-1} * W41T \quad V4^{-1} * W42T \quad V4^{-1} * W43T \end{aligned}$$

where each block is  $= [3 \times 9]$

$$dC \quad [9m \times 1]$$

each block is  $= [9 \times 1]$

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...

## 6.1 The Schur Complement and the Reduced Bundle System

**Schur complement:** Consider the following block triangular matrix factorization:

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ CA^{-1} & 1 \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & \bar{D} \end{pmatrix} \begin{pmatrix} 1 & A^{-1}B \\ 0 & 1 \end{pmatrix}, \quad \bar{D} \equiv D - CA^{-1}B \quad (16)$$

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} 1 & -A^{-1}B \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A^{-1} & 0 \\ 0 & \bar{D}^{-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -CA^{-1} & 1 \end{pmatrix} = \begin{pmatrix} A^{-1} + A^{-1}B\bar{D}^{-1}CA^{-1} & -A^{-1}B\bar{D}^{-1} \\ -\bar{D}^{-1}CA^{-1} & \bar{D}^{-1} \end{pmatrix} \quad (17)$$

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

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} \bar{A}^{-1} & -\bar{A}^{-1}B\bar{D}^{-1} \\ -\bar{D}^{-1}C\bar{A}^{-1} & \bar{D}^{-1} + \bar{D}^{-1}C\bar{A}^{-1}B\bar{D}^{-1} \end{pmatrix}, \quad \bar{A} = A - B\bar{D}^{-1}C$$

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

$$(A \pm B\bar{D}^{-1}C)^{-1} = A^{-1} \mp A^{-1}B(D \pm CA^{-1}B)^{-1}CA^{-1} \quad (18)$$

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

$$\begin{aligned} (U^* - W V^{*-1} W^T) \delta_a &= \epsilon_a - W V^{*-1} \epsilon_b \\ \underbrace{(H_{CC} - H_{PC}^T \underbrace{H_{PP}^{-1}}_{\bar{D}} H_{PC})}_{\bar{D}} dC &= \underbrace{b_C - H_{PC}^T H_{PP}^{-1} b_P}_B \end{aligned}$$



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...

```

L = profile_cholesky_decomp(A)
for i = 1 to n do
  for j = first(i) to i do
    a = Aij -  $\sum_{k=\max(\text{first}(i), \text{first}(j))}^{j-1} L_{ik} L_{jk}$ 
    Lij = (j < i) ? a / Ljj :  $\sqrt{a}$ 

```

```

x = profile_cholesky_forward_subs(A, b)
for i = first(b) to n do
  xi =  $\left( b_i - \sum_{k=\max(\text{first}(i), \text{first}(b))}^{i-1} L_{ik} x_k \right) / L_{ii}$ 

```

---

```

y = profile_cholesky_back_subs(A, x)
y = x
for i = last(b) to 1 step -1 do
  for k = max(first(i), first(y)) to i do
    yk = yk - yi Lik
  yi = yi / Lii

```

Figure 10: A complete implementation of profile Cholesky decomposition.

**cholesky:**

$\_A\_ = L * D * L^T$   
 $= L * \sqrt{D} * \sqrt{D} * L^T$

let C = L \*  $\sqrt{D}$

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)  $L * y = b \implies y$  via forward subst

(3)  $L^* * x = y \implies x$  via backward subst

[http://users.ics.forth.gr/~argyros/mypapers/2004\\_08\\_tr340\\_forth\\_sba.pdf](http://users.ics.forth.gr/~argyros/mypapers/2004_08_tr340_forth_sba.pdf)

The Design and Implementation of a Generic  
Sparse Bundle Adjustment Software Package  
Based on the Levenberg-Marquardt Algorithm†

*Manolis I.A. Lourakis and Antonis A. Argyros*

In all cases, the function pointed to by `proj` is assumed to estimate in `xij` the projection in image `j` of the point `i`. Arguments `aj` and `bi` are respectively the parameters of the `j`-th camera and `i`-th point. In other words, `proj` implements the parameterizing function `Q()`. Similarly, `projac` is assumed to compute in `Aij` and `Bij` the functions  $\frac{\partial Q(\mathbf{a}_j, \mathbf{b}_i)}{\partial \mathbf{a}_j}$  and  $\frac{\partial Q(\mathbf{a}_j, \mathbf{b}_i)}{\partial \mathbf{b}_i}$ , i.e. the jacobians with respect to `aj` and `bi` of the projection of point `i` in image `j`. If `projac` is NULL, the jacobians are

