# 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. January 5, 2021



Open Informatics Master's Course

# Module II

## **Perspective Camera**

**covered by**

    [H&Z] Secs: 2.1, 2.2, 3.1, 6.1, 6.2, 8.6, 2.5, Example: 2.19

## ▶Basic Geometric Entities, their Representation, and Notation

- entities have <u>names</u> and <u>representations</u>
- names and their components:

| entity | in 2-space | in 3-space |
|--------|------------|------------|
| point | $m = (u, v)$ | $X = (x, y, z)$ |
| line | $n$ | $O$ |
| plane | | $\pi,\ \varphi$ |

- associated vector representations

$$\mathbf{m} = \begin{bmatrix} u \\ v \end{bmatrix} = [u, v]^{\top}, \quad \mathbf{X} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{n}$$

will also be written in an 'in-line' form as $\mathbf{m} = (u, v)$, $\mathbf{X} = (x, y, z)$, etc.

- vectors are always meant to be columns $\mathbf{x} \in \mathbb{R}^{n \times 1}$
- associated homogeneous representations

$$\underline{\mathbf{m}} = [m_1, m_2, m_3]^{\top}, \quad \underline{\mathbf{X}} = [x_1, x_2, x_3, x_4]^{\top}, \quad \underline{\mathbf{n}}$$

'in-line' forms: $\underline{\mathbf{m}} = (m_1, m_2, m_3)$, $\underline{\mathbf{X}} = (x_1, x_2, x_3, x_4)$, etc.

- matrices are $\mathbf{Q} \in \mathbb{R}^{m \times n}$, linear map of a $\mathbb{R}^{n \times 1}$ vector is $\mathbf{y} = \mathbf{Q}\mathbf{x}$
- $j$-th element of vector $\mathbf{m}_i$ is $(\mathbf{m}_i)_j$; element $i, j$ of matrix $\mathbf{P}$ is $\mathbf{P}_{ij}$

# ▶Image Line (in 2D)

a finite line in the 2D $(u, v)$ plane $\qquad\qquad a\,u + b\,v + c = 0$

has a parameter (homogeneous) vector $\qquad\qquad \underline{\mathbf{n}} \simeq (a,\,b,\,c), \quad \|\underline{\mathbf{n}}\| \neq 0$

and there is an equivalence class for $\lambda \in \mathbb{R}, \lambda \neq 0 \quad (\lambda a,\,\lambda b,\,\lambda c) \simeq (a,\,b,\,c)$

## 'Finite' lines
- standard representative for <u>finite</u> $\underline{\mathbf{n}} = (n_1, n_2, n_3)$ is $\lambda\underline{\mathbf{n}}$, where $\lambda = \frac{\mathbf{1}}{\sqrt{n_1^2 + n_2^2}}$
  assuming $n_1^2 + n_2^2 \neq 0$; $\mathbf{1}$ is the unit, usually $\mathbf{1} = 1$

## 'Infinite' line
- we augment the set of lines for a special entity called the **line at infinity** (ideal line)

  $$\underline{\mathbf{n}}_\infty \simeq (0, 0, 1) \qquad \text{(standard representative)}$$

- the set of equivalence classes of vectors in $\mathbb{R}^3 \setminus (0, 0, 0)$ forms the projective space $\mathbb{P}^2$
  a set of rays →21

- line at infinity is a proper member of $\mathbb{P}^2$

- I may sometimes wrongly use $=$ instead of $\simeq$, if you are in doubt, ask me

## ►Image Point

Finite point $\mathbf{m} = (u, v)$ is incident on a finite line $\underline{\mathbf{n}} = (a, b, c)$ <u>iff</u>     iff = works either way!

$$a\,u + b\,v + c = 0$$

can be rewritten as (with scalar product):    $(u, v, \mathbf{1}) \cdot (a, b, c) = \underline{\mathbf{m}}^\top \underline{\mathbf{n}} = 0$

### 'Finite' points

- a finite point is <u>also</u> represented by a homogeneous vector $\underline{\mathbf{m}} \simeq (u, v, \mathbf{1})$,    $\|\underline{\mathbf{m}}\| \neq 0$
- the equivalence class for $\lambda \in \mathbb{R}$, $\lambda \neq 0$ is $(m_1, m_2, m_3) = \lambda\underline{\mathbf{m}} \simeq \underline{\mathbf{m}}$
- the standard representative for <u>finite</u> point $\underline{\mathbf{m}}$ is $\lambda\underline{\mathbf{m}}$, where $\lambda = \frac{1}{m_3}$   assuming $m_3 \neq 0$
- when $\mathbf{1} = 1$ then units are pixels and $\lambda\underline{\mathbf{m}} = (u, v, 1)$
- when $\mathbf{1} = f$ then all elements have a similar magnitude, $f \sim$ image diagonal

  use $\mathbf{1} = 1$ unless you know what you are doing;
  all entities participating in a formula must be expressed in the same units

### 'Infinite' points

- we augment for **points at infinity** (ideal points) $\underline{\mathbf{m}}_\infty \simeq (m_1, m_2, 0)$

  proper members of $\mathbb{P}^2$

- all such points lie on the line at infinity (ideal line) $\underline{\mathbf{n}}_\infty \simeq (0, 0, 1)$, i.e. $\underline{\mathbf{m}}_\infty^\top \underline{\mathbf{n}}_\infty = 0$

# ▶ Line Intersection and Point Join

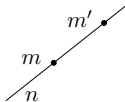The point of **intersection** $m$ of image lines $n$ and $n'$, $n \not\simeq n'$ is



$$\underline{\mathbf{m}} \simeq \underline{\mathbf{n}} \times \underline{\mathbf{n}}'$$
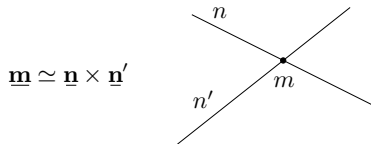
**proof:** If $\underline{\mathbf{m}} = \underline{\mathbf{n}} \times \underline{\mathbf{n}}'$ is the intersection point, it must be incident on both lines. Indeed, using known equivalences from vector algebra

$$\underline{\mathbf{n}}^\top \underbrace{(\underline{\mathbf{n}} \times \underline{\mathbf{n}}')}_{\underline{\mathbf{m}}} \equiv \underline{\mathbf{n}}'^\top \underbrace{(\underline{\mathbf{n}} \times \underline{\mathbf{n}}')}_{\underline{\mathbf{m}}} \equiv 0$$

The **join** $n$ of two image points $m$ and $m'$, $m \not\simeq m'$ is

$$\underline{\mathbf{n}} \simeq \underline{\mathbf{m}} \times \underline{\mathbf{m}}'$$



Parallel lines intersect (somewhere) on the line at infinity $\underline{\mathbf{n}}_\infty \simeq (0,0,1)$:
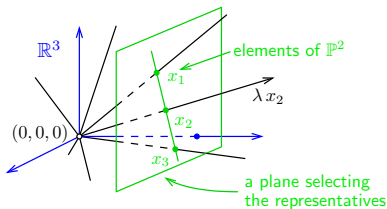
$$a\,u + b\,v + c = 0,$$
$$a\,u + b\,v + d = 0, \qquad\qquad d \neq c$$
$$(a,b,c) \times (a,b,d) \simeq (b,-a,0)$$

- all such intersections lie on $\underline{\mathbf{n}}_\infty$
- line at infinity therefore represents the set of (unoriented) directions in the plane
- Matlab: m = cross(n, n_prime);

$\mathbb{R}^3$

elements of $\mathbb{P}^2$

$x_1$

$\lambda\, x_2$

$(0,0,0)$

$x_2$

$x_3$

a plane selecting
the representatives

Projective plane $\mathbb{P}^2$: Vector space of dimension 3 <u>excluding the zero vector</u>, $\mathbb{R}^3 \setminus (0,0,0)$, factorized to linear equivalence classes ('rays'), $\underline{x} \simeq \lambda\underline{x},\ \lambda \neq 0$

including 'points at infinity'
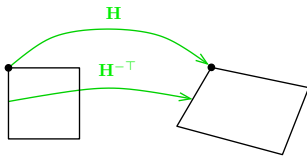
we call $\underline{x} \in \mathbb{P}^2$  'points'

**Homography in $\mathbb{P}^2$:** Non-singular linear mapping in $\mathbb{P}^2$      an analogic definition for $\mathbb{P}^3$

$$\underline{x}' \simeq \mathbf{H}\,\underline{x}, \quad \mathbf{H} \in \mathbb{R}^{3,3} \text{ non-singular}$$

**Defining properties**

- collinear points are mapped to collinear points

  lines of points are mapped to lines of points

- concurrent lines are mapped to concurrent lines

  concurrent = intersecting at a point

- <u>and</u> point-line incidence is preserved

  e.g. line intersection points mapped to line intersection points

- $\mathbf{H}$ is a $3 \times 3$ <u>non-singular</u> matrix, $\lambda\, \mathbf{H} \simeq \mathbf{H}$ equivalence class, 8 degrees of freedom
- homogeneous matrix representant: $\det \mathbf{H} = 1$
- what we call homography here is often called 'projective collineation' in mathematics

$$\underline{\mathbf{m}}' \simeq \mathbf{H}\,\underline{\mathbf{m}} \qquad \text{(image) point}$$

$$\underline{\mathbf{n}}' \simeq \mathbf{H}^{-\top}\underline{\mathbf{n}} \qquad \text{(image) line} \qquad \mathbf{H}^{-\top} = (\mathbf{H}^{-1})^{\top} = (\mathbf{H}^{\top})^{-1}$$

- incidence is preserved: $(\underline{\mathbf{m}}')^{\top}\underline{\mathbf{n}}' \simeq \underline{\mathbf{m}}^{\top}\mathbf{H}^{\top}\mathbf{H}^{-\top}\underline{\mathbf{n}} = \underline{\mathbf{m}}^{\top}\underline{\mathbf{n}} = 0$

**Mapping a finite 2D point $\mathbf{m} = (u, v)$ to $\underline{\mathbf{m}} = (u', v')$**

1. extend the Cartesian (pixel) coordinates to homogeneous coordinates, $\underline{\mathbf{m}} = (u, v, \mathbf{1})$
2. map by homography, $\underline{\mathbf{m}}' = \mathbf{H}\,\underline{\mathbf{m}}$
3. if $m_3' \neq 0$ convert the result $\underline{\mathbf{m}}' = (m_1', m_2', m_3')$ back to Cartesian coordinates (pixels),

$$u' = \frac{m_1'}{m_3'}\,\mathbf{1}, \qquad v' = \frac{m_2'}{m_3'}\,\mathbf{1}$$

- note that, typically, $m_3' \neq 1$          $m_3' = 1$ when $\mathbf{H}$ is affine
- an infinite point $\underline{\mathbf{m}} = (u, v, 0)$ maps the same way

# Some Homographic Tasters

**Rectification of camera rotation**: →59 (geometry), →128 (homography estimation)





$$\mathbf{H} \simeq \mathbf{K}\mathbf{R}^{\top}\mathbf{K}^{-1}$$     maps from image plane to facade plane

**Homographic Mouse for Visual Odometry**: [Mallis 2007]





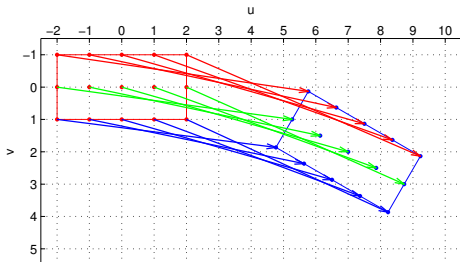illustrations courtesy of AMSL Racing Team, Meiji University and LIBVISO: Library for VISual Odometry

$$\mathbf{H} \simeq \mathbf{K}\left(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^{\top}}{d}\right)\mathbf{K}^{-1}$$     [H&Z, p. 327]

# ►Homography Subgroups: Euclidean Mapping (aka Rigid Motion)

- Euclidean mapping (EM): rotation, translation and their combination

$$\mathbf{H} = \begin{bmatrix} \cos\phi & -\sin\phi & t_x \\ \sin\phi & \cos\phi & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

- eigenvalues $\left(1,\, e^{-i\phi},\, e^{i\phi}\right)$



rotation by $30°$, then translation by $(7, 2)$

**EM = The most general homography preserving**

1. **areas**: $\det\mathbf{H} = 1 \implies$ unit Jacobian
2. **lengths**: Let $\underline{x}'_i = \mathbf{H}\underline{x}_i$ (check we can use $=$ instead of $\simeq$). Let $(x_i)_3 = 1$, Then
$$\|\underline{x}'_2 - \underline{x}'_1\| = \|\mathbf{H}\underline{x}_2 - \mathbf{H}\underline{x}_1\| = \|\mathbf{H}(\underline{x}_2 - \underline{x}_1)\| = \cdots = \|\underline{x}_2 - \underline{x}_1\|$$
3. **angles**    check the dot-product of normalized differences from a point $(\mathbf{x} - \mathbf{z})^\top(\mathbf{y} - \mathbf{z})$   (Cartesian(!))

- eigenvectors when $\phi \neq k\pi$, $k = 0, 1, \ldots$ (columnwise)

$$\mathbf{e}_1 \simeq \begin{bmatrix} t_x + t_y \cot\frac{\phi}{2} \\ t_y - t_x \cot\frac{\phi}{2} \\ 2 \end{bmatrix}, \quad \mathbf{e}_2 \simeq \begin{bmatrix} i \\ 1 \\ 0 \end{bmatrix}, \quad \m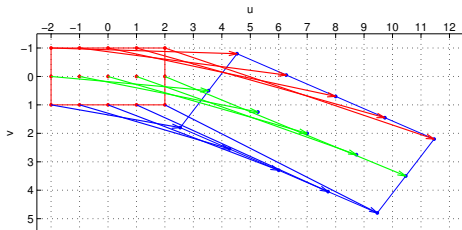athbf{e}_3 \simeq \begin{bmatrix} -i \\ 1 \\ 0 \end{bmatrix} \qquad \mathbf{e}_2, \mathbf{e}_3 \text{ – circular points, } i \text{ – imaginary unit}$$

4. **circular points**: points at infinity $(i, 1, 0)$, $(-i, 1, 0)$ (preserved even by similarity)
- **similarity**: scaled Euclidean mapping (does not preserve lengths, areas)

3D Computer Vision: II. Perspective Camera (p. 24/196) ↺↻↺    R. Šára, CMP; rev. 5–Jan–2021

rotation by $30°$
then scaling by $\mathrm{diag}(1, 1.5, 1)$
then translation by $(7, 2)$

$$\mathbf{H} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

**AM = The most general homography preserving**
- parallelism
- ratio of areas
- ratio of lengths on <u>parallel lines</u>
- linear combinations of vectors (e.g. midpoints)
- convex hull
- line at infinity $\underline{\mathbf{n}}_\infty$ (not pointwise)

**does not preserve**    observe $\mathbf{H}^\top \underline{\mathbf{n}}_\infty \simeq \begin{bmatrix} a_{11} & a_{21} & 0 \\ a_{12} & a_{22} & 0 \\ t_x & t_y & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \underline{\mathbf{n}}_\infty \quad \Rightarrow \quad \underline{\mathbf{n}}_\infty \simeq \mathbf{H}^{-\top} \underline{\mathbf{n}}_\infty$
- lengths
- angles
- areas
- circular points

Euclidean mappings preserve all properties affine mappings preserve, of course
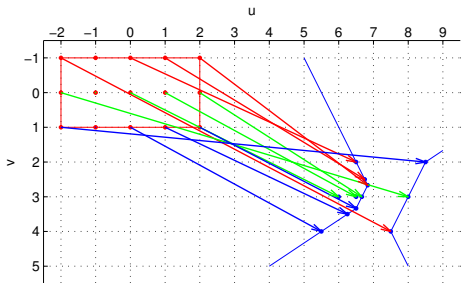
$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

**preserves only**

- incidence and concurrency
- collinearity
- cross-ratio on the line          →46

**does not preserve**

- lengths
- areas
- parallelism
- ratio of areas
- ratio of lengths
- linear combinations of vectors (midpoints, etc.)
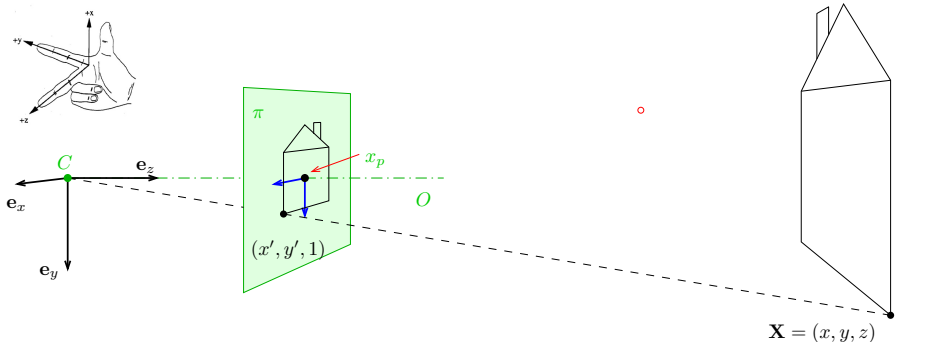- convex hull
- line at infinity $\underline{\mathbf{n}}_\infty$



$$\mathbf{H} = \begin{bmatrix} 7 & -0.5 & 6 \\ 3 & 1 & 3 \\ 1 & 0 & 1 \end{bmatrix}$$

line $\underline{\mathbf{n}} = (1, 0, 1)$ is mapped to $\underline{\mathbf{n}}_\infty$:  $\mathbf{H}^{-\top} \underline{\mathbf{n}} \simeq \underline{\mathbf{n}}_\infty$

(where in the picture is the line $n$?)

# ►Canonical Perspective Camera (Pinhole Camera, Camera Obscura)



1. in this picture we are looking 'down the street'
2. right-handed canonical coordinate system $(x, y, z)$ with unit vectors $\mathbf{e}_x$, $\mathbf{e}_y$, $\mathbf{e}_z$
3. origin = center of projection $C$
4. image plane $\pi$ at <u>unit</u> distance from $C$
5. optical axis $O$ is perpendicular to $\pi$
6. principal point $x_p$: intersection of $O$ and $\pi$
7. perspective camera is given by $C$ and $\pi$
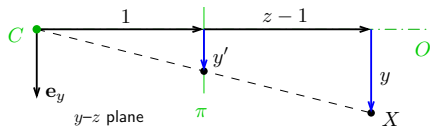


projected point in the natural image coordinate system:

$$\frac{y'}{1} = y' = \frac{y}{1 + z - 1} = \frac{y}{z}, \qquad x' = \frac{x}{z}$$
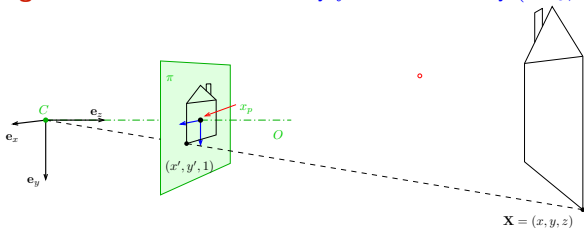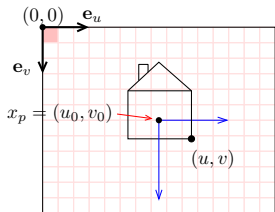
projected point **in canonical camera** $(z \neq 0)$

$$(x', y', 1) = \left(\frac{x}{z}, \frac{y}{z}, 1\right) = \frac{1}{z}(x, y, z) \simeq \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{P}_0 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{P}_0 \, \underline{\mathbf{X}}$$

projected point **in scanned image**                    scale by $f$ and translate by $(-u_0, -v_0)$



$$u = f\frac{x}{z} + u_0$$
$$v = f\frac{y}{z} + v_0$$

$$\frac{1}{z}\begin{bmatrix} f\,x + z\,u_0 \\ f\,y + z\,v_0 \\ z \end{bmatrix} \simeq \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{P}_0 \, \underline{\mathbf{X}} = \mathbf{P} \, \underline{\mathbf{X}}$$

- 'calibration' matrix $\mathbf{K}$ transforms <u>canonical</u> $\mathbf{P}_0$ to <u>standard</u> perspective camera $\mathbf{P}$

$$\underbrace{\begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx + u_0 z \\ fy + v_0 z \\ z \end{bmatrix} \simeq \underbrace{\begin{bmatrix} x + \frac{z}{f} u_0 \\ y + \frac{z}{f} v_0 \\ \frac{z}{f} \end{bmatrix}}_{(a)} \simeq \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \mathbf{\underline{m}}$$

$$\frac{m_1}{m_3} = \frac{f\,x}{z} + u_0 = u, \qquad \frac{m_2}{m_3} = \frac{f\,y}{z} + v_0 = v \quad \text{when} \quad m_3 \neq 0$$

$f$ – 'focal length' – converts length ratios to pixels, $\quad [f] = \text{px}, \; f > 0$

$(u_0, v_0)$ – principal point in pixels

**Perspective Camera:**

1. dimension reduction $\hfill$ since $\mathbf{P} \in \mathbb{R}^{3,4}$

2. nonlinear unit change $\mathbf{1} \mapsto \mathbf{1} \cdot z/f,$ $\quad$ see (a)
   for convenience we use $P_{11} = P_{22} = f$ rather than $P_{33} = 1/f$ and the $u_0, v_0$ in relative units

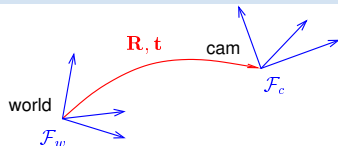3. $m_3 = 0$ represents points at infinity in image plane $\pi$ $\hfill$ i.e. points with $z = 0$

A transformation of a point from the world to camera coordinate system:

$$\mathbf{X}_c = \mathbf{R}\,\mathbf{X}_w + \mathbf{t}$$



$\mathbf{R}$ – camera rotation matrix — world orientation in the camera coordinate frame $\mathcal{F}_c$
$\mathbf{t}$ – camera translation vector — world origin in the camera coordinate frame $\mathcal{F}_c$

$$\mathbf{P}\,\underline{\mathbf{X}}_c = \mathbf{K}\mathbf{P}_0\begin{bmatrix}\mathbf{X}_c\\1\end{bmatrix} = \mathbf{K}\mathbf{P}_0\begin{bmatrix}\mathbf{R}\mathbf{X}_w+\mathbf{t}\\1\end{bmatrix} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\underbrace{\begin{bmatrix}\mathbf{R} & \mathbf{t}\\\mathbf{0}^\top & 1\end{bmatrix}}_{\mathbf{T}}\begin{bmatrix}\mathbf{X}_w\\1\end{bmatrix} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\underline{\mathbf{X}}_w$$

$\mathbf{P}_0$ (a $3 \times 4$ mtx) discards the last row of $\mathbf{T}$

- $\mathbf{R}$ is rotation, $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$, $\det\mathbf{R} = +1$      $\mathbf{I} \in \mathbb{R}^{3,3}$ identity matrix
- 6 **extrinsic parameters**: 3 rotation angles (Euler theorem), 3 translation components
- alternative, often used, camera representations

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix}$$

$\mathbf{C}$ – camera position in the world reference frame $\mathcal{F}_w$      $\mathbf{t} = -\mathbf{R}\mathbf{C}$
$\mathbf{r}_3^\top$ – optical axis in the world reference frame $\mathcal{F}_w$      third row of $\mathbf{R}$: $\mathbf{r}_3 = \mathbf{R}^{-1}[0,0,1]^\top$
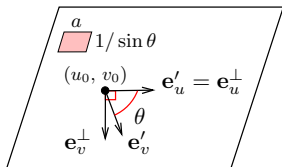
- we can save some conversion and computation by noting that $\mathbf{K}\mathbf{R}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix}\underline{\mathbf{X}} = \mathbf{K}\mathbf{R}(\mathbf{X} - \mathbf{C})$

**The general form of calibration matrix K includes**
- skew angle $\theta$ of the digitization raster
- pixel aspect ratio $a$



$$\mathbf{K} = \begin{bmatrix} a\,f & -a\,f\,\cot\theta & u_0 \\ 0 & f/\sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

units: $[f] = \mathrm{px}$, $[u_0] = \mathrm{px}$, $[v_0] = \mathrm{px}$, $[a] = 1$

⊛ H1; 2pt: Verify this **K**; deadline LD+2 wk

Hints:

1. image projects to orthogonal system $F^{\perp}$, then it maps by skew to $F'$, then by scale $a\,f$, $f$ to $F''$, then by translation by $u_0$, $v_0$ to $F'''$

2. Skew: Express point $\mathbf{x}$ as
$$\mathbf{x} = u'\mathbf{e}_{u'} + v'\mathbf{e}_{v'} = u^{\perp}\mathbf{e}_u^{\perp} + v^{\perp}\mathbf{e}_v^{\perp}$$
   $\mathbf{e}_{:}$ are unit basis vectors

3. **K** maps from $F^{\perp}$ to $F'''$ as
$$w'''\,[u''', v''', 1]^{\top} = \mathbf{K}[u^{\perp}, v^{\perp}, 1]^{\top}$$

4. figure drawn 'after the transformation' but in $F^{\perp}$, pink polygon is a unit square mapped by **K**

$$\underline{\mathbf{m}} \simeq \mathbf{P}\underline{\mathbf{X}}, \qquad \mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{q} \end{bmatrix} \simeq \mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I} & -\mathbf{C} \end{bmatrix}$$

a recipe for filling $\mathbf{P}$

**general finite perspective camera has 11 parameters:**

- 5 intrinsic parameters: $f$, $u_0$, $v_0$, $a$, $\theta$    finite camera: $\det \mathbf{K} \neq 0$
- 6 extrinsic parameters: $\mathbf{t}$, $\mathbf{R}(\alpha, \beta, \gamma)$

Representation Theorem: The set of projection matrices $\mathbf{P}$ of finite perspective cameras is isomorphic to the set of homogeneous $3 \times 4$ matrices with the left $3 \times 3$ submatrix $\mathbf{Q}$ non-singular.

random finite camera: `Q = rand(3,3); while det(Q)==0, Q = rand(3,3); end, P = [Q, rand(3,1)];`

# ►Projection Matrix Decomposition

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{q} \end{bmatrix} \quad \longrightarrow \quad \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$$

$\mathbf{Q} \in \mathbb{R}^{3,3}$    <u>full rank</u>    (if finite perspective camera; see [H&Z, Sec. 6.3] for cameras at infinity)

$\mathbf{K} \in \mathbb{R}^{3,3}$    upper triangular with <u>positive diagonal elements</u>

$\mathbf{R} \in \mathbb{R}^{3,3}$    <u>rotation</u>:    $\mathbf{R}^\top \mathbf{R} = \mathbf{I}$ and $\det \mathbf{R} = +1$

1. $\begin{bmatrix} \mathbf{Q} & \mathbf{q} \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{KR} & \mathbf{Kt} \end{bmatrix}$      also →35

2. RQ decomposition of $\mathbf{Q} = \mathbf{KR}$ using three Givens rotations      [H&Z, p. 579]

$$\mathbf{K} = \mathbf{Q} \underbrace{\mathbf{R}_{32}\mathbf{R}_{31}\mathbf{R}_{21}}_{\mathbf{R}^{-1}} \qquad \mathbf{QR}_{32} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot \end{bmatrix}, \; \mathbf{QR}_{32}\mathbf{R}_{31} = \begin{bmatrix} \cdot & \cdot & \cdot \\ 0 & 0 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}, \; \mathbf{QR}_{32}\mathbf{R}_{31}\mathbf{R}_{21} = \begin{bmatrix} \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot \\ 0 & 0 & \cdot \end{bmatrix}$$

$\mathbf{R}_{ij}$ zeroes element $ij$ in $\mathbf{Q}$ affecting only columns $i$ and $j$ and the sequence preserves previously zeroed elements, e.g. (see next slide for derivation details)

$$\mathbf{R}_{32} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix} \text{ gives } \quad \begin{matrix} c^2 + s^2 = 1 \\ 0 = k_{32} = c\,q_{32} + s\,q_{33} \end{matrix} \Rightarrow c = \frac{q_{33}}{\sqrt{q_{32}^2 + q_{33}^2}} \quad s = \frac{-q_{32}}{\sqrt{q_{32}^2 + q_{33}^2}}$$

⊛ P1; 1pt: Multiply known matrices $\mathbf{K}$, $\mathbf{R}$ and then decompose back; discuss numerical errors

- RQ decomposition nonuniqueness: $\mathbf{KR} = \mathbf{KT}^{-1}\mathbf{TR}$, where $\mathbf{T} = \mathrm{diag}(-1, -1, 1)$ is also a rotation, we must correct the result so that the diagonal elements of $\mathbf{K}$ are all positive

           'thin' RQ decomposition

- care must be taken to avoid overflow, see [Golub & van Loan 2013, sec. 5.2]

## RQ Decomposition Step

```
Q = Array [q_{#1,#2} &, {3, 3}];
R32 = {{1, 0, 0}, {0, c, -s}, {0, s, c}}; R32 // MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{pmatrix}$$

```
Q1 = Q.R32 ; Q1 // MatrixForm
```

$$\begin{pmatrix} q_{1,1} & c\,q_{1,2} + s\,q_{1,3} & -s\,q_{1,2} + c\,q_{1,3} \\ q_{2,1} & c\,q_{2,2} + s\,q_{2,3} & -s\,q_{2,2} + c\,q_{2,3} \\ q_{3,1} & c\,q_{3,2} + s\,q_{3,3} & -s\,q_{3,2} + c\,q_{3,3} \end{pmatrix}$$

```
s1 = Solve [{Q1[[3]][[2]] == 0, c^2 + s^2 == 1}, {c, s}][[2]]
```

$$\left\{ c \rightarrow \frac{q_{3,3}}{\sqrt{q_{3,2}^2 + q_{3,3}^2}}, \; s \rightarrow -\frac{q_{3,2}}{\sqrt{q_{3,2}^2 + q_{3,3}^2}} \right\}$$

