Vision 3D artificielle
Multiple view geometry

Pascal Monasse pascal.monasse@enpc.fr

IMAGINE/LIGM, École nationale des ponts et chaussées

https://imagine.enpc.fr/~monasse/Stereo/

Contents

Contents

Reminder: Triangulation

- Let us write again the binocular formulae:

$$\lambda x = K(RX + T) \quad \lambda' x' = K' X$$

- Write $Y^\top = \begin{pmatrix} X^\top & 1 & -\lambda & -\lambda' \end{pmatrix}$:

$$\begin{pmatrix} KR & KT & x & 0_3 \\ K' & 0_3 & 0_3 & x' \end{pmatrix} Y = 0_6$$

  (6 equations$\leftrightarrow$5 unknowns+1 epipolar constraint)
- We can then recover X.

Multi-linear constraints

- Bilinear constraints: fundamental matrix $x^\top F x' = 0$.
- There are trilinear constraints: $x''_i = x^\top T_i x'$, which are *not* combinations of bilinear contraints
- All constraints involving more than 3 views are combinations of 2- and/or 3-view constraints.

Trilinear constraints

- Write $\lambda_i x_i = K_i(R_i X + T_i)$
- Write as $AY = 0$ with $Y = \begin{pmatrix} X^\top & 1 & -\lambda_1 & \cdots & -\lambda_n \end{pmatrix}^\top$
- A has size $3n \times (n+4)$ ($n = 2 \rightarrow 6 \times 6$, $n = 3 \rightarrow 9 \times 7$, $\ldots$): $n > 2 \Rightarrow$ more rows than columns.
- Look at the rank of A (must be $\leq n+4$)

Trilinear constraints

- Assume $R_1 = Id$ and $T_1 = 0$. We write A of size $3n \times (n+4)$:

$$A = \begin{pmatrix} K_1 & 0 & x_1 & 0 & \cdots & 0 \\ K_2 R_2 & K_2 T_2 & 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ K_n R_n & K_n T_n & 0 & \cdots & 0 & x_n \end{pmatrix}$$

- Subtracting from 3rd column the first column multiplied by $K_1^{-1} x_1$, rank of A = rank of A' with:

$$A' = \begin{pmatrix} K_1 & 0 & 0 & 0 & \cdots & 0 \\ K_2 R_2 & K_2 T_2 & -K_2 R_2 K_1^{-1} x_1 & x_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ K_n R_n & K_n T_n & -K_n R_n K_1^{-1} x_1 & \cdots & 0 & x_n \end{pmatrix}$$

- Since K_1 is invertible, we have to look at the rank of the lower-right $3(n-1) \times (n+1)$ submatrix.

Trilinear constraints

- Rank of A=3+rank of B with

$$B = \begin{pmatrix} K_2 T_2 & K_2 R_2 K_1^{-1} x_1 & x_2 & 0 & 0 & \cdots & 0 \\ K_3 T_3 & K_3 R_3 K_1^{-1} x_1 & 0 & x_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ K_{n-1} T_{n-1} & K_{n-1} R_{n-1} K_1^{-1} x_1 & 0 & \cdots & 0 & x_{n-1} & 0 \\ K_n T_n & K_n R_n K_1^{-1} x_1 & 0 & \cdots & 0 & 0 & x_n \end{pmatrix}$$

- Size of B: $3(n-1) \times (n+1)$.

Trilinear constraints

- DB has same rank as B since D is full rank $3(n-1)$:

$$D = \begin{pmatrix} x_2^\top & 0 & 0 & \cdots & 0 \\ 0 & x_3^\top & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & x_{n-1}^\top & 0 \\ 0 & \cdots & 0 & 0 & x_n^\top \\ [x_2]_\times & 0 & 0 & \cdots & 0 \\ 0 & [x_3]_\times & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & [x_{n-1}]_\times & 0 \\ 0 & \cdots & 0 & 0 & [x_n]_\times \end{pmatrix}$$

- D has size $4(n-1) \times 3(n-1)$
- It is easy to check that the kernel of D is $\{0\}$.

