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. December 19, 2023

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 names and representations
- names and their components:

entity	in 2-space	in 3-space
point	$m = (u, v)$	$X = (x, y, z)$
line	n	O
plane		π, φ

- 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 $\qquad (u, v) \in \mathbb{R}^2 \quad \text{s.t.} \quad a\,u + b\,v + c = 0$

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

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

'Finite' lines

- standard representative for finite $\underline{\mathbf{n}} = (n_1, n_2, n_3)$ 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)

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

- 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 $\underline{\mathbf{n}} = (a, b, c)$ iff

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 also 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 finite point $\underline{\mathbf{m}}$ is $\lambda\,\underline{\mathbf{m}}$, where $\lambda = \frac{\mathbf{1}}{m_3}$ — 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

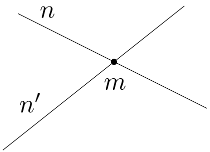
'Infinite' points

- 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

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

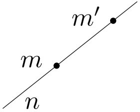


proof: If $\underline{\mathbf{m}} = \underline{\mathbf{n}} \times \underline{\mathbf{n}}'$ is the intersection point, it must be incident on both lines. Indeed, using known equivalences from vector algebra

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

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

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

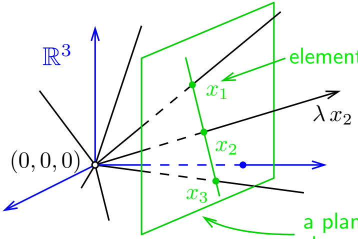


Paralel lines intersect (somewhere) on the line at infinity $\underline{\mathbf{n}}_\infty \simeq (0, 0, 1)$:

$$a\,u + b\,v + c = 0,$$
$$a\,u + b\,v + d = 0, \qquad d \neq c$$
$$(a, b, c) \times (a, b, d) \simeq (b, -a, 0)$$

- 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);`

▶Homography in $\mathbb{P}^2$



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

we call $\underline{\mathbf{x}} \in \mathbb{P}^2$ 'points'

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

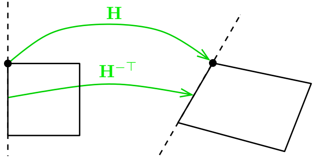
$$\underline{\mathbf{x}}' \simeq \mathbf{H}\, \underline{\mathbf{x}}, \quad \mathbf{H} \in \mathbb{R}^{3,3} \text{ non-singular}$$

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
- and point-line incidence is preserved — e.g. line intersection points mapped to line intersection points

- $\mathbf{H}$ is a 3×3 non-singular matrix, $\lambda\, \mathbf{H} \simeq \mathbf{H}$ equivalence class, 8 degrees of freedom
- homogeneous matrix representative: $\det \mathbf{H} = 1$ — $\mathbf{H} \in \mathrm{SL}(3)$
- what we call homography here is often called 'projective collineation' in mathematics

▶Mapping 2D Points and Lines by Homography



$\underline{\mathbf{m}}' \simeq \mathbf{H}\,\underline{\mathbf{m}}$	(image) point	
$\underline{\mathbf{n}}' \simeq \mathbf{H}^{-\top}\underline{\mathbf{n}}$	(image) line	$\mathbf{H}^{-\top} = (\mathbf{H}^{-1})^{\top} = (\mathbf{H}^{\top})^{-1}$

- 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

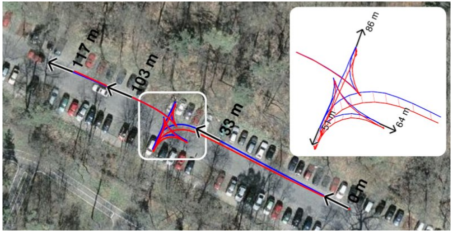
Some Homographic Tasters

Rectification of camera rotation: →59 (geometry), →132 (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]

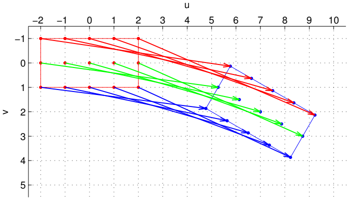
illustrations courtesy of AMSL Racing Team, Meiji University and LIBVISO: Library for VISual Odometry

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

▶Homography Subgroups: Euclidean Mapping (aka Rigid Motion)

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

$$\mathbf{H} = \begin{bmatrix} \cos\phi & -\sin\phi & t_x \\ \sin\phi & \cos\phi & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathrm{SE}(2)$$



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

- note: action $H(\mathbf{x}) = \mathbf{R}\mathbf{x} + \mathbf{t} \colon \mathbb{R}^2 \to \mathbb{R}^2$, not commutative

EM = The most general homography preserving

1. **lengths**: Let $\mathbf{x}'_i = H(\mathbf{x}_i)$. Then

$$\|\mathbf{x}'_2 - \mathbf{x}'_1\| = \|H(\mathbf{x}_2) - H(\mathbf{x}_1)\| \overset{\circledast\ \text{P1; 1pt}}{=} \cdots = \|\mathbf{x}_2 - \mathbf{x}_1\|$$

2. **angles** check the dot-product of normalized differences from a point $(\mathbf{x}-\mathbf{z})^\top(\mathbf{y}-\mathbf{z})$ (Cartesian(!))
3. **areas**: $\det \mathbf{H} = 1 \ \Rightarrow$ unit determinant of the action's Jacobian $\mathbf{J}$ it follows from: $\mathbf{J} = \mathbf{R}$, $\det \mathbf{R} = 1$

- eigenvalues $(1,\ e^{-i\phi},\ e^{i\phi})$
- eigenvectors when $\phi \neq k\pi,\ k = 0, 1, \ldots$ (columnwise)

$$\mathbf{e}_1 \simeq \begin{bmatrix} t_x + t_y \cot\frac{\phi}{2} \\ t_y - t_x \cot\frac{\phi}{2} \\ 2 \end{bmatrix}, \quad \mathbf{e}_2 \simeq \begin{bmatrix} i \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 \simeq \begin{bmatrix} -i \\ 1 \\ 0 \end{bmatrix}$$

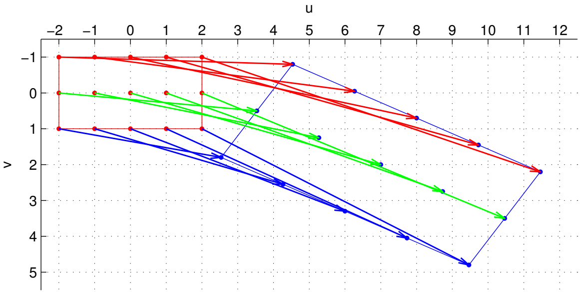
$\mathbf{e}_2$, $\mathbf{e}_3$ – circular points, i – imaginary unit

4. **circular points**: complex points at infinity $(i, 1, 0)$, $(-i, 1, 0)$ (preserved even by similarity)

- **similarity**: scaled Euclidean mapping (does not preserve lengths, areas)

▶Homography Subgroups: Affine Mapping (Affinity)

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



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

Affinity = The most general homography preserving

- parallelism
- ratio of areas
- ratio of lengths on parallel lines
- linear combinations of vectors (e.g. midpoints, centers of gravity)
- convex hull
- line at infinity $\underline{\mathbf{n}}_\infty$ (not pointwise)

observe $\mathbf{H}^\top \underline{\mathbf{n}}_\infty \simeq \begin{bmatrix} a_{11} & a_{21} & 0 \\ a_{12} & a_{22} & 0 \\ t_x & t_y & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \underline{\mathbf{n}}_\infty \quad \Rightarrow \quad \underline{\mathbf{n}}_\infty \simeq \mathbf{H}^{-\top} \underline{\mathbf{n}}_\infty$

does not preserve

- lengths
- angles
- areas
- circular points

►Homography Subgroups: General Homography

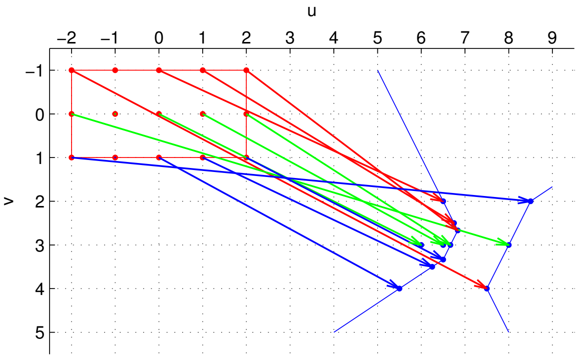
$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \qquad \mathbf{H} \in \mathrm{SL}(3)$$

preserves only

- incidence and concurrency
- collinearity
- cross-ratio (ratio of ratios) on the line →46

does not preserve

- lengths
- areas
- parallelism
- ratio of areas
- ratio of lengths
- linear combinations of vectors
- convex hull
- line at infinity $\underline{\mathbf{n}}_\infty$



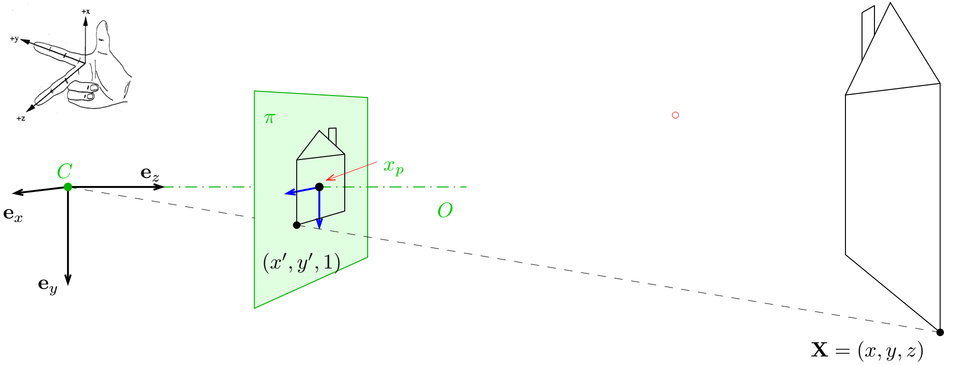
$$\mathbf{H} = \begin{bmatrix} 7 & -0.5 & 6 \\ 3 & 1 & 3 \\ 1 & 0 & 1 \end{bmatrix}$$

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

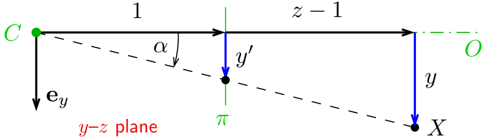
where in the picture is the line n?

$1 \cdot u + 0 \cdot v + 1 = 0$

►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 π at unit distance from C
5. optical axis O is perpendicular to π
6. principal point x_p: intersection of O and π
7. perspective camera is given by C and π



projected point in the natural image coordinate system:

$$\tan\alpha = \frac{y'}{1} = y' = \frac{y}{1+z-1} = \frac{y}{z}, \qquad x' = \frac{x}{z}$$

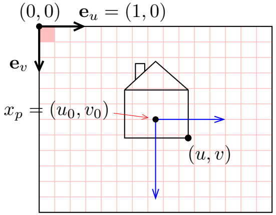
▶Natural and Canonical Image Coordinate Systems

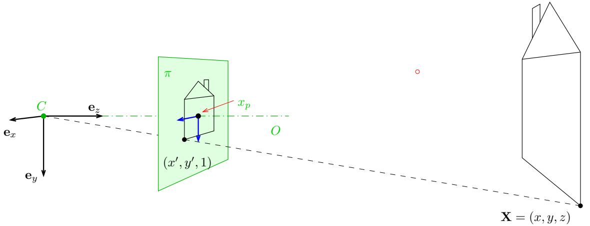
projected point **in canonical camera** $(z \neq 0)$

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

projected point **in scanned image**

scale by f and translate origin to image corner





$$u = f\,\frac{x}{z} + u_0$$

$$v = f\,\frac{y}{z} + v_0$$

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

- ‘calibration’ matrix $\mathbf{K}$ transforms canonical $\mathbf{P}_0$ to standard perspective camera $\mathbf{P}$

▶Computing with Perspective Camera Projection Matrix

Projection from world to image in standard camera $\mathbf{P}$:

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

cross-check: $\dfrac{m_1}{m_3} = \dfrac{f\,x}{z} + u_0 = u, \qquad \dfrac{m_2}{m_3} = \dfrac{f\,y}{z} + v_0 = v$ when $m_3 \neq 0$

f – ‘focal length’ – converts length ratios to pixels, $[f] = \mathrm{px}$, $f > 0$

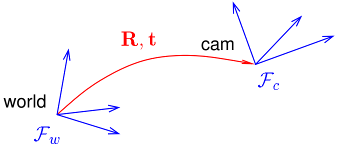
(u_0, v_0) – principal point in pixels

Perspective Camera:

1. dimension reduction — since $\mathbf{P} \in \mathbb{R}^{3,4}$
2. nonlinear unit change $\mathbf{1} \mapsto \mathbf{1} \cdot z/f$, 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. $(m_1, m_2, 0)$ represents points at infinity in image plane π — i.e. points with $z = 0$

▶Changing The Outer (World) Reference Frame



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

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

$\mathbf{R}$ – rotation matrix world orientation in the camera coordinate frame $\mathcal{F}_c$

$\mathbf{t}$ – translation vector world origin in the camera coordinate frame $\mathcal{F}_c$

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

$\mathbf{P}_0$ (a 3×4 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

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix} \qquad \text{i.e. } \mathbf{C} = -\mathbf{R}^\top\mathbf{t}$$

$\mathbf{C}$ – 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 $\mathcal{F}_w$ cam: $\mathbf{o}_c = (1, 0, 0)$, world: $\mathbf{o}_w = -\mathbf{R}^\top\mathbf{o}_c = \mathbf{r}_3^\top$ third row of $\mathbf{R}$

- 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 $\mathbf{K}$ includes

- skew angle θ of the digitization raster
- pixel aspect ratio a

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

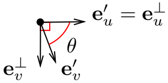
units: $[f] = \mathrm{px}$, $[u_0] = \mathrm{px}$, $[v_0] = \mathrm{px}$, $[a] = 1$

