3D Computer Vision

Radim Šára   Martin Matoušek

Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague

https://cw.fel.cvut.cz/wiki/courses/tdv/start

http://cmp.felk.cvut.cz
mailto:sara@cmp.felk.cvut.cz
phone ext. 7203

rev. October 27, 2020

Open Informatics Master's Course

►7-Point Algorithm for Estimating Fundamental Matrix

Problem: Given a set $\{(x_i, y_i)\}_{i=1}^{k}$ of $k = 7$ correspondences, estimate f. m. $\mathbf{F}$.

$$\underline{\mathbf{y}}_i^\top \mathbf{F}\, \underline{\mathbf{x}}_i = 0, \quad i = 1, \dots, k, \qquad \underline{\text{known}}: \ \underline{\mathbf{x}}_i = (u_i^1, v_i^1, 1), \ \ \underline{\mathbf{y}}_i = (u_i^2, v_i^2, 1)$$

terminology: correspondence = truth, later: match = algorithm's result; hypothesized corresp.

Solution:

$$\underline{\mathbf{y}}_i^\top \mathbf{F}\, \underline{\mathbf{x}}_i = (\underline{\mathbf{y}}_i \underline{\mathbf{x}}_i^\top) : \mathbf{F} = \left(\operatorname{vec}(\underline{\mathbf{y}}_i \underline{\mathbf{x}}_i^\top)\right)^\top \operatorname{vec}(\mathbf{F}),$$ rotation property of matrix trace

$$\operatorname{vec}(\mathbf{F}) = \begin{bmatrix} f_{11} & f_{21} & f_{31} & \dots & f_{33} \end{bmatrix}^\top \in \mathbb{R}^9$$ column vector from matrix

$$\mathbf{D} = \begin{bmatrix} \left(\operatorname{vec}(\underline{\mathbf{y}}_1 \underline{\mathbf{x}}_1^\top)\right)^\top \\ \left(\operatorname{vec}(\underline{\mathbf{y}}_2 \underline{\mathbf{x}}_2^\top)\right)^\top \\ \left(\operatorname{vec}(\underline{\mathbf{y}}_3 \underline{\mathbf{x}}_3^\top)\right)^\top \\ \vdots \\ \left(\operatorname{vec}(\underline{\mathbf{y}}_k \underline{\mathbf{x}}_k^\top)\right)^\top \end{bmatrix} = \begin{bmatrix} u_1^1 u_1^2 & u_1^1 v_1^2 & u_1^1 & u_1^2 v_1^1 & v_1^1 v_1^2 & v_1^1 & u_1^2 & v_1^2 & 1 \\ u_2^1 u_2^2 & u_2^1 v_2^2 & u_2^1 & u_2^2 v_2^1 & v_2^1 v_2^2 & v_2^1 & u_2^2 & v_2^2 & 1 \\ u_3^1 u_3^2 & u_3^1 v_3^2 & u_3^1 & u_3^2 v_3^1 & v_3^1 v_3^2 & v_3^1 & u_3^2 & v_3^2 & 1 \\ \vdots & & & & & & & & \vdots \\ u_k^1 u_k^2 & u_k^1 v_k^2 & u_k^1 & u_k^2 v_k^1 & v_k^1 v_k^2 & v_k^1 & u_k^2 & v_k^2 & 1 \end{bmatrix} \in \mathbb{R}^{k,9}$$

$$\boxed{\mathbf{D} \operatorname{vec}(\mathbf{F}) = \mathbf{0}}$$

▶7-Point Algorithm Continued

$$\mathbf{D}\,\mathrm{vec}(\mathbf{F}) = \mathbf{0}, \quad \mathbf{D} \in \mathbb{R}^{k,9}$$

- for $k = 7$ we have a rank-deficient system, the null-space of $\mathbf{D}$ is 2-dimensional
- but we know that $\det \mathbf{F} = 0$, hence
  1. find a basis of the null space of $\mathbf{D}$: $\mathbf{F}_1$, $\mathbf{F}_2$ — by SVD or QR factorization
  2. get up to 3 real solutions for α from

     $$\det(\alpha \mathbf{F}_1 + (1-\alpha)\mathbf{F}_2) = 0 \qquad \text{cubic equation in } \alpha$$

  3. get up to 3 fundamental matrices $\mathbf{F} = \alpha_i \mathbf{F}_1 + (1-\alpha_i)\mathbf{F}_2$
  4. if $\mathrm{rank}\,\mathbf{F} < 2$ then fail