Trilinear constraints

- We get:

$$DB = \begin{pmatrix} x_2^\top K_2 T_2 & x_2^\top K_2 R_2 K_1^{-1} x_1 & x_2^\top x_2 & 0 & \cdots & 0 \\ x_3^\top K_3 T_3 & x_3^\top K_3 R_3 K_1^{-1} x_1 & 0 & x_3^\top x_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ x_n^\top K_n T_n & x_n^\top K_n R_n K_1^{-1} x_1 & 0 & \cdots & 0 & x_n^\top x_n \\ M_1 & M_2 & 0 & \ldots & 0 & 0 \end{pmatrix}$$

- Since $x_i^\top x_i > 0$, rank of $B = (n-1) +$ Rank of M (size $3(n-1) \times 2$):

$$M = \begin{pmatrix} [x_2]_\times K_2 R_2 K_1^{-1} x_1 & [x_2]_\times K_2 T_2 \\ \vdots & \vdots \\ [x_n]_\times K_n R_n K_1^{-1} x_1 & [x_n]_\times K_n T_n \end{pmatrix}$$

- We should have: rank of $M = 1$, so that rank of $A = n + 3$.
- Write that 2×2 submatrices of M should have determinant 0

Trilinear constraints

- Proposition Let M a $3n \times 2$ matrix written in blocks of 3 rows:

$$M = \begin{pmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{pmatrix}$$

Rank of $M < 2$ iff $\forall i, j$, $a_i b_j^\top - b_i a_j^\top = 0$.

Trilinear constraints

- Proposition Let M a $3n \times 2$ matrix written in blocks of 3 rows:

$$M = \begin{pmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{pmatrix}$$

  Rank of $M < 2$ iff $\forall i, j$, $a_i b_j^\top - b_i a_j^\top = 0$.
- Proof: The case $b = 0$ is trivial, assume $b \neq 0$.
  - $\Rightarrow$ We have $a = \lambda b$, and $a_i b_j^\top - b_i a_j^\top = \lambda(b_i b_j^\top - b_i b_j^\top) = 0$.
  - $\Leftarrow$ We have $(a_i b_j^\top - b_i a_j^\top)^{kl} = a_i^k b_j^l - b_i^k a_j^l = \begin{vmatrix} a_i^k & b_i^k \\ a_j^l & b_j^l \end{vmatrix}$. We get that all 2×2 submatrices of M have null determinant.

Trilinear constraints

- Proposition Let M a $3n \times 2$ matrix written in blocks of 3 rows:

$$M = \begin{pmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{pmatrix}$$

  Rank of $M < 2$ iff $\forall i, j,\ a_i b_j^\top - b_i a_j^\top = 0$.
- For $i = j$, $([x_i]_\times K_i R_i K_1^{-1} x_1) \times ([x_i]_\times K_i T_i) = 0$ amounts to

$$([x_i]_\times K_i T_i)^\top (K_i R_i K_1^{-1} x_1) = \begin{vmatrix} x_i & K_i T_i & K_i R_i K_1^{-1} x_1 \end{vmatrix} = 0 \text{ or}$$

$$x_i^T K_i^{-\top} [T_i]_\times R_i K_1^{-1} x_1 = 0 \text{ (epipolar constraint)}$$

- $[x_i]_\times K_i R_i K_1^{-1} x_1 ([x_j]_\times K_j T_j)^\top - [x_i]_\times K_i T_i ([x_j]_\times K_j R_j K_1^{-1} x_1)^\top = 0$

$$[x_i]_\times \left(\sum_k x_1^k \mathcal{T}_{ij}^k \right) [x_j]_\times = 0 \text{ (9 trilinear constraints)}$$

Trilinear constraints

- A triplet (x_1, x_i, x_j) imposes at most 4 independent constraints on $\mathcal{T}_{ij}^k$ because of the cross-products.
- Rank of $M = 0$ (multiple solutions X) means

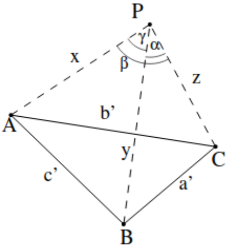
$$\forall i, [x_i]_\times K_i R_i K_1^{-1} x_1 = [x_i]_\times K_i T_i = 0$$

so that $K_i R_i K_1^{-1} x_1$ and $K_i T_i$ are proportional, hence $x_1 = \lambda K_1 R_i^\top T_i$ (epipole in image 1 wrt image i), so that all camera centers are aligned and X is on this line.