⊛ H1; 2pt: Give the parameters f, a, θ, u_0, v_0 a precise meaning by decomposing $\mathbf{K}$ to simple maps, derived from first principles; 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: Do not confuse it with the shear mapping. Express point $\mathbf{x}$ as

$$\mathbf{x} = u'\mathbf{e}_{u'} + v'\mathbf{e}_{v'} = u^\perp\mathbf{e}_u^\perp + v^\perp\mathbf{e}_v^\perp\,, \qquad u, v \in \mathbb{R}$$



$\mathbf{e}_:$ are unit-length basis vectors $\mathbf{e}_u^\perp = \mathbf{e}_u' = (1,0)$, $\mathbf{e}_v^\perp = (0,1), \ldots$

consider their four pairwise dot-products $(\mathbf{e}_u')^\top\mathbf{e}_u^\perp = 0$, $(\mathbf{e}_u')^\top\mathbf{e}_v' = \cos(\theta), \ldots$

3. $\mathbf{K}$ maps from $F^\perp$ to F''' as

$$w'''\,[u''', v''', 1]^\top = \mathbf{K}[u^\perp, v^\perp, 1]^\top$$

►Summary: Projection Matrix of a General Finite Perspective Camera

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

a recipe for filling $\mathbf{P}$

general finite perspective camera has 11 parameters:

- 5 intrinsic parameters: f, u_0, v_0, a, θ
- 6 extrinsic parameters: $\mathbf{t}$, $\mathbf{R}(\alpha, \beta, \gamma)$

finite camera: $\det \mathbf{K} \neq 0$

Representation Theorem: The set of projection matrices $\mathbf{P}$ of finite perspective cameras is isomorphic to the set of homogeneous 3×4 matrices with the left 3×3 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

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

$\mathbf{Q} \in \mathbb{R}^{3,3}$	full rank	(if finite perspective camera; see [H&Z, Sec. 6.3] for cameras at infinity)
$\mathbf{K} \in \mathbb{R}^{3,3}$	upper triangular with positive diagonal elements	
$\mathbf{R} \in \mathbb{R}^{3,3}$	rotation mtx: $\mathbf{R}^\top \mathbf{R} = \mathbf{I}$ 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]

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

$\mathbf{R}_{ij}$ zeroes element ij in $\mathbf{Q}$ affecting only columns i and j and the sequence preserves previously zeroed elements, e.g.

(see the next slide for derivation details)