```
Q1 /. s1 // Simplify // MatrixForm
```

$$\begin{pmatrix} q_{1,1} & \frac{-q_{1,3}\,q_{3,2} + q_{1,2}\,q_{3,3}}{\sqrt{q_{3,2}^2 + q_{3,3}^2}} & \frac{q_{1,2}\,q_{3,2} + q_{1,3}\,q_{3,3}}{\sqrt{q_{3,2}^2 + q_{3,3}^2}} \\[2ex] q_{2,1} & \frac{-q_{2,3}\,q_{3,2} + q_{2,2}\,q_{3,3}}{\sqrt{q_{3,2}^2 + q_{3,3}^2}} & \frac{q_{2,2}\,q_{3,2} + q_{2,3}\,q_{3,3}}{\sqrt{q_{3,2}^2 + q_{3,3}^2}} \\[2ex] q_{3,1} & 0 & \sqrt{q_{3,2}^2 + q_{3,3}^2} \end{pmatrix}$$

## ►Center of Projection (Optical Center)

**Observation:** finite $\mathbf{P}$ has a non-trivial right null-space <span style="float:right">rank 3 but 4 columns</span>

### Theorem
*Let $\mathbf{P}$ be a camera and let there be $\underline{\mathbf{B}} \neq \mathbf{0}$ s.t. $\mathbf{P}\,\underline{\mathbf{B}} = \mathbf{0}$. Then $\underline{\mathbf{B}}$ is equivalent to the projection center $\underline{\mathbf{C}}$ (homogeneous, in world coordinate frame).*

### Proof.
1. Consider spatial line $AB$ ($B$ is given, $A \neq B$). We can write

$$\underline{\mathbf{X}}(\lambda) \simeq \lambda\,\underline{\mathbf{A}} + (1-\lambda)\,\underline{\mathbf{B}}, \qquad \lambda \in \mathbb{R}$$



2. it projects to

$$\mathbf{P}\underline{\mathbf{X}}(\lambda) \simeq \lambda\,\mathbf{P}\,\underline{\mathbf{A}} + (1-\lambda)\,\mathbf{P}\,\underline{\mathbf{B}} \simeq \mathbf{P}\,\underline{\mathbf{A}}$$

- the entire line projects to a single point $\Rightarrow$ it must pass through the projection center of $\mathbf{P}$
- this holds for any choice of $A \neq B \Rightarrow$ the only common point of the lines is the $C$, i.e. $\underline{\mathbf{B}} \simeq \underline{\mathbf{C}}$ □
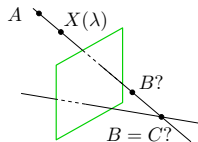
Hence

$$\mathbf{0} = \mathbf{P}\,\underline{\mathbf{C}} = \begin{bmatrix} \mathbf{Q} & \mathbf{q} \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ 1 \end{bmatrix} = \mathbf{Q}\,\mathbf{C} + \mathbf{q} \;\Rightarrow\; \boxed{\mathbf{C} = -\mathbf{Q}^{-1}\mathbf{q}}$$

$\underline{\mathbf{C}} = (c_j)$, where $c_j = (-1)^j \det \mathbf{P}^{(j)}$, in which $\mathbf{P}^{(j)}$ is $\mathbf{P}$ with column $j$ dropped
Matlab: `C_homo = null(P);` or `C = -Q\q;`

Optical ray: Spatial line that projects to a single image point.

1. consider the following line

   $\mathbf{d}$ unit line direction vector, $\|\mathbf{d}\| = 1$, $\lambda \in \mathbb{R}$, Cartesian representation

   $$\mathbf{X}(\lambda) = \mathbf{C} + \lambda\,\mathbf{d}$$

2. the projection of the (finite) point $X(\lambda)$ is

   $$\underline{\mathbf{m}} \simeq \begin{bmatrix} \mathbf{Q} & \mathbf{q} \end{bmatrix} \begin{bmatrix} \mathbf{X}(\lambda) \\ 1 \end{bmatrix} = \mathbf{Q}(\mathbf{C} + \lambda\mathbf{d}) + \mathbf{q} = \lambda\,\mathbf{Q}\,\mathbf{d} =$$

   $$= \lambda \begin{bmatrix} \mathbf{Q} & \mathbf{q} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ 0 \end{bmatrix}$$

   . . . which is also the image of a point at infinity in $\mathbb{P}^3$

- optical ray line corresponding to image point $m$ is the set

   $$\mathbf{X}(\mu) = \mathbf{C} + \mu\,\mathbf{Q}^{-1}\underline{\mathbf{m}}, \qquad \mu \in \mathbb{R} \qquad (\mu = 1/\lambda)$$

- optical ray direction may be represented by a point at infinity $(\mathbf{d}, 0)$ in $\mathbb{P}^3$
- optical ray is expressed in world coordinate frame
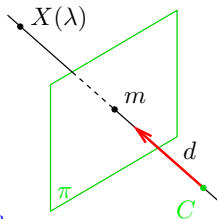
# ►Optical Axis

Optical axis: Optical ray that is perpendicular to image plane $\pi$

1. points $X$ on a given line $N$ parallel to $\pi$ project to a
   point at infinity $(u, v, 0)$ in $\pi$:

$$\begin{bmatrix} u \\ v \\ 0 \end{bmatrix} \simeq \mathbf{P}\underline{\mathbf{X}} = \begin{bmatrix} \mathbf{q}_1^\top & q_{14} \\ \mathbf{q}_2^\top & q_{24} \\ \mathbf{q}_3^\top & q_{34} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$



2. therefore the set of points $X$ is parallel to $\pi$ iff

$$\mathbf{q}_3^\top \mathbf{X} + q_{34} = 0$$

3. this is a plane with $\pm\mathbf{q}_3$ as the normal vector
4. <u>optical axis direction:</u> substitution $\mathbf{P} \mapsto \lambda\mathbf{P}$ must not change the direction
5. we select (assuming $\det(\mathbf{R}) > 0$)

$$\boxed{\mathbf{o} = \det(\mathbf{Q})\,\mathbf{q}_3}$$

if $\mathbf{P} \mapsto \lambda\mathbf{P}$ then $\det(\mathbf{Q}) \mapsto \lambda^3 \det(\mathbf{Q})$ and $\mathbf{q}_3 \mapsto \lambda\,\mathbf{q}_3$                    [H&Z, p. 161]

• the axis is expressed in world coordinate frame
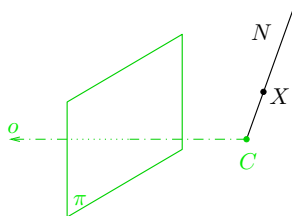
Principal point: The intersection of image plane and the optical axis

1. as we saw, $\mathbf{q}_3$ is the directional vector of optical axis
2. we take point at infinity on the optical axis that must project to the principal point $m_0$



3. then

$$\underline{\mathbf{m}}_0 \simeq \begin{bmatrix} \mathbf{Q} & \mathbf{q} \end{bmatrix} \begin{bmatrix} \mathbf{q}_3 \\ 0 \end{bmatrix} = \mathbf{Q}\,\mathbf{q}_3$$
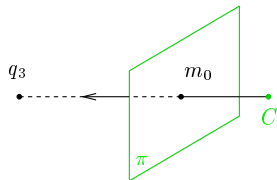
principal point:     $\boxed{\underline{\mathbf{m}}_0 \simeq \mathbf{Q}\,\mathbf{q}_3}$

• principal point is also the center of radial distortion

A spatial plane with normal $p$ containing the projection center $C$ and a given image line $n$.
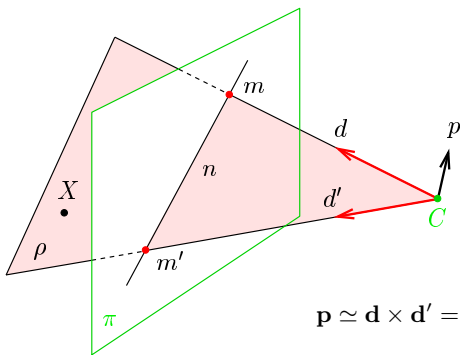


optical ray given by $m$    $\mathbf{d} \simeq \mathbf{Q}^{-1}\underline{\mathbf{m}}$

optical ray given by $m'$    $\mathbf{d}' \simeq \mathbf{Q}^{-1}\underline{\mathbf{m}}'$

$$\mathbf{p} \simeq \mathbf{d} \times \mathbf{d}' = (\mathbf{Q}^{-1}\underline{\mathbf{m}}) \times (\mathbf{Q}^{-1}\underline{\mathbf{m}}') = \mathbf{Q}^{\top}(\underline{\mathbf{m}} \times \underline{\mathbf{m}}') = \mathbf{Q}^{\top}\underline{\mathbf{n}}$$

• note the way $\mathbf{Q}$ factors out!

hence, $0 = \mathbf{p}^{\top}(\mathbf{X} - \mathbf{C}) = \underline{\mathbf{n}}^{\top} \underbrace{\mathbf{Q}(\mathbf{X} - \mathbf{C})}_{\rightarrow 30} = \underline{\mathbf{n}}^{\top}\mathbf{P}\underline{\mathbf{X}} = (\mathbf{P}^{\top}\underline{\mathbf{n}})^{\top}\underline{\mathbf{X}}$   for every $X$ in plane $\rho$

optical plane is given by $n$:    $\boxed{\underline{\boldsymbol{\rho}} \simeq \mathbf{P}^{\top}\underline{\mathbf{n}}}$      $\rho_1\,x + \rho_2\,y + \rho_3\,z + \rho_4 = 0$

optical plane normal given by $n$ $\qquad$ $\mathbf{p} = \mathbf{Q}^\top \underline{\mathbf{n}}$

optical plane normal given by $n'$ $\qquad$ $\mathbf{p}' = \mathbf{Q}^\top \underline{\mathbf{n}}'$

$$\mathbf{d} = \mathbf{p} \times \mathbf{p}' = (\mathbf{Q}^\top \underline{\mathbf{n}}) \times (\mathbf{Q}^\top \underline{\mathbf{n}}') = \mathbf{Q}^{-1}(\underline{\mathbf{n}} \times \underline{\mathbf{n}}') = \mathbf{Q}^{-1}\underline{\mathbf{m}}$$

General (finite) camera

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1^\top & q_{14} \\ \mathbf{q}_2^\top & q_{24} \\ \mathbf{q}_3^\top & q_{34} \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} = \mathbf{K}\,\mathbf{R} \begin{bmatrix} \mathbf{I} & -\mathbf{C} \end{bmatrix}$$

$\underline{\mathbf{C}} \simeq \mathrm{rnull}(\mathbf{P}), \quad \mathbf{C} = -\mathbf{Q}^{-1}\mathbf{q}$ ............... projection center (world coords.) →35

$\mathbf{d} = \mathbf{Q}^{-1}\,\underline{\mathbf{m}}$ ............... optical ray direction (world coords.) →36

$\mathbf{o} = \det(\mathbf{Q})\,\mathbf{q}_3$ ............... outward optical axis (world coords.) →37

$\underline{\mathbf{m}}_0 \simeq \mathbf{Q}\,\mathbf{q}_3$ ............... principal point (in image plane) →38

$\underline{\boldsymbol{\rho}} = \mathbf{P}^\top\,\underline{\mathbf{n}}$ ............... optical plane (world coords.) →39

$\mathbf{K} = \begin{bmatrix} a\,f & -a\,f\,\cot\theta & u_0 \\ 0 & f/\sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ ...... camera (calibration) matrix ($f$, $u_0$, $v_0$ in pixels) →31

$\mathbf{R}$ ............... camera rotation matrix (cam coords.) →30

$\mathbf{t}$ ............... camera translation vector (cam coords.) →30

# What Can We Do with An 'Uncalibrated' Perspective Camera?



How far is the engine?

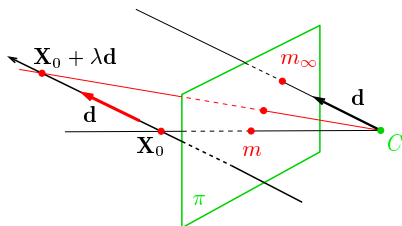<span style="color:blue">distance between sleepers (ties) 0.806m but we cannot count them, the image resolution is too low</span>

We will review some life-saving theory...
...and build a bit of geometric intuition...

**In fact**

- 'uncalibrated' = the image contains a 'calibrating object' that suffices for the task at hand

**Vanishing point:** the limit of the projection of a point that moves along a space line infinitely in one direction. the image of the point at infinity on the line



$$\underline{\mathbf{m}}_\infty \simeq \lim_{\lambda \to \pm\infty} \mathbf{P} \begin{bmatrix} \mathbf{X}_0 + \lambda \mathbf{d} \\ 1 \end{bmatrix} = \cdots \simeq \mathbf{Q}\,\mathbf{d}$$

⊛ P1; 1pt: Prove (use Cartesian coordinates and L'Hôpital's rule)

- the V.P. of a spatial line with directional vector $\mathbf{d}$ is $\underline{\mathbf{m}}_\infty \simeq \mathbf{Q}\,\mathbf{d}$
- V.P. is independent on line position $\mathbf{X}_0$, it depends on its directional vector only
- all parallel (world) lines share the same (image) V.P., including the optical ray defined by $m_\infty$

where is the sun?

what is the wind direction?
(must have video)

fly above the lane,
at constant altitude!

Vanishing line: The set of vanishing points of all lines in a plane

the image of the line at infinity in the plane
and in all parallel planes



$n$ — plane normal

$v_1$          $v_2$

$m$ — line orientation vector
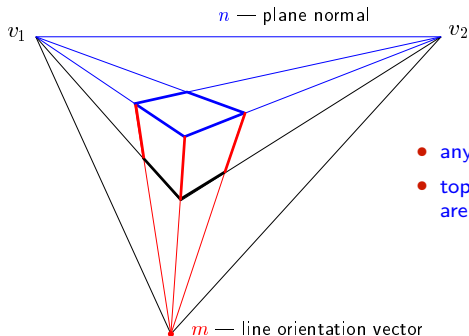
- any box with parallel edges
- top (blue) and bottom (black) box planes are parallel, hence they share V.L. $n$

- V.L. $n$ corresponds to spatial plane of normal vector $\mathbf{p} = \mathbf{Q}^\top \underline{\mathbf{n}}$

because this is the normal vector of a parallel optical plane (!) →39

- a spatial plane of normal vector $\mathbf{p}$ has a V.L. represented by    $\underline{\mathbf{n}} = \mathbf{Q}^{-\top}\mathbf{p}$.

# ►Cross Ratio

Four distinct collinear spatial points $R, S, T, U$ define cross-ratio

$$[RSTU] = \frac{|\overrightarrow{RT}|}{|\overrightarrow{SR}|} \frac{|\overrightarrow{US}|}{|\overrightarrow{TU}|}$$

a mnemonic $(\infty)$
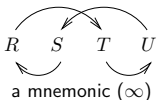


$|\overrightarrow{RT}|$ – signed distance from $R$ to $T$ in the arrow direction

6 cross-ratios from four points:

$[SRUT] = [RSTU]$, $[RSUT] = \dfrac{1}{[RSTU]}$, $[RTSU] = 1 - [RSTU]$, $\cdots$

$v \notin n$
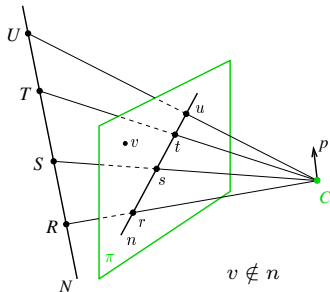
**Obs:** $[RSTU] = \dfrac{|\underline{\mathbf{r}} \ \underline{\mathbf{t}} \ \underline{\mathbf{v}}|}{|\underline{\mathbf{s}} \ \underline{\mathbf{r}} \ \underline{\mathbf{v}}|} \cdot \dfrac{|\underline{\mathbf{u}} \ \underline{\mathbf{s}} \ \underline{\mathbf{v}}|}{|\underline{\mathbf{t}} \ \underline{\mathbf{u}} \ \underline{\mathbf{v}}|}$, $\quad |\underline{\mathbf{r}} \ \underline{\mathbf{t}} \ \underline{\mathbf{v}}| = \det\left[\underline{\mathbf{r}} \ \underline{\mathbf{t}} \ \underline{\mathbf{v}}\right] = (\underline{\mathbf{r}} \times \underline{\mathbf{t}})^{\top} \underline{\mathbf{v}}$ (1)

## Corollaries:

- cross ratio is invariant under homographies $\underline{\mathbf{x}}' \simeq \mathbf{H}\underline{\mathbf{x}}$   plug $\mathbf{H}\underline{\mathbf{x}}$ in (1): $(\mathbf{H}^{-\top}(\underline{\mathbf{r}} \times \underline{\mathbf{t}}))^{\top} \mathbf{H}\underline{\mathbf{v}}$
- cross ratio is invariant under perspective projection: $[RSTU] = [r\,s\,t\,u]$

- 4 collinear points: any perspective camera will "see" the same cross-ratio of their images
- we measure the same cross-ratio in image as on the world line
- one of the points $R, S, T, U$ may be at infinity (we take the limit, in effect $\frac{\infty}{\infty} = 1$)

The 1-D projective coordinate of a point $P$ is defined by the following cross-ratio:

$$[P] = [P_0 \, P_1 \, P \, P_\infty] = [p_0 \, p_1 \, p \, p_\infty] = \frac{|\overrightarrow{p_0 \, p}|}{|\overrightarrow{p_1 \, p_0}|} \, \frac{|\overrightarrow{p_\infty \, p_1}|}{|\overrightarrow{p \, p_\infty}|} = [p]$$
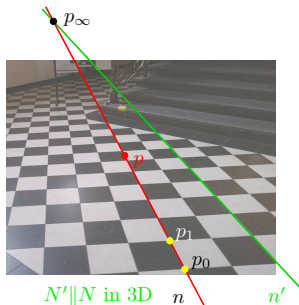


$p_0 \quad p_1 \quad p \quad p_\infty$
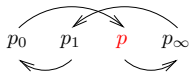
naming convention:

$P_0$ – the origin $\qquad\qquad [P_0] = 0$

$P_1$ – the unit point $\qquad\qquad [P_1] = 1$

$P_\infty$ – the supporting point $\qquad [P_\infty] = \pm\infty$

$$[P] = [p]$$

$[P]$ is equal to Euclidean coordinate along $N$
$[p]$ is its measurement in the image plane



$N' \| N$ in 3D $\quad n \qquad\qquad n'$

**Applications**

- Given the image of a 3D line $N$, the origin, the unit point, and the vanishing point, then the Euclidean coordinate of any point $P \in N$ can be determined $\qquad$ →48
- Finding v.p. of a line through a regular object $\qquad$ →49
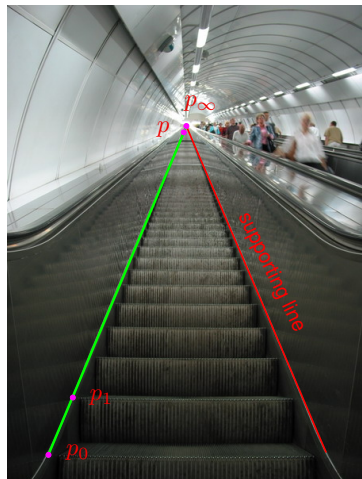
- Namesti Miru underground station in Prague
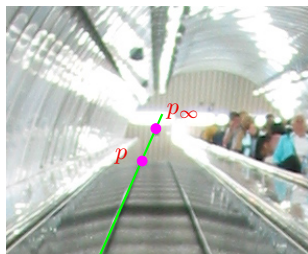


detail around the vanishing point

**Result:** $[P] = 214$ steps (correct answer is 216 steps)

4Mpx camera
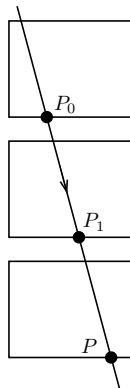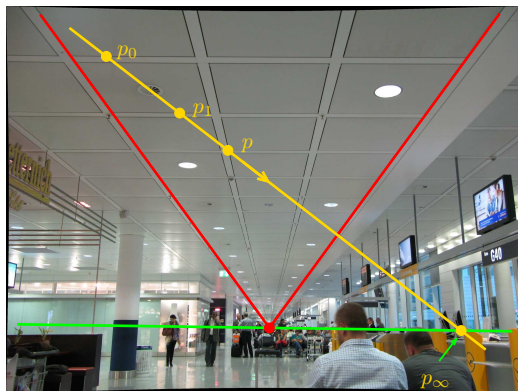
in 3D: $|P_0P| = 2|P_0P_1|$ then $\quad$ [H&Z, p. 218]

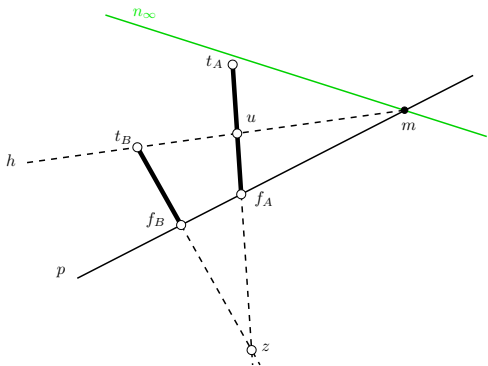$$[P_0P_1PP_\infty] = \frac{|P_0P|}{|P_1P_0|} = 2 \quad \Rightarrow \quad x_\infty = \frac{x_0\,(2\,x - x_1) - x\,x_1}{x + x_0 - 2\,x_1}$$
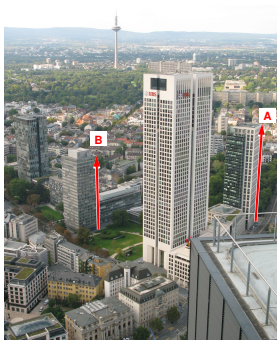
- $x$ – 1D coordinate along the yellow line, positive in the arrow direction
- could be applied to counting steps ($\rightarrow$48) if there was no supporting line

⊛ P1; 1pt: How high is the camera above the floor?

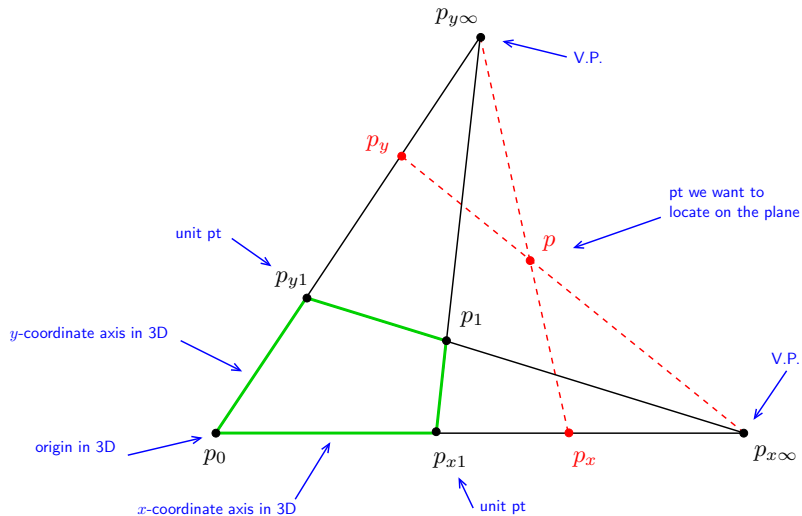⊛ H2; 3pt: What is the ratio of heights of Building A to Building B?

- expected: conceptual solution; use notation from this figure
- deadline: LD+2 weeks



**Hints**

1. What are the interesting properties of line $h$ connecting the top $t_B$ of Buiding B with the point $m$ at which the horizon intersects the line $p$ joining the foots $f_A$, $f_B$ of both buildings? [1 point]
2. How do we actually get the horizon $n_\infty$? (we do not see it directly, there are some hills there...) [1 point]
3. Give the formula for measuring the length ratio. [formula = 1 point]

$$[P_x] = [P_0 \ P_{x1} \ P_x \ P_{x\infty}] \qquad\qquad [P_y] = [P_0 \ P_{y1} \ P_y \ P_{y\infty}]$$

# Application: Measuring on the Floor (Wall, etc)



San Giovanni in Laterano, Rome

- measuring distances on the floor in terms of tile units
- what are the dimensions of the seal? Is it circular (assuming square tiles)?
- needs no explicit camera calibration

  because we can see the calibrating object (vanishing points)

# **Computing with a Single Camera**

**3.1** Calibration: Internal Camera Parameters from Vanishing Points and Lines

**3.2** Camera Resection: Projection Matrix from 6 Known Points

**3.3** Exterior Orientation: Camera Rotation and Translation from 3 Known Points

**3.4** Relative Orientation Problem: Rotation and Translation between Two Point Sets

**covered by**

[1] [H&Z] Secs: 8.6, 7.1, 22.1

[2] Fischler, M.A. and Bolles, R.C . Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24(6):381–395, 1981

[3] [Golub & van Loan 2013, Sec. 2.5]

# Obtaining Vanishing Points and Lines

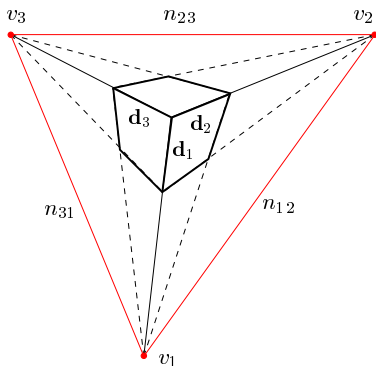- orthogonal direction pairs can be collected from more images by camera rotation
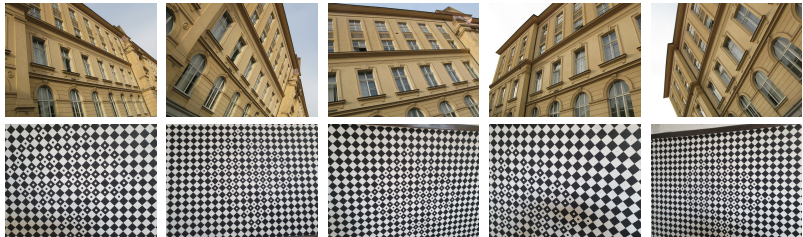


- vanishing line can be obtained from vanishing points and/or regularities (→49)
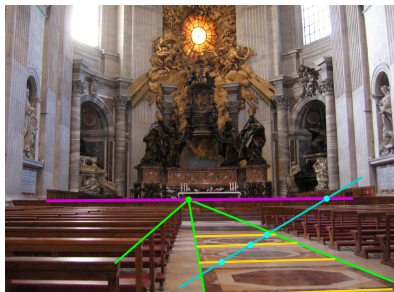
# ►Camera Calibration from Vanishing Points and Lines

**Problem:** Given finite vanishing points and/or vanishing lines, compute $\mathbf{K}$



$$\mathbf{d}_i = \lambda_i \mathbf{Q}^{-1} \underline{\mathbf{v}}_i, \qquad\qquad i = 1, 2, 3 \quad \rightarrow 43$$

$$\mathbf{p}_{ij} = \mu_{ij} \mathbf{Q}^{\top} \underline{\mathbf{n}}_{ij}, \quad i, j = 1, 2, 3, \ i \neq j \quad \rightarrow 39 \qquad (2)$$

- simple method: solve (2) after eliminating $\lambda_i$, $\mu_{ij}$.

**Special Configurations**

1. orthogonal rays $\mathbf{d}_1 \perp \mathbf{d}_2$ in space then

$$0 = \mathbf{d}_1^{\top} \mathbf{d}_2 = \underline{\mathbf{v}}_1^{\top} \mathbf{Q}^{-\top} \mathbf{Q}^{-1} \underline{\mathbf{v}}_2 = \underline{\mathbf{v}}_1^{\top} \underbrace{(\mathbf{K}\mathbf{K}^{\top})^{-1}}_{\boldsymbol{\omega} \text{ (IAC)}} \underline{\mathbf{v}}_2$$

2. orthogonal planes $\mathbf{p}_{ij} \perp \mathbf{p}_{ik}$ in space

$$0 = \mathbf{p}_{ij}^{\top} \mathbf{p}_{ik} = \underline{\mathbf{n}}_{ij}^{\top} \mathbf{Q}\mathbf{Q}^{\top} \underline{\mathbf{n}}_{ik} = \underline{\mathbf{n}}_{ij}^{\top} \boldsymbol{\omega}^{-1} \underline{\mathbf{n}}_{ik}$$

3. orthogonal ray and plane $\mathbf{d}_k \parallel \mathbf{p}_{ij}$, $k \neq i, j$        *normal parallel to optical ray*

$$\mathbf{p}_{ij} \simeq \mathbf{d}_k \quad \Rightarrow \quad \mathbf{Q}^{\top} \underline{\mathbf{n}}_{ij} = \frac{\lambda_i}{\mu_{ij}} \mathbf{Q}^{-1} \underline{\mathbf{v}}_k \quad \Rightarrow \quad \underline{\mathbf{n}}_{ij} = \varkappa \mathbf{Q}^{-\top} \mathbf{Q}^{-1} \underline{\mathbf{v}}_k = \varkappa \boldsymbol{\omega} \underline{\mathbf{v}}_k, \quad \varkappa \neq 0$$

- $n_{ij}$ may be constructed from non-orthogonal $v_i$ and $v_j$, e.g. using the cross-ratio
- $\boldsymbol{\omega}$ is a symmetric, positive definite $3 \times 3$ matrix        IAC = Image of Absolute Conic

| | configuration | equation | # constraints |
|---|---|---|---|
| (3) | orthogonal v.p. | $\underline{\mathbf{v}}_i^\top \boldsymbol{\omega} \, \underline{\mathbf{v}}_j = 0$ | 1 |
| (4) | orthogonal v.l. | $\underline{\mathbf{n}}_{ij}^\top \boldsymbol{\omega}^{-1} \underline{\mathbf{n}}_{ik} = 0$ | 1 |
| (5) | v.p. orthogonal to v.l. | $\underline{\mathbf{n}}_{ij} = \varkappa \boldsymbol{\omega} \, \underline{\mathbf{v}}_k$ | 2 |
| (6) | orthogonal image raster $\theta = \pi/2$ | $\omega_{12} = \omega_{21} = 0$ | 1 |
| (7) | unit aspect $a = 1$ when $\theta = \pi/2$ | $\omega_{11} - \omega_{22} = 0$ | 1 |
| (8) | known principal point $u_0 = v_0 = 0$ | $\omega_{13} = \omega_{31} = \omega_{23} = \omega_{32} = 0$ | 2 |

- these are homogeneous linear equations for the 5 parameters in $\boldsymbol{\omega}$ in the form $\mathbf{D}\mathbf{w} = \mathbf{0}$

  $\varkappa$ can be eliminated from (5)

- we need at least 5 constraints for full $\boldsymbol{\omega}$

  symmetric $3 \times 3$

- we get $\mathbf{K}$ from $\boldsymbol{\omega}^{-1} = \mathbf{K}\mathbf{K}^\top$ by Choleski decomposition

  the decomposition returns a positive definite upper triangular matrix

  one avoids solving an explicit set of quadratic equations for the parameters in $\mathbf{K}$

## Examples

Assuming orthogonal raster, unit aspect (ORUA): $\theta = \pi/2$, $a = 1$

$$\boldsymbol{\omega} \simeq \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ -u_0 & -v_0 & f^2 + u_0^2 + v_0^2 \end{bmatrix}$$

### Ex 1:
Assuming ORUA and known $m_0 = (u_0, v_0)$, two <u>finite</u> orthogonal vanishing points give $f$

$$\underline{\mathbf{v}}_1^\top \boldsymbol{\omega}\, \underline{\mathbf{v}}_2 = 0 \quad \Rightarrow \quad f^2 = \left| (\mathbf{v}_1 - \mathbf{m}_0)^\top (\mathbf{v}_2 - \mathbf{m}_0) \right|$$

in this formula, $\mathbf{v}_i$, $\mathbf{m}_0$ are Cartesian (not homogeneous)!

### Ex 2:
Non-orthogonal vanishing points $\mathbf{v}_i$, $\mathbf{v}_j$, known angle $\phi$: $\cos\phi = \dfrac{\underline{\mathbf{v}}_i^\top \boldsymbol{\omega} \underline{\mathbf{v}}_j}{\sqrt{\underline{\mathbf{v}}_i^\top \boldsymbol{\omega} \underline{\mathbf{v}}_i}\sqrt{\underline{\mathbf{v}}_j^\top \boldsymbol{\omega} \underline{\mathbf{v}}_j}}$

- leads to polynomial equations
- e.g. ORUA and $u_0 = v_0 = 0$ gives

$$(f^2 + \mathbf{v}_i^\top \mathbf{v}_j)^2 = (f^2 + \|\mathbf{v}_i\|^2) \cdot (f^2 + \|\mathbf{v}_j\|^2) \cdot \cos^2\phi$$

# ►Camera Orientation from Two Finite Vanishing Points

**Problem:** Given $\mathbf{K}$ and two vanishing points corresponding to two known orthogonal directions $\mathbf{d}_1$, $\mathbf{d}_2$, compute camera orientation $\mathbf{R}$ with respect to the plane.

- 3D coordinate system <u>choice</u>, e.g.:

$$\mathbf{d}_1 = (1,0,0), \quad \mathbf{d}_2 = (0,1,0)$$

- we know that

$$\mathbf{d}_i \simeq \mathbf{Q}^{-1}\underline{v}_i = (\mathbf{KR})^{-1}\underline{v}_i = \mathbf{R}^{-1}\underbrace{\mathbf{K}^{-1}\underline{v}_i}_{\underline{w}_i}$$

$$\mathbf{R}\mathbf{d}_i \simeq \underline{w}_i$$

- knowing $\mathbf{d}_{1,2}$ we conclude that $\underline{w}_i/\|\underline{w}_i\|$ is the $i$-th column $\mathbf{r}_i$ of $\mathbf{R}$

- the third column is orthogonal:
  $\mathbf{r}_3 \simeq \mathbf{r}_1 \times \mathbf{r}_2$

$$\mathbf{R} = \left[ \frac{\underline{w}_1}{\|\underline{w}_1\|} \quad \frac{\underline{w}_2}{\|\underline{w}_2\|} \quad \frac{\underline{w}_1 \times \underline{w}_2}{\|\underline{w}_1 \times \underline{w}_2\|} \right]$$

- in general we have to care about the signs $\pm\underline{w}_i$ (such that $\det \mathbf{R} = 1$)
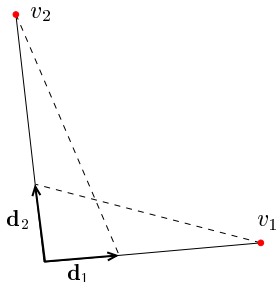


some suitable scenes

# Application: Planar Rectification

**Principle:** Rotate camera (image plane) parallel to the plane of interest.



$$\underline{m} \simeq \mathbf{K}\mathbf{R}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix}\underline{X} \qquad\qquad \underline{m}' \simeq \mathbf{K}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix}\underline{X}$$

$$\underline{m}' \simeq \mathbf{K}(\mathbf{K}\mathbf{R})^{-1}\,\underline{m} = \mathbf{K}\mathbf{R}^{\top}\mathbf{K}^{-1}\,\underline{m} = \mathbf{H}\,\underline{m}$$

- $\mathbf{H}$ is the rectifying homography
- both $\mathbf{K}$ and $\mathbf{R}$ can be calibrated from two finite vanishing points assuming ORUA →57
- not possible when one of them is (or both are) infinite
- without ORUA we would need 4 additional views to calibrate $\mathbf{K}$ as on →54

# ►Camera Resection

Camera <u>calibration</u> and <u>orientation</u> from a known set of $k \geq 6$ reference points and their images $\left\{(X_i, m_i)\right\}_{i=1}^6$.
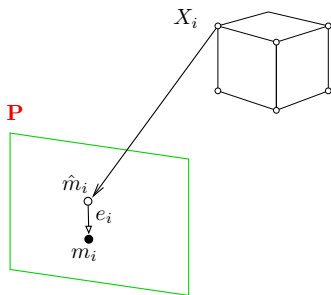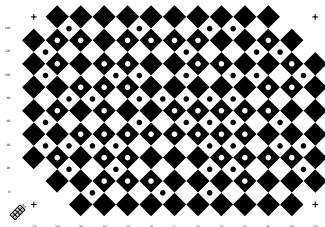


- $X_i$ are considered exact
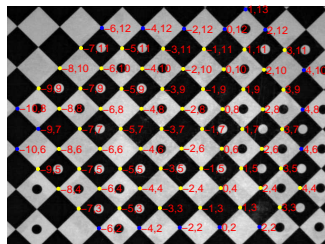
- $m_i$ is a measurement subject to detection error

$$\mathbf{m}_i = \hat{\mathbf{m}}_i + \mathbf{e}_i \qquad \text{Cartesian}$$

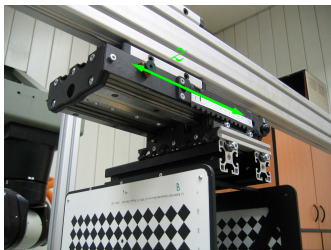- where $\lambda_i \, \hat{\underline{\mathbf{m}}}_i = \mathbf{P} \underline{\mathbf{X}}_i$

# Resection Targets



calibration chart



automatic calibration point detection



resection target with translation stage

- target translated at least once
- by a calibrated (known) translation
- $X_i$ point locations looked up in a table based on their code

# ►The Minimal Problem for Camera Resection

**Problem:** Given $k = 6$ corresponding pairs $\left\{ (X_i, m_i) \right\}_{i=1}^{k}$, find $\mathbf{P}$

$$\lambda_i \underline{\mathbf{m}}_i = \mathbf{P}\underline{\mathbf{X}}_i, \qquad \mathbf{P} = \begin{bmatrix} \mathbf{q}_1^\top & q_{14} \\ \mathbf{q}_2^\top & q_{24} \\ \mathbf{q}_3^\top & q_{34} \end{bmatrix}$$

$$\underline{\mathbf{X}}_i = (x_i, y_i, z_i, 1), \quad i = 1, 2, \ldots, k, \ k = 6$$
$$\underline{\mathbf{m}}_i = (u_i, v_i, 1), \quad \lambda_i \in \mathbb{R}, \ \lambda_i \neq 0, \ |\lambda_i| < \infty$$

easily modifiable for infinite points $X_i$ but be aware of →64

expanded: $\quad \lambda_i u_i = \mathbf{q}_1^\top \mathbf{X}_i + q_{14}, \quad \lambda_i v_i = \mathbf{q}_2^\top \mathbf{X}_i + q_{24}, \quad \lambda_i = \mathbf{q}_3^\top \mathbf{X}_i + q_{34}$

after elimination of $\lambda_i$: $\quad (\mathbf{q}_3^\top \mathbf{X}_i + q_{34}) u_i = \mathbf{q}_1^\top \mathbf{X}_i + q_{14}, \quad (\mathbf{q}_3^\top \mathbf{X}_i + q_{34}) v_i = \mathbf{q}_2^\top \mathbf{X}_i + q_{24}$

**Then**

$$\mathbf{A}\,\mathbf{q} = \begin{bmatrix} \mathbf{X}_1^\top & 1 & \mathbf{0}^\top & 0 & -u_1\mathbf{X}_1^\top & -u_1 \\ \mathbf{0}^\top & 0 & \mathbf{X}_1^\top & 1 & -v_1\mathbf{X}_1^\top & -v_1 \\ \vdots & & & & & \vdots \\ \mathbf{X}_k^\top & 1 & \mathbf{0}^\top & 0 & -u_k\mathbf{X}_k^\top & -u_k \\ \mathbf{0}^\top & 0 & \mathbf{X}_k^\top & 1 & -v_k\mathbf{X}_k^\top & -v_k \end{bmatrix} \cdot \begin{bmatrix} \mathbf{q}_1 \\ q_{14} \\ \mathbf{q}_2 \\ q_{24} \\ \mathbf{q}_3 \\ q_{34} \end{bmatrix} = \mathbf{0} \qquad (9)$$

- we need 11 indepedent parameters for $\mathbf{P}$
- $\mathbf{A} \in \mathbb{R}^{2k,12}, \ \mathbf{q} \in \mathbb{R}^{12}$
- 6 points in a general position give $\text{rank}\,\mathbf{A} = 12$ and there is no (non-trivial) null space
- drop one row to get rank-11 matrix, then the basis vector of the null space of $\mathbf{A}$ gives $\mathbf{q}$

- given the 6 correspondences, we have 12 equations for the 11 parameters
- can we use all the information present in the 6 points?

**Jack-knife estimation**

1. $n := 0$
2. for $i = 1, 2, \ldots, 2k$ do
   a) delete $i$-th row from $\mathbf{A}$, this gives $\mathbf{A}_i$
   b) if $\dim \mathrm{null}\, \mathbf{A}_i > 1$ continue with the next $i$
   c) $n := n + 1$
   d) compute the right null-space $\mathbf{q}_i$ of $\mathbf{A}_i$                          e.g. by 'economy-size' SVD
   e) $\hat{\mathbf{q}}_i := \mathbf{q}_i$ normalized to $q_{34} = 1$ and dimension-reduced     assuming finite cam. with $P_{3,4} = 1$
3. from all $n$ vectors $\hat{\mathbf{q}}_i$ collected in Step 1d compute

$$\mathbf{q} = \frac{1}{n} \sum_{i=1}^{n} \hat{\mathbf{q}}_i, \quad \mathrm{var}[\mathbf{q}] = \frac{n-1}{n} \mathrm{diag} \sum_{i=1}^{n} (\hat{\mathbf{q}}_i - \mathbf{q})(\hat{\mathbf{q}}_i - \mathbf{q})^{\top}$$

regular for $n \geq 11$

variance of the sample mean

- have a solution + an error estimate, per individual elements of $\mathbf{P}$ (except $P_{34}$)
- at least 5 points must be in a general position ($\rightarrow$64)
- large error indicates near degeneracy
- computation not efficient with $k > 6$ points, needs $\binom{2k}{11}$ draws, e.g. $k = 7 \Rightarrow 364$ draws
- better error estimation method: decompose $\mathbf{P}_i$ to $\mathbf{K}_i$, $\mathbf{R}_i$, $\mathbf{t}_i$ ($\rightarrow$33), represent $\mathbf{R}_i$ with 3 parameters (e.g. Euler angles, or in Cayley representation $\rightarrow$142) and compute the errors for the parameters
- even better: use the $\mathrm{SE}(3)$ Lie group for $(\mathbf{R}_i, \mathbf{t}_i)$ and average its Lie-algebra representations

Let $\mathcal{X} = \{X_i;\ i = 1, \ldots\}$ be a set of points and $\mathbf{P}_1 \not\simeq \mathbf{P}_j$ be two regular (rank-3) cameras. Then two configurations $(\mathbf{P}_1, \mathcal{X})$ and $(\mathbf{P}_j, \mathcal{X})$ are _image-equivalent_ if

$$\mathbf{P}_1 \underline{\mathbf{X}}_i \simeq \mathbf{P}_j \underline{\mathbf{X}}_i \quad \text{for all} \quad X_i \in \mathcal{X}$$

there is a non-trivial set of other cameras that see the same image

**Results**



$\mathcal{C}$ $C_1$ $C_2$ $\varkappa$ $C_\infty$

Case 4

- _importantly_: If all calibration points $X_i \in \mathcal{X}$ lie on a plane $\varkappa$ then camera resection is non-unique and all image-equivalent camera centers lie on a spatial line $\mathcal{C}$ with the $C_\infty = \varkappa \cap \mathcal{C}$ excluded

  this also means we cannot resect if all $X_i$ are infinite

- and more: by adding points $X_i \in \mathcal{X}$ to $\mathcal{C}$ we gain nothing

- there are additional image-equivalent configurations, see next

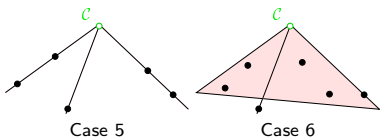  proof sketch in [H&Z, Sec. 22.1.2]
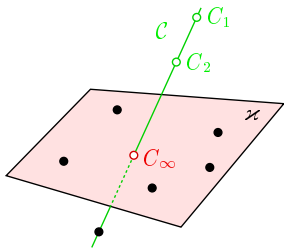
Note that if $\mathbf{Q}$, $\mathbf{T}$ are suitable homographies then $\mathbf{P}_1 \simeq \mathbf{Q}\mathbf{P}_0\mathbf{T}$, where $\mathbf{P}_0$ is canonical and the analysis can be made with $\hat{\mathbf{P}}_j \simeq \mathbf{Q}^{-1}\mathbf{P}_j$

$$\mathbf{P}_0 \underbrace{\mathbf{T}\underline{\mathbf{X}}_i}_{\underline{\mathbf{Y}}_i} \simeq \hat{\mathbf{P}}_j \underbrace{\mathbf{T}\underline{\mathbf{X}}_i}_{\underline{\mathbf{Y}}_i} \quad \text{for all} \quad Y_i \in \mathcal{Y}$$
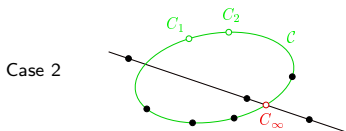
# cont'd (all cases)



- cameras $C_1$, $C_2$ co-located at point $\mathcal{C}$
- points on three optical rays or one optical ray and one optical plane

- Case 5: camera sees 3 isolated point images
- Case 6: cam. sees a line of points and an isolated point
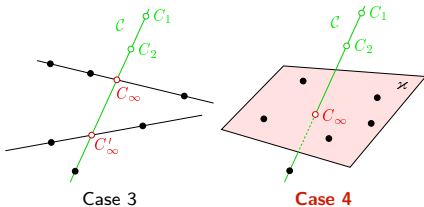


- cameras lie on a line $\mathcal{C} \setminus \{C_\infty, C'_\infty\}$
- points lie on $\mathcal{C}$ and
    1. on two lines meeting $\mathcal{C}$ at $C_\infty$, $C'_\infty$
    2. or on a plane meeting $\mathcal{C}$ at $C_\infty$

- Case 3: camera sees 2 lines of points
- Case 4: **dangerous!**



- cameras lie on a planar conic $\mathcal{C} \setminus \{C_\infty\}$
  not necessarily an ellipse
- points lie on $\mathcal{C}$ and an additional line meeting the conic at $C_\infty$

- Case 2: camera sees 2 lines of points
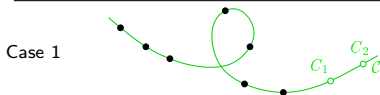


- cameras and points all lie on a twisted cubic $\mathcal{C}$
- Case 1: camera sees points on a conic
  dangerous but unlikely

# ►Three-Point Exterior Orientation Problem (P3P)

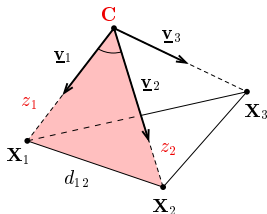Calibrated camera rotation and translation from Perspective images of 3 reference Points.
**Problem:** Given $\mathbf{K}$ and three corresponding pairs $\left\{(m_i, X_i)\right\}_{i=1}^{3}$, find $\mathbf{R}$, $\mathbf{C}$ by solving

$$\lambda_i \underline{\mathbf{m}}_i = \mathbf{K}\mathbf{R}\,(\mathbf{X}_i - \mathbf{C}), \qquad i = 1, 2, 3 \qquad \mathbf{X}_i \text{ Cartesian}$$

1. Transform $\underline{\mathbf{v}}_i \overset{\text{def}}{=} \mathbf{K}^{-1}\underline{\mathbf{m}}_i$. Then

$$\lambda_i \underline{\mathbf{v}}_i = \mathbf{R}\,(\mathbf{X}_i - \mathbf{C}). \tag{10}$$

2. If there was no rotation in (10), the situation would look like this



3. and we could shoot 3 lines from the given points $\mathbf{X}_i$ in given directions $\underline{\mathbf{v}}_i$ to get $\mathbf{C}$
4. given $\mathbf{C}$ we solve (10) for $\lambda_i$, $\mathbf{R}$

**If there is rotation R**

1. Eliminate **R** by taking      rotation preserves length: $\|\mathbf{R}\mathbf{x}\| = \|\mathbf{x}\|$