Contents

PnP

- "PnP" = Perspective from n Points.
- From 2D-3D correspondences (x_i, X_i) and known K, recover $P = K\begin{pmatrix} R & T \end{pmatrix}$ so that $x_i \sim PX_i$
- Remember calibration from a 3D rig: same problem but with unknown K, $P = K\begin{pmatrix} R & T \end{pmatrix}$.
- Minimal problem: $n = 3$, $P3P$ problem, up to 4 solutions:



(c) Gao, Hou, Tang & Cheng

$$\begin{cases} Y^2 + Z^2 - pYZ - a^2 = 0 & (p = 2\cos\alpha) \\ X^2 + Z^2 - qXZ - b^2 = 0 & (q = 2\cos\beta) \\ X^2 + Y^2 - rXY - c^2 = 0 & (r = 2\cos\gamma) \end{cases}$$

Write $x = X/Z$, $y = Y/Z$,
$a' = a^2/c^2$, $b' = b^2/c^2$, $v = c^2/Z^2$.

$$\begin{cases} y^2 + 1 - py - a'v = 0 \\ x^2 + 1 - qx - b'v = 0 \\ x^2 + y^2 - rxy - v = 0 \end{cases} \Rightarrow \begin{cases} (1-a')y^2 - a'x^2 + a'rxy - py + 1 = 0 \\ (1-b')x^2 - b'y^2 + b'rxy - qx + 1 = 0 \\ \text{(intersection of two conics)} \end{cases}$$

PnP, $n \geq 4$

[Lepetit, Moreno-Noguer & Fua, *EPnP: An accurate O(n) solution to the PnP problem*, IJCV 2008]

- Write $X_i = \sum_{j=1\ldots4} \alpha_{ij} C_j^w$ with C_j^w four fixed points.
- Write $C_j = R C_j^w + T$ in camera coordinates.
- Project onto image to obtain a $2n \times 12$ linear system on $\{C_j\}$:

$$[K^{-1}x_i]_\times \sum_{j=1\ldots4} \alpha_{ij} C_j = 0$$

- From $MC = 0$, write $C = \sum_{k=1\ldots N} \beta_k V^k$ with the N last columns of V from SVD of M
  $(n = 4 \rightarrow N = 4,\ n = 5 \rightarrow N = 2,\ n \geq 6 \rightarrow N = 1)$
- Write the conservation of distances (6 equations in β):

$$1 \leq i < j \leq 4 : \left\| \sum_{k=1\ldots N} \beta_k ({V^k}_{3i-2:3i} - {V^k}_{3j-2:3j}) \right\| = \| C_i^w - C_j^w \|$$

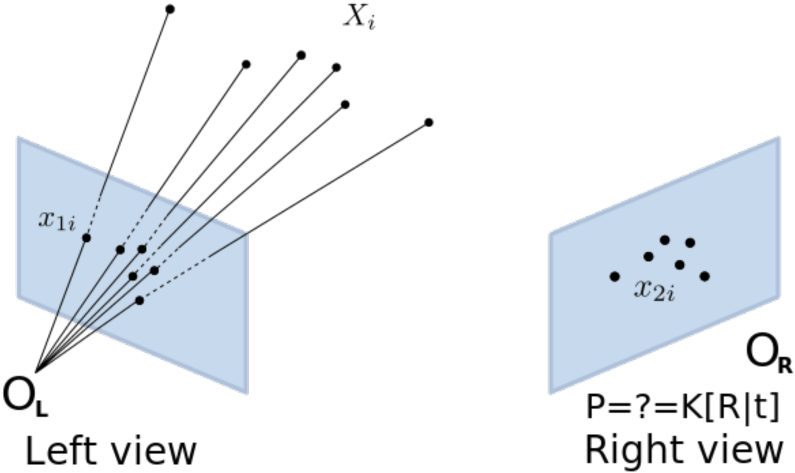
Contents

Incremental multi-view calibration

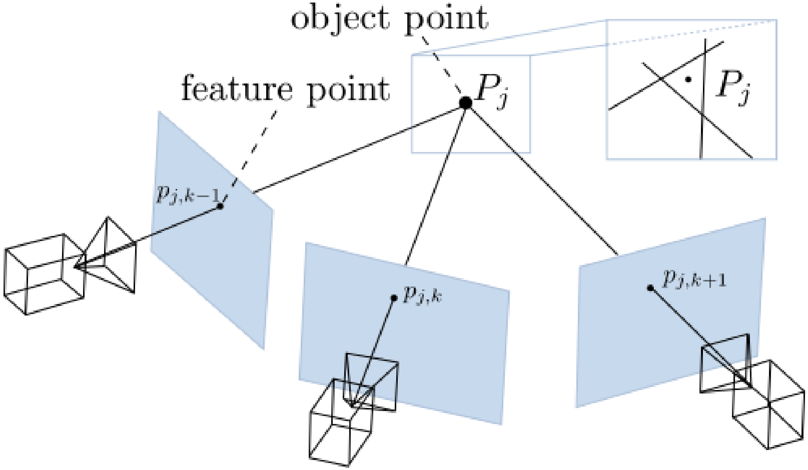
1. Compute two-view correspondences
2. Build tracks (multi-view correspondences)
3. Start from initial pair: compute F, deduce R, T and 3D points (known K)
4. Add image with common points.
5. Estimate pose (R, T)
6. Add new 3D points
7. Bundle adjustment
8. Go to 4