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

⊛ 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]]
```

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

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

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

▶Center of Projection (Optical Center)

Observation: finite $\mathbf{P}$ has a non-trivial right null-space — rank 3 but 4 columns

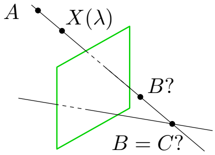
Theorem

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. Let AB be a spatial line (B given from $\mathbf{P}\underline{\mathbf{B}} = \mathbf{0}$, $A \neq B$). Then

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



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

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

⊛ verify from →30

$\underline{\mathbf{C}} = (c_j)$, 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

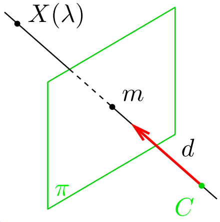
Optical ray: Spatial line that projects to a single image point.

1. Consider the following spatial line (world coordinate frame)

   $\mathbf{d} \in \mathbb{R}^3$ 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 the world coordinate frame

▶Optical Axis

Optical axis: Optical ray that is perpendicular to image plane π

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

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

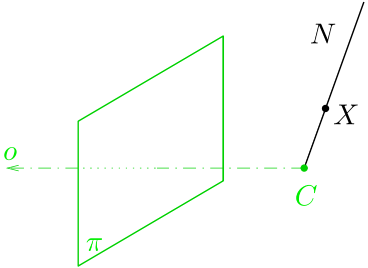
2. therefore the set of points X is parallel to π iff

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

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

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

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



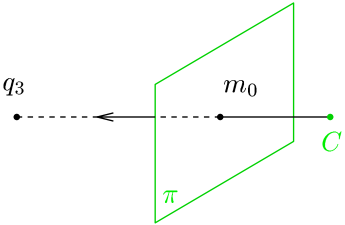
- the axis is expressed in the world coordinate frame

▶Principal Point

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

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

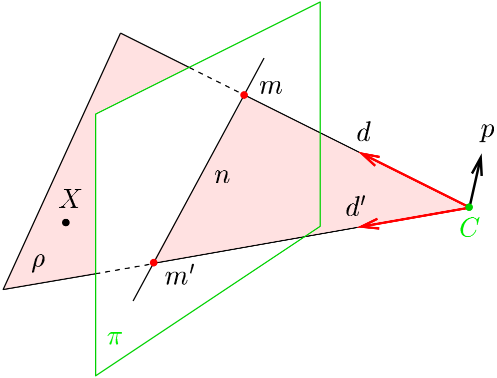


principal point: $\underline{\mathbf{m}}_0 \simeq \mathbf{Q}\,\mathbf{q}_3$

- principal point is also the center of radial distortion

▶Optical Plane

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



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

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

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

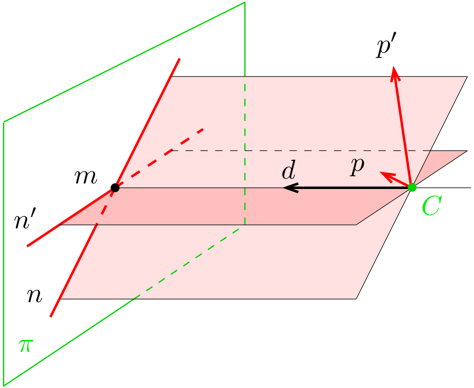
- note the way $\mathbf{Q}$ factors out!

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

optical plane is given by n: $\quad \underline{\boldsymbol{\rho}} \simeq \mathbf{P}^{\top}\underline{\mathbf{n}}$

$\underline{\boldsymbol{\rho}}$ are the plane's parameters: $\quad \rho_1\, x + \rho_2\, y + \rho_3\, z + \rho_4 = 0$

Cross-Check: Optical Ray as Optical Plane Intersection



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

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

The optical ray through their intersection is then

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

▶Summary: Projection Center; Optical Ray, Axis, Plane

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

$\mathbf{R}$ — 3D rotation matrix (cam coords.) →30

$\mathbf{t}$ — 3D translation vector (cam coords.) →30

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

How far is the engine from a given point on the tracks?

the distance between sleepers (ties) is 0.806m but we cannot count them, the image resolution is too low

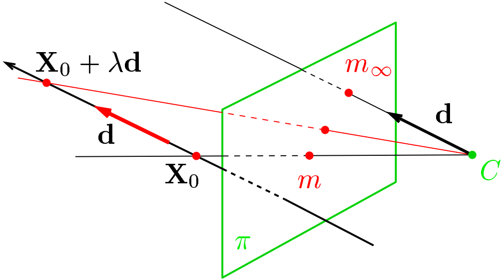
We will review some life-saving theory...
...and build a bit of geometric intuition...

In fact

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

▸Vanishing Point

Vanishing point (V.P.): The limit m_∞ of the projection of a point $\mathbf{X}(\lambda)$ that moves along a space line $\mathbf{X}(\lambda) = \mathbf{X}_0 + \lambda\mathbf{d}$ infinitely in one direction. the image of the point at infinity on the line

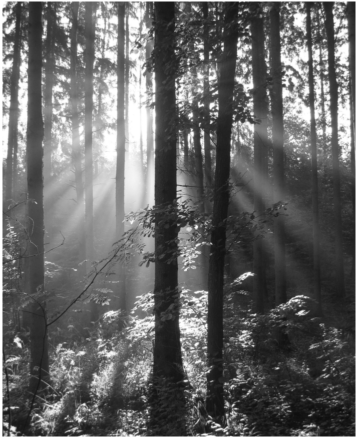


$$\underline{\mathbf{m}}_\infty \simeq \lim_{\lambda\to\pm\infty} \mathbf{P} \begin{bmatrix} \mathbf{X}_0 + \lambda\mathbf{d} \\ 1 \end{bmatrix} = \cdots \simeq \mathbf{Q}\,\mathbf{d}$$

⊛ 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_∞

Some Vanishing Point "Applications"



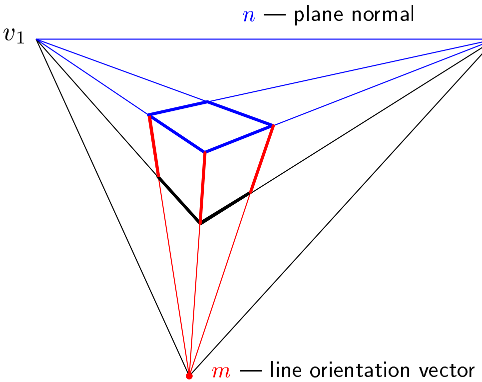
where is the sun?

what is the wind direction?
(must have video)

fly above the lane,
at constant altitude!

▶Vanishing Line

Vanishing line (V.L.): 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 (!)



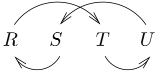
- any box with parallel edges
- top (blue) and bottom (black) box planes are parallel, hence they share V.L. n

- 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

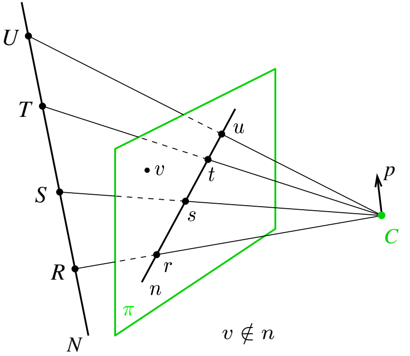
$$[RSTU] = \frac{|\overrightarrow{RT}|}{|\overrightarrow{SR}|}\,\frac{|\overrightarrow{US}|}{|\overrightarrow{TU}|}$$



a mnemonic (∞)

- $|\overrightarrow{RT}|$ – signed distance from R to T in the arrow direction
- each point X is once in numerator and once in denominator
- if X is 1st in a numerator term, it is 2nd in a denominator term
- there are six cross-ratios from four points:

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



Obs: $$[RSTU] = \frac{|\underline{\mathbf{r}}\ \underline{\mathbf{t}}\ \underline{\mathbf{v}}|}{|\underline{\mathbf{s}}\ \underline{\mathbf{r}}\ \underline{\mathbf{v}}|}\cdot\frac{|\underline{\mathbf{u}}\ \underline{\mathbf{s}}\ \underline{\mathbf{v}}|}{|\underline{\mathbf{t}}\ \underline{\mathbf{u}}\ \underline{\mathbf{v}}|}, \qquad |\underline{\mathbf{r}}\ \underline{\mathbf{t}}\ \underline{\mathbf{v}}| = \det\left[\underline{\mathbf{r}}\ \underline{\mathbf{t}}\ \underline{\mathbf{v}}\right] = (\underline{\mathbf{r}}\times\underline{\mathbf{t}})^\top\underline{\mathbf{v}} \quad \text{mixed product} \tag{1}$$

Corollaries:

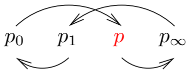
- cross ratio is invariant under homographies $\underline{\mathbf{x}}' \simeq \mathbf{H}\underline{\mathbf{x}}$ — proof: plug $\mathbf{H}\underline{\mathbf{x}}$ in (1): $(\mathbf{H}^{-\top}(\underline{\mathbf{r}}\times\underline{\mathbf{t}}))^\top\mathbf{H}\underline{\mathbf{v}}$
- cross ratio is invariant 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 $\frac{\infty}{\infty} = 1$)

▶1D Projective Coordinates

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

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



naming convention:

P_0 – the origin $\qquad [P_0] = 0$

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

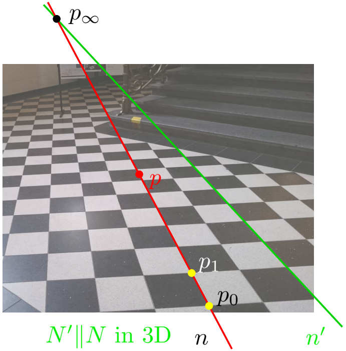
P_∞ – the supporting point $\qquad [P_\infty] = \pm\infty$

$$[P] = [p]$$

$[P]$ is equal to Euclidean coordinate along N

$[p]$ is its measurement in the image plane

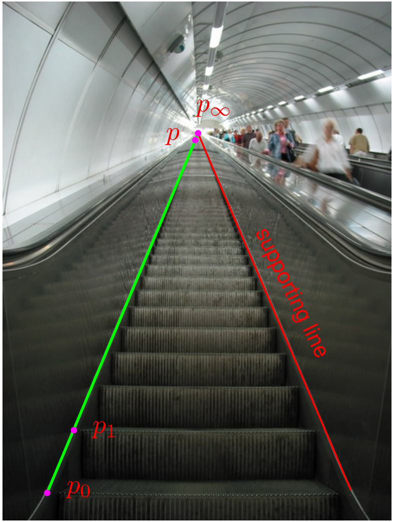
if the sign is not of interest, any cross-ratio containing $|p_0\, p|$ does the job



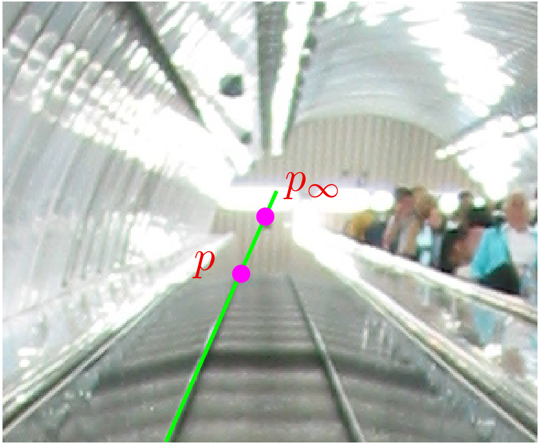
Applications

- 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 →48
- Finding V.P. of a line through a regular object →49

Application: Counting Steps



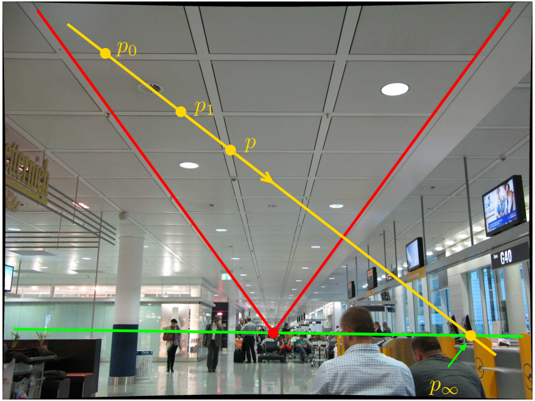
- Namesti Miru underground station in Prague

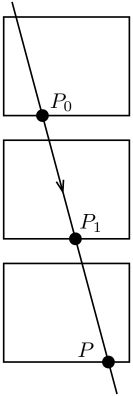


detail around the vanishing point (w/ strong aliasing)

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

4Mpx camera





in 3D: $|P_0 P| = 2|P_0 P_1|$ then [H&Z, p. 218]

$$[P_0 P_1 P P_\infty] = \frac{|P_0 P|}{|P_1 P_0|} = 2 \quad \Rightarrow \quad x_\infty = \frac{x_0\,(2\,x - x_1) - x\,x_1}{x + x_0 - 2\,x_1}$$

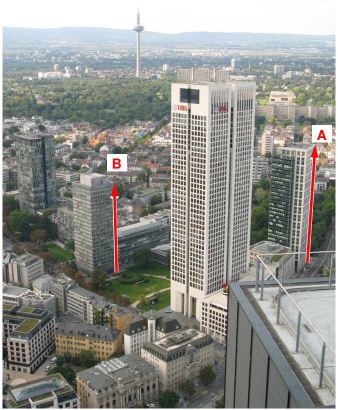
- x – 1D coordinate along the yellow line, positive in the arrow direction
- could be applied to counting steps (→48) if there was no supporting line

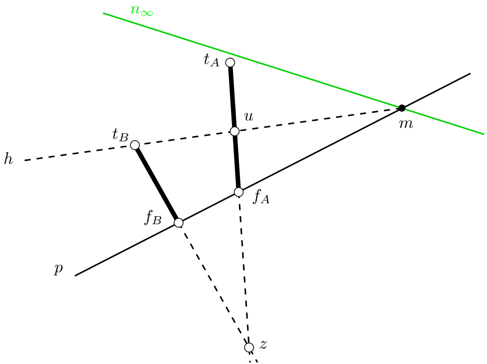
⊛ P1; 1pt: How high is the camera above the floor?

Homework Problem

⊛ 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

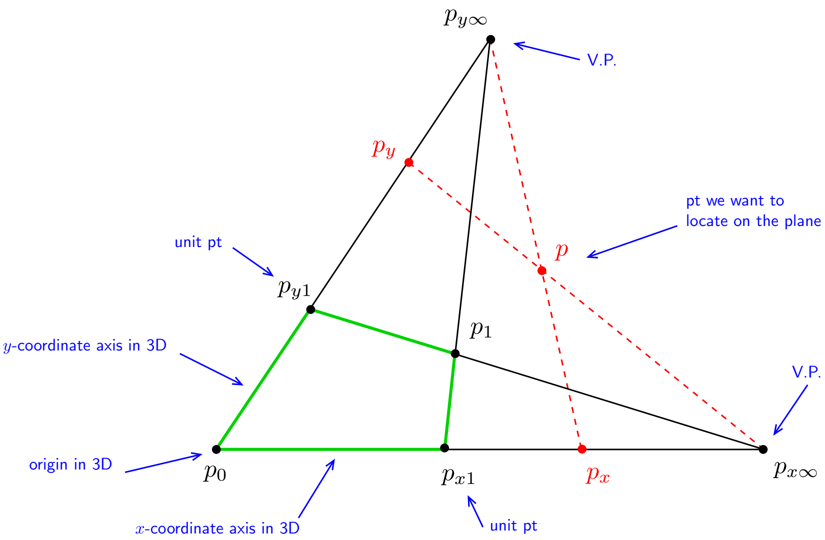




Hints

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_∞? (we do not see it directly, there are some hills there. . .) [1 point]
3. Give a formula for measuring the length ratio. Make sure you distinguish points in 3D from their images. [formula = 1 point]

2D Projective Coordinates



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

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)

Module III

Computing with a Single Camera

covered by

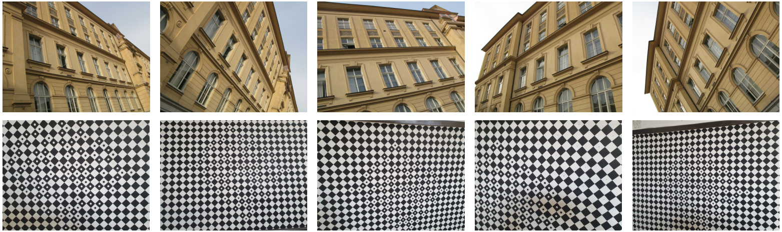
[1] [H&Z] Secs: 8.6, 7.1, 22.1

[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

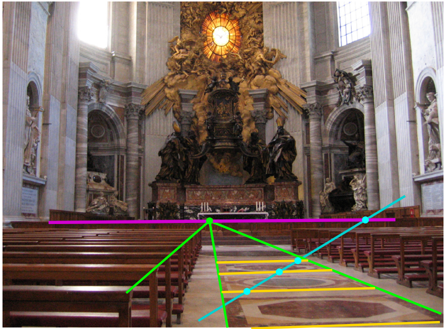
[3] [Golub & van Loan 2013, Sec. 2.5]

Obtaining Vanishing Points and Lines

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

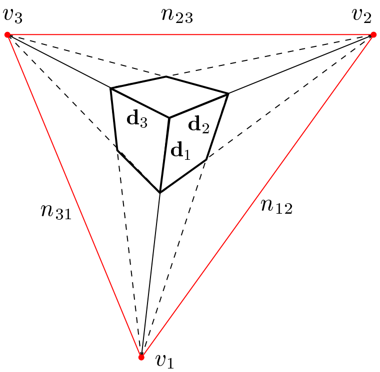


- vanishing line can be obtained from vanishing points and/or regularities (→49)



▸Camera Calibration from Vanishing Points and Lines

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



$$\mathbf{d}_i = \lambda_i \mathbf{Q}^{-1} \underline{\mathbf{v}}_i, \qquad i = 1,2,3 \quad \to 43$$
$$\mathbf{p}_{ij} = \mu_{ij} \mathbf{Q}^\top \underline{\mathbf{n}}_{ij}, \quad i,j = 1,2,3,\ i \neq j \quad \to 39 \tag{2}$$

- method: eliminate λ_i, μ_{ij}, $\mathbf{R}$ from (2) and solve for $\mathbf{K}$.

Configurations allowing elimination of $\mathbf{R}$

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 homogeneous, symmetric, definite 3×3 matrix (5 DoF) — IAC = Image of Absolute Conic
- equations are quadratic in $\mathbf{K}$ but linear in $\boldsymbol{\omega}$

►cont'd

	configuration	equation	# constraints
(3)	orthogonal vanishing points	$\underline{\mathbf{v}}_i^\top \boldsymbol{\omega}\, \underline{\mathbf{v}}_j = 0$	1
(4)	orthogonal vanishing lines	$\underline{\mathbf{n}}_{ij}^\top \boldsymbol{\omega}^{-1} \underline{\mathbf{n}}_{ik} = 0$	1
(5)	vanishing points orthogonal to vanishing lines	$\underline{\mathbf{n}}_{ij} = \varkappa\, \boldsymbol{\omega}\, \underline{\mathbf{v}}_k$	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 $\boldsymbol{\omega}$ or $\boldsymbol{\omega}^{-1}$ — $\varkappa$ can be eliminated from (5)
- When $\mathbf{w} = \mathrm{vec}(\boldsymbol{\omega}) \in \mathbb{R}^6$, it has the form of $\mathbf{D}\mathbf{w} = \mathbf{0}$, $\mathbf{D} \in \mathbb{R}^{k \times 6}$
- With $k = 5$ constraints, we have $\mathrm{rank}(\mathbf{D}) = 5$, hence there is a unique solution for the homogeneous $\mathbf{w}$.
- 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 finite 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, $\mathbf{v}_{1,2}$, $\mathbf{m}_0$ are Cartesian (not homogeneous)!

Ex 2:

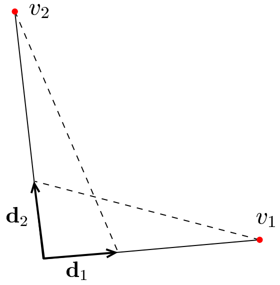
Non-orthogonal vanishing points $\mathbf{v}_i$, $\mathbf{v}_j$, known angle ϕ: $\cos\phi = \dfrac{\underline{\mathbf{v}}_i^\top \boldsymbol{\omega} \underline{\mathbf{v}}_j}{\sqrt{\underline{\mathbf{v}}_i^\top \boldsymbol{\omega} \underline{\mathbf{v}}_i}\sqrt{\underline{\mathbf{v}}_j^\top \boldsymbol{\omega}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