$$|\lambda_i| \cdot \|\underline{\mathbf{v}}_i\| = \|\mathbf{X}_i - \mathbf{C}\| \overset{\text{def}}{=} z_i \qquad (11)$$

2. Consider only angles among $\underline{\mathbf{v}}_i$ and apply Cosine Law per triangle $(\mathbf{C}, \mathbf{X}_i, \mathbf{X}_j)$ $i, j = 1, 2, 3$, $i \neq j$
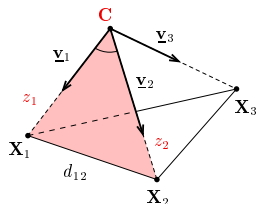
$$d_{ij}^2 = z_i^2 + z_j^2 - 2 z_i z_j c_{ij},$$

$$z_i = \|\mathbf{X}_i - \mathbf{C}\|, \ d_{ij} = \|\mathbf{X}_j - \mathbf{X}_i\|, \ c_{ij} = \cos(\angle \underline{\mathbf{v}}_i \underline{\mathbf{v}}_j)$$

4. Solve system of 3 quadratic eqs in 3 unknowns $z_i$

[Fischler & Bolles, 1981]
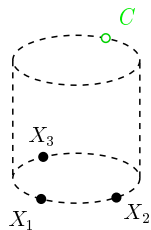
there may be no real root; there are up to 4 solutions that cannot be ignored                    (verify on additional points)

5. Compute **C** by trilateration (3-sphere intersection) from $\mathbf{X}_i$ and $z_i$; then $\lambda_i$ from (11) and **R** from (10)

Similar problems (P4P with unknown $f$) at `http://cmp.felk.cvut.cz/minimal/` (with code)

# Degenerate (Critical) Configurations for Exterior Orientation
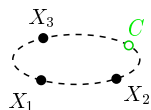


**unstable solution**

- center of projection $C$ located on the orthogonal circular cylinder with base circumscribing the three points $X_i$

unstable: a small change of $X_i$ results in a large change of $C$
can be detected by error propagation

**degenerate**

- camera $C$ is coplanar with points $(X_1, X_2, X_3)$ but is not on the circumscribed circle of $(X_1, X_2, X_3)$

camera sees points on a line



**no solution**

1. $C$ cocyclic with $(X_1, X_2, X_3)$    camera sees points on a line

- additional critical configurations depend on the quadratic equations solver

[Haralick et al. IJCV 1994]

| problem | given | unknown | slide |
|---|---|---|---|
| camera resection | 6 world–img correspondences $\left\{(X_i, m_i)\right\}_{i=1}^{6}$ | **P** | 62 |
| exterior orientation | $\mathbf{K}$, 3 world–img correspondences $\left\{(X_i, m_i)\right\}_{i=1}^{3}$ | **R, C** | 66 |
| relative orientation | 3 world-world correspondences $\left\{(X_i, Y_i)\right\}_{i=1}^{3}$ | **R, t** | 70 |

- camera resection and exterior orientation are similar problems in a sense:
  - we do resectioning when our camera is uncalibrated
  - we do orientation when our camera is calibrated
- relative orientation involves no camera (see next)
- more problems to come

## The Relative Orientation Problem

**Problem:** Given point triples $(X_1, X_2, X_3)$ and $(Y_1, Y_2, Y_3)$ in a general position in $\mathbf{R}^3$ such that the correspondence $X_i \leftrightarrow Y_i$ is known, determine the relative orientation $(\mathbf{R}, \mathbf{t})$ that maps $\mathbf{X}_i$ to $\mathbf{Y}_i$, i.e.

$$\mathbf{Y}_i = \mathbf{R}\mathbf{X}_i + \mathbf{t}, \quad i = 1, 2, 3.$$

**Applies to:**
- 3D scanners
- partial reconstructions from different viewpoints

**Obs:** Let the centroid be $\bar{\mathbf{X}} = \frac{1}{3}\sum_i \mathbf{X}_i$ and analogically for $\bar{\mathbf{Y}}$. Then

$$\bar{\mathbf{Y}} = \mathbf{R}\bar{\mathbf{X}} + \mathbf{t}.$$

Therefore

$$\mathbf{Z}_i \stackrel{\mathrm{def}}{=} (\mathbf{Y}_i - \bar{\mathbf{Y}}) = \mathbf{R}(\mathbf{X}_i - \bar{\mathbf{X}}) \stackrel{\mathrm{def}}{=} \mathbf{R}\mathbf{W}_i$$

If all dot products are equal, $\mathbf{Z}_i^\top \mathbf{Z}_j = \mathbf{W}_i^\top \mathbf{W}_j$ for $i, j = 1, 2, 3$, we have

$$\mathbf{R}^* = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 & \mathbf{W}_3 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Z}_1 & \mathbf{Z}_2 & \mathbf{Z}_3 \end{bmatrix}$$

Otherwise (in practice) we setup a minimization problem

$$\mathbf{R}^* = \arg\min_{\mathbf{R}} \sum_{i=1}^{3} \|\mathbf{Z}_i - \mathbf{R}\mathbf{W}_i\|^2 \quad \text{s.t.} \quad \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \quad \det \mathbf{R} = 1$$

$$\arg\min_{\mathbf{R}} \sum_i \|\mathbf{Z}_i - \mathbf{R}\mathbf{W}_i\|^2 = \arg\min_{\mathbf{R}} \sum_i \left( \|\mathbf{Z}_i\|^2 - 2\mathbf{Z}_i^\top \mathbf{R}\mathbf{W}_i + \|\mathbf{W}_i\|^2 \right) = \cdots$$

$$\cdots = \arg\max_{\mathbf{R}} \sum \mathbf{Z}_i^\top \mathbf{R}\mathbf{W}_i$$

## cont'd (What is Linear Algebra Telling Us?)

**Obs 1:** Let $\mathbf{A} : \mathbf{B} = \sum_{i,j} a_{ij} b_{ij}$ be the dot-product (Frobenius inner product) over real matrices. Then

$$\mathbf{A} : \mathbf{B} = \mathbf{B} : \mathbf{A} = \operatorname{tr}(\mathbf{A}^\top \mathbf{B})$$

**Obs 2:** (cyclic property for matrix trace)

$$\operatorname{tr}(\mathbf{ABC}) = \operatorname{tr}(\mathbf{CAB})$$

**Obs 3:** ($\mathbf{Z}_i$, $\mathbf{W}_i$ are vectors)

$$\mathbf{Z}_i^\top \mathbf{R} \mathbf{W}_i = \operatorname{tr}(\mathbf{Z}_i^\top \mathbf{R} \mathbf{W}_i) = \operatorname{tr}(\mathbf{W}_i \mathbf{Z}_i^\top \mathbf{R}) = (\mathbf{Z}_i \mathbf{W}_i^\top) : \mathbf{R} = \mathbf{R} : (\mathbf{Z}_i \mathbf{W}_i^\top)$$

Let the SVD be

$$\sum_i \mathbf{Z}_i \mathbf{W}_i^\top \stackrel{\mathrm{def}}{=} \mathbf{M} = \mathbf{U} \mathbf{D} \mathbf{V}^\top$$

Then

$$\mathbf{R} : \mathbf{M} = \mathbf{R} : (\mathbf{U} \mathbf{D} \mathbf{V}^\top) = \operatorname{tr}(\mathbf{R}^\top \mathbf{U} \mathbf{D} \mathbf{V}^\top) = \operatorname{tr}(\mathbf{V}^\top \mathbf{R}^\top \mathbf{U} \mathbf{D}) = (\mathbf{U}^\top \mathbf{R} \mathbf{V}) : \mathbf{D}$$

We are solving

$$\mathbf{R}^* = \arg\max_{\mathbf{R}} \sum_i \mathbf{Z}_i^\top \mathbf{R} \mathbf{W}_i = \arg\max_{\mathbf{R}} \left( \mathbf{U}^\top \mathbf{R} \mathbf{V} \right) : \mathbf{D}$$

**A particular solution is found as follows:**
- $\mathbf{U}^\top \mathbf{R} \mathbf{V}$ must be (1) orthogonal, and most similar to (2) diagonal, (3) positive definite
- Since $\mathbf{U}$, $\mathbf{V}$ are orthogonal matrices then the solution to the problem is among $\mathbf{R}^* = \mathbf{U} \mathbf{S} \mathbf{V}^\top$, where $\mathbf{S}$ is diagonal and orthogonal, i.e. one of

$$\pm \operatorname{diag}(1,1,1), \quad \pm \operatorname{diag}(1,-1,-1), \quad \pm \operatorname{diag}(-1,1,-1), \quad \pm \operatorname{diag}(-1,-1,1)$$

- $\mathbf{U}^\top \mathbf{V}$ is not necessarily positive definite
- We choose $\mathbf{S}$ so that $(\mathbf{R}^*)^\top \mathbf{R}^* = \mathbf{I}$

**Alg:**
1. Compute matrix $\mathbf{M} = \sum_i \mathbf{Z}_i \mathbf{W}_i^\top$.
2. Compute SVD $\mathbf{M} = \mathbf{U} \mathbf{D} \mathbf{V}^\top$.
3. Compute all $\mathbf{R}_k = \mathbf{U} \mathbf{S}_k \mathbf{V}^\top$ that give $\mathbf{R}_k^\top \mathbf{R}_k = \mathbf{I}$.
4. Compute $\mathbf{t}_k = \bar{\mathbf{Y}} - \mathbf{R}_k \bar{\mathbf{X}}$.

- The algorithm can be used for more than 3 points
- Triple pairs can be pre-filtered based on motion invariants (lengths, angles)
- The P3P problem is very similar but not identical

Module IV

# Computing with a Camera Pair

**4.1** Camera Motions Inducing Epipolar Geometry

**4.2** Estimating Fundamental Matrix from 7 Correspondences

**4.3** Estimating Essential Matrix from 5 Correspondences

**4.4** Triangulation: 3D Point Position from a Pair of Corresponding Points

**covered by**

[1] [H&Z] Secs: 9.1, 9.2, 9.6, 11.1, 11.2, 11.9, 12.2, 12.3, 12.5.1

[2] H. Li and R. Hartley. Five-point motion estimation made easy. In *Proc ICPR* 2006, pp. 630–633

**additional references**

H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293 (5828):133–135, 1981.

# ►Geometric Model of a Camera Pair

**Epipolar geometry:**

- brings constraints necessary for inter-image matching
- its parametric form encapsulates information about the relative pose of two cameras



two-camera setup
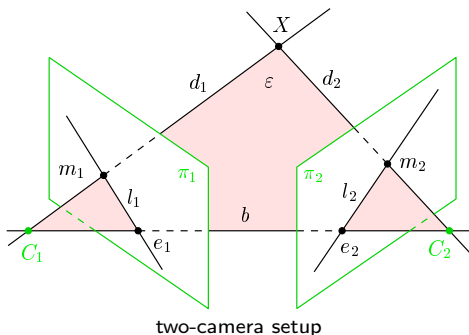
**Description**

- <u>baseline</u> $b$ joins projection centers $C_1$, $C_2$
$$\mathbf{b} = \mathbf{C}_2 - \mathbf{C}_1$$

- <u>epipole</u> $e_i \in \pi_i$ is the image of $C_j$:
$$\underline{\mathbf{e}}_1 \simeq \mathbf{P}_1 \underline{\mathbf{C}}_2, \quad \underline{\mathbf{e}}_2 \simeq \mathbf{P}_2 \underline{\mathbf{C}}_1$$

- $l_i \in \pi_i$ is the image of <u>epipolar plane</u>
$$\varepsilon = (C_2, X, C_1)$$

- $l_j$ is the <u>epipolar line</u> in image $\pi_j$ induced by $m_i$ in image $\pi_i$

**Epipolar constraint:** corresponding $d_2$, $b$, $d_1$ are coplanar

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{q}_i \end{bmatrix} = \mathbf{K}_i \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix} = \mathbf{K}_i \mathbf{R}_i \begin{bmatrix} \mathbf{I} & -\mathbf{C}_i \end{bmatrix} \quad i = 1, 2 \qquad \text{→31}$$
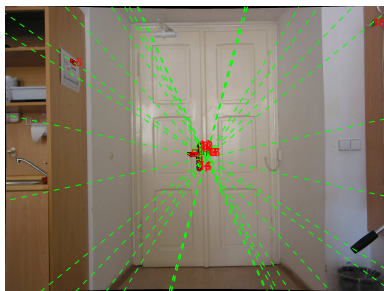
image 1                                    image 2
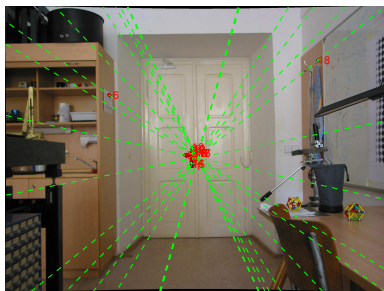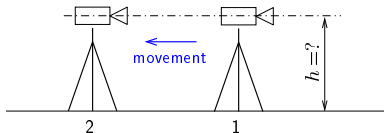
- red: correspondences
- green: epipolar line pairs per correspondence

<span style="color:blue">click on the image to see their IDs<br>same ID in both images</span>
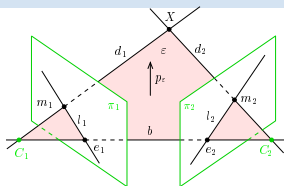
How high was the camera above the floor?

• There is an equivalence $\mathbf{b} \times \mathbf{m} = [\mathbf{b}]_{\times} \mathbf{m}$, where $[\mathbf{b}]_{\times}$ is a $3 \times 3$ skew-symmetric matrix

$$[\mathbf{b}]_{\times} = \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix}, \qquad \text{assuming} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

**Some properties**

1. $[\mathbf{b}]_{\times}^{\top} = -[\mathbf{b}]_{\times}$            <span style="color:blue">the general antisymmetry property</span>

2. $\mathbf{A}$ is skew-symmetric iff $\mathbf{x}^{\top}\mathbf{A}\mathbf{x} = 0$ for all $\mathbf{x}$     <span style="color:blue">skew-sym mtx generalizes cross products</span>

3. $[\mathbf{b}]_{\times}^{3} = -\|\mathbf{b}\|^2 \cdot [\mathbf{b}]_{\times}$

4. $\|[\mathbf{b}]_{\times}\|_F = \sqrt{2}\,\|\mathbf{b}\|$      <span style="color:blue">Frobenius norm ($\|\mathbf{A}\|_F = \sqrt{\mathrm{tr}(\mathbf{A}^{\top}\mathbf{A})} = \sqrt{\sum_{i,j}|a_{ij}|^2}$)</span>

5. $[\mathbf{b}]_{\times}\mathbf{b} = \mathbf{0}$

6. $\mathrm{rank}\,[\mathbf{b}]_{\times} = 2$ iff $\|\mathbf{b}\| > 0$          <span style="color:blue">check minors of $[\mathbf{b}]_{\times}$</span>

7. eigenvalues of $[\mathbf{b}]_{\times}$ are $(0, \lambda, -\lambda)$

8. for any $3 \times 3$ regular $\mathbf{B}$ :   $\mathbf{B}^{\top}[\mathbf{Bz}]_{\times}\mathbf{B} = \det \mathbf{B}\,[\mathbf{z}]_{\times}$    <span style="color:blue">follows from the factoring on →39</span>

9. in particular:   if $\mathbf{R}\mathbf{R}^{\top} = \mathbf{I}$ then $[\mathbf{Rb}]_{\times} = \mathbf{R}[\mathbf{b}]_{\times}\mathbf{R}^{\top}$

    • note that if $\mathbf{R}_b$ is rotation about $\mathbf{b}$ then $\mathbf{R}_b\mathbf{b} = \mathbf{b}$

    • note $[\mathbf{b}]_{\times}$ is not a homography; it is not a rotation matrix    <span style="color:blue">it is the logarithm of a rotation mtx</span>

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{q}_i \end{bmatrix} = \mathbf{K}_i \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix}, \; i = 1, 2$$

$\mathbf{R}_{21}$ – relative camera rotation, $\mathbf{R}_{21} = \mathbf{R}_2 \mathbf{R}_1^\top$

$\mathbf{t}_{21}$ – relative camera translation, $\mathbf{t}_{21} = \mathbf{t}_2 - \mathbf{R}_{21}\mathbf{t}_1 = -\mathbf{R}_2 \mathbf{b} \to 74$

$\mathbf{b}$ – baseline vector (world coordinate system)

remember: $\mathbf{C} = -\mathbf{Q}^{-1}\mathbf{q} = -\mathbf{R}^\top \mathbf{t}$      $\to 33$ and $35$

$$0 = \underbrace{\mathbf{d}_2^\top}_{\text{normal of } \varepsilon} \mathbf{p}_\varepsilon \simeq \underbrace{(\mathbf{Q}_2^{-1}\underline{\mathbf{m}}_2)^\top}_{\text{optical ray}} \underbrace{\mathbf{Q}_1^\top \mathbf{l}_1}_{\text{optical plane}} = \underline{\mathbf{m}}_2^\top \underbrace{\mathbf{Q}_2^{-\top}\mathbf{Q}_1^\top (\mathbf{e}_1 \times \underline{\mathbf{m}}_1)}_{\text{image of } \varepsilon \text{ in } \pi_2} = \underline{\mathbf{m}}_2^\top \underbrace{\left( \mathbf{Q}_2^{-\top}\mathbf{Q}_1^\top [\mathbf{e}_1]_\times \right)}_{\text{fundamental matrix } \mathbf{F}} \underline{\mathbf{m}}_1$$

**Epipolar constraint**    $\boxed{\underline{\mathbf{m}}_2^\top \mathbf{F} \underline{\mathbf{m}}_1 = 0}$     is a point-line incidence constraint

- point $\underline{\mathbf{m}}_2$ is incident on epipolar line $\mathbf{l}_2 \simeq \mathbf{F}\underline{\mathbf{m}}_1$
- point $\underline{\mathbf{m}}_1$ is incident on epipolar line $\mathbf{l}_1 \simeq \mathbf{F}^\top \underline{\mathbf{m}}_2$
- $\mathbf{F}\mathbf{e}_1 = \mathbf{F}^\top \mathbf{e}_2 = \mathbf{0}$ (non-trivially)
- all epipolars meet at the epipole

$$\mathbf{e}_1 \simeq \mathbf{Q}_1 \mathbf{C}_2 + \mathbf{q}_1 = \mathbf{Q}_1 \mathbf{C}_2 - \mathbf{Q}_1 \mathbf{C}_1 = \mathbf{K}_1 \mathbf{R}_1 \mathbf{b} = -\mathbf{K}_1 \mathbf{R}_1 \mathbf{R}_2^\top \mathbf{t}_{21} = -\mathbf{K}_1 \mathbf{R}_{21}^\top \mathbf{t}_{21}$$

$$\mathbf{F} = \mathbf{Q}_2^{-\top}\mathbf{Q}_1^\top [\mathbf{e}_1]_\times = \mathbf{Q}_2^{-\top}\mathbf{Q}_1^\top [-\mathbf{K}_1 \mathbf{R}_{21}^\top \mathbf{t}_{21}]_\times \overset{\circledast \, 1}{= \cdots} \simeq \mathbf{K}_2^{-\top}[-\mathbf{t}_{21}]_\times \mathbf{R}_{21} \mathbf{K}_1^{-1} \text{ fundamental}$$

$$\mathbf{E} = [-\mathbf{t}_{21}]_\times \mathbf{R}_{21} = \underbrace{[\mathbf{R}_2 \mathbf{b}]_\times}_{\text{baseline in Cam 2}} \mathbf{R}_{21} = \mathbf{R}_{21} \underbrace{[\mathbf{R}_1 \mathbf{b}]_\times}_{\text{baseline in Cam 1}} = \mathbf{R}_{21}[-\mathbf{R}_{21}^\top \mathbf{t}_{21}]_\times \quad \text{essential}$$

$$\mathbf{F} = ( \underbrace{\mathbf{Q}_2 \mathbf{Q}_1^{-1}}_{\text{epipolar homography } \mathbf{H}_e} )^{-\top} [\mathbf{e}_1]_\times = \underbrace{\mathbf{K}_2^{-\top} \mathbf{R}_{21} \mathbf{K}_1^{\top}}_{\mathbf{H}_e^{-\top}} \overbrace{[\mathbf{e}_1]_\times}^{\text{left epipole}} \overset{76}{\simeq} [\overbrace{\mathbf{H}_e \mathbf{e}_1}^{\text{right epipole}}]_\times \mathbf{H}_e = \mathbf{K}_2^{-\top} \underbrace{[-\mathbf{t}_{21}]_\times \mathbf{R}_{21}}_{\text{essential matrix } \mathbf{E}} \mathbf{K}_1^{-1}$$

1. $\mathbf{E}$ captures relative camera pose only           [Longuet-Higgins 1981]

       (the change of the world coordinate system does not change $\mathbf{E}$)

$$\begin{bmatrix} \mathbf{R}'_i & \mathbf{t}'_i \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i \mathbf{R} & \mathbf{R}_i \mathbf{t} + \mathbf{t}_i \end{bmatrix},$$

then

$$\mathbf{R}'_{21} = \mathbf{R}'_2 {\mathbf{R}'_1}^\top = \cdots = \mathbf{R}_{21} \qquad\qquad \mathbf{t}'_{21} = \mathbf{t}'_2 - \mathbf{R}'_{21} \mathbf{t}'_1 = \cdots = \mathbf{t}_{21}$$
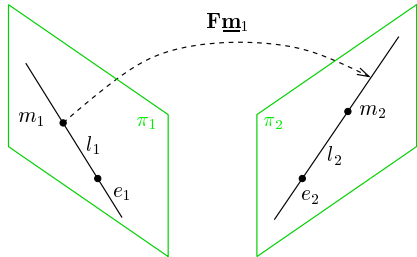
2. the translation length $\mathbf{t}_{21}$ is <u>lost</u> since $\mathbf{E}$ is homogeneous
3. $\mathbf{F}$ maps points to lines and it is not a homography
4. $\mathbf{H}_e$ maps epipoles to epipoles, $\mathbf{H}_e^{-\top}$ epipolar lines to epipolar lines: $\mathbf{l}_2 \simeq \mathbf{H}_e^{-\top} \mathbf{l}_1$
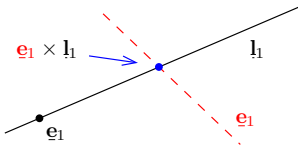


- replacement for $\mathbf{H}_e^{-\top}$ for epipolar line map: $\mathbf{l}_2 \simeq \mathbf{F}[\mathbf{e}_1]_\times \mathbf{l}_1$
- proof by point/line 'transmutation' (left)
- point $\underline{\mathbf{e}}_1$ does not lie on line $\underline{\mathbf{e}}_1$ (dashed): $\mathbf{e}_1^\top \mathbf{e}_1 \neq 0$
- $\mathbf{F}[\mathbf{e}_1]_\times$ is not a homography, unlike $\mathbf{H}_e^{-\top}$ but it does the same job for epipolar line mapping

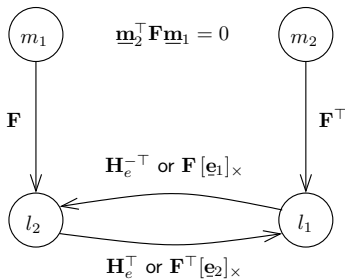$$0 = \underline{\mathbf{m}}_2^\top \mathbf{F} \, \underline{\mathbf{m}}_1$$

$$\underline{\mathbf{e}}_1 \simeq \mathrm{null}(\mathbf{F}), \qquad \underline{\mathbf{e}}_2 \simeq \mathrm{null}(\mathbf{F}^\top)$$

$$\underline{\mathbf{e}}_1 \simeq \mathbf{H}_e^{-1} \mathbf{e}_2 \qquad \underline{\mathbf{e}}_2 \simeq \mathbf{H}_e \mathbf{e}_1$$

$$\underline{\mathbf{l}}_1 \simeq \mathbf{F}^\top \underline{\mathbf{m}}_2 \qquad \underline{\mathbf{l}}_2 \simeq \mathbf{F} \underline{\mathbf{m}}_1$$

$$\underline{\mathbf{l}}_1 \simeq \mathbf{H}_e^\top \mathbf{l}_2 \qquad \underline{\mathbf{l}}_2 \simeq \mathbf{H}_e^{-\top} \mathbf{l}_1$$

$$\underline{\mathbf{l}}_1 \simeq \mathbf{F}^\top [\underline{\mathbf{e}}_2]_\times \mathbf{l}_2 \qquad \underline{\mathbf{l}}_2 \simeq \mathbf{F} [\underline{\mathbf{e}}_1]_\times \mathbf{l}_1$$



- $\mathbf{F}[\underline{\mathbf{e}}_1]_\times$ maps lines to lines but it is not a homography

- $\mathbf{H}_e = \mathbf{Q}_2 \mathbf{Q}_1^{-1}$ is the epipolar homography→78
  $\mathbf{H}_e^{-\top}$ maps epipolar lines to epipolar lines, where

$$\mathbf{H}_e = \mathbf{Q}_2 \mathbf{Q}_1^{-1} = \mathbf{K}_2 \mathbf{R}_{21} \mathbf{K}_1^{-1}$$

you have seen this →59

## ►Representation Theorem for Fundamental Matrices

**Def:** $\mathbf{F}$ is fundamental when $\mathbf{F} \simeq \mathbf{H}^{-\top}[\mathbf{e}_1]_\times$, where $\mathbf{H}$ is regular and $\mathbf{e}_1 \simeq \operatorname{null}\mathbf{F} \neq \mathbf{0}$.

**Theorem:** A $3 \times 3$ matrix $\mathbf{A}$ is fundamental iff it is of rank 2.

**Proof.**

Direct: By the geometry, $\mathbf{H}$ is full-rank, $\mathbf{e}_1 \neq \mathbf{0}$, hence $\mathbf{H}^{-\top}[\mathbf{e}_1]_\times$ is a $3 \times 3$ matrix of rank 2.

Converse:

1. let $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the SVD of $\mathbf{A}$ of rank 2; then $\mathbf{D} = \operatorname{diag}(\lambda_1, \lambda_2, 0)$, $\lambda_1 \geq \lambda_2 > 0$

2. we write $\mathbf{D} = \mathbf{B}\mathbf{C}$, where $\mathbf{B} = \operatorname{diag}(\lambda_1, \lambda_2, \lambda_3)$, $\mathbf{C} = \operatorname{diag}(1, 1, 0)$, $\lambda_3 = \lambda_2$ (w.l.o.g.)

3. then $\mathbf{A} = \mathbf{U}\mathbf{B}\mathbf{C}\mathbf{V}^\top = \mathbf{U}\mathbf{B}\mathbf{C}\underbrace{\mathbf{W}\mathbf{W}^\top}_{\mathbf{I}}\mathbf{V}^\top$ with $\mathbf{W}$ rotation

4. we look for a rotation $\mathbf{W}$ that maps $\mathbf{C}$ to a skew-symmetric $\mathbf{S}$, i.e. $\mathbf{S} = \mathbf{C}\mathbf{W}$

5. then $\mathbf{W} = \begin{bmatrix} 0 & \alpha & 0 \\ -\alpha & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $|\alpha| = 1$, and $\mathbf{S} = [\mathbf{s}]_\times$, $\mathbf{s} = (0, 0, 1)$

6. we write

   $\mathbf{v}_3$ – 3rd column of $\mathbf{V}$, $\mathbf{u}_3$ – 3rd column of $\mathbf{U}$

   $$\mathbf{A} = \mathbf{U}\mathbf{B}[\mathbf{s}]_\times \mathbf{W}^\top \mathbf{V}^\top \overset{\circledast 1}{=} \cdots = \underbrace{\mathbf{U}\mathbf{B}(\mathbf{V}\mathbf{W})^\top}_{\simeq \mathbf{H}^{-\top}}[\mathbf{v}_3]_\times \simeq \underbrace{[\mathbf{H}\mathbf{v}_3]_\times}_{\simeq [\mathbf{u}_3]_\times}\mathbf{H}, \tag{12}$$

7. $\mathbf{H}$ regular, $\mathbf{A}\mathbf{v}_3 = \mathbf{0}$, $\mathbf{u}_3\mathbf{A} = \mathbf{0}$ for $\mathbf{v}_3 \neq \mathbf{0}$, $\mathbf{u}_3 \neq \mathbf{0}$ □

• we also got a (non-unique: $\alpha = \pm 1$) decomposition formula for fundamental matrices
• it follows there is no constraint on $\mathbf{F}$ except the rank

**Theorem**

*Let $\mathbf{E}$ be a $3 \times 3$ matrix with SVD $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$. Then $\mathbf{E}$ is essential iff $\mathbf{D} \simeq \mathrm{diag}(1, 1, 0)$.*

**Proof.**

Direct:

If $\mathbf{E}$ is an essential matrix, then the epipolar homography matrix is a rotation matrix ($\rightarrow$78), hence $\mathbf{H}^{-\top} \simeq \mathbf{U}\mathbf{B}(\mathbf{V}\mathbf{W})^\top$ in (12) must be ($\lambda$-scaled) orthogonal, therefore $\mathbf{B} = \lambda \mathbf{I}$.

Converse:

$\mathbf{E}$ is fundamental with $\mathbf{D} = \lambda \, \mathrm{diag}(1, 1, 0)$ then we do not need $\mathbf{B}$ (as if $\mathbf{B} = \lambda \mathbf{I}$) in (12) and $\mathbf{U}(\mathbf{V}\mathbf{W})^\top$ is orthogonal, as required.

$\square$

## ►Essential Matrix Decomposition

We are decomposing $\mathbf{E}$ to $\mathbf{E} \simeq [-\mathbf{t}_{21}]_\times \mathbf{R}_{21} = \mathbf{R}_{21}[-\mathbf{R}_{21}^\top \mathbf{t}_{21}]_\times$      [H&Z, sec. 9.6]

1. compute SVD of $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ and verify $\mathbf{D} = \lambda \operatorname{diag}(1,1,0)$
2. ensure $\mathbf{U}$, $\mathbf{V}$ are rotation matrices by $\mathbf{U} \mapsto \det(\mathbf{U})\mathbf{U}$, $\mathbf{V} \mapsto \det(\mathbf{V})\mathbf{V}$
3. compute

$$\mathbf{R}_{21} = \mathbf{U} \underbrace{\begin{bmatrix} 0 & \alpha & 0 \\ -\alpha & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{W}} \mathbf{V}^\top, \quad \mathbf{t}_{21} = -\beta\,\mathbf{u}_3, \qquad |\alpha| = 1, \quad \beta \neq 0 \tag{13}$$

**Notes**

- $\mathbf{v}_3 \simeq \mathbf{R}_{21}^\top \mathbf{t}_{21}$ by (12), hence $\mathbf{R}_{21}\mathbf{v}_3 \simeq \mathbf{t}_{21} \simeq \mathbf{u}_3$ since it must fall in left null space by $\mathbf{E} \simeq [\mathbf{u}_3]_\times \mathbf{R}_{21}$

- $\mathbf{t}_{21}$ is recoverable up to scale $\beta$ and direction $\operatorname{sign}\beta$

- the result for $\mathbf{R}_{21}$ is unique up to $\alpha = \pm 1$      <span style="color:gray">despite non-uniqueness of SVD</span>

- the change of sign in $\alpha$ rotates the solution by $180°$ about $\mathbf{t}_{21}$

    $\mathbf{R}(\alpha) = \mathbf{U}\mathbf{W}\mathbf{V}^\top$, $\mathbf{R}(-\alpha) = \mathbf{U}\mathbf{W}^\top\mathbf{V}^\top \Rightarrow \mathbf{T} = \mathbf{R}(-\alpha)\mathbf{R}^\top(\alpha) = \cdots = \mathbf{U}\operatorname{diag}(-1,-1,1)\mathbf{U}^\top$
    which is a rotation by $180°$ about $\mathbf{u}_3 \simeq \mathbf{t}_{21}$:      <span style="color:gray">show that $\mathbf{u}_3$ is the rotation axis</span>

    $$\mathbf{U}\operatorname{diag}(-1,-1,1)\mathbf{U}^\top\mathbf{u}_3 = \mathbf{U}\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{u}_3$$
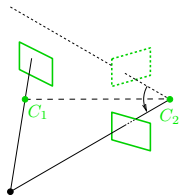
- 4 solution sets for 4 sign combinations of $\alpha$, $\beta$      <span style="color:gray">see next for geometric interpretation</span>
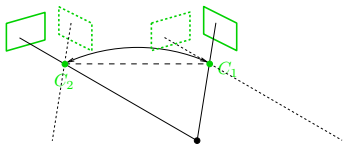
Transform the world coordinate system so that the origin is in Camera 2. Then $\mathbf{t}_{21} = -\mathbf{b}$ and $\mathbf{W}$ rotates about the baseline $\mathbf{b}$.
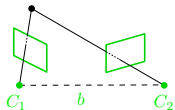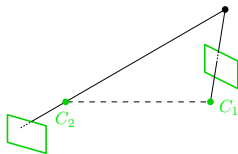
$\alpha, \beta$    FF

$-\alpha, \beta$ (twisted by $\mathbf{W}$)    BF

$\alpha, -\beta$ (baseline reversal)    BB

$-\alpha, -\beta$ (combination of both)    BF

- chirality constraint: all 3D points are in front of both cameras
- this singles-out the upper left case      [H&Z, Sec. 9.6.3]

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

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

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

**Solution:**

$$\underline{\mathbf{y}}_i^\top \mathbf{F} \, \underline{\mathbf{x}}_i = (\underline{\mathbf{y}}_i \underline{\mathbf{x}}_i^\top) : \mathbf{F} = \big(\text{vec}(\underline{\mathbf{y}}_i \underline{\mathbf{x}}_i^\top)\big)^\top \text{vec}(\mathbf{F}), \qquad \text{rotation property of matrix trace}$$

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

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

$$\mathbf{D} \, \text{vec}(\mathbf{F}) = \mathbf{0}$$

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

- 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 $\alpha$ 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 $\operatorname{rank} \mathbf{F} < 2$ then fail

- the result may depend on image (domain) transformations
- normalization improves conditioning      →92
- this gives a good starting point for the full algorithm      →110
- dealing with mismatches need not be a part of the 7-point algorithm      →111

## ▶Degenerate Configurations for Fundamental Matrix Estimation

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

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

   - in both cases: epipolar geometry is not defined
   - we get an <u>arbitrary</u> solution from the 7-point algorithm in the form of $\mathbf{F} = [\underline{\mathbf{s}}]_\times \mathbf{H}$
     $\qquad\qquad$ note that $[\underline{\mathbf{s}}]_\times \mathbf{H} \simeq \mathbf{H}'[\underline{\mathbf{s}}']_\times \rightarrow 76$

   

   - 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: $y_i \in l_i$, $\quad \mathbf{l}_i \simeq \underline{\mathbf{s}} \times \mathbf{H}\underline{\mathbf{x}}_i$
     $$0 = \underline{\mathbf{y}}_i^\top (\underline{\mathbf{s}} \times \mathbf{H}\underline{\mathbf{x}}_i) = \underline{\mathbf{y}}_i^\top [\underline{\mathbf{s}}]_\times \mathbf{H}\underline{\mathbf{x}}_i \quad \text{for any } \underline{\mathbf{x}}_i, \underline{\mathbf{y}}_i, \underline{\mathbf{s}} \, (!)$$
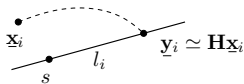
2. both camera centers and all 3D points lie on a ruled quadric
   $\qquad\qquad\qquad$ hyperboloid of one sheet, cones, cylinders, two planes
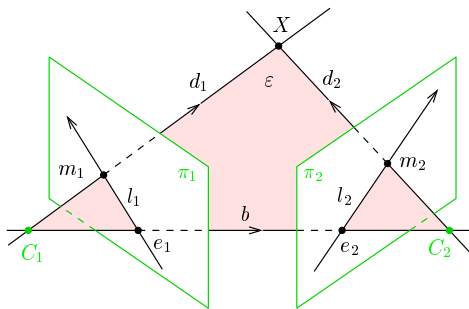   - there are 3 solutions for $\mathbf{F}$

**notes**
   - estimation of $\mathbf{E}$ <u>can</u> 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



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

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

- we can read the constraint as $(\mathbf{e}_2 \times \underline{\mathbf{m}}_2) \underset{\sim}{+} \mathbf{H}_e^{-\top} (\mathbf{e}_1 \times \underline{\mathbf{m}}_1)$
- 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 →111                    [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 $3 \times 3$ homogeneous matrix with two equal singular numbers
2. $\mathbf{R}$ – 3 DOF, $\mathbf{t}$ – 2 DOF only, in total 5 DOF $\rightarrow$ 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 $\rightarrow$81

**This gives an equation system:**

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

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

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

⊛ 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 $\mathbf{v}_i^\top \mathbf{E} \mathbf{v}_i' = 0$ 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 ($\rightarrow$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**

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

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

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 ($\rightarrow$63) not recommended $\qquad\qquad\qquad$ sensitive to small error
- we will use SVD ($\rightarrow$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

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

- 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     <span style="color:blue">projection centers in mm, image points in px</span>

$$\begin{bmatrix} 10^3 & 0 & 10^3 & 10^6 \\ 0 & 10^3 & 10^3 & 10^6 \\ 10^3 & 0 & 10^3 & 10^6 \\ 0 & 10^3 & 10^3 & 10^6 \end{bmatrix}$$
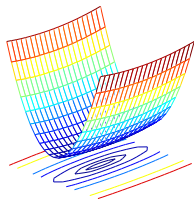
**Quick fix:**

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

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

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

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

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

   • <span style="color:blue">when SVD is used in camera resection, conditioning is essential for success</span>     →62

# Algebraic Error vs Reprojection Error

- algebraic error ($c$ – camera index, $(u^c, v^c)$ – image coordinates)  <span style="float:right">from SVD $\rightarrow$91</span>

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

- 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 $\Leftrightarrow$ reprojection error zero  <span style="float:right">$\sigma_4 = 0 \Rightarrow$ non-trivial null space</span>
- epipolar constraint satisfied $\Rightarrow$ 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 $\rightarrow$105

**Ex:**
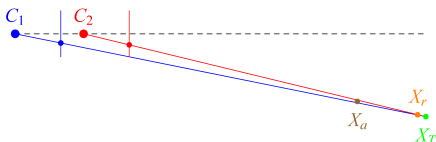


- forward camera motion
- error $f/50$ in image 2, orthogonal to epipolar plane

  $X_T$ – noiseless ground truth position
  $X_r$ – reprojection error minimizer
  $X_a$ – algebraic error minimizer
  $m$ – measurement ($m_T$ with noise in $v^2$)
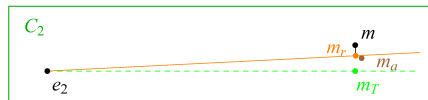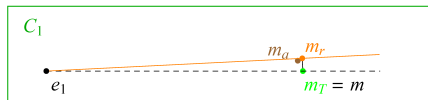
| problem | given | unknown | slide |
|---------|-------|---------|-------|
| camera resection | 6 world–img correspondences $\left\{(X_i, m_i)\right\}_{i=1}^{6}$ | **P** | 62 |
| exterior orientation | $\mathbf{K}$, 3 world–img correspondences $\left\{(X_i, m_i)\right\}_{i=1}^{3}$ | **R**, **t** | 66 |
| relative pointcloud orientation | 3 world-world correspondences $\left\{(X_i, Y_i)\right\}_{i=1}^{3}$ | **R**, **t** | 70 |
| fundamental matrix | 7 img–img correspondences $\left\{(m_i, m_i')\right\}_{i=1}^{7}$ | **F** | 84 |
| relative camera orientation | $\mathbf{K}$, 5 img–img correspondences $\left\{(m_i, m_i')\right\}_{i=1}^{5}$ | **R**, **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 →118)

- 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'

# Optimization for 3D Vision

**5.1** The Concept of Error for Epipolar Geometry
**5.2** Levenberg-Marquardt's Iterative Optimization
**5.3** The Correspondence Problem
**5.4** Optimization by Random Sampling

**covered by**
[1] [H&Z] Secs: 11.4, 11.6, 4.7
[2] Fischler, M.A. and Bolles, R.C . Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24(6):381–395, 1981

**additional references**

P. D. Sampson. Fitting conic sections to 'very scattered' data: An iterative refinement of the Bookstein algorithm. *Computer Vision, Graphics, and Image Processing*, 18:97–108, 1982.

O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *Proc DAGM*, LNCS 2781:236–243. Springer-Verlag, 2003.

O. Chum, T. Werner, and J. Matas. Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint. In *Proc ICPR*, vol 1:112–115, 2004.

**Background problems:** (1) Given at least 8 matched points $x_i \leftrightarrow y_j$ in a general position, estimate the most 'likely' fundamental matrix $\mathbf{F}$; (2) given $\mathbf{F}$ triangulate 3D point from $x_i \leftrightarrow y_j$.

$$\mathbf{x}_i = (u_i^1, v_i^1), \quad \mathbf{y}_i = (u_i^2, v_i^2), \qquad i = 1, 2, \ldots, k, \quad k \geq 8$$



image 1        image 2
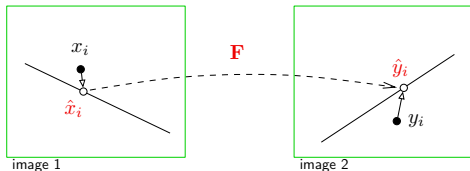
- <u>detected points</u> (measurements) $x_i$, $y_i$