See open source software Bundler: SfM for Unordered Image Collections (`http://www.cs.cornell.edu/~snavely/bundler/`)

Incremental multi-view calibration



Incremental multi-view calibration



Bundle adjustment

We have the equations

$$x_{ij} = DK \begin{pmatrix} R_j & T_j \end{pmatrix} X_i$$

- i: 3D point index
- j: view index
- x_{ij}: 2D projection in view j of point X_i
- D: geometric distortion model
- K: internal parameters of camera

Minimize by Levenberg-Marquardt the error

$$E = \sum_{ij} d(x_{ij}, DK \begin{pmatrix} R_j & T_j \end{pmatrix} X_i)^2$$

Global calibration

- Compute E_{ij}, essential matrices between all views i and j
- Extract R_{ij} and T_{ij} from E_{ij}
- Rotation alignment: recover $\{R_i\}$, global rotation with respect to $R_0 = Id$, such that $R_i = R_{ij} R_j$ for all i, j
  - [Martinec&Pajdla CVPR 2007]: write R_{ij} as unitary quaternions, find minimum of

    $$\sum_{ij} \|q_i - q_{ij} q_j\|^2 \text{ with } \| \begin{pmatrix} q_1^\top & \cdots & q_n^\top \end{pmatrix}^\top \| = n$$

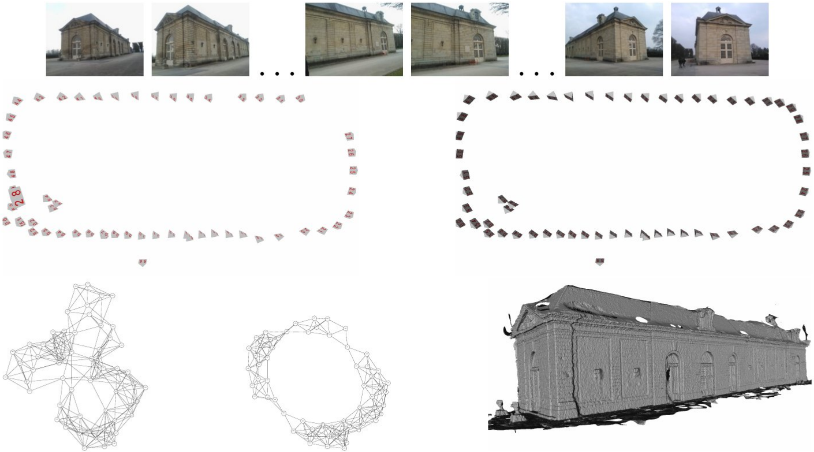
    But this does not ensure $\|q_i\| = 1$, condition for a quaternion to represent a rotation...
  - No exact solution since R_{ij} can have an error. How to close loops?
  - What about outliers among the R_{ij}?
- Translation alignment: recover $\{T_i\}$, global translation with respect to $T_0 = 0$, such that $T_i = R_{ij} T_j + \lambda_{ij} T_{ij}$

Global calibration

- Compute E_{ij}, essential matrices between all views i and j
- Extract R_{ij} and T_{ij} from E_{ij}
- Rotation alignment: recover $\{R_i\}$, global rotation with respect to $R_0 = Id$, such that $R_i = R_{ij}R_j$ for all i, j
- Translation alignment: recover $\{T_i\}$, global translation with respect to $T_0 = 0$, such that $T_i = R_{ij}T_j + \lambda_{ij}T_{ij}$
  - [Moulon&Monasse&Marlet ICCV 2013]: solve the linear programme