- 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 choice, e.g.:

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

- we know that

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

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

- knowing $\mathbf{d}_{1,2}$ we conclude that $\underline{\mathbf{w}}_i/\|\underline{\mathbf{w}}_i\|$ is the i-th column $\mathbf{r}_i$ of $\mathbf{R}$
- the third column is orthogonal: $\mathbf{r}_3 \simeq \mathbf{r}_1 \times \mathbf{r}_2$

$$\mathbf{R} = \begin{bmatrix} \frac{\underline{\mathbf{w}}_1}{\|\underline{\mathbf{w}}_1\|} & \frac{\underline{\mathbf{w}}_2}{\|\underline{\mathbf{w}}_2\|} & \frac{\underline{\mathbf{w}}_1 \times \underline{\mathbf{w}}_2}{\|\underline{\mathbf{w}}_1 \times \underline{\mathbf{w}}_2\|} \end{bmatrix}$$

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

some suitable scenes

Application: Planar Rectification

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

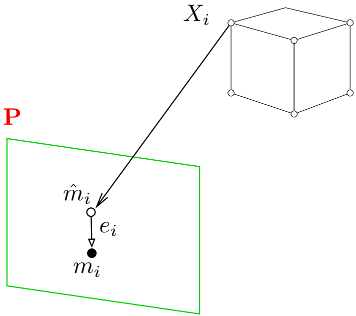
$$\underline{\mathbf{m}} \simeq \mathbf{K}\mathbf{R}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix}\underline{\mathbf{X}} \qquad \underline{\mathbf{m}}' \simeq \mathbf{K}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix}\underline{\mathbf{X}}$$

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

- $\mathbf{H}$ 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 calibration and orientation from a known set of $k \geq 6$ reference points and their images $\{(X_i, m_i)\}_{i=1}^{6}$.

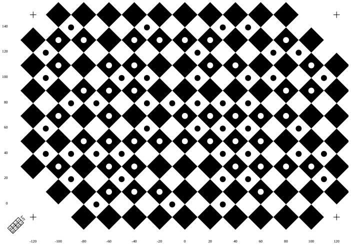


- X_i are considered exact
- m_i is a measurement subject to detection error

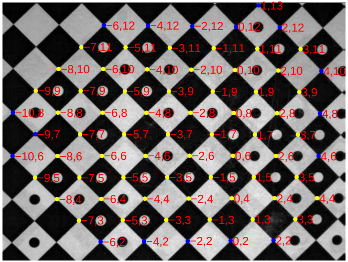
$$\mathbf{m}_i = \hat{\mathbf{m}}_i + \mathbf{e}_i \qquad \text{Cartesian}$$

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

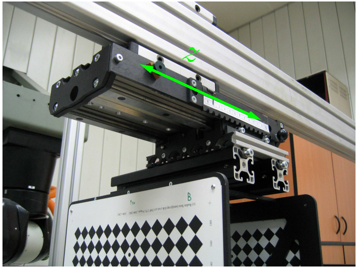
Resection Targets



calibration chart



automatic calibration point detection
based on a distributed bitcode ($2 \times 4 = 8$ bits)



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 bitcode

▶The Minimal Problem for Camera Resection

Problem: Given $k = 6$ corresponding pairs $\{(X_i, m_i)\}_{i=1}^k$, 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: $\lambda_i u_i = \mathbf{q}_1^\top \mathbf{X}_i + q_{14}, \quad \lambda_i v_i = \mathbf{q}_2^\top \mathbf{X}_i + q_{24}, \quad \lambda_i = \mathbf{q}_3^\top \mathbf{X}_i + q_{34}$

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

Then

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

- 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 $\operatorname{rank} \mathbf{A} = 12$ 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 $\mathbf{q}$

▸The Jack-Knife Solution for $k = 6$

- 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 $\dim \operatorname{null} \mathbf{A}_i > 1$ 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 2.e compute

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

regular for $n \geq 11$
variance of the sample mean

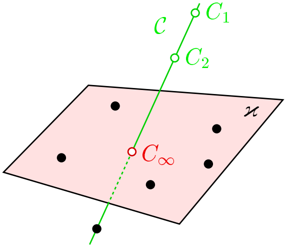
- have a solution + an error estimate, per individual elements of $\mathbf{P}$ (except P_{34})
- at least 5 points must be in a general position (→64)
- large error indicates near degeneracy
- computation not efficient with $k > 6$ points, needs $\binom{2k}{11}$ draws, e.g. $k = 7 \Rightarrow 364$ draws
- better error estimation method: decompose $\mathbf{P}_i$ to $\mathbf{K}_i$, $\mathbf{R}_i$, $\mathbf{t}_i$ (→33), represent $\mathbf{R}_i$ with 3 parameters (e.g. Euler angles, or in exponential map representation →147) 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 group-theoretic representations (the procedure is iterative)

▶Degenerate (Critical) Configurations for Camera Resection

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

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

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



Case 4

Results

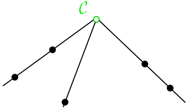
- importantly: If all calibration points $X_i \in \mathcal{X}$ lie on a plane $\varkappa$ then camera resection is non-unique and all image-equivalent camera centers lie on a spatial line $\mathcal{C}$ with the $C_\infty = \varkappa \cap \mathcal{C}$ excluded

  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

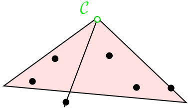
Proof sketch: If $\mathbf{Q}$, $\mathbf{T}$ are suitable homographies then $\mathbf{P}_1 \simeq \mathbf{Q}\mathbf{P}_0\mathbf{T}$, where $\mathbf{P}_0$ is canonical and the analysis can be made with $\hat{\mathbf{P}}_j \simeq \mathbf{Q}^{-1}\mathbf{P}_j$

$$\mathbf{P}_0 \underbrace{\mathbf{T}\underline{\mathbf{X}}_i}_{\underline{\mathbf{Y}}_i} \simeq \hat{\mathbf{P}}_j \underbrace{\mathbf{T}\underline{\mathbf{X}}_i}_{\underline{\mathbf{Y}}_i} \quad \text{for all} \quad Y_i \in \mathcal{Y}$$

see [H&Z, Sec. 22.1.2] for a full prof

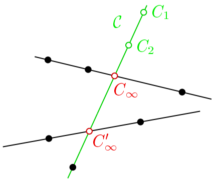


Case 5

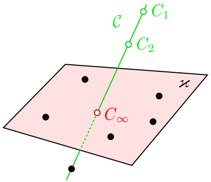


Case 6

- points lie on three optical rays or one optical ray and one optical plane
- cameras C_1, C_2 co-located at point $\mathcal{C}$
- Case 5: camera sees 3 isolated point images
- Case 6: cam. sees a line of points and an isolated point



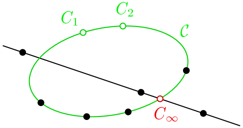
Case 3



Case 4

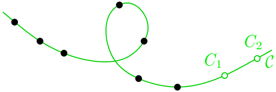
- points lie on a <u>line</u> $\mathcal{C}$ and
  1. on two lines meeting $\mathcal{C}$ at C_∞, C'_∞
  2. or on a plane meeting $\mathcal{C}$ at C_∞
- cameras lie on a line $\mathcal{C} \setminus \{C_\infty, C'_\infty\}$
- Case 3: camera sees 2 lines of points
- Case 4: **dangerous!**

Case 2



- points lie on a <u>planar conic</u> $\mathcal{C}$ and an additional line meeting $\mathcal{C}$ at C_∞
- cameras lie on $\mathcal{C} \setminus \{C_\infty\}$ — not necessarily an ellipse
- Case 2: camera sees 2 lines of points

Case 1



- points and cameras all lie on a <u>twisted cubic</u> $\mathcal{C}$
- Case 1: camera sees points on a conic

  dangerous but unlikely to occur

▶Three-Point Exterior Orientation Problem (P3P)

Calibrated camera rotation and translation from Perspective images of 3 reference Points.

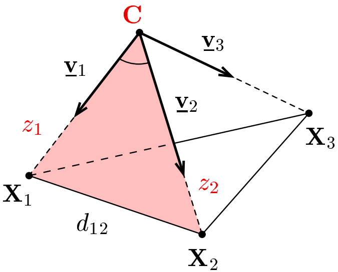
Problem: Given $\mathbf{K}$ and three corresponding pairs $\{(m_i, X_i)\}_{i=1}^{3}$, find $\mathbf{R}$, $\mathbf{C}$ by solving

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

1. Transform $\underline{\mathbf{v}}_i \stackrel{\text{def}}{=} \mathbf{K}^{-1}\underline{\mathbf{m}}_i$. 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 could solve (10) for λ_i

▶P3P cont'd

If there is rotation $\mathbf{R}$

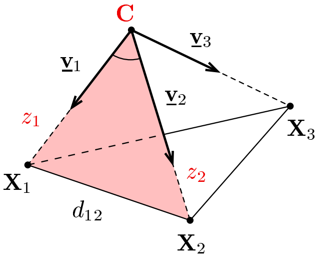
1. Eliminate $\mathbf{R}$ by taking   rotation preserves length: $\|\mathbf{R}\mathbf{x}\| = \|\mathbf{x}\|$

$$|\lambda_i| \cdot \|\underline{\mathbf{v}}_i\| = \|\mathbf{X}_i - \mathbf{C}\| \stackrel{\text{def}}{=} z_i \tag{11}$$

2. Consider only angles among $\underline{\mathbf{v}}_i$ and apply the 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 the system of 3 quadratic eqs in 3 unknowns z_i

[Fischler & Bolles, 1981]

there may be no real root

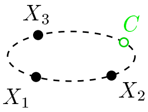
there are up to 4 solutions that cannot be ignored   (verify on additional points)

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

6. Compute $\mathbf{R}$ from (10)   we will solve this problem next →70

Similar problems (P4P with unknown f) at http://aag.ciirc.cvut.cz/minimal/ (papers, code)

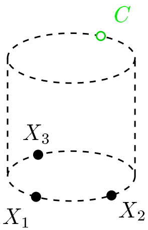
Degenerate (Critical) Configurations for Exterior Orientation



no solution

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

---



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

---

- additional critical configurations depend on the quadratic equations solver [Haralick et al. IJCV 1994]

▶Populating A Little ZOO of Minimal Geometric Problems in CV

problem	given	unknown	slide
camera resection	6 world–image correspondences $\left\{(X_i,\, m_i)\right\}_{i=1}^{6}$	$\mathbf{P}$	→62
exterior orientation	$\mathbf{K}$, 3 world–image correspondences $\left\{(X_i,\, m_i)\right\}_{i=1}^{3}$	$\mathbf{R}$, $\mathbf{C}$	→66
next: relative orientation	3 world-world correspondences $\left\{(X_i,\, Y_i)\right\}_{i=1}^{3}$	$\mathbf{R}$, $\mathbf{t}$	→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) — it is a recurring problem in 3D vision
- 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
- merging partial reconstructions from different viewpoints
- generalization of the last step of P3P

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

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

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

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

Poor man's solver:

- normalize $\mathbf{W}_i$, $\mathbf{Z}_i$ to unit length, use the above formula, and then find the closest rotation matrix
- but this is equivalent to a non-optimal objective — it ignores errors in vector lengths

We setup a minimization problem

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

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

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

$$\mathbf{A} : \mathbf{B} = \mathbf{B} : \mathbf{A} = \operatorname{tr}(\mathbf{A}^\top\mathbf{B}) = \operatorname{vec}(\mathbf{A})^\top \operatorname{vec}(\mathbf{B}) = \mathbf{a} \cdot \mathbf{b}$$

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)

$$\mathbf{Z}_i^\top\mathbf{R}\mathbf{W}_i = \operatorname{tr}(\mathbf{Z}_i^\top\mathbf{R}\mathbf{W}_i) \overset{\text{O2}}{=} \operatorname{tr}(\mathbf{W}_i\mathbf{Z}_i^\top\mathbf{R}) \overset{\text{O1}}{=} (\mathbf{Z}_i\mathbf{W}_i^\top) : \mathbf{R} = \mathbf{R} : (\mathbf{Z}_i\mathbf{W}_i^\top)$$

- Then we can factor the $\mathbf{R}$ out of the sum

$$\sum_i \mathbf{Z}_i^\top\mathbf{R}\mathbf{W}_i = \mathbf{R} : \left(\sum_i \mathbf{Z}_i\mathbf{W}_i^\top\right) \overset{\text{def}}{=} \mathbf{R} : \mathbf{M}$$

- Consider the SVD of $\mathbf{M}$: $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$. Then

$$\mathbf{R} : \mathbf{M} = \mathbf{R} : (\mathbf{U}\mathbf{D}\mathbf{V}^\top) \overset{\text{O1}}{=} \operatorname{tr}(\mathbf{R}^\top\mathbf{U}\mathbf{D}\mathbf{V}^\top) \overset{\text{O2}}{=} \operatorname{tr}(\mathbf{V}^\top\mathbf{R}^\top\mathbf{U}\mathbf{D}) \overset{\text{O1}}{=} (\mathbf{U}^\top\mathbf{R}\mathbf{V}) : \mathbf{D}$$

cont'd: The Algorithm

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 closest to: (2) diagonal and (3) positive definite $\mathbf{D}$
- 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

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

- $\mathbf{U}^\top \mathbf{V}$ is not necessarily positive definite
- We choose $\mathbf{S}$ so that $\det(\mathbf{R}^*) = 1$

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 $\det \mathbf{R}_k = 1$,