- the result may depend on image (domain) transformations
- normalization improves conditioning →92
- this gives a good starting point for the full algorithm →109
- dealing with mismatches need not be a part of the 7-point algorithm →110

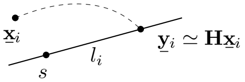
▶Degenerate Configurations for Fundamental Matrix Estimation

When is $\mathbf{F}$ not uniquely determined from any number of correspondences? [H&Z, Sec. 11.9]

1. when images are related by homography
   a) camera centers coincide $\mathbf{t}_{21} = 0$: $\mathbf{H} = \mathbf{K}_2 \mathbf{R}_{21} \mathbf{K}_1^{-1}$ $\mathbf{H}$ – as in epipolar homography
   b) camera moves but all 3D points lie in a plane $(\mathbf{n}, d)$: $\mathbf{H} = \mathbf{K}_2 (\mathbf{R}_{21} - \mathbf{t}_{21} \mathbf{n}^\top / d) \mathbf{K}_1^{-1}$

   - in both cases: epipolar geometry is not defined
   - we get an arbitrary solution from the 7-point algorithm in the form of $\mathbf{F} = [\underline{\mathbf{s}}]_\times \mathbf{H}$

   note that $[\underline{\mathbf{s}}]_\times \mathbf{H} \simeq \mathbf{H}' [\underline{\mathbf{s}}']_\times$ →76



   - given (arbitrary, fixed) $\underline{\mathbf{s}}$
   - and correspondence $x_i \leftrightarrow y_i$
   - y_i is the image of x_i: $\underline{\mathbf{y}}_i \simeq \mathbf{H}\underline{\mathbf{x}}_i$
   - a necessary condition: $y_i \in l_i$, $\underline{\mathbf{l}}_i \simeq \underline{\mathbf{s}} \times \mathbf{H}\underline{\mathbf{x}}_i$

   $$0 = \underline{\mathbf{y}}_i^\top (\underline{\mathbf{s}} \times \mathbf{H}\underline{\mathbf{x}}_i) = \underline{\mathbf{y}}_i^\top [\underline{\mathbf{s}}]_\times \mathbf{H}\underline{\mathbf{x}}_i \quad \text{for any } \underline{\mathbf{x}}_i, \underline{\mathbf{y}}_i, \underline{\mathbf{s}} \text{ (!)}$$

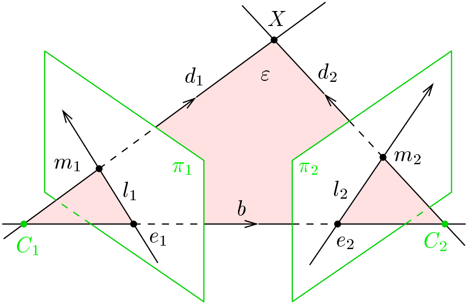
2. both camera centers and all 3D points lie on a ruled quadric
   hyperboloid of one sheet, cones, cylinders, two planes
   - there are 3 solutions for $\mathbf{F}$

notes

- estimation of $\mathbf{E}$ can deal with planes: $[\underline{\mathbf{s}}]_\times \mathbf{H}$ is essential, then $\mathbf{H} = \mathbf{R} - \mathbf{t}\mathbf{n}^\top / d$, and $\underline{\mathbf{s}} \simeq \mathbf{t}$ not arbitrary
- a complete treatment with additional degenerate configurations in [H&Z, sec. 22.2]
- a stronger epipolar constraint could reject some configurations

A Note on Oriented Epipolar Constraint

- a tighter epipolar constraint preserves orientations
- requires all points and cameras be on the same side of the plane at infinity



$$(\underline{\mathbf{e}}_2 \times \underline{\mathbf{m}}_2) \overset{+}{\sim} \mathbf{F}\,\underline{\mathbf{m}}_1$$

notation: $\underline{\mathbf{m}} \overset{+}{\sim} \underline{\mathbf{n}}$ means $\underline{\mathbf{m}} = \lambda \underline{\mathbf{n}}$, $\lambda > 0$

- we can read the constraint as $(\underline{\mathbf{e}}_2 \times \underline{\mathbf{m}}_2) \overset{+}{\sim} \mathbf{H}_e^{-\top} (\mathbf{e}_1 \times \underline{\mathbf{m}}_1)$
- note that the constraint is not invariant to the change of either sign of $\underline{\mathbf{m}}_i$
- all 7 correspondence in 7-point alg. must have the same sign — see later
- this may help reject some wrong matches, see →110 — [Chum et al. 2004]
- an even more tight constraint: scene points in front of both cameras — expensive

this is called chirality constraint

▶5-Point Algorithm for Relative Camera Orientation

Problem: Given $\{m_i, m_i'\}_{i=1}^5$ corresponding image points and calibration matrix $\mathbf{K}$, recover the camera motion $\mathbf{R}$, $\mathbf{t}$.