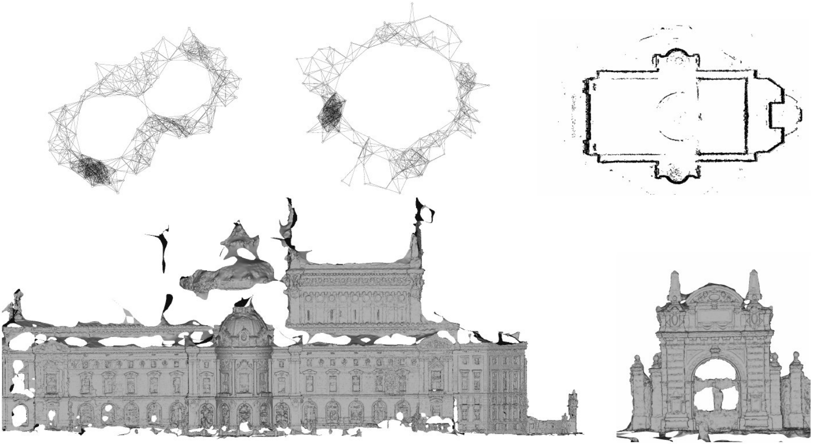
$$\min_{\{T_i\},\{\lambda_{ij}\},\gamma} \gamma \text{ with } \lambda_{ij} \geq 1, \|T_i - R_{ij}T_j - \lambda_{ij}T_{ij}\|_\infty \leq \gamma$$

Some results



Orangerie dataset

Some results

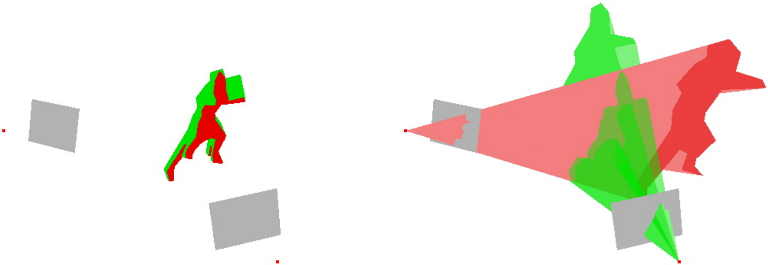


Opera Garnier dataset

Contents

Visual hull

- We assume we are able to segment the object of interest in each view
- From the silhouette, we can restrict the location inside a cone
- Intersect cones from all views
- The result is called the visual hull

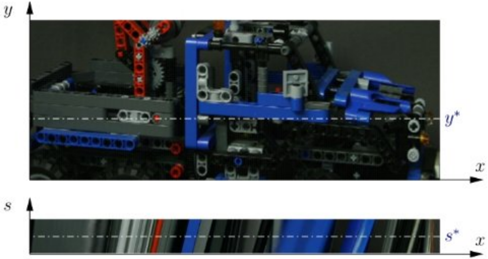


Source: Wikipedia `http://en.wikipedia.org/wiki/Visual_hull`

Epipolar plane imagery

A technique for depth estimation from a movie with controlled motion

- Assume a uniform motion of camera along the horizontal line
- Consider 2D cuts (x, y^*, t) of the volume
- Edges move along lines, whose slope is the disparity
- Advantage: large baseline between distant time steps (accurate estimation) and small baseline between close times steps (easier tracking)



Source: `http://www.informatik.uni-konstanz.de/cvia/research/light-field-analysis/consistent-depth-estimation/`

Software

Infrastructure:

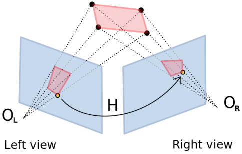
- *Eigen*: C++ library for linear algebra
- Google's *Ceres Solver* for bundle adjustment (automatic differentiation)

SfM pipelines:

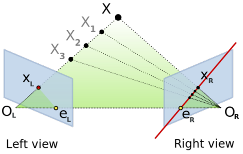
- *Bundler* (2008, open source, University of Washington)
- *PhotoScan* (2010, commercial, Agisoft)
- *VisualSfM* (2011, open source, University of Wahington): GPU
- *OpenMVG* (2012, open source, École des Ponts ParisTech)
- *ColMap* (2016, open source, University of North Carolina)

Conclusion

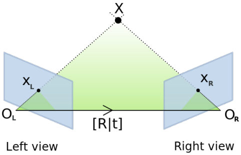
- Multi-view reconstruction is an active and lively field of research, but less explored than 2-view stereo correspondence
- Project: openMVG (incremental and global pipelines)



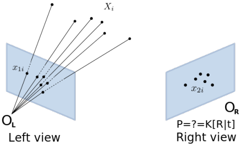
Homography



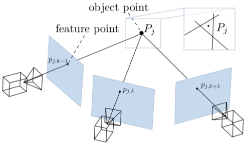
Fundamental matrix



Essential matrix



Resection



Triangulation