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)
- Can be used for the last step of the exterior orientation (P3P) problem →66

Module IV

Computing with a Camera Pair

4.1 Camera Motions Inducing Epipolar Geometry, Fundamental and Essential Matrices

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

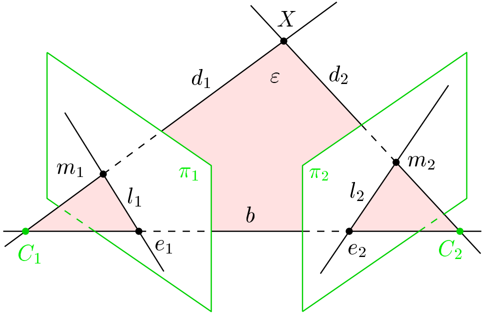
H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

►Geometric Model of a Camera Stereo Pair

$$\mathbf{P}_i = \begin{bmatrix}\mathbf{Q}_i & \mathbf{q}_i\end{bmatrix} = \mathbf{K}_i \begin{bmatrix}\mathbf{R}_i & \mathbf{t}_i\end{bmatrix} = \mathbf{K}_i\mathbf{R}_i \begin{bmatrix}\mathbf{I} & -\mathbf{C}_i\end{bmatrix} \quad i = 1, 2 \qquad \rightarrow 31$$

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

- baseline b joins projection centers C_1, C_2
$$\mathbf{b} = \mathbf{C}_2 - \mathbf{C}_1$$
- epipole $e_i \in \pi_i$ 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 optical ray d_j, $j \neq i$ and also the epipolar plane
$$\varepsilon = (C_2, X, C_1)$$
- l_j is the epipolar line ('epipolar') in image π_j induced by m_i in image π_i

Epipolar constraint relates $\underline{\mathbf{m}}_1$ and $\underline{\mathbf{m}}_2$: corresponding d_2, b, d_1 are coplanar

a necessary condition →88

Epipolar Geometry Example: Forward Motion

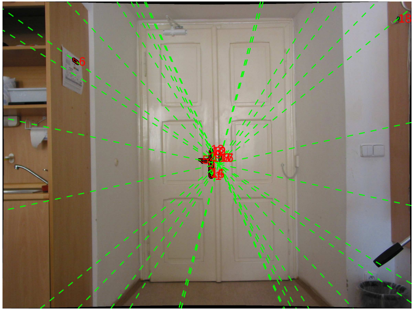


image 1

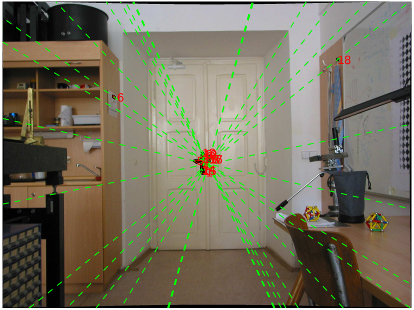
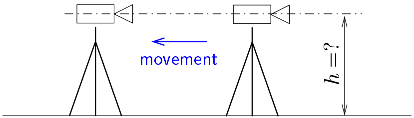


image 2

- red: correspondences
- green: epipolar line pairs per correspondence

click on the image to see their IDs
same ID in both images

Epipole is the image of the other camera's center.
How high was the camera above the floor?



▶Cross Products and Maps by Skew-Symmetric 3×3 Matrices

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

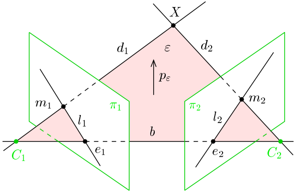
$$[\mathbf{b}]_\times = \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix}, \qquad \text{assuming} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Some properties

1. $[\mathbf{b}]_\times^\top = -[\mathbf{b}]_\times$ — the general antisymmetry property
2. $\mathbf{A}$ is skew-symmetric iff $\mathbf{x}^\top \mathbf{A} \mathbf{x} = 0$ for all $\mathbf{x}$ — skew-sym mtx generalizes cross products
3. $[\mathbf{b}]_\times^3 = -\|\mathbf{b}\|^2 \cdot [\mathbf{b}]_\times$
4. $\|[\mathbf{b}]_\times\|_F = \sqrt{2}\,\|\mathbf{b}\|$ — Frobenius norm ($\|\mathbf{A}\|_F = \sqrt{\operatorname{tr}(\mathbf{A}^\top \mathbf{A})} = \sqrt{\sum_{i,j} |a_{ij}|^2}$)
5. $\operatorname{rank}[\mathbf{b}]_\times = 2$ iff $\|\mathbf{b}\| > 0$ — check minors of $[\mathbf{b}]_\times$
6. $[\mathbf{b}]_\times \mathbf{b} = \mathbf{0}$
7. eigenvalues of $[\mathbf{b}]_\times$ are $(0, \lambda, -\lambda)$
8. for any 3×3 regular $\mathbf{B}$: $\mathbf{B}^\top [\mathbf{B}\mathbf{z}]_\times \mathbf{B} = \det \mathbf{B}\, [\mathbf{z}]_\times$ — follows from the factoring on →39
9. in particular: if $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$ then $[\mathbf{R}\mathbf{z}]_\times = \mathbf{R}[\mathbf{z}]_\times \mathbf{R}^\top$

- 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 — it is the logarithm of a rotation mtx

▶Expressing the Epipolar Constraint Algebraically: Fundamental Matrix



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

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

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

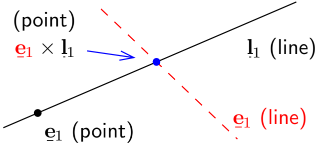
$$\mathbf{F} = (\underbrace{\mathbf{Q}_2\mathbf{Q}_1^{-1}}_{\text{epipolar homography } \mathbf{H}_e})^{-\top}[\underline{\mathbf{e}}_1]_\times = \mathbf{H}_e^{-\top}\overbrace{[\underline{\mathbf{e}}_1]_\times}^{\text{left epipole}} \overset{\rightarrow 76}{\simeq} \overbrace{[\mathbf{H}_e \underline{\mathbf{e}}_1]_\times}^{\text{right epipole}} \mathbf{H}_e$$

- point $\underline{\mathbf{m}}_2$ is incident on epipolar line $\underline{\mathbf{l}}_2 \simeq \mathbf{F}\underline{\mathbf{m}}_1$
- point $\underline{\mathbf{m}}_1$ is incident on epipolar line $\underline{\mathbf{l}}_1 \simeq \mathbf{F}^\top\underline{\mathbf{m}}_2$
- all epipolars meet at the epipole
- epipolar homography maps epipolar lines by $\mathbf{H}_e^{-\top}$
- epipolar homography maps epipoles by $\mathbf{H}_e$

$$\mathbf{F} = (\underbrace{\mathbf{Q}_2\mathbf{Q}_1^{-1}}_{\text{epipolar homography } \mathbf{H}_e})^{-\top}[\underline{\mathbf{e}}_1]_\times = \mathbf{H}_e^{-\top}[\overbrace{\underline{\mathbf{e}}_1}^{\text{left epipole}}]_\times \overset{\to 76}{\simeq} [\overbrace{\mathbf{H}_e\underline{\mathbf{e}}_1}^{\text{right epipole } \underline{\mathbf{e}}_2}]_\times \mathbf{H}_e$$

- epipole $\underline{\mathbf{e}}_1$ falls in the nullspace of $\mathbf{F}$: $\mathbf{F}\underline{\mathbf{e}}_1 = \mathbf{H}_e^{-\top}[\underline{\mathbf{e}}_1]_\times \underline{\mathbf{e}}_1 = \mathbf{0}$, also $\underline{\mathbf{e}}_2^\top \mathbf{F} = \mathbf{0}$
- $\mathbf{F}$ maps points to lines and it is not a homography

- $\mathbf{H}_e^{-\top}$ maps epipolars to epipolars: $\underline{\mathbf{l}}_2 \simeq \mathbf{H}_e^{-\top}\underline{\mathbf{l}}_1$
- there is another useful map that does the job for the epipolars: $\underline{\mathbf{l}}_2 \simeq \mathbf{F}[\underline{\mathbf{e}}_1]_\times \underline{\mathbf{l}}_1 = \mathbf{F}(\underline{\mathbf{e}}_1 \times \underline{\mathbf{l}}_1)$



proof by point/line 'transmutation' (left):

- point $\underline{\mathbf{e}}_1$ does not lie on line $\underline{\mathbf{e}}_1$ (dashed): $\underline{\mathbf{e}}_1^\top \underline{\mathbf{e}}_1 \neq 0$
- $\underline{\mathbf{e}}_1 \times \underline{\mathbf{l}}_1$ is a point on $\underline{\mathbf{l}}_1$
- $\mathbf{F}$ maps that point to $\underline{\mathbf{l}}_2$

- the composition $\mathbf{F}[\underline{\mathbf{e}}_1]_\times$ is not a homography
- usefulness: no need to decompose $\mathbf{F}$ to obtain $\mathbf{H}_e$

▶The Essential Matrix

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

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

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

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

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

- the epipole is the image of the (projection center) of the other camera

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

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

$$\mathbf{E} = [-\mathbf{t}_{21}]_\times \mathbf{R}_{21} = \underbrace{[\mathbf{R}_2 \mathbf{b}]_\times}_{\text{baseline in Cam 2}} \mathbf{R}_{21} \overset{\to 76/9}{=} \mathbf{R}_{21} \underbrace{[\mathbf{R}_1 \mathbf{b}]_\times}_{\text{baseline in Cam 1}} = \mathbf{R}_{21} [-\mathbf{R}_{21}^\top \mathbf{t}_{21}]_\times \quad \text{essential} \tag{12}$$

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

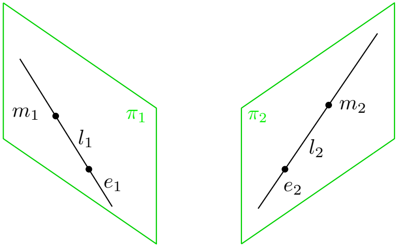
(the change of the world coordinate system by $(\mathbf{R}, \mathbf{t})$ does not change $\mathbf{E}$)

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

then

$$\mathbf{R}_{21}' = \mathbf{R}_2' \mathbf{R}_1'^\top = \cdots = \mathbf{R}_{21} \qquad \mathbf{t}_{21}' = \mathbf{t}_2' - \mathbf{R}_{21}' \mathbf{t}_1' = \cdots = \mathbf{t}_{21}$$

- the translation length $\|\mathbf{t}_{21}\|$ is lost, since $\mathbf{E}$ is homogeneous



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

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

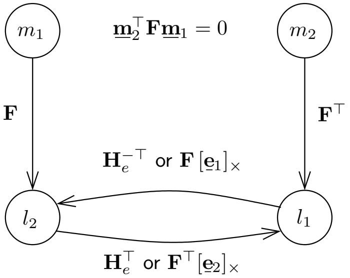
$$\underline{\mathbf{e}}_1 \simeq \mathbf{H}_e^{-1}\underline{\mathbf{e}}_2 \qquad \underline{\mathbf{e}}_2 \simeq \mathbf{H}_e\underline{\mathbf{e}}_1$$

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

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

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

epipolar constraint and the maps



- $\mathbf{H}_e = \mathbf{Q}_2\mathbf{Q}_1^{-1}$ is the epipolar homography→79
  $\mathbf{H}_e^{-\top}$ 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

- $\mathbf{F}[\underline{\mathbf{e}}_1]_\times$ maps epipolar lines to epipolar lines but it is not a homography
- The essential matrix is the 'calibrated fundamental matrix'

▶Representation Theorem for Fundamental Matrices

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

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

Proof.

Direct: $\mathbf{H}$ is full-rank, $\underline{\mathbf{e}}_1 \neq \mathbf{0}$, $\operatorname{rank}[\underline{\mathbf{e}}_1]_\times \overset{\to 76/5}{\simeq} 2 \quad \Rightarrow \quad \mathbf{H}^{-\top}[\underline{\mathbf{e}}_1]_\times$ is a 3×3 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 $\mathbf{B} = \operatorname{diag}(\lambda_1, \lambda_2, \lambda_3)$, $\mathbf{C} = \operatorname{diag}(1, 1, 0)$, $\lambda_3 > 0$
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 matrix
4. we look for a rotation mtx $\mathbf{W}$ that maps $\mathbf{C}$ to a skew-symmetric $\mathbf{S}$, i.e. $\mathbf{S} = \mathbf{C}\mathbf{W}$ (if it exists)
5. then $\mathbf{W} = \begin{bmatrix} 0 & \alpha & 0 \\ -\alpha & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $|\alpha| = 1$, and $\mathbf{S} = \mathbf{C}\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \alpha & 0 \\ -\alpha & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \cdots = [\mathbf{s}]_\times$, where $\mathbf{s} = (0, 0, 1)$
6. we write $\qquad$ $\mathbf{v}_3$ – 3rd column of $\mathbf{V}$, $\mathbf{u}_3$ – 3rd column of $\mathbf{U}$

$$\mathbf{A} = \mathbf{U}\mathbf{B}\underbrace{[\mathbf{s}]_\times}_{\mathbf{C}\mathbf{W}}\mathbf{W}^\top\mathbf{V}^\top = \overset{\circledast 1}{\cdots} = \underbrace{\mathbf{U}\mathbf{B}(\mathbf{V}\mathbf{W})^\top}_{\simeq \mathbf{H}^{-\top}}\underbrace{[\mathbf{v}_3]_\times}_{\text{3rd col } \mathbf{V}} \overset{\to 76/9}{\simeq} \underbrace{[\mathbf{H}\mathbf{v}_3]_\times}_{\simeq [\mathbf{u}_3]_\times}\mathbf{H}, \tag{13}$$

7. $\mathbf{H}$ regular, $\mathbf{A}\mathbf{v}_3 = \mathbf{0}$, $\mathbf{u}_3\mathbf{A} = \mathbf{0}$ for $\mathbf{v}_3 \neq \mathbf{0}$, $\mathbf{u}_3 \neq \mathbf{0}$, $\qquad$ $\mathbf{H} = \mathbf{U}\mathbf{B}^{-1}(\mathbf{V}\mathbf{W})^\top$ □

- we also got a (non-unique: α, λ_3) decomposition formula for fundamental matrices
- it follows there is no constraint on $\mathbf{F}$ except for the rank

▶Representation Theorem for Essential Matrices

Theorem

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

- we know that $\mathbf{E} \overset{(12)}{=} \mathbf{R}_{21}[\mathbf{z}]_\times \overset{\to 78}{=} \mathbf{H}_e^{-\top}[\mathbf{z}]_\times$ for some $\mathbf{z}$

Proof.

Direct:

If $\mathbf{E}$ is an essential matrix, then the epipolar homography matrix $\mathbf{H}_e$ is a rotation matrix (→79), hence $\mathbf{H}_e^{-\top} \simeq \mathbf{U}\mathbf{B}(\mathbf{V}\mathbf{W})^\top$ in (13) must be (1) regular, and (2) (λ-scaled) orthonormal.

$\mathbf{B}$ is diagonal by definition, it follows $\mathbf{B} = \lambda\mathbf{I}$. note this fixed the missing λ_3 in (13)