The employed world coordinate frame is taken to be aligned with the initial camera location. All subsequent camera motions are defined relative to the initial location, through the combination of a 3D rotation and a 3D translation. A 3D rotation by an angle  $\theta$  about a unit vector  $\mathbf{u} = (u_1, u_2, u_3)^T$  is represented by the quaternion  $\mathbf{R} = (\cos(\frac{\theta}{2}), u_1 \sin(\frac{\theta}{2}), u_2 \sin(\frac{\theta}{2}), u_3 \sin(\frac{\theta}{2}))$  [26]. A 3D translation is defined by a vector  $\mathbf{t}$ . A 3D point is represented by its Euclidean coordinate vector  $\mathbf{M}$ . Thus, the parameters of each camera `j` and point `i` are  $\mathbf{a}_j = (\mathbf{R}_j, \mathbf{t}_j^T)^T$  and  $\mathbf{b}_i = \mathbf{M}_i$ , respectively. With the previous definitions, the predicted projection of point `i` on image `j` is

$$\mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i) = \mathbf{K} (\mathbf{R}_j \mathbf{N}_i \mathbf{R}_j^{-1} + \mathbf{t}_j), \quad (28)$$

where  $\mathbf{K}$  is the  $3 \times 3$  intrinsic camera calibration matrix and  $\mathbf{N}_i = (0, \mathbf{M}_i^T)$  is the vector quaternion corresponding to the 3D point  $\mathbf{M}_i$ . The expression  $\mathbf{R}_j \mathbf{N}_i \mathbf{R}_j^{-1}$  corresponds to point  $\mathbf{M}_i$  rotated by an angle  $\theta_j$  about unit vector  $\mathbf{u}_j$ , as specified by the quaternion  $\mathbf{R}_j$ . Source file `eucsbademo.c` accompanying the `sba` package im-

[http://users.ics.forth.gr/~argyros/mypapers/2004\\_08\\_tr340\\_forth\\_sba.pdf](http://users.ics.forth.gr/~argyros/mypapers/2004_08_tr340_forth_sba.pdf)

The Design and Implementation of a Generic  
Sparse Bundle Adjustment Software Package  
Based on the Levenberg-Marquardt Algorithm†

Manolis I.A. Lourakis and Antonis A. Argyros

algorithm<sup>3</sup>. This procedure can be embedded into the LM algorithm of section 2 at the point indicated by the rectangular box in Fig. 1, leading to a sparse bundle adjustment algorithm.

Figure 2: Algorithm for solving the sparse normal equations arising in generic bundle adjustment; see text for details.

12 / 23

**Input:** The current parameter vector partitioned into  $m$  camera parameter vectors  $\mathbf{a}_j$  and  $n$  3D point parameter vectors  $\mathbf{b}_i$ , a function  $\mathbf{Q}$  employing the  $\mathbf{a}_j$  and  $\mathbf{b}_i$  to compute the predicted projections  $\hat{\mathbf{x}}_{ij}$  of the  $i$ -th point on the  $j$ -th image, the observed image point locations  $\mathbf{x}_{ij}$  and a damping term  $\mu$  for LM.

**Output:** The solution  $\delta$  to the normal equations involved in LM-based bundle adjustment.

**Algorithm:**

Compute the derivative matrices  $\mathbf{A}_{ij} := \frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{a}_j} = \frac{\partial \mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i)}{\partial \mathbf{a}_j}$ ,  $\mathbf{B}_{ij} := \frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{b}_i} = \frac{\partial \mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i)}{\partial \mathbf{b}_i}$  and the error vectors  $\epsilon_{ij} := \mathbf{x}_{ij} - \hat{\mathbf{x}}_{ij}$ , where  $i$  and  $j$  assume values in  $\{1, \dots, n\}$  and  $\{1, \dots, m\}$  respectively.