Obs:

1. $\mathbf{E}$ – 9 numbers but 7 DOF   rank-deficient 3×3 homogeneous matrix with two equal singular numbers
2. $\mathbf{R}$ – 3 DOF, $\mathbf{t}$ – 2 DOF only, in total 5 DOF $\to$ we need $8 - 5 = 3$ constraints on $\mathbf{E}$
3. $\mathbf{E}$ essential iff it has two equal singular values and the third is zero $\to$81

This gives an equation system:

$$\underline{\mathbf{v}}_i^\top \mathbf{E}\, \underline{\mathbf{v}}_i' = 0 \qquad \text{5 linear constraints } (\underline{\mathbf{v}} \simeq \mathbf{K}^{-1}\underline{\mathbf{m}})$$

$$\det \mathbf{E} = 0 \qquad \text{1 cubic constraint}$$

$$\mathbf{E}\mathbf{E}^\top\mathbf{E} - \frac{1}{2}\operatorname{tr}(\mathbf{E}\mathbf{E}^\top)\mathbf{E} = \mathbf{0} \qquad \text{9 cubic constraints, 2 independent}$$

⊛ P1; 1pt: verify this equation from $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$, $\mathbf{D} = \lambda\, \operatorname{diag}(1, 1, 0)$

1. estimate $\mathbf{E}$ by SVD from $\underline{\mathbf{v}}_i^\top \mathbf{E}\, \underline{\mathbf{v}}_i' = 0$ by the null-space method   4D null space
2. this gives $\mathbf{E} \simeq x\mathbf{E}_1 + y\mathbf{E}_2 + z\mathbf{E}_3 + \mathbf{E}_4$
3. at most 10 (complex) solutions for x, y, z from the cubic constraints

- when all 3D points lie on a plane: at most 2 real solutions (twisted-pair)   can be disambiguated in 3 views or by chirality constraint ($\to$83) unless all 3D points are closer to one camera
- 6-point problem for unknown f [Kukelova et al. BMVC 2008]
- resources at `http://cmp.felk.cvut.cz/minimal/5_pt_relative.php`

▶The Triangulation Problem

Problem: Given cameras $\mathbf{P}_1$, $\mathbf{P}_2$ and a correspondence $x \leftrightarrow y$ compute a 3D point $\mathbf{X}$ projecting to x and y

$$\lambda_1\, \underline{\mathbf{x}} = \mathbf{P}_1 \underline{\mathbf{X}}, \qquad \lambda_2\, \underline{\mathbf{y}} = \mathbf{P}_2 \underline{\mathbf{X}}, \qquad \underline{\mathbf{x}} = \begin{bmatrix} u^1 \\ v^1 \\ 1 \end{bmatrix}, \qquad \underline{\mathbf{y}} = \begin{bmatrix} u^2 \\ v^2 \\ 1 \end{bmatrix}, \qquad \mathbf{P}_i = \begin{bmatrix} (\mathbf{p}_1^i)^\top \\ (\mathbf{p}_2^i)^\top \\ (\mathbf{p}_3^i)^\top \end{bmatrix}$$

Linear triangulation method

$$u^1\, (\mathbf{p}_3^1)^\top \underline{\mathbf{X}} = (\mathbf{p}_1^1)^\top \underline{\mathbf{X}}, \qquad u^2\, (\mathbf{p}_3^2)^\top \underline{\mathbf{X}} = (\mathbf{p}_1^2)^\top \underline{\mathbf{X}},$$

$$v^1\, (\mathbf{p}_3^1)^\top \underline{\mathbf{X}} = (\mathbf{p}_2^1)^\top \underline{\mathbf{X}}, \qquad v^2\, (\mathbf{p}_3^2)^\top \underline{\mathbf{X}} = (\mathbf{p}_2^2)^\top \underline{\mathbf{X}},$$