Then

$$\mathbf{R}_{21} = \mathbf{H}_e^{-\top} \simeq \mathbf{U}\mathbf{W}^\top\mathbf{V}^\top \simeq \mathbf{U}\mathbf{W}\mathbf{V}^\top \qquad \text{note that } \mathbf{R}_{21}^{-\top} = \mathbf{R}_{21} \tag{14}$$

Converse:

$\mathbf{E}$ is fundamental with

$$\mathbf{D} = \mathrm{diag}(\lambda, \lambda, 0) = \underbrace{\lambda\mathbf{I}}_{\mathbf{B}}\ \underbrace{\mathrm{diag}(1, 1, 0)}_{\mathbf{D}}$$

then $\mathbf{B} = \lambda\mathbf{I}$ in (13) and $\mathbf{U}(\mathbf{V}\mathbf{W})^\top$ is orthogonal, as required. □

▶Essential Matrix Decomposition

We are decomposing $\mathbf{E}$ to $\mathbf{E} \overset{(13)}{\simeq} [\mathbf{u}_3]_\times \mathbf{H} \overset{(12)}{\simeq} [-\mathbf{t}_{21}]_\times \mathbf{R}_{21} \overset{(12)}{=} \mathbf{R}_{21}[-\mathbf{R}_{21}^\top \mathbf{t}_{21}]_\times \overset{(13)}{\simeq} \mathbf{H}^{-\top}[\mathbf{v}_3]_\times$ [H&Z, sec. 9.6]

1. compute SVD of $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ and verify $\mathbf{D} = \lambda\,\mathrm{diag}(1,1,0)$
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

$$\mathbf{R}_{21} \overset{(14)}{=} \mathbf{U} \underbrace{\begin{bmatrix} 0 & \alpha & 0 \\ -\alpha & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{W}} \mathbf{V}^\top, \quad \mathbf{t}_{21} \overset{(13)}{=} -\beta\,\mathbf{u}_3, \qquad |\alpha| = 1, \quad \beta \neq 0 \tag{15}$$

Notes

- $\mathbf{v}_3 \simeq \mathbf{R}_{21}^\top \mathbf{t}_{21}$ by (13), 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 β and direction $\mathrm{sign}\,\beta$
- the result for $\mathbf{R}_{21}$ is unique up to $\alpha = \pm 1$ despite non-uniqueness of SVD
- the change of sign in α rotates the solution by 180° about $\mathbf{t}_{21}$

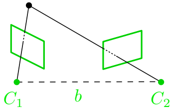
  $\mathbf{R}(\alpha) = \mathbf{U}\mathbf{W}\mathbf{V}^\top \;\Rightarrow\; \mathbf{R}(-\alpha) = \mathbf{U}\mathbf{W}^\top\mathbf{V}^\top \Rightarrow \mathbf{T} = \mathbf{R}(-\alpha)\mathbf{R}^\top(\alpha) = \cdots = \mathbf{U}\,\mathrm{diag}(-1,-1,1)\mathbf{U}^\top$
  which is a rotation by 180° about $\mathbf{u}_3 \simeq \mathbf{t}_{21}$: show that $\mathbf{u}_3$ is the rotation axis

$$\mathbf{U}\,\mathrm{diag}(-1,-1,1)\underbrace{\mathbf{U}^\top\mathbf{u}_3}_{\text{orthonormality(!)}} = \mathbf{U}\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{u}_3$$

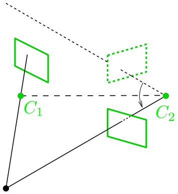
- 4 solution sets for 4 sign combinations of α, β see next for geometric interpretation

▶Four Solutions to Essential Matrix Decomposition

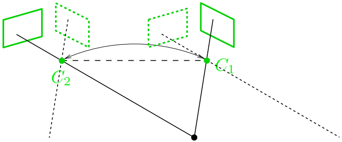
Transform the world coordinate system so that the origin is in Camera 2. Then $\mathbf{t}_{21} = -\mathbf{b}$ and $\mathbf{W}$ rotates about the baseline $\mathbf{b}$. →77



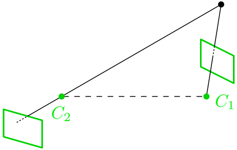
we chose α, β s.t. front – front



then $-\alpha$, β (twisted by $\mathbf{W}$) back – front



α, $-\beta$ (baseline reversal) back – back



$-\alpha$, $-\beta$ (combination of both) back – front

How to disambiguate?

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

►7-Point Algorithm for Estimating Fundamental Matrix

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

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

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

Solution:

$$\underline{\mathbf{y}}_i^\top \mathbf{F}\, \underline{\mathbf{x}}_i \overset{\to 71}{=} (\underline{\mathbf{y}}_i \underline{\mathbf{x}}_i^\top) : \mathbf{F} = \left(\operatorname{vec}(\underline{\mathbf{y}}_i \underline{\mathbf{x}}_i^\top)\right)^\top \operatorname{vec}(\mathbf{F}),$$

rotation property of matrix trace →71

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

column vector from matrix

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

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

▶7-Point Algorithm Continued

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

- for $k = 8$ we have $\mathrm{rank}(\mathbf{D}) = 8$, then there is a non-trivial solution for $\mathbf{F}$ but it is not necessarily a f. m.
- for $k = 7$ we have $\mathrm{rank}(\mathbf{D}) = 7$, the null-space of $\mathbf{D}$ is 2-dimensional
- but we know that $\det(\mathbf{F}) = 0$, hence
  1. find a basis of the null space of $\mathbf{D}$: $\mathbf{F}_1$, $\mathbf{F}_2$ — by SVD or QR factorization
  2. get up to 3 real solutions for α from

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

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

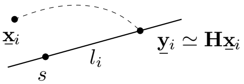
- the result may depend on image (domain) transformations
- normalization of $\mathbf{D}$ improves conditioning →93
- this gives a good starting point for the full algorithm →113
- dealing with mismatches need not be a part of the 7-point algorithm →118

▸Degenerate Configurations for Fundamental Matrix Estimation

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

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

- in either case: epipolar geometry is not uniquely defined
- we get an arbitrary solution from the 7-point algorithm, in the form of $\mathbf{F} = [\underline{\mathbf{s}}]_\times \mathbf{H}$ — note that $[\underline{\mathbf{s}}]_\times \mathbf{H} \simeq \mathbf{H}'[\underline{\mathbf{s}}']_\times$ →76



If $\mathbf{H}$ is a homography, then any correspondence satisfies $\underline{\mathbf{y}}_i^\top [\underline{\mathbf{s}}]_\times \mathbf{H}\underline{\mathbf{x}}_i = 0$ for any $\mathbf{s}$

- given (arbitrary, fixed) point $\underline{\mathbf{s}}$
- 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: $y_i \in l_i$, $\quad \underline{\mathbf{l}}_i \simeq \underline{\mathbf{s}} \times \mathbf{H}\underline{\mathbf{x}}_i$

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

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

notes

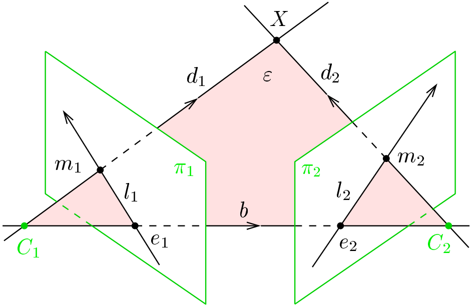
- estimation of $\mathbf{E}$ can deal with planes: $[\underline{\mathbf{s}}]_\times \mathbf{H}$ is essential, then $\mathbf{H} = \mathbf{R} - \mathbf{t}\mathbf{n}^\top/d$, and $\underline{\mathbf{s}} \simeq \mathbf{t}$ not arbitrary

$$\mathbf{E} = [\underline{\mathbf{s}}]_\times \mathbf{R} = [\underline{\mathbf{s}}]_\times \mathbf{H} = [\underline{\mathbf{s}}]_\times (\mathbf{R} - \mathbf{t}\mathbf{n}^\top/d) \overset{!}{\simeq} [\mathbf{t}]_\times \mathbf{R}$$

- a complete treatment with additional degenerate configurations in [H&Z, sec. 22.2]
- a stronger epipolar constraint could reject some configurations (see next)

A Note on Oriented Epipolar Constraint

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



- oriented epipolars
- notation: $\underline{\mathbf{m}} \overset{+}{\sim} \underline{\mathbf{n}}$ means $\underline{\mathbf{m}} = \lambda \underline{\mathbf{n}}$, $\lambda > 0$
- then we define

$$(\underline{\mathbf{e}}_2 \times \underline{\mathbf{m}}_2) \overset{+}{\sim} \mathbf{H}_e^{-\top} (\underline{\mathbf{e}}_1 \times \underline{\mathbf{m}}_1) = \mathbf{F}\underline{\mathbf{m}}_1$$

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

- 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 →118 — [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. $\mathbf{E}$ – homogeneous 3×3 matrix; 9 numbers up to scale
2. $\mathbf{R}$ – 3 DOF, $\mathbf{t}$ – 2 DOF only, in total 5 DOF → we need $9 - 1 - 5 = 3$ constraints on $\mathbf{E}$
3. idea: $\mathbf{E}$ essential iff it has two equal singular values and the third is zero →82

This gives an equation system:

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

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

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

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

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

- when all 3D points lie on a plane: at most 2 real solutions (twisted-pair) — can be disambiguated in 3 views or by chirality constraint (→84) unless all 3D points are closer to one camera
- 6-point problem for unknown f — [Kukelova et al. BMVC 2008]
- resources at http://aag.ciirc.cvut.cz/minimal/

▶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 after eliminating λ_1, λ_2

$$u^1\,(\mathbf{p}_3^1)^\top \underline{\mathbf{X}} = (\mathbf{p}_1^1)^\top \underline{\mathbf{X}}, \qquad u^2\,(\mathbf{p}_3^2)^\top \underline{\mathbf{X}} = (\mathbf{p}_1^2)^\top \underline{\mathbf{X}},$$
$$v^1\,(\mathbf{p}_3^1)^\top \underline{\mathbf{X}} = (\mathbf{p}_2^1)^\top \underline{\mathbf{X}}, \qquad v^2\,(\mathbf{p}_3^2)^\top \underline{\mathbf{X}} = (\mathbf{p}_2^2)^\top \underline{\mathbf{X}}$$

Gives

$$\mathbf{D}\underline{\mathbf{X}} = \mathbf{0}, \qquad \mathbf{D} = \begin{bmatrix} u^1\,(\mathbf{p}_3^1)^\top - (\mathbf{p}_1^1)^\top \\ v^1\,(\mathbf{p}_3^1)^\top - (\mathbf{p}_2^1)^\top \\ u^2\,(\mathbf{p}_3^2)^\top - (\mathbf{p}_1^2)^\top \\ v^2\,(\mathbf{p}_3^2)^\top - (\mathbf{p}_2^2)^\top \end{bmatrix}, \qquad \mathbf{D} \in \mathbb{R}^{4,4}, \quad \underline{\mathbf{X}} \in \mathbb{R}^4 \tag{16}$$

- typically, $\mathbf{D}$ has full rank (!)
- what else: back-projected rays will generally not intersect due to image error, see next
- what else: using Jack-knife (→63) not recommended — sensitive to small error
- idea: we will grind our teeth and use SVD (comes next: →91)
- 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 (16) 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

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

- let $\mathbf{d}_i$ be the i-th row of $\mathbf{D}$ reshaped as a column vector, then

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

- 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 $\min_{\mathbf{q}, \|\mathbf{q}\|=1} \mathbf{q}^\top \mathbf{Q}\, \mathbf{q} = \sigma_4^2$ and $\underline{\mathbf{X}} = \arg \min_{\mathbf{q}, \|\mathbf{q}\|=1} \mathbf{q}^\top \mathbf{Q}\, \mathbf{q} = \mathbf{u}_4$ — $\mathbf{u}_4$ – the last column of $\mathbf{U}$ from SVD($\mathbf{Q}$)

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

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

since $\sigma_j \geq \sigma_4$ □

▶cont'd

- 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)

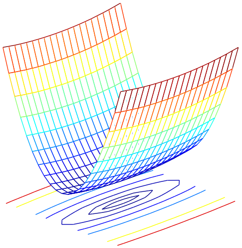
Matlab code for the least-squares solver:

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

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

▶Numerical Conditioning

- The equation $\mathbf{D}\underline{\mathbf{X}} = \mathbf{0}$ in (16) may be ill-conditioned for numerical computation, which results in a poor estimate for $\underline{\mathbf{X}}$.



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

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

Quick fix:

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

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

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

$$\mathbf{S} = \mathrm{diag}(10^{-3}, 10^{-3}, 10^{-3}, 10^{-6})$$