- we introduce <u>matches</u> $\mathbf{Z}_i = (u_i^1, v_i^1, u_i^2, v_i^2) \in \mathbb{R}^4$; $\quad Z = \left\{ \mathbf{Z}_i \right\}_{i=1}^k$
- <u>corrected points</u> $\hat{x}_i$, $\hat{y}_i$; $\quad \hat{\mathbf{Z}}_i = (\hat{u}_i^1, \hat{v}_i^1, \hat{u}_i^2, \hat{v}_i^2)$; $\hat{Z} = \left\{ \hat{\mathbf{Z}}_i \right\}_{i=1}^k$ are <u>correspondences</u>
- correspondences satisfy the epipolar geometry exactly $\hat{\mathbf{y}}_i^\top \mathbf{F} \hat{\mathbf{x}}_i = 0$, $i = 1, \ldots, k$
- small correction is more probable
- let $\mathbf{e}_i(\cdot)$ be the '<u>reprojection error</u>' (vector) per match $i$,

$$\mathbf{e}_i(x_i, y_i \mid \hat{x}_i, \hat{y}_i, \mathbf{F}) = \begin{bmatrix} \mathbf{x}_i - \hat{\mathbf{x}}_i \\ \mathbf{y}_i - \hat{\mathbf{y}}_i \end{bmatrix} = \mathbf{e}_i(\mathbf{Z}_i \mid \hat{\mathbf{Z}}_i, \mathbf{F}) = \mathbf{Z}_i - \hat{\mathbf{Z}}_i(\mathbf{F})$$

(15)

$$\|\mathbf{e}_i(\cdot)\|^2 \overset{\text{def}}{=} \mathbf{e}_i^2(\cdot) = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 + \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2 = \|\mathbf{Z}_i - \hat{\mathbf{Z}}_i(\mathbf{F})\|^2$$

- the total reprojection error (of all data) then is

$$L(Z \mid \hat{Z}, \mathbf{F}) = \sum_{i=1}^{k} \mathbf{e}_i^2(x_i, y_i \mid \hat{x}_i, \hat{y}_i, \mathbf{F}) \quad = \sum_{i=1}^{k} \mathbf{e}_i^2(\mathbf{Z}_i \mid \hat{\mathbf{Z}}_i, \mathbf{F})$$

- and the optimization problem is

$$(\hat{Z}^*, \mathbf{F}^*) = \arg \min_{\substack{\mathbf{F} \\ \mathrm{rank}\, \mathbf{F} = 2}} \min_{\substack{\hat{Z} \\ \hat{\mathbf{y}}_i^\top \mathbf{F} \hat{\mathbf{x}}_i = 0}} \sum_{i=1}^{k} \mathbf{e}_i^2(x_i, y_i \mid \hat{x}_i, \hat{y}_i, \mathbf{F}) \tag{16}$$

### Three possible approaches

- they differ in how the correspondences $\hat{x}_i$, $\hat{y}_i$ are obtained:
  1. direct optimization of reprojection error over all variables $\hat{Z}$, $\mathbf{F}$
  2. Sampson optimal correction = partial correction of $\mathbf{Z}_i$ towards $\hat{\mathbf{Z}}_i$ used in an iterative minimization over $\mathbf{F}$
  3. removing $\hat{x}_i$, $\hat{y}_i$ altogether = marginalization of $L(Z, \hat{Z} \mid \mathbf{F})$ over $\hat{Z}$ followed by minimization over $\mathbf{F}$      not covered, the marginalization is difficult

# Method 1: Reprojection Error Optimization

- we need to encode the constraints $\underline{\hat{\mathbf{y}}}_i\, \mathbf{F}\, \underline{\hat{\mathbf{x}}}_i = 0$, $\operatorname{rank} \mathbf{F} = 2$
- <u>idea</u>: reconstruct 3D point via equivalent projection matrices and use reprojection error
- equivalent projection matrices are                 see [H&Z,Sec. 9.5] for complete characterization

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} [\mathbf{e}_2]_\times \mathbf{F} + \mathbf{e}_2 \mathbf{e}_1^\top & \mathbf{e}_2 \end{bmatrix} \tag{17}$$

⊛ H3; 2pt: Given $\mathbf{F}$, let $\mathbf{e}_1$, $\mathbf{e}_2$ be the right and left nullspace basis vectors of $\mathbf{F}$, respectively (i.e. the epipoles). Verify that $\mathbf{F}$ is a fundamental matrix of $\mathbf{P}_1$, $\mathbf{P}_2$ from (17). Hint: $\mathbf{A}$ is skew symmetric iff $\mathbf{x}^\top \mathbf{A} \mathbf{x} = 0$ for all vectors $\mathbf{x}$.

---

1. compute $\mathbf{F}^{(0)}$ by the 7-point algorithm →84; construct camera $\mathbf{P}_2^{(0)}$ from $\mathbf{F}^{(0)}$ using (17)
2. triangulate 3D points $\hat{\mathbf{X}}_i^{(0)}$ from matches $(x_i, y_i)$ for all $i = 1, \ldots, k$                 →89
3. starting from $\mathbf{P}_2^{(0)}$, $\hat{\mathbf{X}}^{(0)}$ minimize the reprojection error (15)

$$(\hat{\mathbf{X}}^*, \mathbf{P}_2^*) = \arg \min_{\mathbf{P}_2, \hat{\mathbf{X}}} \sum_{i=1}^k \mathbf{e}_i^2(\mathbf{Z}_i \mid \hat{\mathbf{Z}}_i(\hat{\mathbf{X}}_i, \mathbf{P}_2))$$

   where
$$\hat{\mathbf{Z}}_i = (\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i) \ \text{(Cartesian)}, \quad \underline{\hat{\mathbf{x}}}_i \simeq \mathbf{P}_1 \underline{\hat{\mathbf{X}}}_i, \ \underline{\hat{\mathbf{y}}}_i \simeq \mathbf{P}_2 \underline{\hat{\mathbf{X}}}_i \ \text{(homogeneous)}$$

   Non-linear, non-convex problem
4. compute $\mathbf{F}$ from $\mathbf{P}_1$, $\mathbf{P}_2^*$

- $3k + 12$ parameters to be found: latent: $\hat{\mathbf{X}}_i$, for all $i$ (correspondences!), non-latent: $\mathbf{P}_2$
- minimal representation: $3k + 7$ parameters, $\mathbf{P}_2 = \mathbf{P}_2(\mathbf{F})$                 →149
- there are pitfalls; this is essentially bundle adjustment; we will return to this later                 →137

**An elegant method for solving problems like** (16):

- we will get rid of the latent parameters $\hat{X}$ needed for obtaining the correction

  [H&Z, p. 287], [Sampson 1982]

- we will recycle the algebraic error $\varepsilon = \underline{\mathbf{y}}^\top \mathbf{F} \underline{\mathbf{x}}$ from →84

- consider matches $\mathbf{Z}_i$, correspondences $\hat{\mathbf{Z}}_i$, and reprojection error $\mathbf{e}_i = \|\mathbf{Z}_i - \hat{\mathbf{Z}}_i\|^2$

- correspondences satisfy $\underline{\hat{\mathbf{y}}}_i^\top \mathbf{F} \underline{\hat{\mathbf{x}}}_i = 0$,  $\hat{\mathbf{x}}_i = (\hat{u}^1, \hat{v}^1, 1)$, $\hat{\mathbf{y}}_i = (\hat{u}^2, \hat{v}^2, 1)$

- this is a manifold $\mathcal{V}_F \in \mathbb{R}^4$: a set of points $\hat{\mathbf{Z}} = (\hat{u}^1, \hat{v}^1, \hat{u}^2, \hat{v}^2)$ consistent with $\mathbf{F}$

- algebraic error vanishes for $\hat{\mathbf{Z}}_i$:  $\boxed{\mathbf{0} = \varepsilon_i(\hat{\mathbf{Z}}_i) = \underline{\hat{\mathbf{y}}}_i^\top \mathbf{F} \underline{\hat{\mathbf{x}}}_i}$
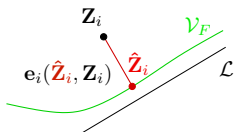


**Sampson's idea:** Linearize the algebraic error $\varepsilon(\mathbf{Z})$ at $\mathbf{Z}_i$ (where it is non-zero) and evaluate the resulting linear function at $\hat{\mathbf{Z}}_i$ (where it is zero). The zero-crossing replaces $\mathcal{V}_F$ by a linear manifold $\mathcal{L}$. The point on $\mathcal{V}_F$ closest to $\mathbf{Z}_i$ is replaced by the closest point on $\mathcal{L}$.

$$\varepsilon_i(\hat{\mathbf{Z}}_i) \approx \varepsilon_i(\mathbf{Z}_i) + \frac{\partial \varepsilon_i(\mathbf{Z}_i)}{\partial \mathbf{Z}_i} (\hat{\mathbf{Z}}_i - \mathbf{Z}_i)$$

# ►Sampson's Approximation of Reprojection Error

- linearize $\varepsilon(\mathbf{Z})$ at match $\mathbf{Z}_i$, evaluate it at correspondence $\hat{\mathbf{Z}}_i$

$$\varepsilon_i(\mathbf{Z}_i) + \underbrace{\frac{\partial \varepsilon_i(\mathbf{Z}_i)}{\partial \mathbf{Z}_i}}_{\mathbf{J}_i(\mathbf{Z}_i)} \underbrace{(\hat{\mathbf{Z}}_i - \mathbf{Z}_i)}_{\mathbf{e}_i(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)} \overset{\text{def}}{=} \varepsilon_i(\mathbf{Z}_i) + \mathbf{J}_i(\mathbf{Z}_i)\,\mathbf{e}_i(\hat{\mathbf{Z}}_i, \mathbf{Z}_i) = \varepsilon_i(\hat{\mathbf{Z}}_i) \overset{!}{=} 0$$

- goal: compute <u>function</u> $\mathbf{e}_i(\cdot)$ from $\varepsilon_i(\cdot)$, where $\mathbf{e}_i(\cdot)$ is the distance of $\hat{\mathbf{Z}}_i$ from $\mathbf{Z}_i$
- we have a linear <u>underconstrained</u> equation for $\mathbf{e}_i(\cdot)$
- we look for a minimal $\mathbf{e}_i(\cdot)$ per match $i$

$$\mathbf{e}_i(\cdot)^* = \arg\min_{\mathbf{e}_i(\cdot)} \|\mathbf{e}_i(\cdot)\|^2 \quad \text{subject to} \quad \varepsilon_i(\cdot) + \mathbf{J}_i(\cdot)\,\mathbf{e}_i(\cdot) = 0$$

- which has a closed-form solution <span style="color:blue">note that $\mathbf{J}_i(\cdot)$ is not invertible!</span> <span style="color:green">⊛ P1; 1pt: derive $\mathbf{e}_i^*(\cdot)$</span>

$$\mathbf{e}_i^*(\cdot) = -\mathbf{J}_i^\top (\mathbf{J}_i \mathbf{J}_i^\top)^{-1} \varepsilon_i(\cdot) \qquad \text{pseudo-inverse}$$
$$\|\mathbf{e}_i^*(\cdot)\|^2 = \varepsilon_i^\top(\cdot)(\mathbf{J}_i \mathbf{J}_i^\top)^{-1} \varepsilon_i(\cdot) \tag{18}$$

- this maps $\varepsilon_i(\cdot)$ to an estimate of $\mathbf{e}_i(\cdot)$ per correspondence
- we often do not need $\mathbf{e}_i$, just $\|\mathbf{e}_i\|^2$ <span style="color:blue">exception: triangulation →105</span>
- the unknown parameters $\mathbf{F}$ are inside: $\mathbf{e}_i = \mathbf{e}_i(\mathbf{F})$, $\varepsilon_i = \varepsilon_i(\mathbf{F})$, $\mathbf{J}_i = \mathbf{J}_i(\mathbf{F})$

## ►Example: Fitting A Circle To Scattered Points

**Problem:** Fit an origin-centered circle $\mathcal{C}$: $\|\mathbf{x}\|^2 - r^2 = 0$ to a set of 2D points $Z = \{x_i\}_{i=1}^k$

1. consider radial error as the 'algebraic error' $\quad \varepsilon(\mathbf{x}) = \|\mathbf{x}\|^2 - r^2 \qquad$ 'arbitrary' choice

2. linearize it at $\hat{\mathbf{x}}$ $\qquad\qquad\qquad\qquad$ we are dropping $i$ in $\varepsilon_i$, $\mathbf{e}_i$ etc for clarity

$$\varepsilon(\hat{\mathbf{x}}) \approx \varepsilon(\mathbf{x}) + \underbrace{\frac{\partial \varepsilon(\mathbf{x})}{\partial \mathbf{x}}}_{\mathbf{J}(\mathbf{x}) = 2\mathbf{x}^\top} \underbrace{(\hat{\mathbf{x}} - \mathbf{x})}_{\mathbf{e}(\hat{\mathbf{x}}, \mathbf{x})} = \cdots = 2\mathbf{x}^\top \hat{\mathbf{x}} - (r^2 + \|\mathbf{x}\|^2) \stackrel{\text{def}}{=} \varepsilon_L(\hat{\mathbf{x}})$$

$\varepsilon_L(\hat{\mathbf{x}}) = 0$ is a line with normal $\frac{\mathbf{x}}{\|\mathbf{x}\|}$ and intercept $\frac{r^2 + \|\mathbf{x}\|^2}{2\|\mathbf{x}\|}$ $\qquad$ not tangent to $\mathcal{C}$, outside!

3. using (18), express error approximation $\mathbf{e}^*$ as

$$\|\mathbf{e}^*\|^2 = \varepsilon^\top (\mathbf{J}\mathbf{J}^\top)^{-1} \varepsilon = \frac{(\|\mathbf{x}\|^2 - r^2)^2}{4\|\mathbf{x}\|^2}$$

4. fit circle
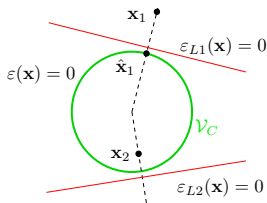


$$r^* = \arg\min_r \sum_{i=1}^k \frac{(\|\mathbf{x}_i\|^2 - r^2)^2}{4\|\mathbf{x}_i\|^2} = \cdots = \left( \frac{1}{k} \sum_{i=1}^k \frac{1}{\|\mathbf{x}_i\|^2} \right)^{-\frac{1}{2}}$$

- this example results in a convex quadratic optimization problem
- note that

$$\arg\min_r \sum_{i=1}^k (\|\mathbf{x}_i\|^2 - r^2)^2 = \left( \frac{1}{k} \sum_{i=1}^k \|\mathbf{x}_i\|^2 \right)^{\frac{1}{2}}$$

# Circle Fitting: Some Results



| medium radial noise | big radial noise | medium isotropic noise | big isotropic noise |
|---|---|---|---|
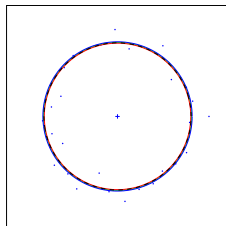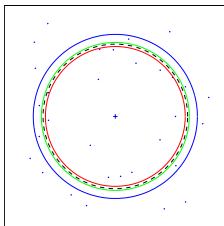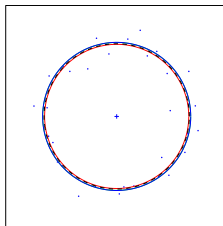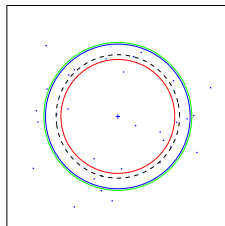
opt: 1.8, Smp: 1.9, dir: 2.3     1.6, 1.8, 2.6     1.8, 2.0, 2.2     1.6, 2.0, 2.4

mean ranks over 10 000 random trials with $k = 32$ samples

green – ground truth
red – Sampson error $\mathbf{e}$ minimizer
blue – direct radial error $\boldsymbol{\varepsilon}$ minimizer
black – optimal estimator for isotropic error

optimal estimator for isotropic error (black, dashed):

$$r \approx \frac{3}{4k} \sum_{i=1}^{k} \|\mathbf{x}_i\| + \sqrt{\left( \frac{3}{4k} \sum_{i=1}^{k} \|\mathbf{x}_i\| \right)^2 - \frac{1}{2k} \sum_{i=1}^{k} \|\mathbf{x}_i\|^2}$$

## which method is better?

- error should model noise, radial noise and isotropic noise behave differently
- ground truth: Normally distributed isotropic error, Gamma-distributed radial error
- Sampson: better for the radial distribution model; Direct: better for the isotropic model
- no matter how corrected, the algebraic error minimizer is not an unbiased parameter estimator
  Cramér-Rao bound tells us how close one can get with unbiased estimator and given $k$

# Discussion: On The Art of Probabilistic Model Design...

- a few models for fitting zero-centered circle $C$ of radius $r$ to points in $\mathbb{R}^2$

| | marginalized over $C$ | orthogonal deviation from $C$ | Sampson approximation |
|---|---|---|---|
| **error model** |  $N(\mathbf{0}, \sigma^2 \mathbf{I})$ |  $\Gamma(\cdot, \cdot)$ |  $N(\mathbf{0}, \sigma^2 \mathbf{I})$ |
| **radial p.d.f.** |  |  |  |
| **random sample** |  |  |  |
| $p(\mathbf{x} \mid r)$ | $\approx \dfrac{1}{\sigma\sqrt{(2\pi)^3 r \, \|\mathbf{x}\|}} \, e^{-\frac{(\|\mathbf{x}\|-r)^2}{2\sigma^2}}$ | $\dfrac{1}{2\pi\Gamma(\frac{r^2}{\sigma})} \dfrac{1}{\|\mathbf{x}\|^2} \left(\dfrac{r\|\mathbf{x}\|}{\sigma}\right)^{\frac{r^2}{\sigma}} e^{-\frac{r\|\mathbf{x}\|}{\sigma}}$ | $\dfrac{1}{r\sigma\sqrt{(2\pi)^3}} \, e^{-\frac{e^2(\mathbf{x};r)}{2\sigma^2}}$ |
| | • mode inside the circle<br>• models the inside well<br>• tends to normal distrib. | • peak at the center<br>• unusable for small radii<br>• tends to Dirac distrib. | • mode at the circle<br>• hole at the center<br>• tends to normal distrib. |
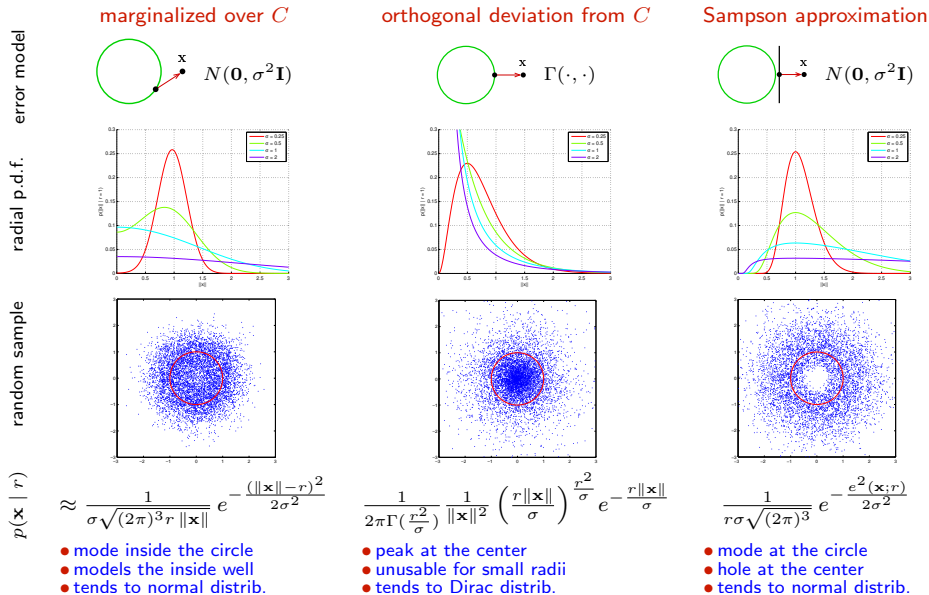
The epipolar algebraic error is

$$\varepsilon_i(\mathbf{F}) = \underline{\mathbf{y}}_i^\top \mathbf{F} \underline{\mathbf{x}}_i, \quad \mathbf{x}_i = (u_i^1, v_i^1), \quad \mathbf{y}_i = (u_i^2, v_i^2), \qquad \varepsilon_i \in \mathbb{R}$$

Let $\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 \end{bmatrix}$ (per columns) $= \begin{bmatrix} (\mathbf{F}^1)^\top \\ (\mathbf{F}^2)^\top \\ (\mathbf{F}^3)^\top \end{bmatrix}$ (per rows), $\quad \mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, then

**Sampson**

$$\mathbf{J}_i(\mathbf{F}) = \left[ \frac{\partial \varepsilon_i(\mathbf{F})}{\partial u_i^1}, \ \frac{\partial \varepsilon_i(\mathbf{F})}{\partial v_i^1}, \ \frac{\partial \varepsilon_i(\mathbf{F})}{\partial u_i^2}, \ \frac{\partial \varepsilon_i(\mathbf{F})}{\partial v_i^2} \right] \qquad\qquad \mathbf{J}_i \in \mathbb{R}^{1,4} \qquad \text{derivatives over point coordinates}$$

$$= \left[ (\mathbf{F}_1)^\top \underline{\mathbf{y}}_i, \ (\mathbf{F}_2)^\top \underline{\mathbf{y}}_i, \ (\mathbf{F}^1)^\top \underline{\mathbf{x}}_i, \ (\mathbf{F}^2)^\top \underline{\mathbf{x}}_i \right] = \begin{bmatrix} \mathbf{S}\mathbf{F}^\top \underline{\mathbf{y}}_i \\ \mathbf{S}\mathbf{F}\underline{\mathbf{x}}_i \end{bmatrix}^\top$$

$$\mathbf{e}_i(\mathbf{F}) = -\frac{\mathbf{J}_i(\mathbf{F})\,\varepsilon_i(\mathbf{F})}{\|\mathbf{J}_i(\mathbf{F})\|^2} \qquad\qquad\qquad \mathbf{e}_i(\mathbf{F}) \in \mathbb{R}^4 \qquad \text{Sampson error vector}$$

$$e_i(\mathbf{F}) \stackrel{\text{def}}{=} \|\mathbf{e}_i(\mathbf{F})\| = \frac{\varepsilon_i(\mathbf{F})}{\|\mathbf{J}_i(\mathbf{F})\|} = \frac{\underline{\mathbf{y}}_i^\top \mathbf{F}\underline{\mathbf{x}}_i}{\sqrt{\|\mathbf{S}\mathbf{F}\underline{\mathbf{x}}_i\|^2 + \|\mathbf{S}\mathbf{F}^\top \underline{\mathbf{y}}_i\|^2}} \qquad e_i(\mathbf{F}) \in \mathbb{R} \qquad \text{scalar Sampson error}$$

- Sampson error 'normalizes' the algebraic error
- automatically copes with multiplicative factors $\mathbf{F} \mapsto \lambda \mathbf{F}$
- actual optimization not yet covered →109

Given $\mathbf{P}_1$, $\mathbf{P}_2$ and a correspondence $x \leftrightarrow y$, look for 3D point $\mathbf{X}$ projecting to $x$ and $y$ →89

**Idea:**

1. if not given, compute $\mathbf{F}$ from $\mathbf{P}_1$, $\mathbf{P}_2$, e.g. $\mathbf{F} = (\mathbf{Q}_1 \mathbf{Q}_2^{-1})^\top [\mathbf{q}_1 - (\mathbf{Q}_1 \mathbf{Q}_2^{-1})\mathbf{q}_2]_\times$

2. correct the measurement by the linear estimate of the correction vector →100

$$
\begin{bmatrix} \hat{u}^1 \\ \hat{v}^1 \\ \hat{u}^2 \\ \hat{v}^2 \end{bmatrix} \approx \begin{bmatrix} u^1 \\ v^1 \\ u^2 \\ v^2 \end{bmatrix} - \frac{\varepsilon}{\|\mathbf{J}\|^2} \mathbf{J}^\top = \begin{bmatrix} u^1 \\ v^1 \\ u^2 \\ v^2 \end{bmatrix} - \frac{\mathbf{y}^\top \mathbf{F}\underline{\mathbf{x}}}{\|\mathbf{SF}\underline{\mathbf{x}}\|^2 + \|\mathbf{SF}^\top\underline{\mathbf{y}}\|^2} \begin{bmatrix} (\mathbf{F}_1)^\top \mathbf{y} \\ (\mathbf{F}_2)^\top \mathbf{y} \\ (\mathbf{F}^1)^\top \underline{\mathbf{x}} \\ (\mathbf{F}^2)^\top \underline{\mathbf{x}} \end{bmatrix}
$$

3. use the SVD triangulation algorithm with numerical conditioning →90

**Ex** (cont'd from →93):
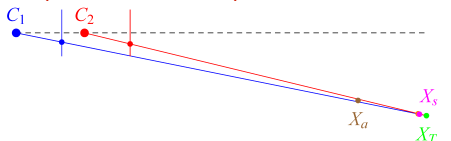


$X_T$ – noiseless ground truth position
● – reprojection error minimizer
$X_s$ – Sampson-corrected algebraic error minimizer
$X_a$ – algebraic error minimizer
$m$ – measurement ($m_T$ with noise in $v^2$)
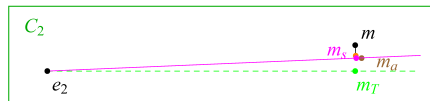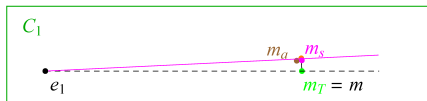
## ▶Back to Fundamental Matrix Estimation

**Goal:** Given a set $X = \{(x_i, y_i)\}_{i=1}^k$ of $k \gg 7$ inlier correspondences, compute a statistically efficient estimate for fundamental matrix $\mathbf{F}$.

### What we have so far

- 7-point algorithm for $\mathbf{F}$ (5-point algorithm for $\mathbf{E}$) →84
- definition of Sampson error per correspondence $e_i(\mathbf{F} \mid x_i, y_i)$ →104
- triangulation requiring an <u>optimal</u> $\mathbf{F}$

### What we need

- an optimization algorithm for

$$\mathbf{F}^* = \arg \min_{\mathbf{F}} \sum_{i=1}^k e_i^2(\mathbf{F} \mid X)$$

- the 7-point estimate is a good starting point $\mathbf{F}_0$

## Levenberg-Marquardt (LM) Iterative Estimation in a Nutshell

Consider error function $\quad \mathbf{e}_i(\boldsymbol{\theta}) = f(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}) \in \mathbb{R}^m, \quad$ with $\mathbf{x}_i, \mathbf{y}_i$ given, $\boldsymbol{\theta} \in \mathbb{R}^q$ unknown

$$\theta = \mathbf{F}, \; q = 9, \; m = 1 \text{ for f.m. estimation}$$

**Our goal:** $\quad \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{k} \|\mathbf{e}_i(\boldsymbol{\theta})\|^2$

**Idea 1** (Gauss-Newton approximation): proceed iteratively for $s = 0, 1, 2, \ldots$

$$\boldsymbol{\theta}^{s+1} := \boldsymbol{\theta}^s + \mathbf{d}_s, \quad \text{where} \quad \mathbf{d}_s = \arg\min_{\mathbf{d}} \sum_{i=1}^{k} \|\mathbf{e}_i(\boldsymbol{\theta}^s + \mathbf{d})\|^2 \tag{19}$$

$$\mathbf{e}_i(\boldsymbol{\theta}^s + \mathbf{d}) \approx \mathbf{e}_i(\boldsymbol{\theta}^s) + \mathbf{L}_i \, \mathbf{d},$$

$$(\mathbf{L}_i)_{jl} = \frac{\partial \big(\mathbf{e}_i(\boldsymbol{\theta})\big)_j}{\partial (\boldsymbol{\theta})_l}, \qquad \mathbf{L}_i \in \mathbb{R}^{m,q} \qquad \text{typically a long matrix, } m \ll q$$

Then the solution to Problem (19) is a set of 'normal' eqs

$$-\underbrace{\sum_{i=1}^{k} \mathbf{L}_i^{\top} \mathbf{e}_i(\boldsymbol{\theta}^s)}_{\mathbf{e} \in \mathbb{R}^{q,1}} = \underbrace{\left( \sum_{i=1}^{k} \mathbf{L}_i^{\top} \mathbf{L}_i \right)}_{\mathbf{L} \in \mathbb{R}^{q,q}} \mathbf{d}_s, \tag{20}$$

- $\mathbf{d}_s$ can be solved for by Gaussian elimination using Choleski decomposition of $\mathbf{L}$
  $\mathbf{L}$ symmetric PSD $\Rightarrow$ use Choleski, almost $2\times$ faster than Gauss-Seidel, see bundle adjustment $\rightarrow$140
- such updates do not lead to stable convergence $\longrightarrow$ ideas of Levenberg and Marquardt

**Idea 2** (Levenberg): replace $\sum_i \mathbf{L}_i^\top \mathbf{L}_i$ with $\sum_i \mathbf{L}_i^\top \mathbf{L}_i + \lambda \mathbf{I}$ for some damping factor $\lambda \geq 0$

**Idea 3** (Marquardt): replace $\lambda \mathbf{I}$ with $\lambda \sum_i \mathrm{diag}(\mathbf{L}_i^\top \mathbf{L}_i)$ to adapt to local curvature:

$$-\sum_{i=1}^{k} \mathbf{L}_i^\top \mathbf{e}_i(\boldsymbol{\theta}^s) = \left( \sum_{i=1}^{k} \left( \mathbf{L}_i^\top \mathbf{L}_i + \lambda \, \mathrm{diag}(\mathbf{L}_i^\top \mathbf{L}_i) \right) \right) \mathbf{d}_s$$

**Idea 4** (Marquardt): adaptive $\lambda$ $\qquad$ small $\lambda \to$ Gauss-Newton, large $\lambda \to$ gradient descend

1. choose $\lambda \approx 10^{-3}$ and compute $\mathbf{d}_s$
2. if $\sum_i \|\mathbf{e}_i(\boldsymbol{\theta}^s + \mathbf{d}_s)\|^2 < \sum_i \|\mathbf{e}_i(\boldsymbol{\theta}^s)\|^2$ then accept $\mathbf{d}_s$ and set $\lambda := \lambda/10$, $s := s + 1$
3. otherwise set $\lambda := 10\lambda$ and recompute $\mathbf{d}_s$

- sometimes different constants are needed for the 10 and $10^{-3}$
- note that $\mathbf{L}_i \in \mathbb{R}^{m,q}$ (long matrix) but each contribution $\mathbf{L}_i^\top \mathbf{L}_i$ is a square singular $q \times q$ matrix (always singular for $k < q$)
- error can be made robust to outliers, see the trick $\to$112
- we have approximated the least squares Hessian by ignoring second derivatives of the error function (Gauss-Newton approximation) $\qquad$ See [Triggs et al. 1999, Sec. 4.3]
- $\lambda$ helps avoid the consequences of gauge freedom $\to$142
- modern variants of LM are Trust Region methods

# LM with Sampson Error for Fundamental Matrix Estimation

**Sampson** (derived by linearization over point coordinates $u^1, v^1, u^2, v^2$)

$$e_i(\mathbf{F}) = \frac{\varepsilon_i}{\|\mathbf{J}_i\|} = \frac{\underline{\mathbf{y}}_i^\top \mathbf{F} \underline{\mathbf{x}}_i}{\sqrt{\|\mathbf{S}\mathbf{F}\underline{\mathbf{x}}_i\|^2 + \|\mathbf{S}\mathbf{F}^\top \underline{\mathbf{y}}_i\|^2}} \quad \text{where} \quad \mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

**LM** (by linearization over parameters $\mathbf{F}$)

$$\mathbf{L}_i = \frac{\partial e_i(\mathbf{F})}{\partial \mathbf{F}} = \cdots = \frac{1}{2\|\mathbf{J}_i\|} \left[ \left( \underline{\mathbf{y}}_i - \frac{2e_i}{\|\mathbf{J}_i\|} \mathbf{S}\mathbf{F}\underline{\mathbf{x}}_i \right) \underline{\mathbf{x}}_i^\top + \underline{\mathbf{y}}_i \left( \underline{\mathbf{x}}_i - \frac{2e_i}{\|\mathbf{J}_i\|} \mathbf{S}\mathbf{F}^\top \underline{\mathbf{y}}_i \right)^\top \right] \quad (21)$$

- $\mathbf{L}_i$ in (21) is a $3 \times 3$ matrix, must be reshaped to dimension-9 vector $\mathrm{vec}(\mathbf{L}_i)$ to be used in LM
- $\underline{\mathbf{x}}_i$ and $\underline{\mathbf{y}}_i$ in Sampson error are normalized to unit homogeneous coordinate   (21) relies on this
- reinforce $\mathrm{rank}\,\mathbf{F} = 2$ after each LM update to stay in the fundamental matrix manifold and $\|\mathbf{F}\| = 1$ to avoid gauge freedom                                    by SVD →110
- LM linearization could be done by numerical differentiation (we have a small dimension here)

**Summary so far**

- Given a set $X = \{(x_i, y_i)\}_{i=1}^k$ of $k \gg 7$ inlier correspondences, compute a statistically efficient estimate for fundamental matrix $\mathbf{F}$.
  1. Find the conditioned ($\rightarrow$92) 7-point $\mathbf{F}_0$ ($\rightarrow$84) from a suitable 7-tuple
  2. Improve the $\mathbf{F}_0^*$ using the LM optimization ($\rightarrow$107–108) and the Sampson error ($\rightarrow$109) on all inliers, reinforce rank-2, unit-norm $\mathbf{F}_k^*$ after each LM iteration using SVD

**Partial conceptualization**

- inlier = correspondence
- outlier = non-correspondence
- binary inlier/outlier labels are hidden
- we can get a likely estimate only, with respect to a model

**We are not yet done**

- if there are no wrong correspondences (mismatches, outliers), this gives a local optimum given the 7-point initial estimate

- the algorithm breaks under contamination of (inlier) correspondences by outliers

- the full problem involves finding the inliers!

- in addition, we need a mechanism for jumping out of local minima (and exploring the space of all fundamental matrices)

**Problem:** Given image point sets $X = \{x_i\}_{i=1}^m$ and $Y = \{y_j\}_{j=1}^n$ and their descriptors $D$, find the most probable

1. inlier keypoints $S_X \subseteq X$, $S_Y \subseteq Y$
2. one-to-one perfect matching $M: S_X \to S_Y$    *perfect matching: 1-factor of the bipartite graph*
3. fundamental matrix $\mathbf{F}$ such that $\operatorname{rank} \mathbf{F} = 2$
4. such that for each $x_i \in S_X$ and $y_j = M(x_i)$ it is probable that
   a) the image descriptor $D(x_i)$ is similar to $D(y_j)$, and
   b) the total reprojection error $E = \sum_{ij} e_{ij}^2 (\mathbf{F})$ is small    *note a slight change in notation: $e_{ij}$*
5. inlier-outlier and outlier-outlier matches are improbable
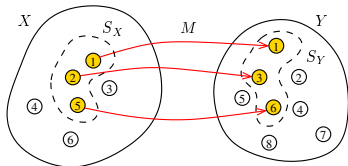


$$(M^*, \mathbf{F}^*) = \arg\max_{M, \mathbf{F}} p(E, D, \mathbf{F} \mid M) \, P(M) \tag{22}$$

- probabilistic model: an efficient language for problem formulation    *it also unifies 4.a and 4.b*
- the (22) is a Bayesian probabilistic model    *there is a constant number of random variables!*
- binary matching table $M_{ij} \in \{0, 1\}$ of fixed size $m \times n$
  - each row/column contains at most one unity
  - zero rows/columns correspond to unmatched point $x_i/y_j$

## Deriving A Robust Matching Model by Approximate Marginalization

For algorithmic efficiency, instead of $(M^*, \mathbf{F}^*) = \arg\max_{M,\mathbf{F}} p(E, D, \mathbf{F} \mid M)\, P(M)$ solve

$$\mathbf{F}^* = \arg\max_{\mathbf{F}} p(E, D, \mathbf{F}) \tag{23}$$

by <u>marginalization</u> of $p(E, D, \mathbf{F} \mid M)\, P(M)$ over $M$       <span style="color:blue">this changes the problem!</span>

drop the assumption that $M$ are 1:1 matchings, assume correspondence-wise independence:

$$p(E, D, \mathbf{F} \mid M)P(M) = \prod_{i=1}^{m}\prod_{j=1}^{n} p_e(e_{ij}, d_{ij}, \mathbf{F} \mid m_{ij})P(m_{ij})$$

- $e_{ij}$ represents (reprojection) error for match $x_i \leftrightarrow y_i$: $e_{ij}(x_i, y_i, \mathbf{F})$
- $d_{ij}$ represents descriptor similarity for match $x_i \leftrightarrow y_i$: $d_{ij} = \|\mathbf{d}(x_i) - \mathbf{d}(y_j)\|$

<u>Approximate marginalization:</u>       <span style="color:blue">take all the $2^{mn}$ terms in place of $M$</span>

$$p(E, D, \mathbf{F}) \approx \sum_{m_{11} \in \{0,1\}} \sum_{m_{12}} \cdots \sum_{m_{mn}} p(E, D, \mathbf{F} \mid M)P(M) =$$

$$= \sum_{m_{11}} \cdots \sum_{m_{mn}} \prod_{i=1}^{m}\prod_{j=1}^{n} p_e(e_{ij}, d_{ij}, \mathbf{F} \mid m_{ij})P(m_{ij}) \;\overset{\circledast\,1}{=}\; \cdots =$$

$$= \prod_{i=1}^{m}\prod_{j=1}^{n} \underbrace{\sum_{m_{ij} \in \{0,1\}} p_e(e_{ij}, d_{ij}, \mathbf{F} \mid m_{ij})P(m_{ij})}_{\text{we will continue with this term}}$$

$$\sum_{m_{ij} \in \{0,1\}} p_e(e_{ij}, d_{ij}, \mathbf{F} \mid m_{ij}) P(m_{ij}) =$$

$$= \underbrace{p_e(e_{ij}, d_{ij}, \mathbf{F} \mid m_{ij} = 1)}_{p_1(e_{ij}, d_{ij}, \mathbf{F})} \underbrace{P(m_{ij} = 1)}_{1 - P_0} + \underbrace{p_e(e_{ij}, d_{ij}, \mathbf{F} \mid m_{ij} = 0)}_{p_0(e_{ij}, d_{ij}, \mathbf{F})} \underbrace{P(m_{ij} = 0)}_{P_0} =$$

$$= (1 - P_0)\, p_1(e_{ij}, d_{ij}, \mathbf{F}) + P_0\, p_0(e_{ij}, d_{ij}, \mathbf{F}) \quad (24)$$

- the $p_0(e_{ij}, d_{ij}, \mathbf{F})$ is a penalty for 'missing a correspondence' but it should be a p.d.f. (cannot be a constant) <span>($\rightarrow$114 for a simplification)</span>

$$\text{choose} \quad P_0 \rightarrow 1, \quad p_0(\cdot) \rightarrow 0 \quad \text{so that} \quad \frac{P_0}{1 - P_0}\, p_0(\cdot) \approx \text{const}$$

- the $p_1(e_{ij}, d_{ij}, \mathbf{F})$ is typically an easy-to-design term: assuming independence of reprojection error and descriptor similarity:

$$p_1(e_{ij}, d_{ij}, \mathbf{F}) = p_1(e_{ij} \mid \mathbf{F})\, p_F(\mathbf{F})\, p_1(d_{ij})$$

- we choose, e.g.

$$p_1(e_{ij} \mid \mathbf{F}) = \frac{1}{T_e(\sigma_1)}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}}, \quad p_1(d_{ij}) = \frac{1}{T_d(\sigma_d, \dim \mathbf{d})}\, e^{-\frac{\|\mathbf{d}(x_i) - \mathbf{d}(y_j)\|^2}{2\sigma_d^2}} \quad (25)$$

- $\mathbf{F}$ is a random variable and $\sigma_1$, $\sigma_d$, $P_0$ are parameters
- the form of $T(\sigma_1)$ depends on error definition, it may depend on $x_i$, $y_j$ but not on $\mathbf{F}$
- we will continue with the result from (24)

# ▶Simplified Robust Energy (Error) Function

- assuming the choice of $p_1$ as in (25), we are simplifying

$$p(E, D, \mathbf{F}) = p(E, D \mid \mathbf{F})\, p_F(\mathbf{F}) =$$
$$= p_F(\mathbf{F}) \prod_{i=1}^{m} \prod_{j=1}^{n} \Big[ (1 - P_0)\, p_1(e_{ij}, d_{ij} \mid \mathbf{F}) + P_0\, p_0(e_{ij}, d_{ij} \mid \mathbf{F}) \Big]$$

- we choose $\sigma_0 \gg \sigma_1$ and omit $d_{ij}$ for simplicity; then the square-bracket term is

$$\frac{1 - P_0}{T_e(\sigma_1)}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + \frac{P_0}{T_e(\sigma_0)}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_0^2}}$$

- we define the 'potential function' as: $V(x) = -\log p(x)$, then

$$V(E, D \mid \mathbf{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \Bigg[ \underbrace{-\log \frac{1 - P_0}{T_e(\sigma_1)}}_{\Delta\, =\, \text{const}} - \log\Big( e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + \underbrace{\frac{P_0}{1 - P_0}\, \frac{T_e(\sigma_1)}{T_e(\sigma_0)}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_0^2}}}_{t\, \approx\, \text{const}} \Big) \Bigg] =$$

$$= m\, n\, \Delta + \sum_{i=1}^{m} \sum_{j=1}^{n} \underbrace{-\log\Big( e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + t \Big)}_{\hat{V}(e_{ij})} \quad (26)$$

- note we are summing over <u>all</u> $m\,n$ matches ($m, n$ are constant!)
- when $t = 0$ we have quadratic error function $\hat{V}(e_{ij}) = e_{ij}^2(\mathbf{F})/(2\sigma_1^2)$

**Example for $\hat{V}(e_{ij})$ from (26):**



$\sigma_1 = 1$
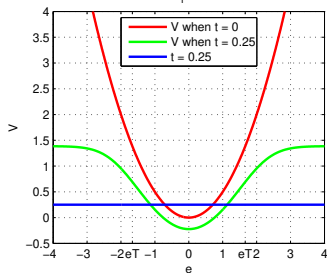
red – the (non-robust) quadratic error    $\hat{V}(e_{ij})$ when $t = 0$

blue – the rejected match penalty $t$

green – robust $\hat{V}(e_{ij})$ from (26)

- if the error of a correspondence exceeds a limit, it is ignored
- then $\hat{V}(e_{ij}) = \text{const}$ and we just count outliers in (26)
- $t$ controls the 'turn-off' point
- the inlier/outlier threshold is $e_T$ – the error for which
  $(1 - P_0)\, p_1(e_T) = P_0\, p_0(e_T)$:    note that $t \approx 0$

$$e_T = \sigma_1 \sqrt{-\log t^2}, \quad t = e^{-\frac{1}{2}\left(\frac{e_T}{\sigma_1}\right)^2} \text{ e.g. } e_T = 4\sigma_1 \quad (27)$$

**The full optimization problem (23) uses (26):**

$$\mathbf{F}^* = \arg\max_{\mathbf{F}} \underbrace{\frac{\overbrace{p(E, D \mid \mathbf{F})}^{\text{data model}} \cdot \overbrace{p(\mathbf{F})}^{\text{prior}}}{\underbrace{p(E, D)}_{\text{evidence}}} \approx \arg\min_{\mathbf{F}} \left[ V(\mathbf{F}) + \sum_{i=1}^{m} \sum_{j=1}^{n} \log\left( e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + t \right) \right]$$

- $\pi(\mathbf{F})$ – a shorthand for the argument of the maximization
- typically we take $V(\mathbf{F}) = -\log p(\mathbf{F}) = 0$ unless we need to stabilize a computation, e.g. when video camera moves smoothly (on a high-mass vehicle) and we have a prediction for $\mathbf{F}$
- evidence is not needed unless we want to compare different models (e.g. homography vs. epipolar geometry)

# How To Find the Global Maxima (Modes) of a PDF?



- given the function $p(x)$ at left

p.d.f. on $[0, 1]$, mode at $0.1$

**consider several methods:**

1. exhaustive search