Gives

$$\mathbf{D}\underline{\mathbf{X}} = \mathbf{0}, \qquad \mathbf{D} = \begin{bmatrix} u^1\, (\mathbf{p}_3^1)^\top - (\mathbf{p}_1^1)^\top \\ v^1\, (\mathbf{p}_3^1)^\top - (\mathbf{p}_2^1)^\top \\ u^2\, (\mathbf{p}_3^2)^\top - (\mathbf{p}_1^2)^\top \\ v^2\, (\mathbf{p}_3^2)^\top - (\mathbf{p}_2^2)^\top \end{bmatrix}, \qquad \mathbf{D} \in \mathbb{R}^{4,4}, \quad \underline{\mathbf{X}} \in \mathbb{R}^4 \qquad (14)$$

- back-projected rays will generally not intersect due to image error, see next
- using Jack-knife (→63) not recommended — sensitive to small error
- we will use SVD (→90)
- but the result will not be invariant to projective frame
  replacing $\mathbf{P}_1 \mapsto \mathbf{P}_1\mathbf{H}$, $\mathbf{P}_2 \mapsto \mathbf{P}_2\mathbf{H}$ does not always result in $\underline{\mathbf{X}} \mapsto \mathbf{H}^{-1}\underline{\mathbf{X}}$
- note the homogeneous form in (14) can represent points $\underline{\mathbf{X}}$ at infinity

▶The Least-Squares Triangulation by SVD

- if $\mathbf{D}$ is full-rank we may minimize the algebraic least-squares error

$$\varepsilon^2(\underline{\mathbf{X}}) = \|\mathbf{D}\underline{\mathbf{X}}\|^2 \quad \text{s.t.} \quad \|\underline{\mathbf{X}}\| = 1, \qquad \underline{\mathbf{X}} \in \mathbb{R}^4$$

- let $\mathbf{D}_i$ be the i-th row of $\mathbf{D}$, then

$$\|\mathbf{D}\underline{\mathbf{X}}\|^2 = \sum_{i=1}^{4} (\mathbf{D}_i\,\underline{\mathbf{X}})^2 = \sum_{i=1}^{4} \underline{\mathbf{X}}^\top \mathbf{D}_i^\top \mathbf{D}_i\,\underline{\mathbf{X}} = \underline{\mathbf{X}}^\top \mathbf{Q}\,\underline{\mathbf{X}}, \text{ where } \mathbf{Q} = \sum_{i=1}^{4} \mathbf{D}_i^\top \mathbf{D}_i = \mathbf{D}^\top \mathbf{D} \in \mathbb{R}^{4,4}$$

- we write the SVD of $\mathbf{Q}$ as $\mathbf{Q} = \sum_{j=1}^{4} \sigma_j^2\,\mathbf{u}_j \mathbf{u}_j^\top$, in which [Golub & van Loan 2013, Sec. 2.5]

$$\sigma_1^2 \geq \cdots \geq \sigma_4^2 \geq 0 \quad \text{and} \quad \mathbf{u}_l^\top \mathbf{u}_m = \begin{cases} 0 & \text{if } l \neq m \\ 1 & \text{otherwise} \end{cases}$$

- then $\underline{\mathbf{X}} = \arg\min_{\mathbf{q}, \|\mathbf{q}\|=1} \mathbf{q}^\top \mathbf{Q}\,\mathbf{q} = \mathbf{u}_4$

Proof (by contradiction).

Let $\bar{\mathbf{q}} = \sum_{i=1}^{4} a_i \mathbf{u}_i$ s.t. $\sum_{i=1}^{4} a_i^2 = 1$, then $\|\bar{\mathbf{q}}\| = 1$, as desired, and

$$\bar{\mathbf{q}}^\top \mathbf{Q}\,\bar{\mathbf{q}} = \sum_{j=1}^{4} \sigma_j^2\,\bar{\mathbf{q}}^\top \mathbf{u}_j\,\mathbf{u}_j^\top \bar{\mathbf{q}} = \sum_{j=1}^{4} \sigma_j^2\,(\mathbf{u}_j^\top \bar{\mathbf{q}})^2 = \cdots = \sum_{j=1}^{4} a_j^2 \sigma_j^2 \geq \sum_{j=1}^{4} a_j^2 \sigma_4^2 = \sigma_4^2$$