```
S = diag(1./max(abs(D), [], 1))
```

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

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

problem	given	unknown	slide
camera resection	6 world–img correspondences $\{(X_i, m_i)\}_{i=1}^{6}$	$\mathbf{P}$	62
exterior orientation	$\mathbf{K}$, 3 world–img correspondences $\{(X_i, m_i)\}_{i=1}^{3}$	$\mathbf{R}$, $\mathbf{t}$	66
relative pointcloud orientation	3 world-world correspondences $\{(X_i, Y_i)\}_{i=1}^{3}$	$\mathbf{R}$, $\mathbf{t}$	70
fundamental matrix	7 img–img correspondences $\{(m_i, m'_i)\}_{i=1}^{7}$	$\mathbf{F}$	85
relative camera orientation	$\mathbf{K}$, 5 img–img correspondences $\{(m_i, m'_i)\}_{i=1}^{5}$	$\mathbf{R}$, $\mathbf{t}$	89
triangulation	$\mathbf{P}_1$, $\mathbf{P}_2$, 1 img–img correspondence (m, m')	$\underline{\mathbf{X}}$	90

A bigger ZOO at http://aag.ciirc.cvut.cz/minimal/

calibrated problems

- have fewer degenerate configurations
- can do with fewer points (good for geometry proposal generators →125)

- 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

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.

▶Algebraic Error vs Reprojection Error

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

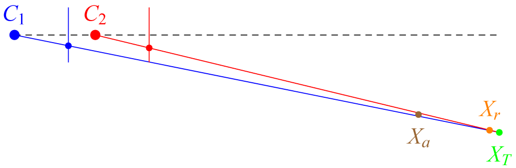
$$\varepsilon^2(\underline{\mathbf{X}}) = \|\mathbf{D}\underline{\mathbf{X}}\|^2 = \sum_{c=1}^{2} \left[\left(u^c (\mathbf{p}_3^c)^\top \underline{\mathbf{X}} - (\mathbf{p}_1^c)^\top \underline{\mathbf{X}} \right)^2 + \left(v^c (\mathbf{p}_3^c)^\top \underline{\mathbf{X}} - (\mathbf{p}_2^c)^\top \underline{\mathbf{X}}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

- reprojection error

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

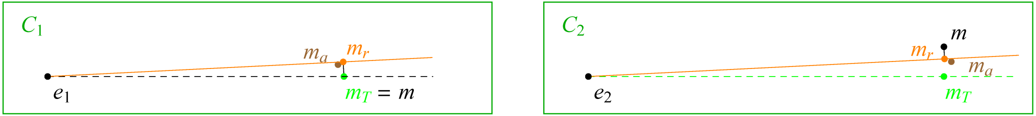
- algebraic error zero ⇔ reprojection error zero $\qquad \sigma_4 = 0 \Rightarrow$ non-trivial null space
- epipolar constraint satisfied ⇒ equivalent results
- in general: minimizing algebraic error is cheap but it gives inferior results
- minimizing reprojection error is expensive but it gives good results
- the midpoint of the common perpendicular to both optical rays gives about 50% greater error in 3D
- the golden standard method – deferred to →108

Algebraic Error vs Reprojection Error: Example



- 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)

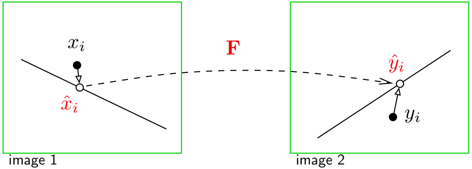


- this demonstrates a difficult configuration (forward camera motion) and a random correspondence
- noise-free ground-truth triangulation from m_T is X_T
- reprojection error minimizer X_r has an error due to simulated noise in image detections (black m)
- algebraic error minimizer X_a essentially failed

►The Concept of Error for Epipolar Geometry

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 $x_i \leftrightarrow y_j$.

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



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

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

$$\|\mathbf{e}_i(\cdot)\|^2 \stackrel{\text{def}}{=} \mathbf{e}_i^2(\cdot) = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 + \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2 = \|\mathbf{Z}_i - \hat{\mathbf{Z}}_i(\mathbf{F})\|^2 \quad \in \mathbb{R}^4 \tag{17}$$

▶cont'd

Consider the estimation of $\mathbf{F}$

- the total reprojection error (of all data) 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

$$(\hat{Z}^*, \mathbf{F}^*) = \arg\min_{\mathbf{F}, \hat{Z}} L(Z \mid \hat{Z}, \mathbf{F}) \quad \text{s.t.} \quad \operatorname{rank} \mathbf{F} = 2, \quad \underline{\hat{\mathbf{y}}}_i^\top \mathbf{F} \, \underline{\hat{\mathbf{x}}}_i = 0, \quad (\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i) \in \hat{\mathbf{Z}}_i \tag{18}$$

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 $\hat{Z}$, $\mathbf{F}$ needs a good parameterization for $\mathbf{F}$ →100
  2. Sampson optimal correction = partial correction of $\mathbf{Z}_i$ towards $\hat{\mathbf{Z}}_i$ used in an iterative minimization over $\mathbf{F}$ →102

Method 1: Reprojection Error Optimization: Idea

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

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad \mathbf{P}_2(\mathbf{F}) = \begin{bmatrix} [\underline{\mathbf{e}}_2]_\times \mathbf{F} + \underline{\mathbf{e}}_2 \underline{\mathbf{e}}_1^\top & \underline{\mathbf{e}}_2 \end{bmatrix}, \quad \text{s.t.} \quad \mathbf{F}\mathbf{e}_1 = \mathbf{0}, \; \mathbf{e}_2^\top \mathbf{F} = \mathbf{0} \tag{19}$$

⊛ H3; 2pt: Given rank-2 matrix $\mathbf{F}$, let $\underline{\mathbf{e}}_1$, $\underline{\mathbf{e}}_2$ be the right and left nullspace basis vectors of $\mathbf{F}$, respectively. Verify that such $\mathbf{F}$ is a fundamental matrix of $\mathbf{P}_1$, $\mathbf{P}_2$ from (19).

Hints:

(1) consider $\hat{\underline{\mathbf{x}}}_i = \mathbf{P}_1 \underline{\mathbf{X}}_i$ and $\hat{\underline{\mathbf{y}}}_i = \mathbf{P}_2 \underline{\mathbf{X}}_i$

(2) $\mathbf{A}$ is skew symmetric iff $\mathbf{x}^\top \mathbf{A} \mathbf{x} = 0$ for all vectors $\mathbf{x}$.

(cont'd) Reprojection Error Optimization: Algorithm

1. compute $\mathbf{F}^{(0)}$ by the 7-point algorithm →85; construct camera $\mathbf{P}_2^{(0)}$ from $\mathbf{F}^{(0)}$ using (19)
2. triangulate 3D points $\hat{\mathbf{X}}_i^{(0)}$ from matches (x_i, y_i) for all $i = 1, \ldots, k$ by the SVD alg. →90
3. starting from $\mathbf{P}_2^{(0)}$, $\hat{\mathbf{X}}_{1:k}^{(0)}$ minimize the reprojection error (17)

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

where

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

- non-linear, non-convex problem
- solves $\mathbf{F}$ estimation and triangulation of all k points jointly
- the solver would be quite slow
- $3k + 7$ parameters to be found: latent: $\hat{\mathbf{X}}_i$, for all i (correspondences!), non-latent: $\mathbf{F}$
- we need minimal representations for $\hat{\mathbf{X}}_i$ and $\mathbf{F}$ →154 or introduce constraints
- there are other pitfalls; this is essentially bundle adjustment; we will return to this later →142

▶Method 2: First-Order Error Approximation

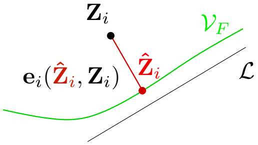
An elegant method for solving problems like (18):

- 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 $\boldsymbol{\varepsilon} = \underline{\mathbf{y}}^\top \mathbf{F}\, \underline{\mathbf{x}}$ from →85

- 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 $\underline{\hat{\mathbf{y}}}_i^\top \mathbf{F}\, \underline{\hat{\mathbf{x}}}_i = 0$, $\quad \underline{\hat{\mathbf{x}}}_i = (\hat{u}^1, \hat{v}^1, 1)$, $\underline{\hat{\mathbf{y}}}_i = (\hat{u}^2, \hat{v}^2, 1)$
- this is a manifold $\mathcal{V}_F \in \mathbb{R}^4$: a set of points $\hat{\mathbf{Z}} = (\hat{u}^1, \hat{v}^1, \hat{u}^2, \hat{v}^2) \in \mathbb{R}^4$ consistent with $\mathbf{F}$
- algebraic error vanishes for $\hat{\mathbf{Z}}_i$: $\quad \mathbf{0} = \boldsymbol{\varepsilon}_i(\hat{\mathbf{Z}}_i) = \underline{\hat{\mathbf{y}}}_i^\top \mathbf{F}\, \underline{\hat{\mathbf{x}}}_i$ $\quad$ $\boldsymbol{\varepsilon}(\mathbf{Z})$ is a function of $\mathbf{Z}$



Sampson's idea: Linearize the algebraic error $\boldsymbol{\varepsilon}(\mathbf{Z})$ 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 $\mathcal{L}$.

$$\mathcal{L}: \quad 0 = \boldsymbol{\varepsilon}_i(\hat{\mathbf{Z}}_i) \approx \boldsymbol{\varepsilon}_i(\mathbf{Z}_i) + \frac{\partial \boldsymbol{\varepsilon}_i(\mathbf{Z}_i)}{\partial \mathbf{Z}_i}(\hat{\mathbf{Z}}_i - \mathbf{Z}_i) \qquad \text{linear in } \hat{\mathbf{Z}}_i$$

▶Sampson's Approximation of Reprojection Error

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

$$\boldsymbol{\varepsilon}_i(\mathbf{Z}_i) + \underbrace{\frac{\partial \boldsymbol{\varepsilon}_i(\mathbf{Z}_i)}{\partial \mathbf{Z}_i}}_{\mathbf{J}_i(\mathbf{Z}_i)} \underbrace{(\hat{\mathbf{Z}}_i - \mathbf{Z}_i)}_{\mathbf{e}_i(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)} \stackrel{\text{def}}{=} \underbrace{\boldsymbol{\varepsilon}_i(\mathbf{Z}_i)}_{\text{given}} + \mathbf{J}_i(\mathbf{Z}_i)\, \underbrace{\mathbf{e}_i(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)}_{\text{wanted}} = \boldsymbol{\varepsilon}_i(\hat{\mathbf{Z}}_i) \stackrel{!}{=} 0$$

- goal: compute function $\mathbf{e}_i(\cdot)$ from $\boldsymbol{\varepsilon}_i(\cdot)$, where $\mathbf{e}_i(\cdot)$ is the distance of $\hat{\mathbf{Z}}_i$ from $\mathbf{Z}_i$
- we have a linear underconstrained equation for $\mathbf{e}_i(\cdot)$ e.g. $\boldsymbol{\varepsilon}_i \in \mathbb{R}$, $\mathbf{e}_i \in \mathbb{R}^4$
- we look for a minimal $\mathbf{e}_i(\cdot)$ per match i

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

- which has a closed-form solution note that $\mathbf{J}_i(\cdot)$ is not invertible! ⊛ P1; 1pt: derive $\mathbf{e}_i^*(\cdot)$

$$\mathbf{e}_i^*(\cdot) = -\mathbf{J}_i^\top(\mathbf{J}_i\mathbf{J}_i^\top)^{-1}\boldsymbol{\varepsilon}_i(\cdot) \qquad \text{pseudo-inverse}$$
$$\|\mathbf{e}_i^*(\cdot)\|^2 = \boldsymbol{\varepsilon}_i^\top(\cdot)(\mathbf{J}_i\mathbf{J}_i^\top)^{-1}\boldsymbol{\varepsilon}_i(\cdot) \tag{20}$$

- this maps $\boldsymbol{\varepsilon}_i(\cdot)$ to an estimate of $\mathbf{e}_i(\cdot)$ per correspondence
- we need $\|\mathbf{e}_i\|^2$ for the $\mathbf{F}$ estimation, we will need $\mathbf{e}_i$ for triangulation in the golden-standard alg. →108
- the unknown parameters $\mathbf{F}$ are inside: $\mathbf{e}_i = \mathbf{e}_i(\mathbf{F})$, $\boldsymbol{\varepsilon}_i = \boldsymbol{\varepsilon}_i(\mathbf{F})$, $\mathbf{J}_i = \mathbf{J}_i(\mathbf{F})$

▶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' $\boldsymbol{\varepsilon}(\mathbf{x}) = \|\mathbf{x}\|^2 - r^2$ 'arbitrary' choice
2. linearize it at $\hat{\mathbf{x}}$ we are dropping i in $\boldsymbol{\varepsilon}_i$, $\mathbf{e}_i$ etc for clarity

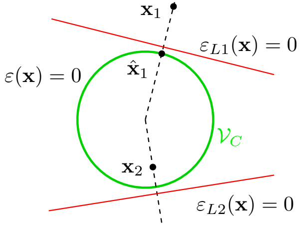
$$\boldsymbol{\varepsilon}(\hat{\mathbf{x}}) \approx \boldsymbol{\varepsilon}(\mathbf{x}) + \underbrace{\frac{\partial \boldsymbol{\varepsilon}(\mathbf{x})}{\partial \mathbf{x}}}_{\mathbf{J}(\mathbf{x})=2\mathbf{x}^\top} \underbrace{(\hat{\mathbf{x}} - \mathbf{x})}_{\mathbf{e}(\hat{\mathbf{x}},\mathbf{x})}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

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

3. using (20), express error approximation $\mathbf{e}^*$ as

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

4. fit circle

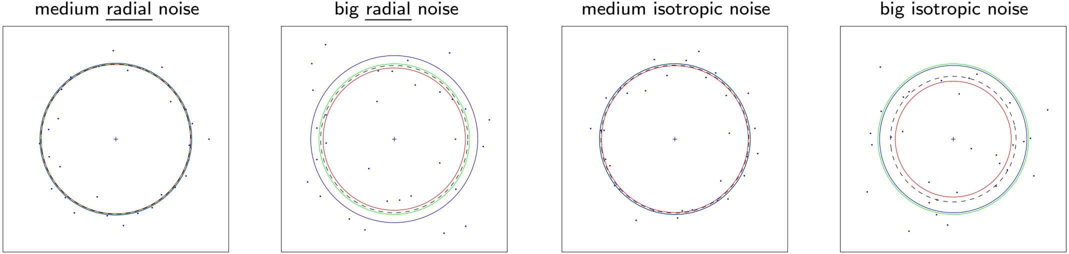


$$r^* = \arg\min_r \sum_{i=1}^k \frac{(\|\mathbf{x}_i\|^2 - r^2)^2}{4\|\mathbf{x}_i\|^2} = \cdots = \left(\frac{1}{k}\sum_{i=1}^k \frac{1}{\|\mathbf{x}_i\|^2}\right)^{-\frac{1}{2}}$$

- this example results in a convex quadratic optimization problem
- note that the 'algebraic error' minimizer is different:

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

Circle Fitting: Some Results



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; smaller is better

solid green – ground truth
solid red – Sampson error $\mathbf{e}$ minimizer
solid blue – direct algebraic radial error $\boldsymbol{\varepsilon}$ minimizer
dashed 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 (!) the devil is hiding there
- 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

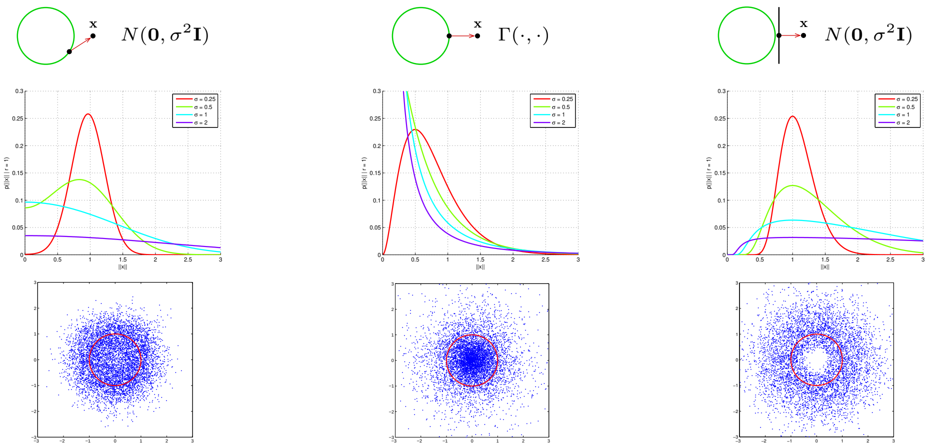
Discussion: On The Art of Probabilistic Model Design...

- a few probabilistic 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



$p(\mathbf{x} \mid r)$:

marginalized over C:
$$\approx \frac{1}{\sigma\sqrt{(2\pi)^3 r\,\|\mathbf{x}\|}}\, e^{-\frac{(\|\mathbf{x}\|-r)^2}{2\sigma^2}}$$

- mode inside the circle
- models the inside well
- tends to normal distribution

orthogonal deviation from C:
$$\frac{1}{2\pi\Gamma(\frac{r^2}{\sigma})}\,\frac{1}{\|\mathbf{x}\|^2}\left(\frac{r\|\mathbf{x}\|}{\sigma}\right)^{\frac{r^2}{\sigma}} e^{-\frac{r\|\mathbf{x}\|}{\sigma}}$$

- peak at the center
- unusable for small radii
- tends to Dirac distribution

Sampson approximation:
$$\frac{1}{r\sigma\sqrt{(2\pi)^3}}\, e^{-\frac{e^2(\mathbf{x};r)}{2\sigma^2}}$$

- mode at the circle
- hole at the center
- tends to normal distribution

▶Sampson Error for Fundamental Matrix Manifold

The (signed) epipolar algebraic error is assuming finite points

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

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), $\mathbf{S} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$, then

Sampson

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

derivatives over point coordinates

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

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

Sampson error vector

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

scalar Sampson error

- generalization for infinite points is easy
- Sampson error ‘normalizes’ the algebraic error
- automatically copes with multiplicative factors $\mathbf{F} \mapsto \lambda\mathbf{F}$
- the actual optimization not yet covered →112

▶Sampson Error for Triangulation: The Golden Standard Triangulation Method

Given $\mathbf{P}_1$, $\mathbf{P}_2$ and a correspondence $x \leftrightarrow y$, look for 3D point $\mathbf{X}$ projecting to x and y →90

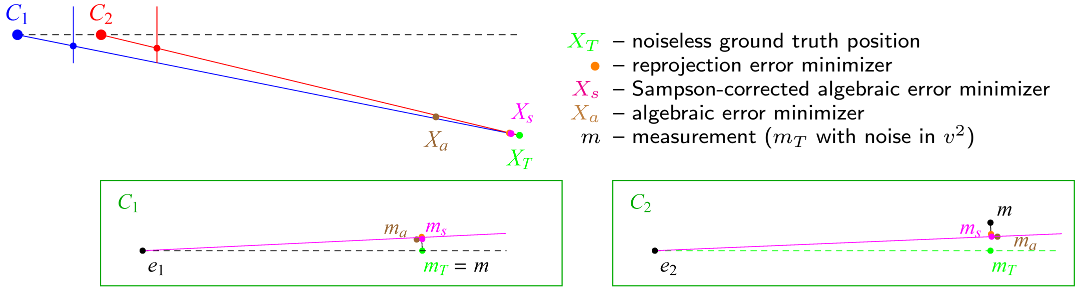
Idea:

1. if not given, compute $\mathbf{F}$ from $\mathbf{P}_1$, $\mathbf{P}_2$, e.g. $\mathbf{F} = (\mathbf{Q}_1\mathbf{Q}_2^{-1})^\top [\mathbf{q}_1 - (\mathbf{Q}_1\mathbf{Q}_2^{-1})\mathbf{q}_2]_\times$ →77
2. correct the measurement by the linear estimate of the correction vector $\mathbf{e}_i(\mathbf{F})$ →103

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