```
step = 1/(iterations-1);
for x = 0:step:1
 if p(x) > bestp
  bestx = x; bestp = p(x);
 end
end
```

- slow algorithm (definite quantization)
- fast to implement

2. randomized search with uniform sampling

```
while t < iterations
 x = rand(1);
 if p(x) > bestp
  bestx = x; bestp = p(x);
 end
 t = t+1;  % time
end
```

- equally slow algorithm
- fast to implement

3. random sampling from $p(x)$ (Gibbs sampler)
   - faster algorithm • fast to implement but often infeasible (e.g. when $p(x)$ is data dependent (our case in correspondence prob.))

4. Metropolis-Hastings sampling
   - almost as fast (with care) • not so fast to implement
   - rarely infeasible • RANSAC belongs here

- averaged over $10^4$ trials
- number of proposals before $|x - x_{\text{true}}| \leq \text{step}$

# How To Generate Random Samples from a Complex Distribution?



target (red) and scaled proposal (blue) distributions
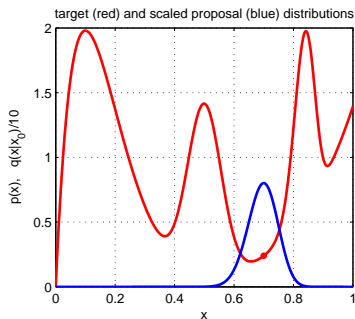
- red: probability density function $\pi(x)$ of the toy distribution on the unit interval    *target distribution*

$$\pi(x) = \sum_{i=1}^{4} \gamma_i \, \mathrm{Be}(x; \alpha_i, \beta_i), \quad \sum_{i=1}^{4} \gamma_i = 1, \ \gamma_i \geq 0$$

$$\mathrm{Be}(x; \alpha, \beta) = \frac{1}{\mathrm{B}(\alpha, \beta)} \cdot x^{\alpha-1}(1-x)^{\beta-1}$$

- alg. for generating samples from $\mathrm{Be}(x; \alpha, \beta)$ is known
- $\Rightarrow$ we can generate samples from $\pi(x)$    how?

- suppose we cannot sample from $\pi(x)$ but we can sample from some 'simple' <u>proposal distribution</u> $q(x \mid x_0)$, given the previous sample $x_0$ (blue)

$$q(x \mid x_0) = \begin{cases} \mathrm{U}_{0,1}(x) & \text{(independent) uniform sampling} \\ \mathrm{Be}(x; \frac{x_0}{T}+1, \frac{1-x_0}{T}+1) & \text{'beta' diffusion (crawler)} \quad T - temperature \\ \pi(x) & \text{(independent) Gibbs sampler} \end{cases}$$

- note we have unified all the random sampling methods from the previous slide
- how to redistribute proposal samples $q(x \mid x_0)$ to target distribution $\pi(x)$ samples?

# ►Metropolis-Hastings (MH) Sampling

$C, S$ – configurations (of all variable values)  e.g. $C = x$ and $\pi(C) = \pi(x)$ from →117

**Goal:** Generate a sequence of random samples $\{C_t\}$ from target distribution $\pi(C)$

- setup a Markov chain with a suitable transition probability to generate the sequence

**Sampling procedure**

1. given current config. $C_t$, draw a random config. sample $S$ from $q(S \mid C_t)$
   $q$ may use some information from $C_t$ (Hastings)
2. compute acceptance probability  the evidence term drops out

$$a = \min\left\{1, \ \frac{\pi(S)}{\pi(C_t)} \cdot \frac{q(C_t \mid S)}{q(S \mid C_t)}\right\}$$

3. draw a random number $u$ from unit-interval uniform distribution $\mathrm{U}_{0,1}$
4. if $u \le a$ then $C_{t+1} := S$ else $C_{t+1} := C_t$

**'Programming' an MH sampler**

1. design a proposal distribution (mixture) $q$ and a sampler from $q$
2. write functions $q(C_t \mid S)$ and $q(S \mid C_t)$ that are proper distributions  not always simple

**Finding the mode**

- remember the best sample  fast implementation but must wait long to hit the mode
- use simulated annealing  very slow
- start local optimization from the best sample  good trade-off between speed and accuracy
  an optimal algorithm does not use just the best sample: a Stochastic EM Algorithm (e.g. SAEM)

# MH Sampling Demo
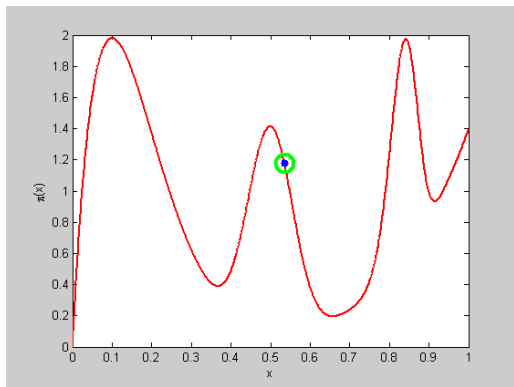


sampling process (video, 7:33, 100k samples)



initial sample



final distribution
of visited states

- blue point: current sample
- green circle: best sample so far $\qquad$ quality $= \pi(x)$
- histogram: current distribution of visited states
- the vicinity of modes are the most often visited states
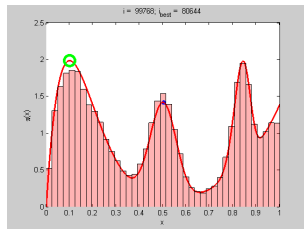
## Demo Source Code (Matlab)