□

- if $\sigma_4 \ll \sigma_3$, there is a unique solution $\underline{\mathbf{X}} = \mathbf{u}_4$ with residual error $(\mathbf{D}\,\underline{\mathbf{X}})^2 = \sigma_4^2$
  the quality (conditioning) of the solution may be expressed as $q = \sigma_3/\sigma_4$ (greater is better)

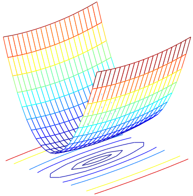
Matlab code for the least-squares solver:

```
[U,O,V] = svd(D);
X = V(:,end);
q = sqrt(O(end-1,end-1)/O(end,end));
```

⊛ P1; 1pt: Why did we decompose $\mathbf{D}$ and not $\mathbf{Q} = \mathbf{D}^\top\mathbf{D}$?

▶Numerical Conditioning

- The equation $\mathbf{D}\underline{\mathbf{X}} = \mathbf{0}$ in (14) may be ill-conditioned for numerical computation, which results in a poor estimate for $\underline{\mathbf{X}}$.



Why: on a row of $\mathbf{D}$ there are big entries together with small entries, e.g. of orders   projection centers in mm, image points in px

$$\begin{bmatrix} 10^3 & 0 & 10^3 & 10^6 \\ 0 & 10^3 & 10^3 & 10^6 \\ 10^3 & 0 & 10^3 & 10^6 \\ 0 & 10^3 & 10^3 & 10^6 \end{bmatrix}$$

Quick fix:

1. re-scale the problem by a regular diagonal conditioning matrix $\mathbf{S} \in \mathbb{R}^{4,4}$

$$\mathbf{0} = \mathbf{D}\,\underline{\mathbf{X}} = \mathbf{D}\,\mathbf{S}\,\mathbf{S}^{-1}\underline{\mathbf{X}} = \bar{\mathbf{D}}\,\bar{\underline{\mathbf{X}}}$$

   choose $\mathbf{S}$ to make the entries in $\hat{\mathbf{D}}$ all smaller than unity in absolute value:

$$\mathbf{S} = \mathrm{diag}(10^{-3}, 10^{-3}, 10^{-3}, 10^{-6})$$ `S = diag(1./max(abs(D), 1))`

2. solve for $\bar{\underline{\mathbf{X}}}$ as before
3. get the final solution as $\underline{\mathbf{X}} = \mathbf{S}\,\bar{\underline{\mathbf{X}}}$

- when SVD is used in camera resection, conditioning is essential for success →62

Algebraic Error vs Reprojection Error

- algebraic error (c – camera index, (u^c, v^c) – image coordinates) from SVD →91

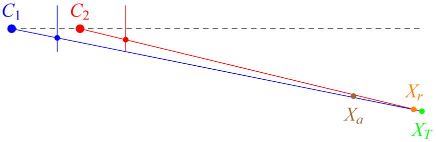
$$\varepsilon^2(\underline{\mathbf{X}}) = \sigma_4^2 = \sum_{c=1}^{2} \left[\left(u^c (\mathbf{p}_3^c)^\top \underline{\mathbf{X}} - (\mathbf{p}_1^c)^\top \underline{\mathbf{X}} \right)^2 + \left(v^c (\mathbf{p}_3^c)^\top \underline{\mathbf{X}} - (\mathbf{p}_2^c)^\top \underline{\mathbf{X}} \right)^2 \right]$$

- reprojection error

$$e^2(\underline{\mathbf{X}}) = \sum_{c=1}^{2} \left[\left(u^c - \frac{(\mathbf{p}_1^c)^\top \underline{\mathbf{X}}}{(\mathbf{p}_3^c)^\top \underline{\mathbf{X}}} \right)^2 + \left(v^c - \frac{(\mathbf{p}_2^c)^\top \underline{\mathbf{X}}}{(\mathbf{p}_3^c)^\top \underline{\mathbf{X}}} \right)^2 \right]$$

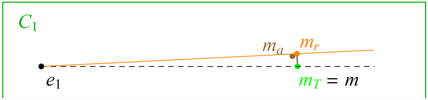
- algebraic error zero ⇔ reprojection error zero $\qquad \sigma_4 = 0 \Rightarrow$ non-trivial null space
- epipolar constraint satisfied ⇒ equivalent results
- in general: minimizing algebraic error is cheap but it gives inferior results
- minimizing reprojection error is expensive but it gives good results
- the midpoint of the common perpendicular to both optical rays gives about 50% greater error in 3D
- the golden standard method – deferred to →104