Compute the following auxiliary variables:

$$\mathbf{U}_j := \sum_i \mathbf{A}_{ij}^T \Sigma_{\mathbf{x}_{ij}}^{-1} \mathbf{A}_{ij} \quad \mathbf{V}_i := \sum_j \mathbf{B}_{ij}^T \Sigma_{\mathbf{x}_{ij}}^{-1} \mathbf{B}_{ij} \quad \mathbf{W}_{ij} := \mathbf{A}_{ij}^T \Sigma_{\mathbf{x}_{ij}}^{-1} \mathbf{B}_{ij}$$

$$\epsilon_{\mathbf{a}_j} := \sum_i \mathbf{A}_{ij}^T \Sigma_{\mathbf{x}_{ij}}^{-1} \epsilon_{ij} \quad \epsilon_{\mathbf{b}_i} := \sum_j \mathbf{B}_{ij}^T \Sigma_{\mathbf{x}_{ij}}^{-1} \epsilon_{ij}$$

Augment  $\mathbf{U}_j$  and  $\mathbf{V}_i$  by adding  $\mu$  to their diagonals to yield  $\mathbf{U}_j^*$  and  $\mathbf{V}_i^*$ .

Compute  $\mathbf{Y}_{ij} := \mathbf{W}_{ij} \mathbf{V}_i^{*-1}$ .

Compute  $\delta_{\mathbf{a}}$  from  $\mathbf{S} (\delta_{\mathbf{a}_1}^T, \delta_{\mathbf{a}_2}^T, \dots, \delta_{\mathbf{a}_m}^T)^T = (\mathbf{e}_1^T, \mathbf{e}_2^T, \dots, \mathbf{e}_m^T)^T$ ,

where  $\mathbf{S}$  is a matrix consisting of  $m \times m$  blocks; block  $jk$  is defined by

$$\mathbf{S}_{jk} = \delta_{jk} \mathbf{U}_j^* - \sum_i \mathbf{Y}_{ij} \mathbf{W}_{ik}^T, \text{ where } \delta_{jk} \text{ is Kronecker's delta}$$

and

$$\mathbf{e}_j = \epsilon_{\mathbf{a}_j} - \sum_i \mathbf{Y}_{ij} \epsilon_{\mathbf{b}_i}.$$

Compute each  $\delta_{\mathbf{b}_i}$  from the equation  $\delta_{\mathbf{b}_i} = \mathbf{V}_i^{*-1} (\epsilon_{\mathbf{b}_i} - \sum_j \mathbf{W}_{ij}^T \delta_{\mathbf{a}_j})$ .

Form  $\delta$  as  $(\delta_{\mathbf{a}}^T, \delta_{\mathbf{b}}^T)^T$ .

[http://users.ics.forth.gr/~argyros/mypapers/2004\\_08\\_tr340\\_forth\\_sba.pdf](http://users.ics.forth.gr/~argyros/mypapers/2004_08_tr340_forth_sba.pdf)

The Design and Implementation of a Generic  
Sparse Bundle Adjustment Software Package  
Based on the Levenberg-Marquardt Algorithm†

Figure 1: Levenberg-Marquardt non-linear least squares algorithm; see text and [16, 20] for details. The reason for enclosing a statement in a rectangular box will be explained in section 3.

6 / 23

Manolis I.A. Lourakis and Antonis A. Argyros

$$\mathbf{p}_{new} := \mathbf{p} + \delta \mathbf{p};$$

$$\rho := (||\epsilon_{\mathbf{p}}||^2 - ||\mathbf{x} - f(\mathbf{p}_{new})||^2) / (\delta_{\mathbf{p}}^T (\mu \delta \mathbf{p} + \mathbf{g}));$$

gain ratio →

**Input:** A vector function  $f : \mathcal{R}^m \rightarrow \mathcal{R}^n$  with  $n \geq m$ , a measurement vector  $\mathbf{x} \in \mathcal{R}^n$  and an initial parameters estimate  $\mathbf{p}_0 \in \mathcal{R}^m$ .  
**Output:** A vector  $\mathbf{p}^+ \in \mathcal{R}^m$  minimizing  $||\mathbf{x} - f(\mathbf{p})||^2$ .  
**Algorithm:**  
 $k := 0; \nu := 2; \mathbf{p} := \mathbf{p}_0;$   
 $\mathbf{A} := \mathbf{J}^T \mathbf{J}; \epsilon_{\mathbf{p}} := \mathbf{x} - f(\mathbf{p}); \mathbf{g} := \mathbf{J}^T \epsilon_{\mathbf{p}};$   
stop:  $(||\mathbf{g}||_{\infty} \leq \varepsilon_1); \mu := \tau * \max_{i=1, \dots, m} (A_{ii});$   
while (not stop) and  $(k < k_{max})$   
 $k := k + 1;$   
repeat  