```matlab
function x = proposal_gen(x0)
% proposal generator q(x | x0)

 T = 0.01; % temperature
 x = betarnd(x0/T+1,(1-x0)/T+1);
end


function p = proposal_q(x, x0)
% proposal distribution q(x | x0)

 T = 0.01;
 p = betapdf(x, x0/T+1, (1-x0)/T+1);
end


function p = target_p(x)
% target distribution p(x)

 % shape parameters:
 a = [2   40  100 6];
 b = [10  40  20  1];

 % mixing coefficients:
 w = [1 0.4 0.253 0.50]; w = w/sum(w);
 p = 0;
 for i = 1:length(a)
  p = p + w(i)*betapdf(x,a(i),b(i));
 end
end
```

```matlab
%% DEMO script

k = 10000;    % number of samples
X = NaN(1,k); % list of samples

x0 = proposal_gen(0.5);
for i = 1:k
 x1 = proposal_gen(x0);
 a = target_p(x1)/target_p(x0) * ...
     proposal_q(x0,x1)/proposal_q(x1,x0);
 if rand(1) < a
  X(i) = x1; x0 = x1;
 else
  X(i) = x0;
 end
end

figure(1)
x = 0:0.001:1;
plot(x, target_p(x), 'r', 'linewidth',2);
hold on
binw = 0.025; % histogram bin width
n = histc(X, 0:binw:1);
h = bar(0:binw:1, n/sum(n)/binw, 'histc');
set(h, 'facecolor', 'r', 'facealpha', 0.3)
xlim([0 1]); ylim([0 2.5])
xlabel 'x'
ylabel 'p(x)'
title 'MH demo'
hold off
```

data-driven = proposals are derived from data

**Then**

1. **primitives** = elementary measurements
   - points in line fitting
   - matches in epipolar geometry or homography estimation

2. **configuration** = $s$-tuple of primitives    minimal subsets necessary for parameter estimate
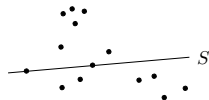


the minimization will be over a <u>discrete</u> set:
   - of point pairs in line fitting (left)
   - of match 7-tuples in epipolar geometry estimation

3. a map from configuration $C$ to parameters $\boldsymbol{\theta} = \boldsymbol{\theta}(C)$ by solving the **minimal problem**
   - line parameters $\mathbf{n}$ from two points
   - fundamental matrix $\mathbf{F}$ from seven matches
   - homography $\mathbf{H}$ from four matches, etc

4. **target likelihood** $p(E, D \mid \boldsymbol{\theta}(C))$ is represented by $\pi(C)$
   - can use log-likelihood: then it is the sum of robust errors $\hat{V}(e_{ij})$ given $\mathbf{F}$ (26)
     - robustified point distance from the line $\boldsymbol{\theta} = \mathbf{n}$
     - robustified Sampson error for $\boldsymbol{\theta} = \mathbf{F}$, etc
   - posterior likelihood $p(E, D \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$ can be used    MAPSAC ($\pi(S)$ includes the prior)

5. parameter distribution follows the **empirical distribution** of $s$-tuples. Since the proposal is done via the minimal problem solver, it is 'data-driven',
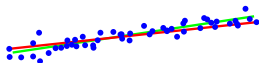


- pairs of points define line distribution $p(\mathbf{n} \mid X)$ (left)
- random correspondence 7-tuples define epipolar geometry distribution $q(\mathbf{F} \mid M)$

6. **proposal distribution** $q(\cdot)$ is just a constant(!) distribution of the $s$-tuples:
   a) $q$ uniform, independent $q(S \mid C_t) = q(S) = \binom{mn}{s}^{-1}$, then $a = \min\left\{1, \frac{p(S)}{p(C_t)}\right\}$
   b) $q$ dependent on descriptor similarity     PROSAC (similar pairs are proposed more often)
   c) $q$ dependent on the current configuration $C_t$                          e.g. 'not far from $C_t$'

7. (optional) hard **inlier/outlier discrimination** by the threshold (27)
$$\hat{V}(e_{ij}) < e_T, \qquad e_T = \sigma_1 \sqrt{-\log t^2}$$

8. **local optimization** from promising proposals
   - can use the hard inliers or just the robust error (26)       (more expensive but more stable)
   - cannot be used to replace $C_t$    (it would violate 'detailed balance' required for the MH scheme)

9. **stopping** based on the probability of proposing an all-inlier configuration       →123

- The number of proposals $N$ needed to hit the "true parameters" = an all-inlier config?

  this will tell us nothing about the accuracy of the result

$P$ ... probability that at least one proposal is all-inlier     $1 - P$ ... all previous $N$ proposals were bad
$\varepsilon$ ... the fraction of inliers among primitives, $\varepsilon \leq 1$
$s$ ... minimal configuration size     2 in line fitting, 7 in 7-point algorithm, 4 in homography fitting,...

$$N \geq \frac{\log(1 - P)}{\log(1 - \varepsilon^s)}$$

- $\varepsilon^s$ ... proposal does not contain an outlier
- $1 - \varepsilon^s$ ... proposal contains at least one outlier
- $(1 - \varepsilon^s)^N$ ... $N$ previous proposals contained an outlier = $1 - P$

$N$ for $s = 7$

| $\varepsilon$ | $P$ | |
|---|---|---|
| | 0.8 | 0.99 |
| 0.5 | 205 | 590 |
| 0.2 | $1.3 \cdot 10^5$ | $3.5 \cdot 10^5$ |
| 0.1 | $1.6 \cdot 10^7$ | $4.6 \cdot 10^7$ |
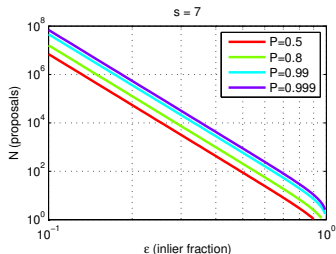


- $N$ can be re-estimated using the current estimate for $\varepsilon$ (if there is LO, then after LO)

  the quasi-posterior estimate for $\varepsilon$ is the average over all samples generated so far
- this shows we have a good reason to limit all possible matches to tentative matches only
- for $\varepsilon \to 0$ we gain nothing over the standard MH-sampler stopping rule

- when we are interested in the best config only. . . and we need fast data exploration. . .

**Simplified sampling procedure**

1. ~~given $C_t$,~~ draw a random sample $S$ from ~~$q(S \mid C_t)$~~ $q(S)$       independent sampling
          no use of information from $C_t$

2. ~~compute acceptance probability~~

$$a = \min \left\{ 1, \ \frac{\pi(S)}{\pi(C_t)} \cdot \frac{q(C_t \mid S)}{q(S \mid C_t)} \right\}$$

3. ~~draw a random number $u$ from unit-interval uniform distribution $U_{0,1}$~~

4. ~~if $u \leq a$ then $C_{t+1} := S$ else $C_{t+1} := C_t$~~

5. if $\pi(S) > \pi(C_{\text{best}})$ then remember $C_{\text{best}} := S$
        Steps 2–4 make no difference when waiting for the best sample configuration

- . . . but getting a good accuracy configuration might take very long this way
- good overall exploration but slow convergence in the vicinity of a mode where $C_t$ could serve as an attractor
- cannot use the past generated configurations to estimate any parameters
- we will fix these problems by (possibly robust) 'local optimization'

# ►RANSAC with Local Optimization and Early Stopping

1. initialize the best configuration as empty $C_{\text{best}} := \emptyset$ and time $t := 0$
2. estimate the number of needed proposals as $N := \binom{n}{s}$  $n$ – No. of primitives, $s$ – minimal config size
3. while $t \leq N$:

   a) propose a minimal random config $S$ of size $s$ from $q(S)$

   

   b) if $\pi(S) > \pi(C_{\text{best}})$ then

      i) update the best config $C_{\text{best}} := S$    $\pi(S)$ marginalized as in (26); $\pi(S)$ includes a prior $\Rightarrow$ MAP

      ii) threshold-out inliers using $e_T$ from (27)
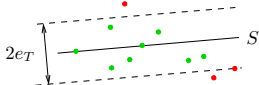
      

      iii) start local optimization from the inliers of $C_{\text{best}}$   LM optimization with robustified ($\rightarrow$114) Sampson error
      possibly weighted by posterior $\pi(m_{ij})$ [Chum et al. 2003]
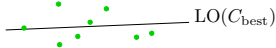
      

      iv) update $C_{\text{best}}$, update inliers using (27), re-estimate $N$ from inlier counts    $\rightarrow$123 for derivation
      $$N = \frac{\log(1 - P)}{\log(1 - \varepsilon^s)}, \quad \varepsilon = \frac{|\text{inliers}(C_{\text{best}})|}{m\,n},$$

   c) $t := t + 1$
4. output $C_{\text{best}}$

   • see ◉MPV course for RANSAC details    see also [Fischler & Bolles 1981], [25 years of RANSAC]

# Example Matching Results for the 7-point Algorithm with RANSAC



input images     interest points (ca. 3600)     tentative corresp. (416)     matching (340)
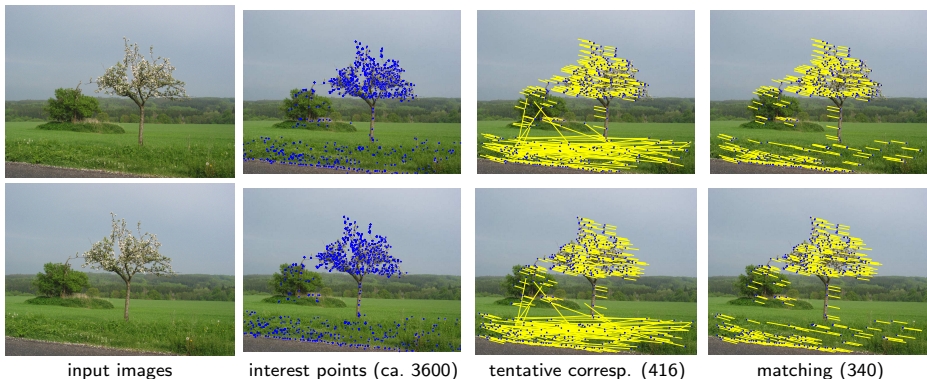
- notice some wrong matches (they have wrong depth, even negative)
- they cannot be rejected without additional constraints or scene knowledge
- without local optimization the minimization is over a <u>discrete set</u> of epipolar geometries proposable from 7-tuples

## Beyond RANSAC

By marginalization in (23) we have lost constraints on $M$ (e.g. uniqueness). One can choose a better model when not marginalizing:

$$\pi(M, \mathbf{F}, E, D) = \underbrace{p(E \mid M, \mathbf{F})}_{\text{reprojection error}} \cdot \underbrace{p(D \mid M)}_{\text{similarity}} \cdot \underbrace{p(\mathbf{F})}_{\text{prior}} \cdot \underbrace{P(M)}_{\text{constraints}}$$

this is a global model: decisions on $m_{ij}$ are no longer independent!

**In the MH scheme**

- one can work with full $p(M, \mathbf{F} \mid E, D)$, then configuration $C = M$     **F** computable from $M$
  - explicit labeling $m_{ij}$ can be done by, e.g. sampling from

    $$q(m_{ij} \mid \mathbf{F}) \sim \big((1 - P_0)\, p_1(e_{ij} \mid \mathbf{F}),\ P_0\, p_0(e_{ij} \mid \mathbf{F})\big)$$

    when $P(M)$ uniform then always accepted, $a = 1$                          ⊛ derive
  - we can compute the posterior probability of each match $p(m_{ij})$ by histogramming $m_{ij}$ from $\{C_i\}$
  - local optimization can then use explicit inliers and $p(m_{ij})$

- error can be estimated for elements of $\mathbf{F}$ from $\{C_i\}$                  does not work in RANSAC!

- large error indicates problem degeneracy               this is not directly available in RANSAC

- good conditioning is not a requirement               we work with the entire distribution $p(\mathbf{F})$

- one can find the most probable number of epipolar geometries       by reversible jump MCMC
  (homographies or other models)                     and Bayesian model selection
                    if there are multiple models explaning data, RANSAC will return one of them randomly

**Task:** Find two vanishing points from line segments detected in input image. Principal point is known, square pixel.



iter: 10 (acc TOT=0.0%, HMC=NaN%); Eavg = 14.597

video

- primitives = line segments
- latent variables
  1. each line has a vanishing point label
     $\lambda_i \in \{\emptyset, 1, 2\}$, $\emptyset$ represents an outlier
  2. 'mother line' parameters $\theta_L$ (they pass through their vanishing points)
- explicit variables
  1. two unknown vanishing points $v_1$, $v_2$
- marginal proposals ($v_i$ fixed, $v_j$ proposed)
- minimal configuration $s = 2$
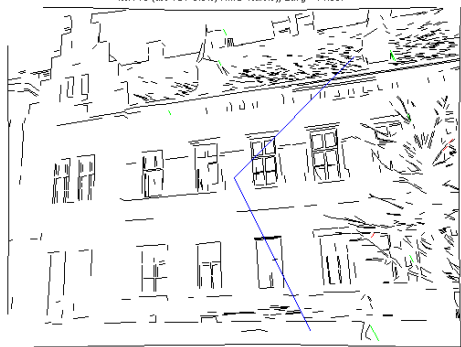
**simplifications**

- vanishing points restricted to the set of all pairwise segment intersections
- mother lines fixed by segment centroid, then $\theta_L$ uniquely given by $\lambda_i$, and the configuration is

$$C = \{v_1, v_2, \Lambda\}$$



$$\arg \min_{v_1, v_2, \Lambda, \theta_L} V(v_1, v_2, \Lambda, \theta_L)$$

Module VI
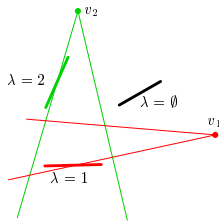
# 3D Structure and Camera Motion

6.1 Reconstructing Camera System

6.2 Bundle Adjustment

**covered by**

[1] [H&Z] Secs: 9.5.3, 10.1, 10.2, 10.3, 12.1, 12.2, 12.4, 12.5, 18.1

[2] Triggs, B. et al. Bundle Adjustment—A Modern Synthesis. In *Proc ICCV Workshop on Vision Algorithms.* Springer-Verlag. pp. 298–372, 1999.

**additional references**

📄 D. Martinec and T. Pajdla. Robust Rotation and Translation Estimation in Multiview Reconstruction. In *Proc CVPR,* 2007

📄 M. I. A. Lourakis and A. A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. ACM Trans Math Software 36(1):1–30, 2009.

**Given:** Calibration matrices $\mathbf{K}_j$ and tentative correspondences per camera <u>triples</u>.

**Initialization**

1. initialize <u>camera cluster</u> $\mathcal{C}$ with $P_1$, $P_2$,

2. find essential matrix $\mathbf{E}_{12}$ and matches $M_{12}$ by the 5-point algorithm →88

3. construct camera pair

$$\mathbf{P}_1 = \mathbf{K}_1 \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}, \ \mathbf{P}_2 = \mathbf{K}_2 \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$$

4. triangulate $\{X_i\}$ per match from $M_{12}$ →105

5. initialize <u>point cloud</u> $\mathcal{X}$ with $\{X_i\}$ satisfying chirality constraint $z_i > 0$ and apical angle constraint $|\alpha_i| > \alpha_T$
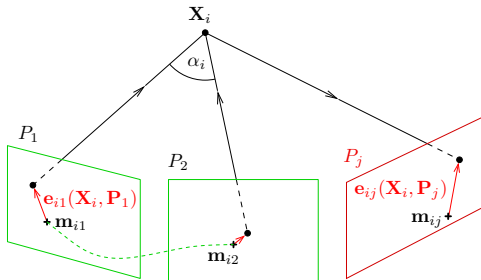


**Attaching camera $P_j \notin \mathcal{C}$**

1. select points $\mathcal{X}_j$ from $\mathcal{X}$ that have matches to $P_j$

2. estimate $\mathbf{P}_j$ using $\mathcal{X}_j$, RANSAC with the 3-pt alg. (P3P), projection errors $\mathbf{e}_{ij}$ in $\mathcal{X}_j$ →66

3. reconstruct 3D points from all tentative matches from $P_j$ to all $P_l$, $l \neq k$ that are <u>not</u> in $\mathcal{X}$

4. filter them by the chirality and apical angle constraints and add them to $\mathcal{X}$

5. add $P_j$ to $\mathcal{C}$

6. perform <u>bundle adjustment</u> on $\mathcal{X}$ and $\mathcal{C}$ coming next →137

**Observation:** Unless $\mathbf{P}_i$ are constrained, then for any number of cameras $i = 1, \ldots, k$

$$\underline{\mathbf{m}}_i \simeq \mathbf{P}_i \underline{\mathbf{X}} = \underbrace{\mathbf{P}_i \mathbf{H}^{-1}}_{\mathbf{P}_i'} \underbrace{\mathbf{H} \underline{\mathbf{X}}}_{\underline{\mathbf{X}}'} = \mathbf{P}_i' \, \underline{\mathbf{X}}'$$

- when $\mathbf{P}_i$ and $\underline{\mathbf{X}}$ are both determined from correspondences (including calibrations $\mathbf{K}_i$), they are given up to a common 3D <u>homography</u> $\mathbf{H}$

  (translation, rotation, scale, shear, pure perspectivity)



$m_1 \qquad\qquad m_2 \qquad\qquad\qquad X \qquad X'$
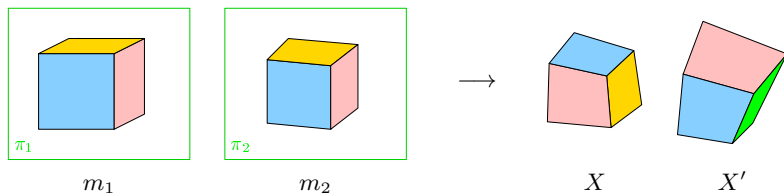
- when cameras are internally calibrated ($\mathbf{K}_i$ known) then $\mathbf{H}$ is restricted to a <u>similarity</u> since it must preserve the calibrations $\mathbf{K}_i$    [H&Z, Secs. 10.2, 10.3], [Longuet-Higgins 1981]
  (translation, rotation, scale)

### ►Analyzing the Camera System Reconstruction Problem

**Problem:** Given a set of $p$ decomposed pairwise essential matrices $\hat{\mathbf{E}}_{ij} = [\hat{\mathbf{t}}_{ij}]_\times \hat{\mathbf{R}}_{ij}$ and calibration matrices $\mathbf{K}_i$ reconstruct the camera system $\mathbf{P}_i$, $i = 1, \ldots, k$
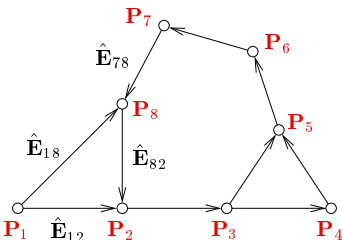
We construct calibrated camera pairs $\hat{\mathbf{P}}_{ij} \in \mathbb{R}^{6,4}$ see (17)

$$\hat{\mathbf{P}}_{ij} = \begin{bmatrix} \mathbf{K}_i^{-1}\hat{\mathbf{P}}_i \\ \mathbf{K}_j^{-1}\hat{\mathbf{P}}_j \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hat{\mathbf{R}}_{ij} & \hat{\mathbf{t}}_{ij} \end{bmatrix} \in \mathbb{R}^{6,4}$$

- singletons $i$, $j$ correspond to graph nodes     $k$ nodes
- pairs $ij$ correspond to graph edges     $p$ edges

$\hat{\mathbf{P}}_{ij}$ are <u>in different coordinate systems</u> but these are related by similarities $\hat{\mathbf{P}}_{ij}\mathbf{H}_{ij} = \mathbf{P}_{ij}$

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hat{\mathbf{R}}_{ij} & \hat{\mathbf{t}}_{ij} \end{bmatrix}}_{\mathbb{R}^{6,4}} \underbrace{\begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\top & s_{ij} \end{bmatrix}}_{\mathbf{H}_{ij}\,\in\mathbb{R}^{4,4}} \overset{!}{=} \underbrace{\begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{R}_j & \mathbf{t}_j \end{bmatrix}}_{\mathbb{R}^{6,4}} \tag{28}$$

- (28) is a linear system of $24p$ eqs. in $7p + 6k$ unknowns     $7p \sim (\mathbf{t}_{ij}, \mathbf{R}_{ij}, s_{ij})$, $6k \sim (\mathbf{R}_i, \mathbf{t}_i)$
- each $\mathbf{P}_i$ appears on the right side as many times as is the degree of node $\mathbf{P}_i$     eg. $P_5$ 3-times

Eq. (28) implies
$$\begin{bmatrix} \mathbf{R}_{ij} \\ \hat{\mathbf{R}}_{ij}\mathbf{R}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i \\ \mathbf{R}_j \end{bmatrix} \qquad \begin{bmatrix} \mathbf{t}_{ij} \\ \hat{\mathbf{R}}_{ij}\mathbf{t}_{ij} + s_{ij}\hat{\mathbf{t}}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_j \end{bmatrix}$$

- $\mathbf{R}_{ij}$ and $\mathbf{t}_{ij}$ can be eliminated:

$$\hat{\mathbf{R}}_{ij}\mathbf{R}_i = \mathbf{R}_j, \qquad \hat{\mathbf{R}}_{ij}\mathbf{t}_i + s_{ij}\hat{\mathbf{t}}_{ij} = \mathbf{t}_j, \qquad s_{ij} > 0 \qquad (29)$$

- note transformations that do not change these equations        assuming no error in $\hat{\mathbf{R}}_{ij}$

      1.   $\mathbf{R}_i \mapsto \mathbf{R}_i\mathbf{R}$,       2.   $\mathbf{t}_i \mapsto \sigma\,\mathbf{t}_i$ and $s_{ij} \mapsto \sigma s_{ij}$,       3.   $\mathbf{t}_i \mapsto \mathbf{t}_i + \mathbf{R}_i\mathbf{t}$

- the global frame is fixed, e.g. by selecting

$$\mathbf{R}_1 = \mathbf{I}, \qquad \sum_{i=1}^{k} \mathbf{t}_i = \mathbf{0}, \qquad \frac{1}{p}\sum_{i,j} s_{ij} = 1 \qquad (30)$$

- rotation equations are decoupled from translation equations
- in principle, $s_{ij}$ could correct the sign of $\hat{\mathbf{t}}_{ij}$ from essential matrix decomposition     

  but $\mathbf{R}_i$ cannot correct the $\alpha$ sign in $\hat{\mathbf{R}}_{ij}$

       $\Rightarrow$ therefore make sure all points are in front of cameras and constrain $s_{ij} > 0$;

+ pairwise correspondences are sufficient
− suitable for well-distributed cameras only (dome-like configurations)

                                              otherwise intractable or numerically unstable
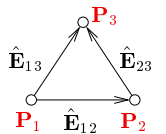
**Task:** Solve $\hat{\mathbf{R}}_{ij}\mathbf{R}_i = \mathbf{R}_j$, $i, j \in V$, $(i,j) \in E$ where $\mathbf{R}$ are a $3 \times 3$ rotation matrix each.
Per columns $c = 1, 2, 3$ of $\mathbf{R}_j$:

$$\hat{\mathbf{R}}_{ij}\mathbf{r}_i^c - \mathbf{r}_j^c = \mathbf{0}, \qquad \text{for all } i, j \tag{31}$$

- fix $c$ and denote $\mathbf{r}^c = \left[\mathbf{r}_1^c, \mathbf{r}_2^c, \ldots, \mathbf{r}_k^c\right]^\top$  $c$-th columns of all rotation matrices stacked; $\mathbf{r}^c \in \mathbb{R}^{3k}$
- then (31) becomes  $\mathbf{D}\,\mathbf{r}^c = \mathbf{0}$  $\mathbf{D} \in \mathbb{R}^{3p,3k}$
- $3p$ equations for $3k$ unknowns $\rightarrow p \geq k$  in a 1-connected graph we have to fix $\mathbf{r}_1^c = [1, 0, 0]$

**Ex:** ($k = p = 3$)



$$\begin{aligned}\hat{\mathbf{R}}_{12}\mathbf{r}_1^c - \mathbf{r}_2^c &= \mathbf{0}\\ \hat{\mathbf{R}}_{23}\mathbf{r}_2^c - \mathbf{r}_3^c &= \mathbf{0}\\ \hat{\mathbf{R}}_{13}\mathbf{r}_1^c - \mathbf{r}_3^c &= \mathbf{0}\end{aligned}$$

$\rightarrow$

$$\mathbf{D}\,\mathbf{r}^c = \begin{bmatrix}\hat{\mathbf{R}}_{12} & -\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \hat{\mathbf{R}}_{23} & -\mathbf{I}\\ \hat{\mathbf{R}}_{13} & \mathbf{0} & -\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{r}_1^c\\ \mathbf{r}_2^c\\ \mathbf{r}_3^c\end{bmatrix} = \mathbf{0}$$

- must hold for any $c$

**Idea:**  [Martinec & Pajdla CVPR 2007]
1. find the space of all $\mathbf{r}^c \in \mathbb{R}^{3k}$ that solve (31)  $\mathbf{D}$ is sparse, use `[V,E] = eigs(D'*D,3,0);` (Matlab)
2. choose 3 unit orthogonal vectors in this space  3 smallest eigenvectors
3. find closest rotation matrices per cam. using SVD  because $\|\mathbf{r}^c\| = 1$ is necessary but insufficient
   $\mathbf{R}_i^* = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{R}_i = \mathbf{U}\mathbf{D}\mathbf{V}^\top$
- global world rotation is arbitrary

From (29) and (30):     $0 < d \leq 3$ – rank of camera center set, $p$ – #pairs, $k$ – #cameras

$$\hat{\mathbf{R}}_{ij}\mathbf{t}_i + s_{ij}\hat{\mathbf{t}}_{ij} - \mathbf{t}_j = \mathbf{0}, \qquad \sum_{i=1}^{k} \mathbf{t}_i = \mathbf{0}, \qquad \sum_{i,j} s_{ij} = p, \qquad s_{ij} > 0, \qquad \mathbf{t}_i \in \mathbb{R}^d$$

- in rank $d$:  $d \cdot p + d + 1$ indep. eqns for $d \cdot k + p$ unknowns $\rightarrow p \geq \frac{d(k-1)-1}{d-1} \stackrel{\text{def}}{=} Q(d,k)$

**Ex: Chains and circuits**     construction from sticks of known orientation and unknown length?



$p = k - 1$         $k = p = 3$         $k = p = 4$         $k = p > 4$

$k \leq 2$ for any $d$     $3 \geq d \geq 2$: non-collinear ok     $3 \geq d \geq 3$: non-planar ok     $3 \geq d \geq k - 1$: impossible
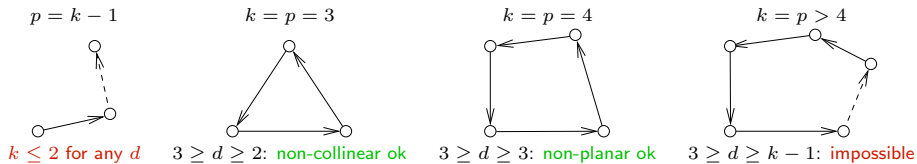
- equations insufficient for chains, trees, or when $d = 1$
- 3-connectivity implies sufficient equations for $d = 3$     cams. in general pos. in 3D     collinear cameras
  - $s$-connected graph has $p \geq \lceil \frac{sk}{2} \rceil$ edges for $s \geq 2$, hence $p \geq \lceil \frac{3k}{2} \rceil \geq Q(3,k) = \frac{3k}{2} - 2$
- 4-connectivity implies sufficient eqns. for any $k$ when $d = 2$     coplanar cams
  - since $p \geq \lceil 2k \rceil \geq Q(2,k) = 2k - 3$
  - maximal planar tringulated graphs have $p = 3k - 6$
  and give a solution for $k \geq 3$     maximal planar triangulated graph example:

## cont'd
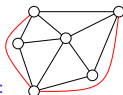
Linear equations in (29) and (30) can be rewritten to

$$\mathbf{D}\mathbf{t} = \mathbf{0}, \qquad \mathbf{t} = \left[\mathbf{t}_1^\top, \mathbf{t}_2^\top, \ldots, \mathbf{t}_k^\top, s_{12}, \ldots, s_{ij}, \ldots\right]^\top$$

assuming measurement errors $\mathbf{D}\mathbf{t} = \boldsymbol{\epsilon}$ and $d = 3$, we have

$$\mathbf{t} \in \mathbb{R}^{3k+p}, \quad \mathbf{D} \in \mathbb{R}^{3p,3k+p} \quad \text{sparse}$$

and

$$\mathbf{t}^* = \underset{\mathbf{t},\, s_{ij} > 0}{\arg\min} \; \mathbf{t}^\top \mathbf{D}^\top \mathbf{D} \, \mathbf{t}$$

- this is a quadratic programming problem (mind the constraints!)

```
z = zeros(3*k+p,1);
t = quadprog(D.'*D, z, diag([zeros(3*k,1); -ones(p,1)]), z);
```

- but check the rank first!

## ►Bundle Adjustment

**Goal:** Use a good (and expensive) error model and improve all estimated parameters

**Given:**
1. set of 3D points $\{\mathbf{X}_i\}_{i=1}^p$
2. set of cameras $\{\mathbf{P}_j\}_{j=1}^c$
3. fixed tentative projections $\mathbf{m}_{ij}$

**Required:**
1. corrected 3D points $\{\mathbf{X}_i'\}_{i=1}^p$
2. corrected cameras $\{\mathbf{P}_j'\}_{j=1}^c$

**Latent:**
1. visibility decision $v_{ij} \in \{0,1\}$ per $\mathbf{m}_{ij}$
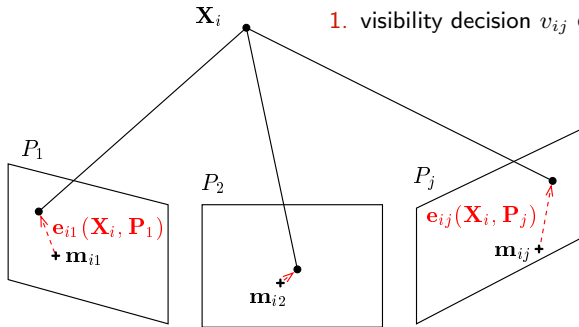


- for simplicity, $\mathbf{X}$, $\mathbf{m}$ are considered Cartesian (not homogeneous)
- we have <u>projection error</u> $\mathbf{e}_{ij}(\mathbf{X}_i, \mathbf{P}_j) = \mathbf{x}_i - \mathbf{m}_i$ per image feature, where $\underline{\mathbf{x}}_i = \mathbf{P}_j \underline{\mathbf{X}}_i$
- for simplicity, we will work with scalar error $e_{ij} = \|\mathbf{e}_{ij}\|$

# Robust Objective Function for Bundle Adjustment

The data model is

$$p(\{\mathbf{e}\} \mid \{\mathbf{P}, \mathbf{X}\}) = \prod_{\text{pts}:i=1}^{p} \prod_{\text{cams}:j=1}^{c} \Big( (1 - P_0) p_1(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j) + P_0 \, p_0(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j) \Big)$$

marginalized negative log-density is (→114)

$$-\log p(\{\mathbf{e}\} \mid \{\mathbf{P}, \mathbf{X}\}) = \sum_i \sum_j \underbrace{-\log\Big( e^{-\frac{e_{ij}^2(\mathbf{X}_i, \mathbf{P}_j)}{2\sigma_1^2}} + t\Big)}_{\rho(e_{ij}^2(\mathbf{X}_i, \mathbf{P}_j)) = \nu_{ij}^2(\mathbf{X}_i, \mathbf{P}_j)} \overset{\text{def}}{=} \sum_i \sum_j \nu_{ij}^2(\mathbf{X}_i, \mathbf{P}_j)$$

- we can use LM, $e_{ij}$ is the projection error (not Sampson error)
- $\nu_{ij}$ is a 'robust' error fcn.; it is non-robust ($\nu_{ij} = e_{ij}$) when $t = 0$
- $\rho(\cdot)$ is a 'robustification function' we often find in M-estimation
- the $\mathbf{L}_{ij}$ in Levenberg-Marquardt changes to vector

$$(\mathbf{L}_{ij})_l = \frac{\partial \nu_{ij}}{\partial \theta_l} = \underbrace{\frac{1}{1 + t\, e^{e_{ij}^2(\theta)/(2\sigma_1^2)}}}_{\text{small for } e_{ij} \gg \sigma_1} \cdot \frac{1}{\nu_{ij}(\theta)} \cdot \frac{1}{4\sigma_1^2} \cdot \frac{\partial e_{ij}^2(\theta)}{\partial \theta_l} \quad (32)$$



but the LM method stays the same as before →107–108

- outliers (wrong $v_{ij}$): almost no impact on $\mathbf{d}_s$ in normal equations because the red term in (32) scales contributions to both sums down for the particular $ij$

$$-\sum_{i,j} \mathbf{L}_{ij}^\top \nu_{ij}(\theta^s) = \Big( \sum_{i,j}^{k} \mathbf{L}_{ij}^\top \mathbf{L}_{ij} \Big) \mathbf{d}_s$$
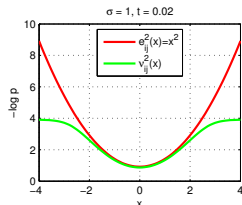
# ►Sparsity in Bundle Adjustment

We have $q = 3p + 11k$ parameters: $\theta = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p; \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_k)$   points, cameras

We will use a multi-index $r = 1, \ldots, z$, $z = p \cdot k$. Then   each $r$ corresponds to some $i, j$

$$\theta^* = \arg\min_\theta \sum_{r=1}^{z} \nu_r^2(\theta), \; \boldsymbol{\theta}^{s+1} := \boldsymbol{\theta}^s + \mathbf{d}_s, \; -\sum_{r=1}^{z} \mathbf{L}_r^\top \nu_r(\theta^s) = \left( \sum_{r=1}^{z} \mathbf{L}_r^\top \mathbf{L}_r + \lambda \operatorname{diag}(\mathbf{L}_r^\top \mathbf{L}_r) \right) \mathbf{d}_s$$

The block form of $\mathbf{L}_r$ in Levenberg-Marquardt ($\rightarrow$107) is zero except in columns $i$ and $j$:

$r$-th error term is $\nu_r^2 = \rho(e_{ij}^2(\mathbf{X}_i, \mathbf{P}_j))$



$\mathbf{L}_r =$

$r = (i, j)$ blocks:
■: $\mathbf{X}_i, 1 \times 3$
■: $\mathbf{P}_j, 1 \times 11$
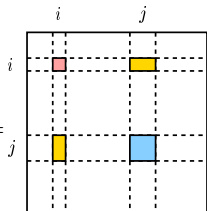
$\mathbf{L}_r^\top \mathbf{L}_r =$

blocks:
■: $\mathbf{X}_i - \mathbf{X}_i, 3 \times 3$
■: $\mathbf{X}_i - \mathbf{P}_j, 3 \times 11$
■: $\mathbf{P}_j - \mathbf{P}_j, 11 \times 11$
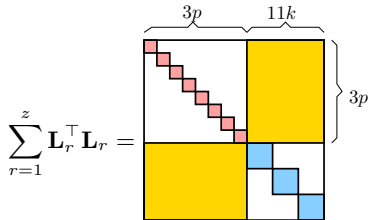
$$\sum_{r=1}^{z} \mathbf{L}_r^\top \mathbf{L}_r =$$

- "points first, then cameras" parameterization scheme

## ►Choleski Decomposition for B. A.

The most expensive computation in B. A. is solving the normal eqs:

$$\text{find } \mathbf{x} \text{ such that } \quad -\sum_{r=1}^{z} \mathbf{L}_r^\top \nu_r(\theta^s) = \Big(\sum_{r=1}^{z} \mathbf{L}_r^\top \mathbf{L}_r + \lambda \operatorname{diag}\big(\mathbf{L}_r^\top \mathbf{L}_r\big)\Big)\mathbf{x}$$

- **A** is very large                  approx. $3 \cdot 10^4 \times 3 \cdot 10^4$ for a small problem of 10000 points and 5 cameras
- **A** is sparse and symmetric, $\mathbf{A}^{-1}$ is dense                  direct matrix inversion is prohibitive

> Choleski: symmetric positive definite matrix **A** can be decomposed to $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$, where **L** is lower triangular. If **A** is sparse then **L** is sparse, too.

1. decompose $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$                  transforms the problem to $\mathbf{L}\underbrace{\mathbf{L}^\top \mathbf{x}}_{\mathbf{c}} = \mathbf{b}$

2. solve for **x** in two passes:

$$\mathbf{L}\,\mathbf{c} = \mathbf{b} \quad \mathbf{c}_i := \mathbf{L}_{ii}^{-1}\Big(\mathbf{b}_i - \sum_{j<i} \mathbf{L}_{ij}\mathbf{c}_j\Big) \quad \text{forward substitution, } i = 1, \ldots, q \text{ (params)}$$

$$\mathbf{L}^\top \mathbf{x} = \mathbf{c} \quad \mathbf{x}_i := \mathbf{L}_{ii}^{-1}\Big(\mathbf{c}_i - \sum_{j>i} \mathbf{L}_{ji}\mathbf{x}_j\Big) \quad \text{back-substitution}$$

- Choleski decomposition is fast (does not touch zero blocks)
   non-zero elements are $9p + 121k + 66pk \approx 3.4 \cdot 10^6$; ca. $250\times$ fewer than all elements
- it can be computed on single elements or on entire blocks
- use profile Choleski for sparse **A** and diagonal pivoting for semi-definite **A**                  see above; [Triggs et al. 1999]
- $\lambda$ controls the definiteness

## Profile Choleski Decomposition is Simple

```
function L = pchol(A)
%
% PCHOL profile Choleski factorization,
%    L = PCHOL(A) returns lower-triangular sparse L such that A = L*L'
%    for sparse square symmetric positive definite matrix A,
%    especially efficient for arrowhead sparse matrices.