Ex:



- forward camera motion
- error $f/50$ in image 2, orthogonal to epipolar plane

X_T – noiseless ground truth position
X_r – reprojection error minimizer
X_a – algebraic error minimizer
m – measurement (m_T with noise in v^2)



►We Have Added to The ZOO (cont'd from →69)

problem	given	unknown	slide
camera resection	6 world–img correspondences $\left\{(X_i,\, m_i)\right\}_{i=1}^{6}$	$\mathbf{P}$	62
exterior orientation	$\mathbf{K}$, 3 world–img correspondences $\left\{(X_i,\, m_i)\right\}_{i=1}^{3}$	$\mathbf{R}$, $\mathbf{t}$	66
relative pointcloud orientation	3 world-world correspondences $\left\{(X_i,\, Y_i)\right\}_{i=1}^{3}$	$\mathbf{R}$, $\mathbf{t}$	70
fundamental matrix	7 img–img correspondences $\left\{(m_i,\, m_i')\right\}_{i=1}^{7}$	$\mathbf{F}$	84
relative camera orientation	$\mathbf{K}$, 5 img–img correspondences $\left\{(m_i,\, m_i')\right\}_{i=1}^{5}$	$\mathbf{R}$, $\mathbf{t}$	88
triangulation	$\mathbf{P}_1$, $\mathbf{P}_2$, 1 img–img correspondence $(m_i,\, m_i')$	X	89

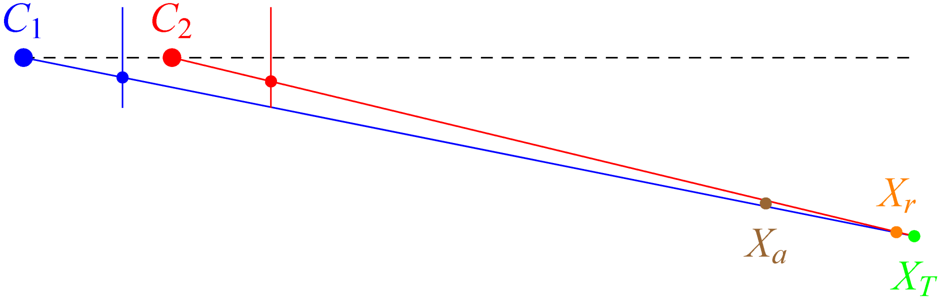
A bigger ZOO at http://cmp.felk.cvut.cz/minimal/

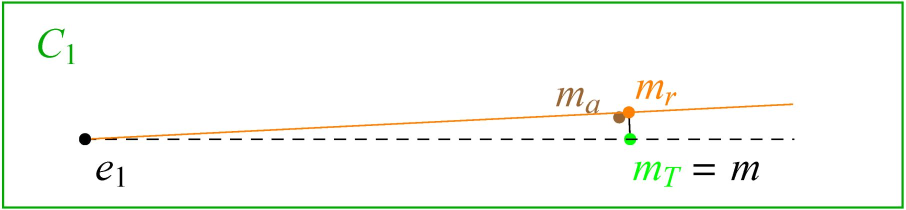
calibrated problems

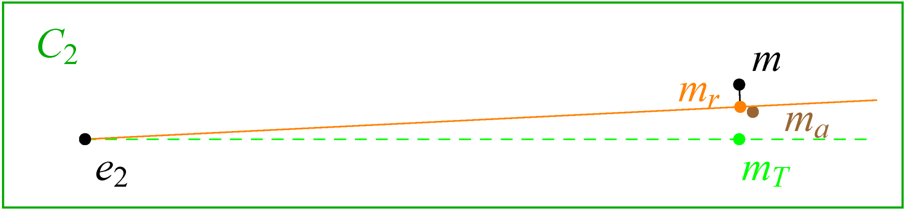
- have fewer degenerate configurations
- can do with fewer points (good for geometry proposal generators →117)

- algebraic error optimization (SVD) makes sense in camera resection and triangulation only
- but it is not the best method; we will now focus on 'optimizing optimally'

Thank You


C_1
C_2
X_r
X_a
X_T


C_1
m_a
m_r
e_1
$m_T = m$


C_2
m
m_r
m_a
e_2
m_T