Solve  $(\mathbf{A} + \mu \mathbf{I}) \delta_{\mathbf{p}} = -\mathbf{g};$

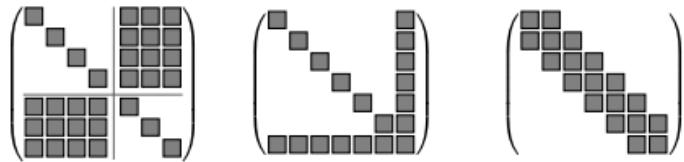
  
if  $(||\delta_{\mathbf{p}}|| \leq \varepsilon_2 ||\mathbf{p}||)$   
stop:=true;  
else  
 $\mathbf{p}_{new} := \mathbf{p} + \delta_{\mathbf{p}};$   
 $\rho := (||\epsilon_{\mathbf{p}}||^2 - ||\mathbf{x} - f(\mathbf{p}_{new})||^2) / (\delta_{\mathbf{p}}^T (\mu \delta \mathbf{p} + \mathbf{g}));$   
if  $\rho > 0$   
 $\mathbf{p} = \mathbf{p}_{new};$   
 $\mathbf{A} := \mathbf{J}^T \mathbf{J}; \epsilon_{\mathbf{p}} := \mathbf{x} - f(\mathbf{p}); \mathbf{g} := \mathbf{J}^T \epsilon_{\mathbf{p}};$   
stop:  $(||\mathbf{g}||_{\infty} \leq \varepsilon_1);$   
 $\mu := \mu * \max(\frac{1}{3}, 1 - (2\rho - 1)^3); \nu := 2;$   
else  
 $\mu := \mu * \nu; \nu := 2 * \nu;$   
endif  
endif  
until  $(\rho > 0)$  or (stop)  
endwhile

factoring an arrowhead Hessian matrix to get a reduced camera system or reduced structure system.

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

We seek variable orderings that approximately minimize the total operation count or fill-in over the whole elimination chain. For many problems a suitable ordering can be fixed in advance, typically giving one of a few standard pattern matrices such as band or arrowhead matrices, perhaps with such structure at several levels.



(25)

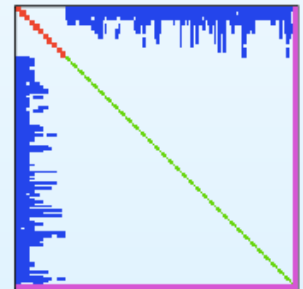
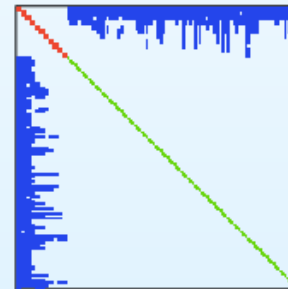
bundle Hessian      arrowhead matrix      block tridiagonal matrix

The most prominent pattern structure in bundle adjustment is the primary subdivision of the Hessian into structure and camera blocks. To get the reduced camera system (19), we treat the Hessian as an arrowhead matrix with a broad final column containing all of the camera parameters. Arrowhead matrices are trivial to factor or reduce by block  $2 \times 2$  Schur complementation, *c.f.* (16, 19). For bundle problems with many independent images and only a few features, one can also complement on the image parameter block to get a reduced *structure* system.

[http://users.ics.forth.gr/~lourakis/sba/PRCV\\_colloq.pdf](http://users.ics.forth.gr/~lourakis/sba/PRCV_colloq.pdf)

Bundle adjustment gone public  
Manolis Lourakis

- A few interesting practical situations violate its underlying assumption regarding the problem's sparsity pattern, rendering it inapplicable. E.g., fixed but unknown intrinsics shared by all cameras
- Example: Hessians corresponding to BA for motion and structure (left) and BA for motion, structure and shared intrinsics (right)