% (c) 2010 Radim Sara (sara@cmp.felk.cvut.cz)

 [p,q] = size(A);
 if p ~= q, error 'Matrix A is not square'; end

 L = sparse(q,q);
 F = ones(q,1);
 for i=1:q
  F(i) = find(A(i,:),1); % 1st non-zero on row i; we are building F gradually
  for j = F(i):i-1
   k = max(F(i),F(j));
   a = A(i,j) - L(i,k:(j-1))*L(j,k:(j-1))';
   L(i,j) = a/L(j,j);
  end
  a = A(i,i) - sum(full(L(i,F(i):(i-1))).^2);
  if a < 0, error 'Matrix A is not positive definite'; end
  L(i,i) = sqrt(a);
 end
end
```

# ►Gauge Freedom

1. The external frame is not fixed:

$$\underline{\mathbf{m}}_{ij} \simeq \mathbf{P}_j \underline{\mathbf{X}}_i = \mathbf{P}_j \mathbf{H}^{-1} \mathbf{H} \underline{\mathbf{X}}_i = \mathbf{P}'_j \underline{\mathbf{X}}'_i$$

2. Some representations are not minimal, e.g.

   - $\mathbf{P}$ is 12 numbers for 11 parameters
   - we may represent $\mathbf{P}$ in decomposed form $\mathbf{K}$, $\mathbf{R}$, $\mathbf{t}$
   - but $\mathbf{R}$ is 9 numbers representing the 3 parameters of rotation

## As a result

- there is no unique solution
- matrix $\sum_r \mathbf{L}_r^\top \mathbf{L}_r$ is singular

## Solutions

1. <u>fixing the external frame</u> (e.g. a selected camera frame) explicitly or by constraints

2. <u>fixing the scale</u> (e.g. $s_{12} = 1$)

3a. either imposing <u>constraints</u> on projective entities
   - cameras, e.g. $\mathbf{P}_{3,4} = 1$        this excludes affine cameras
   - points, e.g. $\|\underline{\mathbf{X}}_i\|^2 = 1$        this way we can represent points at infinity

3b. or using <u>minimal representations</u>
   - points in their Euclidean representation $\mathbf{X}_i$      but finite points may be an unrealistic model
   - rotation matrix can be represented by skew-symmetric matrix    →147

# Implementing Simple Linear Constraints

**What for?**
1. fixing external frame as in $\theta_i = \mathbf{t}_i$, $s_{kl} = 1$ for some $i, k, l$      'trivial gauge'
2. representing additional knowledge as in $\theta_i = \theta_j$    e.g. cameras share calibration matrix $\mathbf{K}$

Introduce <u>reduced parameters</u> $\hat{\theta}$ and
<u>replication matrix</u> $\mathbf{T}$:

$$\theta = \mathbf{T}\hat{\theta} + \mathbf{t}, \quad \mathbf{T} \in \mathbb{R}^{p,\hat{p}}, \quad \hat{p} \leq p$$

then $\mathbf{L}_r$ in LM changes to $\mathbf{L}_r\mathbf{T}$ and
everything else stays the same →107



| | these $\mathbf{T}$, $\mathbf{t}$ represent |
|---|---|
| $\theta_1 = \hat{\theta}_1$ | no change |
| $\theta_2 = \hat{\theta}_2$ | no change |
| $\theta_3 = t_3$ | constancy |
| $\theta_4 = \theta_5 = \hat{\theta}_4$ | equality |

- $\mathbf{T}$ deletes columns of $\mathbf{L}_r$ that correspond to fixed parameters   it reduces the problem size
- consistent initialisation: $\theta^0 = \mathbf{T}\hat{\theta}^0 + \mathbf{t}$      or filter the init by pseudoinverse $\theta^0 \mapsto \mathbf{T}^\dagger\theta^0$
- no need for computing derivatives for $\theta_j$ corresponding to all-zero rows of $\mathbf{T}$     fixed $\theta$

- constraining projective entities →147–149
- more complex constraints tend to make normal equations dense
- implementing constraints is safer than explicit renaming of the parameters, gives a flexibility to experiment
- other methods are much more involved, see [Triggs et al. 1999]
- **BA resource:** http://www.ics.forth.gr/~lourakis/sba/   [Lourakis 2009]

## Matrix Exponential: A path to Minimal Parameterizations

- for any square matrix we define

$$\mathrm{expm}(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{1}{k!} \, \mathbf{A}^k \qquad \text{note: } \mathbf{A}^0 = \mathbf{I}$$

- some properties:

$$\mathrm{expm}\,\mathbf{0} = \mathbf{I}, \quad \boxed{\mathrm{expm}(-\mathbf{A}) = \left(\mathrm{expm}\,\mathbf{A}\right)^{-1}},$$

$$\mathrm{expm}(a\mathbf{A} + b\mathbf{A}) = \mathrm{expm}(a\,\mathbf{A})\,\mathrm{expm}(b\,\mathbf{A}), \quad \mathrm{expm}(\mathbf{A} + \mathbf{B}) \neq \mathrm{expm}(\mathbf{A})\,\mathrm{expm}(\mathbf{B})$$

$$\boxed{\mathrm{expm}(\mathbf{A}^\top) = (\mathrm{expm}\,\mathbf{A})^\top} \;\; \text{hence if } \mathbf{A} \text{ is skew symmetric then } \mathrm{expm}\,\mathbf{A} \text{ is orthogonal:}$$

$$\left(\mathrm{expm}(\mathbf{A})\right)^\top = \mathrm{expm}(\mathbf{A}^\top) = \mathrm{expm}(-\mathbf{A}) = \left(\mathrm{expm}(\mathbf{A})\right)^{-1}$$

$$\boxed{\det\!\left(\mathrm{expm}\,\mathbf{A}\right) = e^{\mathrm{tr}\,\mathbf{A}}}$$

### Some consequences

- traceless matrices map to unit-determinant matrices ⇒ we can represent homogeneous representatives
- skew-symmetric matrices map to orthogonal matrices ⇒ we can represent rotations
- matrix exponential provides the exponential map from the powerful Lie group theory

## Lie Groups Useful in 3D Vision

| group | | matrix | represent |
|---|---|---|---|
| special linear | $\mathrm{SL}(3,\mathbb{R})$ | real $3 \times 3$, unit determinant $\mathbf{H}$ | 2D homography |
| special linear | $\mathrm{SL}(4,\mathbb{R})$ | real $4 \times 4$, unit determinant | 3D homography |
| orthogonal | $\mathrm{SO}(3)$ | real $3 \times 3$ orthogonal $\mathbf{R}$ | 3D rotation |
| special Euclidean | $\mathrm{SE}(3)$ | $4 \times 4 \quad \left[\begin{smallmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{smallmatrix}\right]$, $\mathbf{R} \in \mathrm{SO}(3)$, $\mathbf{t} \in \mathbb{R}^3$ | 3D rigid motion |
| similarity | $\mathrm{Sim}(3)$ | $4 \times 4 \quad \left[\begin{smallmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & s^{-1} \end{smallmatrix}\right]$, $s \in \mathbb{R} \setminus 0$ | rigid motion + scale |

- Lie group $G =$ topological group that is also a smooth manifold with nice properties
- Lie algebra $\mathfrak{g} =$ vector space associated with a Lie group (tangent space of the manifold)
- group: this is where we need to work
- algebra: this is how to represent group elements with a minimal number of parameters
- Exponential map = map between algebra and its group $\exp \colon \mathfrak{g} \to G$
- for matrices $\exp = \mathrm{expm}$
- in most of the above groups we have a closed-form formula for the exponential and for its principal inverse
- also Jacobians are readily available

## Homography

$$\mathbf{H} = \operatorname{expm} \mathbf{Z}$$

- $\mathrm{SL}(3, \mathbb{R})$ group element

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad \text{s.t.} \quad \det \mathbf{H} = 1$$

- $\mathfrak{sl}(3, \mathbb{R})$ algebra element                     8 parameters

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & -(z_{11} + z_{22}) \end{bmatrix}$$

- note that $\operatorname{tr} \mathbf{Z} = 0$

## ▶ Rotation in 3D

$$\mathbf{R} = \operatorname{expm}[\boldsymbol{\phi}]_\times, \quad \boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3) = \varphi\,\mathbf{e}_\varphi, \quad 0 \le \varphi < \pi, \quad \|\mathbf{e}_\varphi\| = 1$$

- SO(3) group element

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \text{s.t.} \quad \mathbf{R}^{-1} = \mathbf{R}^\top$$

- $\mathfrak{so}(3)$ algebra element                                                      3 parameters

$$[\boldsymbol{\phi}]_\times = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}$$

- exponential map in closed form                                          Rodrigues' formula

$$\mathbf{R} = \operatorname{expm}[\boldsymbol{\phi}]_\times = \sum_{n=0}^{\infty} \frac{[\boldsymbol{\phi}]_\times^n}{n!} \overset{\circledast\,1}{=} \mathbf{I} + \frac{\sin\varphi}{\varphi}[\boldsymbol{\phi}]_\times + \frac{1-\cos\varphi}{\varphi^2}[\boldsymbol{\phi}]_\times^2$$

- (principal) logarithm                                          log is a periodic function

$$0 \le \varphi < \pi, \quad \cos\varphi = \frac{1}{2}\left(\operatorname{tr}(\mathbf{R}) - 1\right), \quad [\boldsymbol{\phi}]_\times = \frac{\varphi}{2\sin\varphi}(\mathbf{R} - \mathbf{R}^\top),$$

- $\boldsymbol{\phi}$ is rotation axis vector $\mathbf{e}_\varphi$ scaled by rotation angle $\varphi$ in radians
- finite limits for $\varphi \to 0$ exist: $\sin(\varphi)/\varphi \to 1$, $(1-\cos\varphi)/\varphi^2 \to 1/2$

## 3D Rigid Motion

$$\mathbf{M} = \operatorname{expm}\left[\boldsymbol{\nu}\right]_\wedge$$

- SE(3) group element $\quad$ 4 × 4 matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad \text{s.t.} \quad \mathbf{R} \in \operatorname{SO}(3),\ \mathbf{t} \in \mathbb{R}^3$$

- $\mathfrak{se}(3)$ algebra element $\quad$ 4 × 4 matrix

$$[\boldsymbol{\nu}]_\wedge = \begin{bmatrix} [\boldsymbol{\phi}]_\times & \boldsymbol{\rho} \\ \mathbf{0} & 0 \end{bmatrix} \quad \text{s.t.} \quad \boldsymbol{\phi} \in \mathbb{R}^3,\ \varphi = \|\boldsymbol{\phi}\| < \pi,\ \boldsymbol{\rho} \in \mathbb{R}^3$$

- exponential map in closed form

$$\mathbf{R} = \operatorname{expm}\left[\boldsymbol{\phi}\right]_\times, \quad \mathbf{t} = \operatorname{dexpm}([\boldsymbol{\phi}]_\times)\,\boldsymbol{\rho}$$

$$\operatorname{dexpm}([\boldsymbol{\phi}]_\times) = \sum_{n=0}^{\infty} \frac{[\boldsymbol{\phi}]_\times^n}{(n+1)!} = \mathbf{I} + \frac{1-\cos\varphi}{\varphi^2}[\boldsymbol{\phi}]_\times + \frac{\varphi - \sin\varphi}{\varphi^3}[\boldsymbol{\phi}]_\times^2$$

$$\operatorname{dexpm}^{-1}([\boldsymbol{\phi}]_\times) = \mathbf{I} - \frac{1}{2}[\boldsymbol{\phi}]_\times + \frac{1}{\varphi^2}\left(1 - \frac{\varphi}{2}\cot\frac{\varphi}{2}\right)[\boldsymbol{\phi}]_\times^2$$

- (principal) logarithm via a similar trick as in SO(3)
- finite limits exist: $(\varphi - \sin\varphi)/\varphi^3 \to 1/6$
- this form is preferred to $\operatorname{SO}(3) \times \mathbb{R}^3$

## ►Minimal Representations for Other Entities

- fundamental matrix via $\mathrm{SO}(3) \times \mathrm{SO}(3) \times \mathbb{R}$

$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^\top, \quad \mathbf{D} = \mathrm{diag}(1, d^2, 0), \quad \mathbf{U}, \mathbf{V} \in \mathrm{SO}(3), \quad 3 + 1 + 3 = 7 \text{ DOF}$$

- essential matrix via $\mathrm{SO}(3) \times \mathbb{R}^3$

$$\mathbf{E} = [-\mathbf{t}]_\times \mathbf{R}, \quad \mathbf{R} \in \mathrm{SO}(3), \quad \mathbf{t} \in \mathbb{R}^3, \ \|\mathbf{t}\| = 1, \qquad 3 + 2 = 5 \text{ DOF}$$

- camera via $\mathrm{SO}(3) \times \mathbb{R}^3$ or $\mathrm{SE}(3)$

$$\mathbf{P} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \end{bmatrix} \mathbf{M}, \qquad 5 + 3 + 3 = 11 \text{ DOF}$$

- $\mathrm{Sim}(3)$ useful for SfM without scale
- closed-form formulae still exist but are a bit messy
- a (bit too brief) intro to Lie groups in 3D vision/robotics and SW:

   📄 J. Solà, J. Deray, and D. Atchuthan. A micro Lie theory for state estimation in robotics. arXiv:1812.01537v7 [cs.RO], August 2020.

# Stereovision

**mostly covered by**

Šára, R. How To Teach Stereoscopic Vision. Proc. ELMAR 2010      referenced as [SP]

**additional references**

C. Geyer and K. Daniilidis. Conformal rectification of omnidirectional stereo pairs. In *Proc Computer Vision and Pattern Recognition Workshop*, p. 73, 2003.

J. Gluckman and S. K. Nayar. Rectifying transformations that minimize resampling effects. In *Proc IEEE CS Conf on Computer Vision and Pattern Recognition*, vol. 1:111–117. 2001.

M. Pollefeys, R. Koch, and L. V. Gool. A simple and efficient rectification method for general motion. In *Proc Int Conf on Computer Vision*, vol. 1:496–501, 1999.

- monocular vision already gives a rough 3D sketch because we understand the scene

# What Are The Relative Distances?
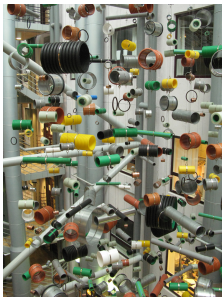


Centrum för teknikstudier at Malmö Högskola, Sweden



The Vyšehrad Fortress, Prague

- left: we have no help from image interpretation
- right: ambiguous interpretation due to a combination of missing texture and occlusion

- when we <u>do not recognize</u> the scene and cannot use high-level constraints the problem <u>seems difficult</u> (right, less so in the center)
- most stereo matching algorithms do not require scene understanding prior to matching
- the success of a model-free stereo matching algorithm is unlikely:
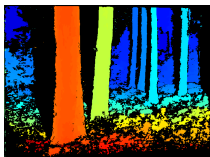


left image     a good disparity map     disparity map from WTA
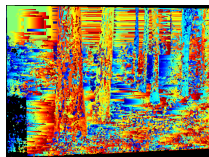
**WTA Matching:**

for every left-image pixel find the most similar right-image pixel along the corresponding epipolar line
[Marroquin 83]

# A Summary of Our Observations and an Outlook

1. simple matching algorithms do not work

2. stereopsis requires image interpretation in sufficiently complex scenes

<span style="color:blue">or another-modality measurement</span>

<div style="background-color:#d5e8d4">we have a tradeoff: model strength ↔ universality</div>

**Outlook:**

1. represent the occlusion constraint:    <span style="color:blue">correspondences are not independent due to occlusions</span>
   - epipolar rectification
   - disparity
   - uniqueness as an occlusion constraint

2. represent piecewise continuity    <span style="color:blue">the weakest of interpretations; piecewise: object boundaries</span>
   - ordering as a weak continuity model

3. use a consistent framework
   - finding the most probable solution (MAP)

# ▶Linear Epipolar Rectification for Easier Correspondence Search

**Obs:**
- if we map epipoles to infinity, epipolar lines become parallel
- we then rotate them to become horizontal
- we then scale the images to make corresponding epipolar lines colinear
- this can be achieved by a pair of (non-unique) homographies applied to the images

**Problem:** Given fundamental matrix $\mathbf{F}$ or camera matrices $\mathbf{P}_1$, $\mathbf{P}_2$, compute a pair of homographies that maps epipolar lines to horizontal with the same row coordinate.

**Procedure:**
1. find a pair of rectification homographies $\mathbf{H}_1$ and $\mathbf{H}_2$.
2. warp images using $\mathbf{H}_1$ and $\mathbf{H}_2$ and transform the fundamental matrix

$$\mathbf{F} \mapsto \mathbf{H}_2^{-\top}\mathbf{F}\mathbf{H}_1^{-1} \quad \text{or the cameras } \mathbf{P}_1 \mapsto \mathbf{H}_1\mathbf{P}_1, \ \ \mathbf{P}_2 \mapsto \mathbf{H}_2\mathbf{P}_2.$$



original pair
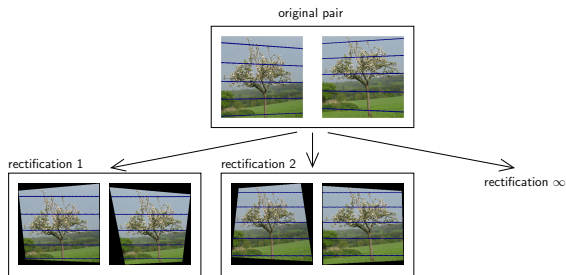
rectification 1    rectification 2

rectification ∞

**Assumption:** Cameras $(\mathbf{P}_1, \mathbf{P}_2)$ are rectified by a homography pair $(\mathbf{H}_1, \mathbf{H}_2)$:

$$\mathbf{P}_i^* = \mathbf{H}_i \mathbf{P}_i = \mathbf{H}_i \mathbf{K}_i \mathbf{R}_i \begin{bmatrix} \mathbf{I} & -\mathbf{C}_i \end{bmatrix}, \quad i = 1, 2$$

rectified entities: $\mathbf{F}^*$, $\mathbf{l}_2^*$, $\mathbf{l}_1^*$, etc:



$u$

$v$     $m_1^* = (u_1^*, v^*)$     $m_2^* = (u_2^*, v^*)$

$l_1^*$      $l_2^*$      $e_2^*$
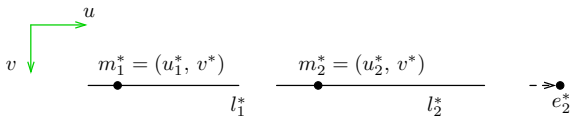
- the rectified location difference $d = u_1^* - u_2^*$ is called <u>disparity</u>

**corresponding epipolar lines must be:**
 1. parallel to image rows $\Rightarrow$ epipoles become $e_1^* = e_2^* = (1, 0, 0)$
 2. equivalent $l_2^* = l_1^*$ :    $\mathbf{l}_1^* \simeq \underline{\mathbf{e}}_1^* \times \underline{\mathbf{m}}_1 = [\underline{\mathbf{e}}_1^*]_\times \underline{\mathbf{m}}_1$,    $\mathbf{l}_2^* \simeq \mathbf{F}^* \underline{\mathbf{m}}_1$    $\Rightarrow$    $\mathbf{F}^* = [\underline{\mathbf{e}}_1^*]_\times$

- therefore the <u>canonical fundamental matrix</u> is

$$\mathbf{F}^* \simeq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

**A two-step rectification procedure**
 1. find <u>some</u> pair of <u>primitive rectification homographies</u> $\hat{\mathbf{H}}_1$, $\hat{\mathbf{H}}_2$
 2. upgrade to a pair of <u>optimal rectification homographies</u> while preserving $\mathbf{F}^*$

## ▶Geometric Interpretation of Linear Rectification

What pair of physical cameras is compatible with $\mathbf{F}^*$?

- we know that $\mathbf{F} = (\mathbf{Q}_1\mathbf{Q}_2^{-1})^\top[\mathbf{e}_1]_\times$
- we choose $\mathbf{Q}_1^* = \mathbf{K}_1^*$, $\mathbf{Q}_2^* = \mathbf{K}_2^*\mathbf{R}^*$; then

$$(\mathbf{Q}_1^*\mathbf{Q}_2^{*-1})^\top[\mathbf{e}_1^*]_\times = (\mathbf{K}_1^*\mathbf{R}^{*\top}\mathbf{K}_2^{*-1})^\top\mathbf{F}^*$$

- we look for $\mathbf{R}^*$, $\mathbf{K}_1^*$, $\mathbf{K}_2^*$ compatible with

$$(\mathbf{K}_1^*\mathbf{R}^{*\top}\mathbf{K}_2^{*-1})^\top\mathbf{F}^* = \lambda\mathbf{F}^*, \qquad \mathbf{R}^*\mathbf{R}^{*\top} = \mathbf{I}, \qquad \mathbf{K}_1^*,\mathbf{K}_2^* \text{ upper triangular}$$

- we also want $\mathbf{b}^*$ from $\underline{\mathbf{e}}_1^* \simeq \mathbf{P}_1^*\underline{\mathbf{C}}_2^* = \mathbf{K}_1^*\mathbf{b}^*$ $\qquad\qquad$ $\mathbf{b}^*$ in cam. 1 frame
- result:

$$\mathbf{R}^* = \mathbf{I}, \quad \mathbf{b}^* = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{K}_1^* = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}_2^* = \begin{bmatrix} k_{21} & k_{22} & k_{23} \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (33)$$

- rectified cameras are in canonical relative pose $\qquad\qquad$ not rotated, canonical baseline
- rectified calibration matrices can differ in the first row only
- when $\mathbf{K}_1^* = \mathbf{K}_2^*$ then the rectified pair is called the standard stereo pair and the homographies standard rectification homographies
- standard rectification homographies: points at infinity have zero disparity

$$\mathbf{P}_i^*\underline{\mathbf{X}}_\infty = \mathbf{K}\begin{bmatrix}\mathbf{I} & -\mathbf{C}_i\end{bmatrix}\underline{\mathbf{X}}_\infty = \mathbf{K}\mathbf{X}_\infty \qquad i = 1, 2$$

- this does not mean that the images are not distorted after rectification

**Goal:** Given fundamental matrix $\mathbf{F}$, derive some simple rectification homographies $\mathbf{H}_1$, $\mathbf{H}_2$

1. Let the SVD of $\mathbf{F}$ be $\mathbf{U}\mathbf{D}\mathbf{V}^\top = \mathbf{F}$, where $\mathbf{D} = \operatorname{diag}(1, d^2, 0)$, $\quad 1 \geq d^2 > 0$

2. Write $\mathbf{D}$ as $\mathbf{D} = \mathbf{A}^\top \mathbf{F}^* \mathbf{B}$ for some regular $\mathbf{A}$, $\mathbf{B}$. For instance  ($\mathbf{F}^*$ is given →156)

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -d & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & d & 0 \end{bmatrix}$$

3. Then

$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^\top = \underbrace{\mathbf{U}\mathbf{A}^\top}_{\hat{\mathbf{H}}_2^\top} \mathbf{F}^* \underbrace{\mathbf{B}\mathbf{V}^\top}_{\hat{\mathbf{H}}_1}$$

and the primitive rectification homographies are

$$\hat{\mathbf{H}}_2 = \mathbf{A}\mathbf{U}^\top, \qquad \hat{\mathbf{H}}_1 = \mathbf{B}\mathbf{V}^\top$$
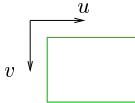
⊛ P1; 1pt: derive some other admissible $\mathbf{A}$, $\mathbf{B}$

- rectification homographies do exist →156

- there are other primitive rectification homographies, these suggested are just simple to obtain

► **The Set of All Rectification Homographies**

**Proposition 1** Homographies $\mathbf{A}_1$ and $\mathbf{A}_2$ are <u>rectification-preserving</u> if the images stay rectified, i.e. if $\mathbf{A}_2^{-\top} \mathbf{F}^* \mathbf{A}_1^{-1} \simeq \mathbf{F}^*$, which gives

$$\mathbf{A}_1 = \begin{bmatrix} l_1 & l_2 & l_3 \\ 0 & s_v & t_v \\ 0 & q & 1 \end{bmatrix}, \qquad \mathbf{A}_2 = \begin{bmatrix} r_1 & r_2 & r_3 \\ 0 & s_v & t_v \\ 0 & q & 1 \end{bmatrix},$$

where $s_v \neq 0$, $t_v$, $l_1 \neq 0$, $l_2$, $l_3$, $r_1 \neq 0$, $r_2$, $r_3$, $q$ are <u>9 free parameters</u>.

| general | transformation | | standard |
|---|---|---|---|
| $l_1, r_1$ | horizontal scales | | $l_1 = r_1$ |
| $l_2, r_2$ | horizontal shears | | $l_2 = r_2$ |
| $l_3, r_3$ | horizontal shifts | | $l_3 = r_3$ |
| $q$ | common special projective | | |
| $s_v$ | common vertical scale | | |
| $t_v$ | common vertical shift | | |
| 9 DoF | | | $9 - 3 = 6$ DoF |

- $q$ is due to a rotation about the baseline
- $s_v$ changes the focal length

proof: find a rotation $\mathbf{G}$ that brings $\mathbf{K}$ to upper triangular form via RQ decomposition: $\mathbf{A}_1 \mathbf{K}_1^* = \hat{\mathbf{K}}_1 \mathbf{G}$ and $\mathbf{A}_2 \mathbf{K}_2^* = \hat{\mathbf{K}}_2 \mathbf{G}$

**Corollary for Proposition 1** Let $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$ be (primitive or other) rectification homographies. Then $\mathbf{H}_1 = \mathbf{A}_1 \bar{\mathbf{H}}_1$, $\quad \mathbf{H}_2 = \mathbf{A}_2 \bar{\mathbf{H}}_2$ are also rectification homographies.

**Proposition 2** Pairs of rectification-preserving homographies $(\mathbf{A}_1, \mathbf{A}_2)$ form a group with group operation $(\mathbf{A}'_1, \mathbf{A}'_2) \circ (\mathbf{A}_1, \mathbf{A}_2) = (\mathbf{A}'_1 \mathbf{A}_1, \mathbf{A}'_2 \mathbf{A}_2)$.

**Proof:**
- closure by Proposition 1
- associativity by matrix multiplication
- identity belongs to the set
- inverse element belongs to the set by $\mathbf{A}_2^\top \mathbf{F}^* \mathbf{A}_1 \simeq \mathbf{F}^* \Leftrightarrow \mathbf{F}^* \simeq \mathbf{A}_2^{-\top} \mathbf{F}^* \mathbf{A}_1^{-1}$

**Obs:** calibrated cameras: $d = 1 \Rightarrow \hat{\mathbf{H}}_1, \hat{\mathbf{H}}_2$ ($\rightarrow$158) are orthonormal

1. determine primitive rectification homographies $(\hat{\mathbf{H}}_1, \hat{\mathbf{H}}_2)$ from the <u>essential</u> matrix
2. choose a suitable common calibration matrix $\mathbf{K}$, e.g.

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad f = \frac{1}{2}(f^1 + f^2), \quad u_0 = \frac{1}{2}(u_0^1 + u_0^2), \text{ etc.}$$

3. the final rectification homographies applied as $\mathbf{P}_i \mapsto \mathbf{H}_i \mathbf{P}_i$ are

$$\mathbf{H}_1 = \mathbf{K}\hat{\mathbf{H}}_1\mathbf{K}_1^{-1}, \quad \mathbf{H}_2 = \mathbf{K}\hat{\mathbf{H}}_2\mathbf{K}_2^{-1}$$

- we got a standard stereo pair ($\rightarrow$157) and non-negative disparity:
  let $\mathbf{K}_i^{-1}\mathbf{P}_i = \mathbf{R}_i \begin{bmatrix} \mathbf{I} & -\mathbf{C}_i \end{bmatrix}, \quad i = 1, 2$      note we started from $\mathbf{E}$, not $\mathbf{F}$

$$\mathbf{H}_1\mathbf{P}_1 = \mathbf{K}\hat{\mathbf{H}}_1\mathbf{K}_1^{-1}\mathbf{P}_1 = \mathbf{K}\underbrace{\mathbf{B}\mathbf{V}^\top\mathbf{R}_1}_{\mathbf{R}^*} \begin{bmatrix} \mathbf{I} & -\mathbf{C}_1 \end{bmatrix} = \mathbf{K}\mathbf{R}^* \begin{bmatrix} \mathbf{I} & -\mathbf{C}_1 \end{bmatrix}$$

$$\mathbf{H}_2\mathbf{P}_2 = \mathbf{K}\hat{\mathbf{H}}_2\mathbf{K}_2^{-1}\mathbf{P}_2 = \mathbf{K}\underbrace{\mathbf{A}\mathbf{U}^\top\mathbf{R}_2}_{\mathbf{R}^*} \begin{bmatrix} \mathbf{I} & -\mathbf{C}_2 \end{bmatrix} = \mathbf{K}\mathbf{R}^* \begin{bmatrix} \mathbf{I} & -\mathbf{C}_2 \end{bmatrix}$$

- one can prove that $\mathbf{B}\mathbf{V}^\top\mathbf{R}_1 = \mathbf{A}\mathbf{U}^\top\mathbf{R}_2$ with the help of essential matrix decomposition (13)
- points at infinity project by $\mathbf{K}\mathbf{R}^*$ in both cameras $\Rightarrow$ they have zero disparity      $\rightarrow$164

standard rectification homographies reproject onto a common image plane parallel to the baseline
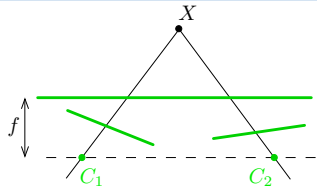
- rectification is done with a pair of homographies (one per image)     →155
  - ⇒ rectified camera centers are equal to the original ones
    - binocular rectification: a 9-parameter family of rectification homographies
    - trinocular rectification: has 9 or 6 free parameters (depending on additional constrains)
    - in general, linear rectification is not possible for more than three cameras
- rectified cameras are in canonical orientation     →157
  - ⇒ rectified image projection planes are coplanar
- equal rectified calibration matrices give standard rectification     →157
  - ⇒ rectified image projection planes are equal
- primitive rectification is standard in calibrated cameras     →161

- known $\mathbf{F}$ used alone does not allow standardization of rectification homographies
- for that we need either of these:
  1. projection matrices, or calibrated cameras, or
  2. a few points at infinity calibrating $k_{1i}$, $k_{2i}$, $i = 1, 2, 3$ in (33)

## Optimal choice for the free parameters

- by minimization of residual image distortion, eg.
  [Gluckman & Nayar 2001]

$$\mathbf{A}_1^* = \arg\min_{\mathbf{A}_1} \iint_\Omega \left(\det J(\mathbf{A}_1 \hat{\mathbf{H}}_1 \underline{\mathbf{x}}) - 1\right)^2 d\mathbf{x}$$

- by minimization of image information loss
  [Matoušek, ICIG 2004]

- non-linear rectification    suitable for forward motion
  non-parametric: [Pollefeys et al. 1999]
  analytic: [Geyer & Daniilidis 2003]



forward egomotion
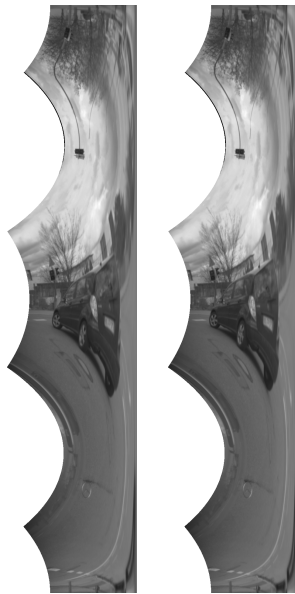


rectified images, Pollefeys' method

# ►Binocular Disparity in Standard Stereo Pair



top view in $xz$ plane

side view in $yz$ plane
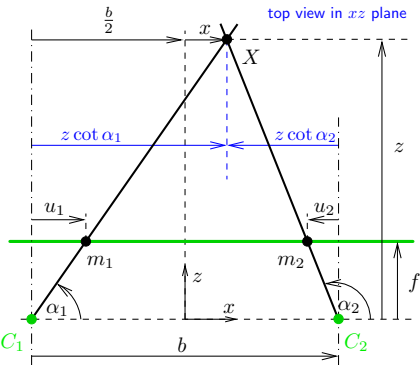
- Assumptions: single image line, standard camera pair

$$b = z \cot \alpha_1 - z \cot \alpha_2$$
$$u_1 = f \cot \alpha_1 \qquad\qquad u_2 = f \cot \alpha_2$$
$$b = \frac{b}{2} + x - z \cot \alpha_2$$

$X = (x, y, z)$ from **disparity** $d = u_1 - u_2$:

$$\boxed{z = \frac{b\,f}{d}}\,, \quad x = \frac{b}{d}\,\frac{u_1 + u_2}{2}, \quad y = \frac{b\,v}{d}$$

$f, d, u, v$ in pixels, $b, x, y, z$ in meters

## Observations

- constant disparity surface is a frontoparallel plane
- distant points have small disparity
- relative error in $z$ is large for small disparity

$$\frac{1}{z}\,\frac{\mathrm{d}z}{\mathrm{d}d} = -\frac{1}{d}$$

- increasing the baseline or the focal length increases disparity and reduces the error
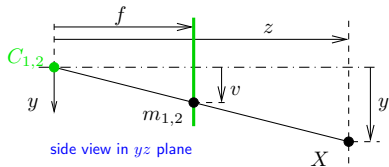
# Structural Ambiguity in Stereovision

- suppose we can recognize local matches independently but have no scene model
- lack of an occlusion model
- lack of a continuity model   $\Rightarrow$   structural ambiguity in the presence of repetitions (or lack of texture)



left image

right image
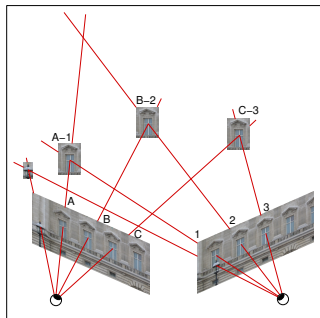


interpretation 1

interpretation 2

occluded

$r_3$

$r_1$

surface pt.

$r_2$

transparent

$l$

$X_2$ $Y_2$ $Y_3$

$Y_1$ $X_1$

half occlusion

$X$

mutual occlusion
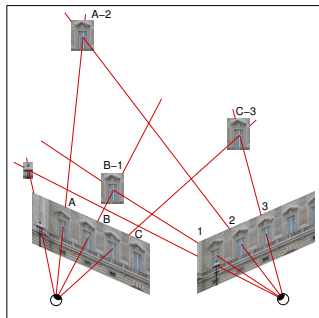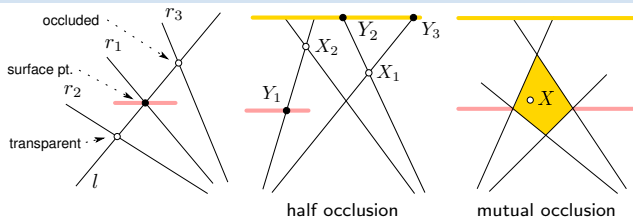
- surface point at the intersection of rays $l$ and $r_1$ occludes a world point at the intersection $(l, r_3)$ and implies the world point $(l, r_2)$ is transparent, therefore

$$(l, r_3) \text{ and } (l, r_2) \text{ are \underline{excluded} by } (l, r_1)$$

- in half-occlusion, every world point such as $X_1$ or $X_2$ is excluded by a binocularly visible surface point such as $Y_1$, $Y_2$, $Y_3$    ⇒ decisions on correspondences <u>are not</u> independent

- in mutual occlusion this is no longer the case: any $X$ in the yellow zone is <u>not excluded</u>
  ⇒ decisions in the zone <u>are</u> independent on the rest
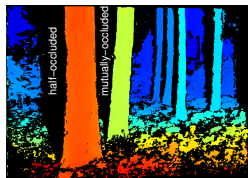
Based on scene opacity and the observation on mutual exclusion we expect each pixel to match at most once.



rays in epipolar plane
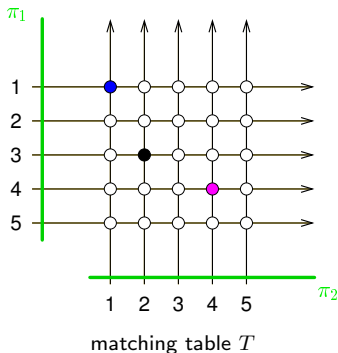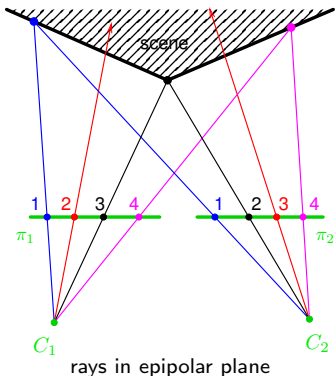


matching table $T$

**matching table**
- rows and columns represent optical rays
- nodes: possible correspondence pairs
- full nodes: matches
- numerical values associated with nodes: descriptor similarities    see next

# ▶Constructing An Image Similarity Cost

- let $p_i = (l, r)$ and $\mathbf{L}(l)$, $\mathbf{R}(r)$ be (left, right) image descriptors (vectors) constructed from local image neighborhood windows

in matching table $T$:
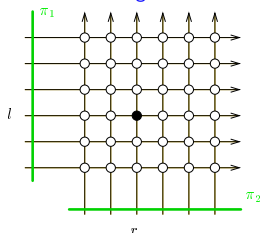


'block' in the left image $\mapsto$ 'signal sample':
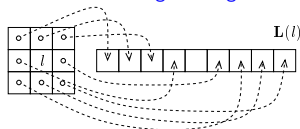


- a simple block similarity is $\mathrm{SAD}(l, r) = \|\mathbf{L}(l) - \mathbf{R}(r)\|_1$     $L_1$ metric (sum of absolute differences)

- a scaled-descriptor similarity is   $\mathrm{sim}(l, r) = \dfrac{\|\mathbf{L}(l) - \mathbf{R}(r)\|^2}{\sigma_I^2(l, r)}$     smaller is better

- $\sigma_I^2$ – the difference <u>scale</u>; a suitable (plug-in) estimate is $\frac{1}{2}\left[\mathrm{var}\big(\mathbf{L}(l)\big) + \mathrm{var}\big(\mathbf{R}(r)\big)\right]$, giving

$$\mathrm{sim}(l, r) = 1 - \underbrace{\frac{2\,\mathrm{cov}\big(\mathbf{L}(l), \mathbf{R}(r)\big)}{\mathrm{var}\big(\mathbf{L}(l)\big) + \mathrm{var}\big(\mathbf{R}(r)\big)}}_{\rho\big(\mathbf{L}(l), \mathbf{R}(r)\big)} \qquad (34)$$

  $\mathrm{var}(\cdot)$, $\mathrm{cov}(\cdot)$ is sample (co-)variance, not invariant to scale difference

- $\rho$ – MNCC – Moravec's Normalized Cross-Correlation statistic   bigger is better   [Moravec 1977]

$$\rho^2 \in [0, 1], \qquad \mathrm{sign}\,\rho \sim \text{'phase'}$$

# How A Scene Looks in The Filled-In Matching Table



scene      left image      right image

$5 \times 5$ window    $11 \times 11$ window    $3 \times 3$ window

a good tradeoff    occlusion artefacts    undiscriminable
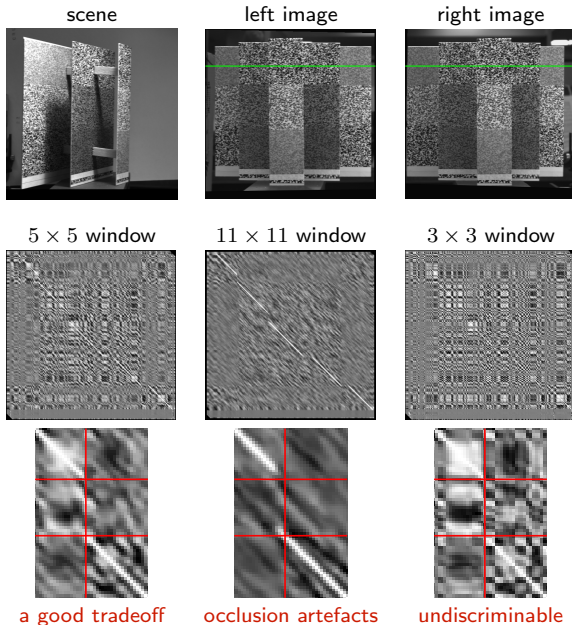
- MNCC $\rho$ used
  ($\alpha = 1.5$, $\beta = 1$) →175
- high-correlation structures correspond to scene objects

**constant disparity**

- a diagonal in matching table
- zero disparity is the main diagonal  nonstd rectification

**depth discontinuity**

- horizontal or vertical jump in matching table

**large image window**

- better correlation
- worse occlusion localization

**repeated texture**

- horizontal and vertical block repetition

# Image Point Descriptors And Their Similarity

**Descriptors:** Image points are tagged by their (viewpoint-invariant) physical properties:

- texture window                                                [Moravec 77]
- a descriptor like DAISY                                        [Tola et al. 2010]
- learned descriptors
- reflectance profile under a moving illuminant
- photometric ratios                              [Wolff & Angelopoulou 93-94]
- dual photometric stereo                                       [Ikeuchi 87]
- polarization signature
- . . .

- similar points are more likely to match
- image similarity values for all 'match candidates' give the 3D <u>matching table</u>
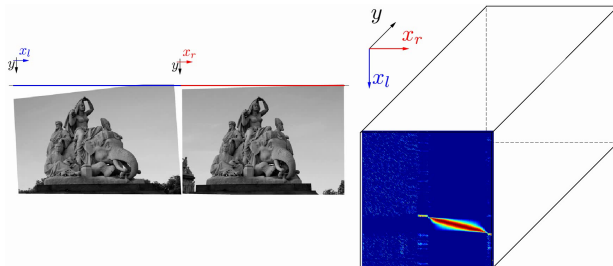


video

**Alg:** Per left-image pixel: The most SAD-similar pixel along the right epipolar line →168

1. select disparity range                                                    this is a critical weak point
2. represent the matching table diagonals in a compact form
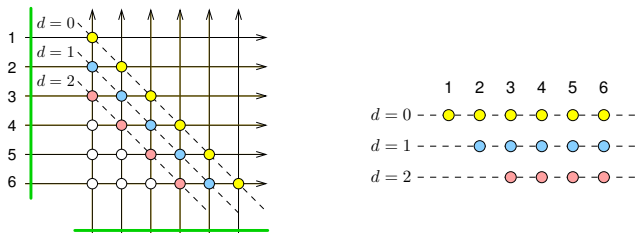


3. use an 'image sliding & cost aggregation algorithm'



image shifted by $d = 1$ pixel
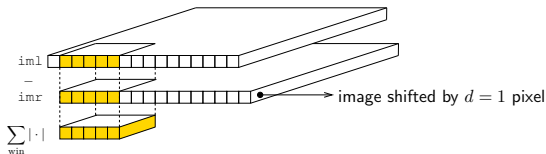
4. take the maximum over disparities $d$
5. threshold results by maximal allowed SAD dissimilarity

## A Matlab Code for WTA

```matlab
function dmap = marroquin(iml, imr, disparityRange)
%        iml, imr - rectified gray-scale images
% disparityRange - non-negative disparity range

% (c) Radim Sara (sara@cmp.felk.cvut.cz) FEE CTU Prague, 10 Dec 12

 thr = 20;                                    % bad match rejection threshold
 r = 2;
 winsize = 2*r+[1 1];                         % 5x5 window (neighborhood) for r=2
 N = boxing(ones(size(iml)), winsize);        % the size of each local patch is
                                              % N = (2r+1)^2 except for boundary pixels

 % --- compute dissimilarity per pixel and disparity --->

 for d = 0:disparityRange                     % cycle over all disparities
  slice = abs(imr(:,1:end-d) - iml(:,d+1:end)); % pixelwise dissimilarity (unscaled SAD)
  V(:,d+1:end,d+1) = boxing(slice, winsize)./N; % window aggregation
 end

 % --- collect winners, threshold, output disparity map --->

 [cmap,dmap] = min(V,[],3);                   % collect winners and their dissimilarities
 dmap(cmap > thr) = NaN;                      % mask-out high dissimilarity pixels
end % of marroquin

function c = boxing(im, wsz)
 % if the mex is not found, run this slow version:
 c = conv2(ones(1,wsz(1)), ones(wsz(2),1), im, 'same');
end % of boxing
```

thr = 20           thr = 10
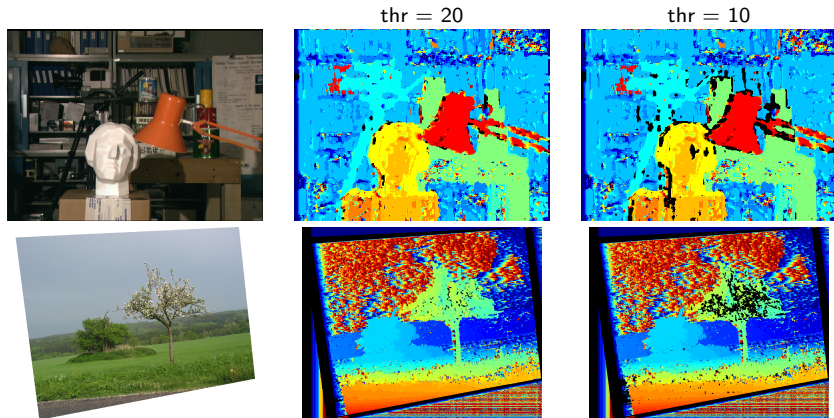
- results are fairly bad
- false matches in textureless image regions and on repetitive structures (book shelf)
- a more restrictive threshold (thr = 10) does not work as expected
- we searched the true disparity range, results get worse if the range is set wider
- chief failure reasons:
  - unnormalized image dissimilarity does not work well
  - no occlusion model (it just ignores the occlusion structure we have discussed →166)

Empirical Distribution of MNCC $\rho$ for Matches (green) and Non-Matches (red)



MNCC, 5×5 window, 50 KITTI images
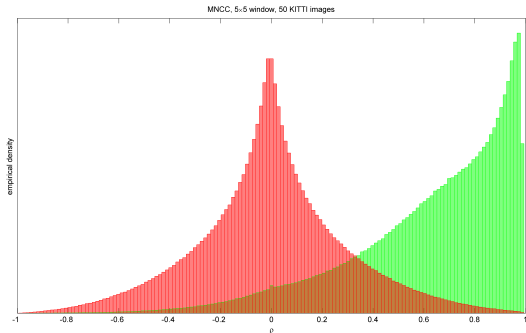
- histograms of $\rho$ computed from $5 \times 5$ correlation window        $\rho$: bigger is better
- KITTI dataset
    - $4.2 \cdot 10^6$ ground-truth (LiDAR) matches for $p_1(\rho)$ (green),
    - $4.2 \cdot 10^6$ random non-matches for $p_0(\rho)$ (red)

**Obs:**
- non-matches (red) may have arbitrarily large $\rho$
- matches (green) may have arbitrarily low $\rho$
- $\rho = 1$ is improbable for matches

# Match Likelihood

- $\rho$ is just a normalized measurement
- we need a probability distribution on $[0, 1]$, e.g. Beta distribution

$$p_1(\rho) = \frac{1}{B(\alpha, \beta)} \, |\rho|^{\alpha-1} (1 - |\rho|)^{\beta-1}$$

- note that uniform distribution is obtained for $\alpha = \beta = 1$
- when $\alpha = 2$ and $\beta = 1$ then $p_1(\cdot) = 2|\rho|$



negative log-likelihoods $V_0$ (red), $V_1$ (blue)

$\alpha$=10, $\beta$=1.5

- the mode is at $\sqrt{\frac{\alpha-1}{\alpha+\beta-2}} \approx 0.9733$ for $\alpha = 10$, $\beta = 1.5$
- if we chose $\beta = 1$ then the mode was at $\rho = 1$
- perfect similarity is 'suspicious' (depends on expected camera noise level)
- from now on we will work with negative log-likelihood cost

$$V_1\big(\rho(l,r)\big) = -\log p_1\big(\rho(l,r)\big) \qquad \text{smaller is better} \tag{35}$$

- we should also define similarity (and negative log-likelihood $V_0(\rho(l,r))$) for non-matches
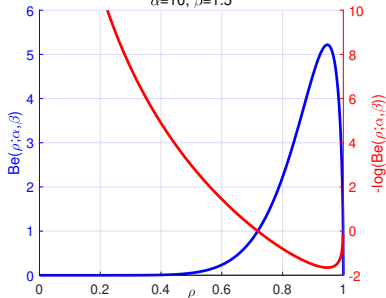
- given matching $M$ what is the likelihood of observed data $D$?
- data – all cost pairs $(V_0, V_1)$ in the matching table $T$
- matches – pairs $p_i = (l_i, r_i), \quad i = 1, \dots, n$
- matching: partitioning matching table $T$ to matched $M$ and excluded $E$ pairs

$$T = M \cup E, \quad M \cap E = \emptyset$$

- matching cost (negative log-likelihood, smaller is better)

$$V(D \mid M) = \sum_{p \in M} V_1(D \mid p) + \sum_{p \in E} V_0(D \mid p)$$

$V_1(D \mid p)$ – negative log-probability of data $D$ at <u>matched</u> pixel $p$ (35)
$V_0(D \mid p)$ – ditto at <u>unmatched</u> pixel $p$ <span style="color:blue">→174 and →175</span>

- matching problem

$$M^* = \arg \min_{M \in \mathcal{M}(T)} V(D \mid M)$$

$\mathcal{M}(T)$ – the set of all matchings in table $T$

- <u>symmetric</u>: formulated over pairs, invariant to left $\leftrightarrow$ right image swap <span style="color:blue">unlike in WTA</span>

- we need to reduce matching to a standard polynomial-complexity problem
- convert the matching cost to an 'easier' sum

$$V(D \mid M) = \sum_{p \in M} V_1(D \mid p) + \sum_{p \in E} V_0(D \mid p) + \overbrace{\sum_{p \in M} V_0(D \mid p) - \sum_{p \in M} V_0(D \mid p)}^{0}$$

$$= \sum_{p \in M} \underbrace{\Big(V_1(D \mid p) - V_0(D \mid p)\Big)}_{-L(D \mid p)} + \underbrace{\sum_{p \in E} V_0(D \mid p) + \sum_{p \in M} V_0(D \mid p)}_{\displaystyle\sum_{p \in T} V_0(D \mid p) = \text{const}}$$

- hence

$$\arg \min_{M \in \mathcal{M}(T)} V(D \mid M) = \arg \max_{M \in \mathcal{M}(T)} \sum_{p \in M} L(D \mid p) \tag{36}$$

$L(D \mid p)$ – logarithm of matched-to-unmatched likelihood ratio (bigger is better)

why this way: we want to use maximum-likelihood but our measurement is all data $D$

- (36) is max-cost matching (maximum assignment) for the maximum-likelihood (ML) matching problem
  - use Hungarian (Munkres) algorithm and threshold the result with $T$: $L(D \mid p) > T \geq 0$
  - or step back: sacrifice symmetry to speed and use dynamic programming
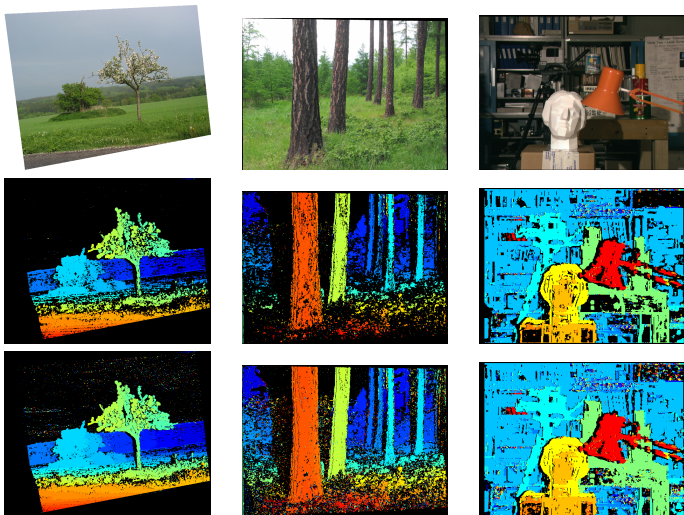
- unlike the WTA we can efficiently control the density/accuracy tradeoff      black = no match
- middle row: threshold $T$ for $L(D \mid p)$ set to achieve error rate of 3% (and 61% density results)
- bottom row: threshold $T$ set to achieve density of 76% (and 4.3% error rate results)

- notice many small isolated errors in the ML matching
- Q: how to reduce the noisiness? A: a stronger model

**Potential models for $M$ (from weaker to stronger)**

1. <u>Uniqueness</u>: Every image point matches at most once
   - excludes semi-transparent objects
   - used in the ML matching algorithm (but not by the WTA algorithm)

2. <u>Monotonicity</u>: Matched pixel ordering is preserved          →180
   - for all $(i, j) \in M, (k, l) \in M, \quad k > i \Rightarrow l > j$
     Notation: $(i, j) \in M$ or $j = M(i)$ – left-image pixel $i$ matches right-image pixel $j$
   - excludes thin objects close to the cameras
   - used in 3-Label Dynamic Programming (3LDP) [SP]

3. <u>Coherence</u>: Objects occupy well-defined 3D volumes
   - concept by [Prazdny 85]
   - algorithms are based on image/disparity map segmentation
   - a popular model (segment-based, bilateral filtering and their successors)
   - used in Stable Segmented 3LDP [Aksoy et al. PRRS 2008]

4. <u>(Piecewise) binocular continuity</u>: The scene images continuously w/o self-occlusions
   - disparities do not differ much in neighboring pixels (except at object boundaries)
   - full binocular continuity too strong, except in some applications
   - piecewise binocular continuity is combined with monotonicity in 3LDP

# Binocular Discontinuities in Matching Table
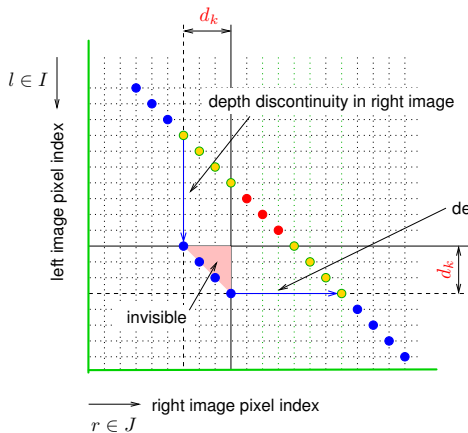


- binocularly visible foreground points
- binocularly visible background pts violating ordering
- monocularly visible points
- $d_k$ critical disparity

$l \in I$ — left image pixel index

$r \in J$ — right image pixel index

$d_k$

depth discontinuity in right image

depth discontinuity in left image

invisible

- this leads to the concept of 'forbidden zone'

$X$-zone and $F$-zone
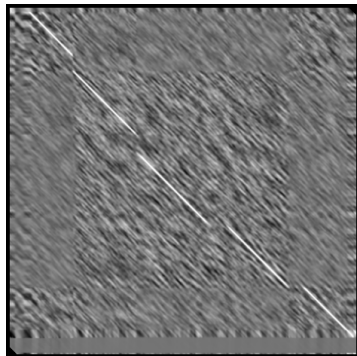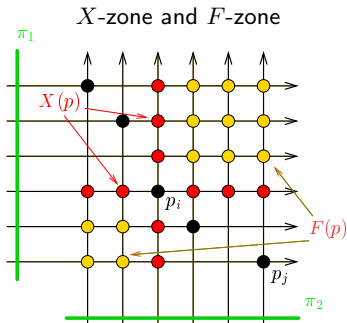


$$p_j \notin X(p_i), \quad p_j \notin F(p_i)$$

- **Uniqueness Constraint:**

  A set of pairs $M = \{p_i\}_{i=1}^n$, $p_i \in T$ is a matching iff
  $$\forall p_i, p_j \in M : p_j \notin X(p_i).$$

  $X$-zone, $p_i \notin X(p_i)$

- **Ordering Constraint:**

  Matching $M$ is monotonic iff
  $$\forall p_i, p_j \in M : p_j \notin F(p_i).$$

  $F$-zone, $p_i \notin F(p_i)$

- ordering constraint: matched points form a monotonic set in both images
- ordering is a powerful constraint: in $n \times n$ table we have monotonic matchings $O(4^n) \ll O(n!)$ all matchings
  - ⊛ 2: how many are there <u>maximal</u> monotonic matchings? (e.g. 27 for $n = 4$; hard!)

- uniqueness constraint is a basic <u>occlusion model</u>
- ordering constraint is a <u>weak continuity model</u>     and partly also an occlusion model
- monotonic matching can be found by **dynamic programming**

left image
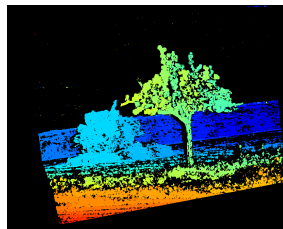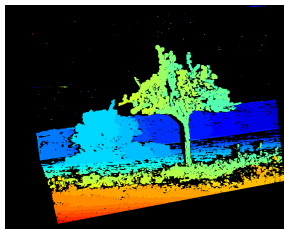
right image

ML →177
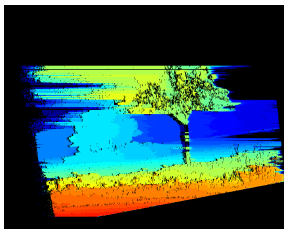
3LDP w/ordering
[SP]

naïve DP
[Cox et al. 1992]
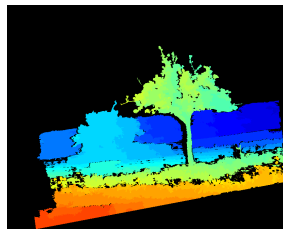
Stable Segmented 3LDP
[Aksoy et al. PRRS 2008]

- **3LDP parameters** $\alpha_i$, $V_e$ **learned on Middlebury stereo data** http://vision.middlebury.edu/stereo/

## Some Results: Larch



left image

right image
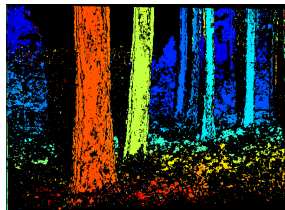
ML →177
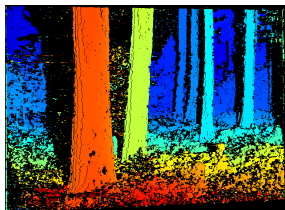
3LDP w/ordering [SP]
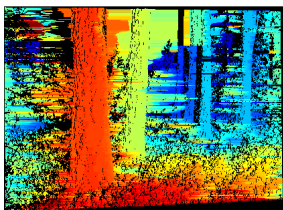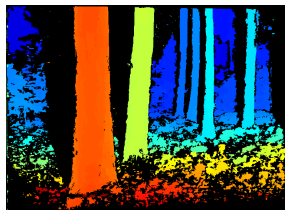
naïve DP

Stable Segmented 3LDP

- naïve DP: no mutual occlusion model, ignores symmetry, has no similarity distribution model
- but even 3LDP has errors in mutually occluded region
- Stable Segmented 3LDP: few errors in mutually occluded region since it uses a coherence model

# Algorithm Comparison

## Marroquin's Winner-Take-All (WTA →171)

- the ur-algorithm *very weak model*
- dense disparity map
- $O(N^3)$ algorithm, simple but it rarely works

## Maximum Likelihood Matching (ML →177)

- semi-dense disparity map
- many small isolated errors
- models basic occlusion
- $O(N^3 \log(NV))$ algorithm *max-flow by cost scaling*

## MAP with Min-Cost Labeled Path (3LDP)

- semi-dense disparity map
- models occlusion in flat, piecewise binocularly continuous scenes
- has 'illusions' if ordering does not hold
- $O(N^3)$ algorithm

## Stable Segmented 3LDP

- better than 3LDP *fewer errors at any given density*
- $O(N^3 \log N)$ algorithm
- requires image segmentation *itself a difficult task*



ROC curves and their average error rate bounds

Legend:
- 3LDP (3.65 ± 0.26)
- WTA (4.71 ± 0.17)
- ML (4.60 ± 0.65)
- GCS (4.29 ± 1.47)
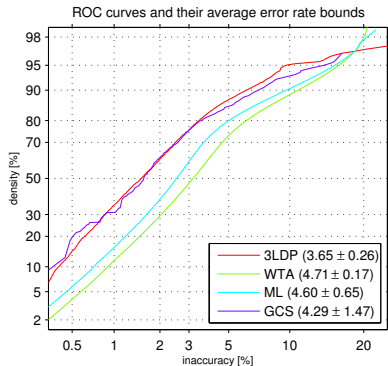
- ROC-like curve captures the density/accuracy tradeoff
- numbers: AUC (smaller is better)
- GCS is the one used in the exercises
- more algorithms at http://vision.middlebury.edu/stereo/ (good luck!)

# A Summary of This Course Highlights

- homography as a two-image model
- epipolar geometry as a two-image model
- core algorithms for 3D vision:
  - simple intrinsic calibration methods
  - 6-pt alg for camera resection and 3-pt alg for exterior orientation (calibrated resection)
  - 7-pt alg for fundamental matrix, 5-pt alg for essential matrix
  - essential matrix decomposition to rotation and translation
  - efficient accurate triangulation
  - robust matching by RANSAC sampling
  - camera system reconstruction
  - efficient bundle adjustment
  - stereoscopic matching
- statistical robustness as a way to work with partially unknown information

**What can we do with these tools?**

- 3D scene reconstruction
- visual odometry
- motion capture
- self-localization and mapping (not covered: 3D aggregation in scene maps)
- 3D scene measurement for robot motion planning
- automatic extrinsic calibration from motion (hand-eye calibration)

Thank You

$C_1$

$m_a$  $m_s$
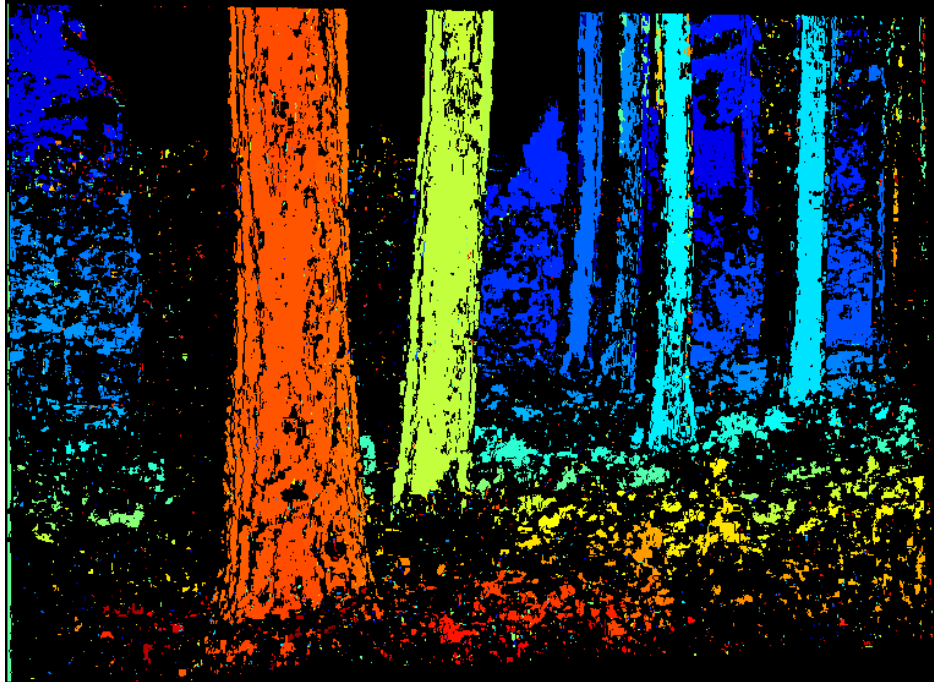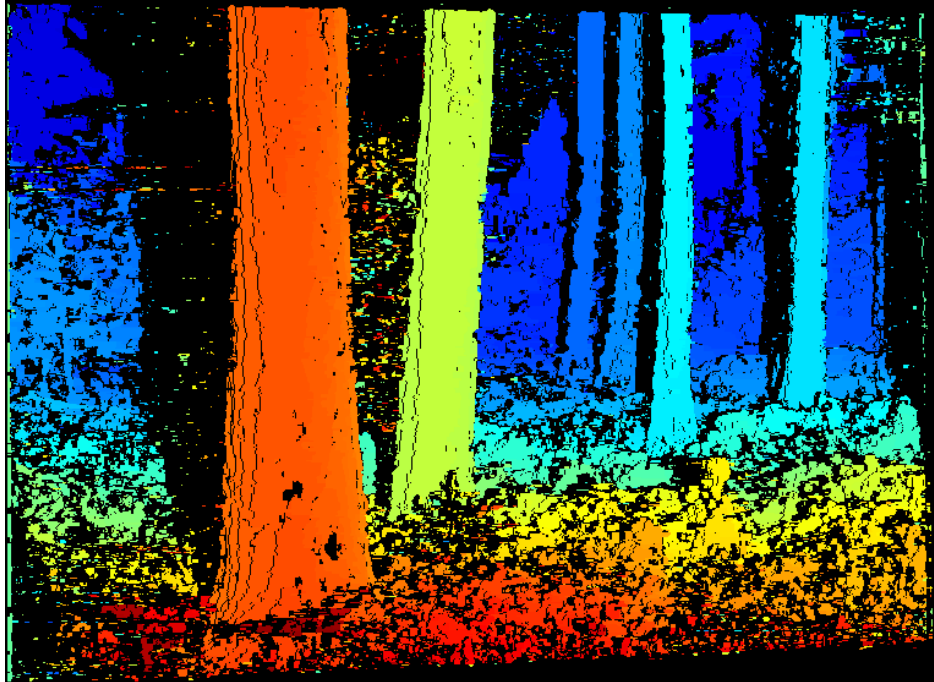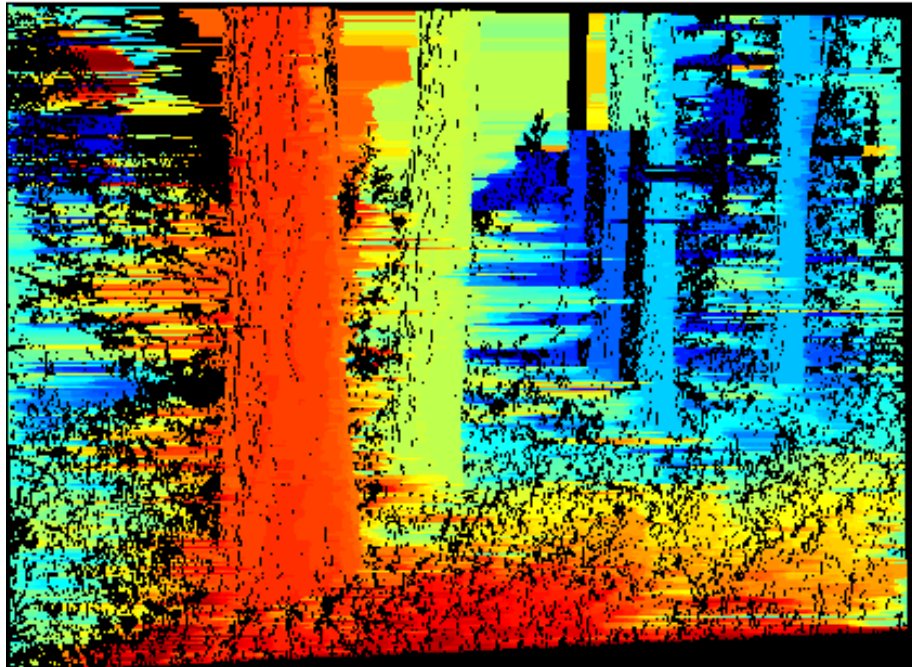
$e_1$

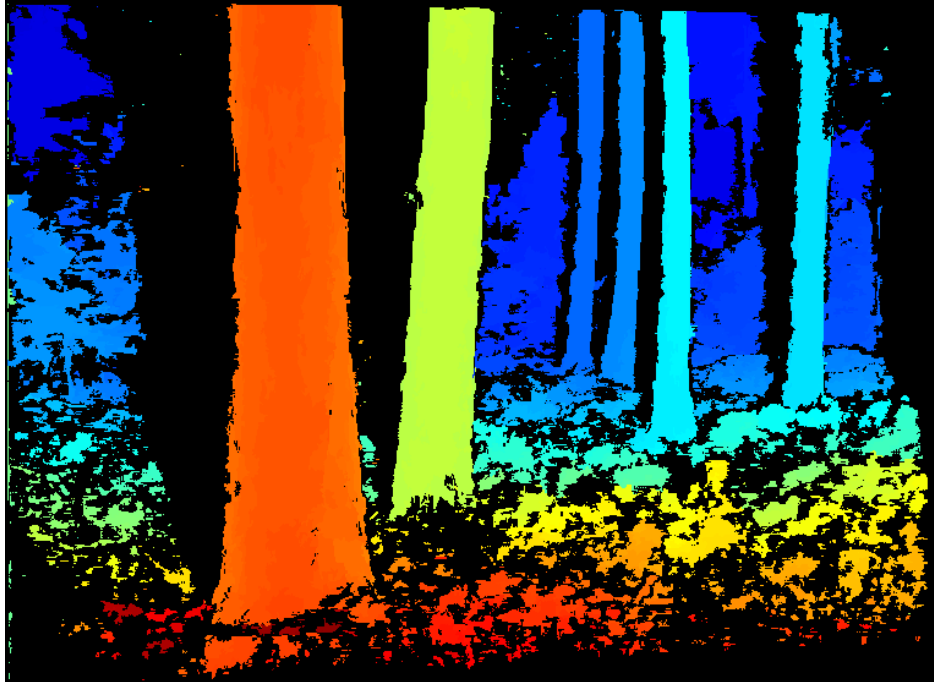$m_T = m$

half-occluded

mutually-occluded

R. Šára, CMP; rev. 5–Jan–2021

ROC curves and their average error rate bounds

- 3LDP (3.65 ± 0.26)
- WTA (4.71 ± 0.17)
- ML (4.60 ± 0.65)
- GCS (4.29 ± 1.47)

x-axis: inaccuracy [%]
y-axis: density [%]