## applying local updates rather than global

### How to Avoid Singularity For Euler Angle Set?

Puneet Singla\*, Daniele Mortari<sup>†</sup>, and John L. Junkins<sup>‡</sup>

Since all rotations are performed about the principal axes of the reference frame, we define  $\mathbf{M}_i = \exp(-[\tilde{\mathbf{e}}_i]\theta_i)$  as an elementary rotation matrix about the  $\mathbf{e}_i$ -body axis. Here,  $[\tilde{\mathbf{e}}_i]$  represents the skew-symmetric cross product matrix given by the following expression:

$$[\tilde{\mathbf{a}}] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (1)$$

From the expression of  $\mathbf{M}_i$ , we can construct the following three elementary rotation matrices:

$$\mathbf{M}_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (2)$$

$$\mathbf{M}_2(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3)$$

$$\mathbf{M}_3(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

if  $i, j$ , and  $k$ , indicate the coordinate axes about which each subsequent rotation is performed, that is, they can be any integer from 1-3, provided that  $i \neq j$  and  $j \neq k$ , are satisfied then the resultant direction cosine matrix can be written as

$$\mathbf{C}_{ijk}(\theta_1, \theta_2, \theta_3) = \mathbf{M}_k(\theta_3) \mathbf{M}_j(\theta_2) \mathbf{M}_i(\theta_1) \quad (5)$$

applying local updates rather than global

**How to Avoid Singularity For Euler Angle Set?**

Puneet Singla\*, Daniele Mortari†, and John L. Junkins‡

It is a fundamental topological fact that singularities can never be eliminated in any 3-dimensional representation of orientation. But we can avoid this singularity by describing the attitude at a particular instant by the Euler angle set *which is farthest away from singularity*. In this paper, we present an algorithm to switch between different sets of Euler angles to avoid this singularity.

Table 1: Classical Parameterizations of Attitude Rotation Matrix

Paramet-rization	Dimension	Attitude Matrix	Kinematic Equations	Singularities	Constraints
DCM, $(C_{ij})$	9	$\mathbf{C} = [C_{ij}]$	$\dot{\mathbf{C}} = -[\tilde{\boldsymbol{\omega}}]\mathbf{C}$	None	$\mathbf{C}^T \mathbf{C} = \mathbf{I}$
Euler Angles EA $(\theta_i)$	3	$\mathbf{C} = \begin{bmatrix} \text{transcendental} \\ \text{functions of} \\ \theta'_i s \end{bmatrix}$	$\dot{\boldsymbol{\theta}} = \begin{bmatrix} \text{transcendental} \\ \text{functions of} \\ \theta'_i s \end{bmatrix} \boldsymbol{\omega}$	$\theta_2 = \pm \frac{\pi}{2}$	None
Euler-Rodrigues Symmetric Parameters ERSP $(q_i)$	4	$\mathbf{C} = \begin{bmatrix} \text{algebraic} \\ \text{functions of} \\ q'_i s \end{bmatrix}$	$\dot{\mathbf{q}} = \begin{bmatrix} \text{linear} \\ \text{functions of} \\ q'_i s \end{bmatrix} \begin{Bmatrix} 0 \\ \boldsymbol{\omega} \end{Bmatrix}$	None	$\mathbf{q}^T \mathbf{q} = 1$
RP $(r_i)$	3	$\mathbf{C} = \begin{bmatrix} \text{quadratic} \\ \text{functions of} \\ r'_i s \end{bmatrix}$	$\dot{\mathbf{r}} = \begin{bmatrix} \text{non-linear} \\ \text{functions of} \\ r'_i s \end{bmatrix} \boldsymbol{\omega}$	$\phi = \pm \pi$	None
MRP $(\sigma_i)$	3	$\mathbf{C} = \begin{bmatrix} \text{quartic} \\ \text{functions of} \\ \sigma'_i s \end{bmatrix}$	$\dot{\boldsymbol{\sigma}} = \begin{bmatrix} \text{non-linear} \\ \text{functions of} \\ \sigma'_i s \end{bmatrix} \boldsymbol{\omega}$	$\phi = \pm 2\pi$	None

applying local updates rather than global

## How to Avoid Singularity For Euler Angle Set?

Puneet Singla<sup>\*</sup>, Daniele Mortari<sup>†</sup>, and John L. Junkins<sup>‡</sup>

second reason comes out from the capability of the Shuster's Method of Sequential Rotations[4] (MSR) to avoid the *un-avoidable* singularity affecting all the minimum attitude parameter. Not only the original QUEST[4] algorithm has taken advantage from the MSR technique, but also some recent attitude determination approaches, like ESOQ2[5] and OLAE[6].



**Snavely's GitHub [bundler\\_sfm/lib/sba-1.5/sba\\_levmar.c](#)**