3. use the SVD triangulation algorithm with numerical conditioning →91

Ex (cont'd from →97):



▶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 $\mathbf{F}$ (or essential matrix $\mathbf{E}$).

What we have so far

- 7-point algorithm for $\mathbf{F}$ (5-point algorithm for $\mathbf{E}$) →85
- definition of Sampson error per correspondence $e_i(\mathbf{F} \mid x_i, y_i)$ →107
- triangulation requiring an optimal $\mathbf{F}$

What we need

- correspondence recognition see later →118
- an optimization algorithm for many $(k \gg 7)$ correspondences comes next

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

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

Levenberg-Marquardt (LM) Iterative Optimization in a Nutshell

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

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

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

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

$$\boldsymbol{\theta}^{s+1} := \boldsymbol{\theta}^s + \mathbf{d}_s, \quad \text{where} \quad \mathbf{d}_s = \arg\min_{\mathbf{d}} \sum_{i=1}^{k} \|\mathbf{e}_i(\boldsymbol{\theta}^s + \mathbf{d})\|^2 \tag{21}$$

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

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

Then the solution to Problem (21) is a set of ‘normal eqs’

$$\underbrace{-\sum_{i=1}^{k} \mathbf{L}_i^\top \mathbf{e}_i(\boldsymbol{\theta}^s)}_{\mathbf{e} \in \mathbb{R}^{q,1}} = \underbrace{\left(\sum_{i=1}^{k} \mathbf{L}_i^\top \mathbf{L}_i\right)}_{\mathbf{L} \in \mathbb{R}^{q,q}} \mathbf{d}_s, \tag{22}$$

- $\mathbf{d}_s$ can be solved for by Gaussian elimination using Choleski decomposition of $\mathbf{L}$
  - $\mathbf{L}$ (large) symmetric PSD $\Rightarrow$ use Choleski, almost $2\times$ faster than Gauss-Seidel, see bundle adjustment →145
- beware of rank defficiency in $\mathbf{L}$ when k is small
- such updates do not lead to stable convergence $\longrightarrow$ ideas of Levenberg and Marquardt

LM (cont'd)

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 $\lambda \sum_i \operatorname{diag}(\mathbf{L}_i^\top \mathbf{L}_i)$ to adapt to local curvature:

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

Idea 4 (Marquardt): adaptive λ 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 $\lambda := \lambda/10$, $s := s + 1$ better: Armijo's rule
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 $k < q$)
- λ helps avoid the consequences of gauge freedom →147
- the error function can be made robust to outliers →119
- we have approximated the least squares Hessian by ignoring second derivatives of the error function (Gauss-Newton approximation) See [Triggs et al. 1999, Sec. 4.3]
- a good book on convex optimization: [Boyd and Vandenberghe(2009)]

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 $\mathbf{F}$)

$$\mathbf{L}_i = \frac{\partial e_i(\mathbf{F})}{\partial \mathbf{F}} = \cdots = \frac{1}{2\|\mathbf{J}_i\|} \left[\left(\underline{\mathbf{y}}_i - \frac{2e_i(\mathbf{F})}{\|\mathbf{J}_i\|} \mathbf{S}\mathbf{F}\underline{\mathbf{x}}_i \right) \underline{\mathbf{x}}_i^\top + \underline{\mathbf{y}}_i \left(\underline{\mathbf{x}}_i - \frac{2e_i(\mathbf{F})}{\|\mathbf{J}_i\|} \mathbf{S}\mathbf{F}^\top \underline{\mathbf{y}}_i \right)^\top \right] \tag{23}$$

- $\mathbf{L}_i$ in (23) is a 3×3 matrix, must be reshaped to dimension-9 vector $\operatorname{vec}(\mathbf{L}_i)$ to be used in LM
- $\underline{\mathbf{x}}_i$ and $\underline{\mathbf{y}}_i$ in Sampson error are normalized to unit homogeneous coordinate — (23) relies on this
- reinforce $\operatorname{rank} \mathbf{F} = 2$ after each LM update to stay on the fundamental matrix manifold and $\|\mathbf{F}\| = 1$ to avoid gauge freedom — by SVD →113
- LM linearization could be done by numerical differentiation (we can afford it, we have a small dimension here)

►Local Optimization for Fundamental Matrix Estimation

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 (→93) 7-point $\mathbf{F}_0$ (→85) from a suitable 7-tuple
  2. Improve the $\mathbf{F}_0^*$ using the LM optimization (→110–111) and the Sampson error (→112) on all inliers, reinforce rank-2, unit-norm $\mathbf{F}_k^*$ after each LM iteration using SVD

Partial conceptualization

- inlier = a correspondence (a true match)
- outlier = a non-correspondence
- binary inlier/outlier labels are hidden
- we can get their 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)

Example Matching Results for the 7-point Algorithm with Random-Sampling Optimization

input images | interest points (ca. 3600) | tentative corresp. (416) | matching (340)

- descriptors used to obtain tentative matches but no descriptors used in the final matching
- without local optimization the minimization is over a discrete set of epipolar geometries proposable from 7-tuples
- notice the mismatches (they have wrong depth, even negative) remember: hidden labels →113
- they are considered as random outliers to the epipolar model
- inlier matches will be treated as correspondences for the SfM problem

▶A Preview: Optimization by Random Sampling of Geometric Primitives

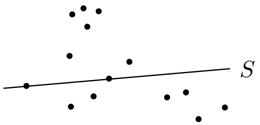
Given an optimization problem, define:

- parameters $\boldsymbol{\theta} \in \mathrm{domain}(\boldsymbol{\theta})$
- primitive geometric element $x_i \in \mathcal{P}$
- generator q of random minimal proposal s-tuples $S \in \mathcal{P}^s$ of primitive elements
- minimal-problem solver computing $\boldsymbol{\theta}$ from the s-tuples: $\mathrm{solver} : \mathcal{P}^s \to \mathrm{domain}(\boldsymbol{\theta})$
- objective function $\pi(\mathcal{P} \mid \boldsymbol{\theta})$

Examples:	$\boldsymbol{\theta}$	primitive	s	solver	$\pi(\cdot)$ terms
line fitting in 2D	$\underline{\mathbf{n}} \in \mathbb{R}^3$	point	2	$\underline{\mathbf{n}} \simeq \underline{\mathbf{x}}_1 \times \underline{\mathbf{x}}_2$	point-to-line distances
plane fitting in 3D	$\underline{\mathbf{p}} \in \mathbb{R}^4$	point	3	$\underline{\mathbf{p}} \simeq \mathrm{null}\big([\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2, \underline{\mathbf{x}}_3]^\top\big)$	point-to-plane distances
fundamental matrix fitting	$\mathbf{F}$	match 2D–2D	7	7-pt alg	Sampson errors
exterior orientation	$(\mathbf{R}, \mathbf{t})$	match 3D–2D	3	P3P alg	projection errors

Algorithm sketch:

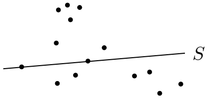
- propose a random s-tuple of primitives S using $q(\cdot)$
- run the solver on S to obtain parameters $\boldsymbol{\theta}$
- compute the value of $\pi(\mathcal{P} \mid \boldsymbol{\theta})$ on all primitives $\mathcal{P}$
- remember the sample which gave the best $\pi(\mathcal{P} \mid \boldsymbol{\theta})$



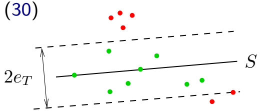
▶A Preview: RANSAC with Local Optimization and Early Stopping

Given: minimal configuration C definition, proposal distribution $q(\cdot)$, minimal-problem solver, objective $\pi(\cdot)$:

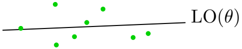
1. initialize the best parameters $\boldsymbol{\theta}_{\text{best}} := \emptyset$, $\pi_{\text{best}} := -\infty$, and proposal index $k := 0$
2. estimate the total number of needed proposals as $N := \binom{n}{s}$ — n – No. of primitives, s – minimal config size
3. while $k \leq N$:



   a) **propose** a random s-tuple S from $q(\cdot)$
   b) solve the minimal problem on S to obtain $\boldsymbol{\theta}$
   c) if $\pi(\mathcal{P} \mid \boldsymbol{\theta}) > \pi_{\text{best}}$ then **accept**
      i) update the best $\boldsymbol{\theta}_{\text{best}} := \boldsymbol{\theta}$ — $\pi(S)$ marginalized as in (29); $\pi(S)$ includes a prior $\Rightarrow$ MAP
      ii) threshold-out outliers using e_T from (30)



      iii) locally **optimize** $\boldsymbol{\theta}$ from the inliers of $\boldsymbol{\theta}_{\text{best}}$ — LM optimization with robustified (→121) Sampson error possibly weighted by posterior $\pi(m_{ij})$ [Chum et al. 2003]



      iv) update $\boldsymbol{\theta}_{\text{best}}$, update inliers using (30), re-estimate the **stopping criterion** N from inlier counts — →117 for derivation

$$N = \frac{\log(1-P)}{\log(1-\varepsilon^s)}, \quad \varepsilon = \frac{|\,\text{inliers}(\boldsymbol{\theta}_{\text{best}})|}{n},$$

   d) $k := k + 1$
4. output C_{best}

- see the MPV course for RANSAC details — see also [Fischler & Bolles 1981], [25 years of RANSAC]

▶Data-Driven Stopping Criterion

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

this will tell us nothing about the accuracy of the result

P . . . probability that the last proposal is an all-inlier for the first time — $1-P$ . . . all previous N proposals contained outlier(s)

ε . . . the fraction of inliers among primitives, $\varepsilon \leq 1$

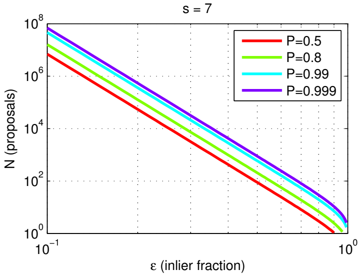
s . . . No. of primitives in a minimal configuration — 2 in line fitting, 7 in 7-point algorithm, 4 in homography fitting,. . .

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

- ε^s . . . proposal is all-inlier
- $1-\varepsilon^s$ . . . proposal contains at least one outlier
- $(1-\varepsilon^s)^N$ . . . N previous proposals contained an outlier $= 1-P$

N for $s=7$

	P	
ε	0.8	0.99
0.5	205	590
0.2	$1.3\cdot10^5$	$3.5\cdot10^5$
0.1	$1.6\cdot10^7$	$4.6\cdot10^7$



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

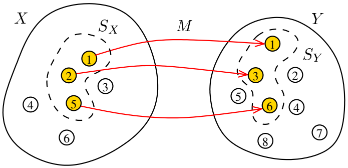
the quasi-posterior estimate for ε is the average over all samples generated so far

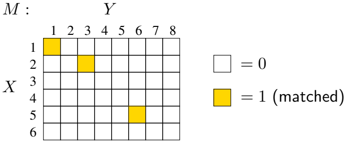
- this shows we have a good reason to limit all possible matches to tentative matches only
- for $\varepsilon \to 0$ we gain nothing over the standard MH-sampler stopping rule — not covered in this course

▶Towards $\pi(\cdot)$: The Full Problem of Matching and Fundamental Matrix Estimation

Problem: Given image keypoint 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





$$(M^*, \mathbf{F}^*) = \arg\max_{M,\mathbf{F}} \pi(E, D, \mathbf{F}, M) \qquad (E, D) \sim \mathcal{P},\ (\mathbf{F}, M) \sim \boldsymbol{\theta} \tag{24}$$

- probabilistic model: an efficient language for problem formulation — it also unifies 4.a and 4.b
- the (24) 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 $(M^*, \mathbf{F}^*) = \arg\max_{M,\mathbf{F}} p(E, D, \mathbf{F}, M)$ solve

$$\mathbf{F}^* = \arg\max_{\mathbf{F}} p(E, D, \mathbf{F}) \tag{25}$$

by <u>marginalization</u> of $p(E, D, \mathbf{F}, M)$ over the set of all matchings $\mathcal{M}$ s.t. $M \in \mathcal{M}$ — this changes the problem!

drop the assumption that M is a 1:1 matching, assume correspondence-wise independence:

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

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

<u>Approximate marginalization:</u> — take all the 2^{mn} terms in place of M

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

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

$$= \prod_{i=1}^{m}\prod_{j=1}^{n} \underbrace{\sum_{m_{ij}\in\{0,1\}} p_e(e_{ij}, d_{ij}, \mathbf{F} \mid m_{ij})P(m_{ij})}_{\text{we will continue with this term}} \tag{26}$$

Robust Matching Model (cont'd)

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

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

$$= (1 - P_0)\, p_1(e_{ij}, d_{ij}, \mathbf{F}) + P_0\, p_0(e_{ij}, d_{ij}, \mathbf{F}) \qquad (27)$$

- 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)

→121 for a simplification

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

- 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.

$$p_1(e_{ij} \mid \mathbf{F}) = \frac{1}{T_e(\sigma_1)}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}}, \quad p_1(d_{ij}) = \frac{1}{T_d(\sigma_d, \dim \mathbf{d})}\, e^{-\frac{\|\mathbf{d}(x_i) - \mathbf{d}(y_j)\|^2}{2\sigma_d^2}} \qquad (28)$$

- $\mathbf{F}$ is a random variable and σ_1, σ_d, P_0 are parameters
- the form of $T_e(\sigma_1)$ depends on the error definition, it may depend on x_i, y_j but not on $\mathbf{F}$
- we will continue with the result from (27)

Simplified Robust Energy (Error) Function

- assuming the choice of p_1 as in (28), we are simplifying (26) to

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

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

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

- we define the 'error function' as: $V(x) = -\log p(x)$ smaller V is better

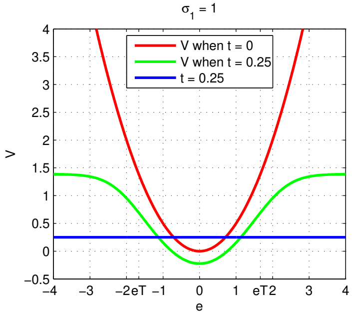
$$V(E, D \mid \mathbf{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left[\underbrace{-\log \frac{1-P_0}{T_e(\sigma_1)}}_{\Delta\, =\, \text{const}} - \log \left(e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + \underbrace{\frac{P_0}{1-P_0}\, \frac{T_e(\sigma_1)}{T_e(\sigma_0)}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_0^2}}}_{t\,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

$$= m\, n\, \Delta + \sum_{i=1}^{m} \sum_{j=1}^{n} \underbrace{-\log \left(e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + t \right)}_{\hat{V}(e_{ij})} \qquad (29)$$

- the terms in (29) are: (constant) + (total robust error for all pairs in M) expensive but explicit matching is avoided
- note we are summing over all $m\,n$ matches (m, n are constant!)
- when $t = 0$ we have quadratic inlier error function $\hat{V}(e_{ij}) = e_{ij}^2(\mathbf{F})/(2\sigma_1^2)$

▶The Action of the Robust Matching Model on Data

Ex: Error function $\hat{V}(e_{ij})$ (29):



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 (29)

- 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 (29)
- 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 $t \approx 0$

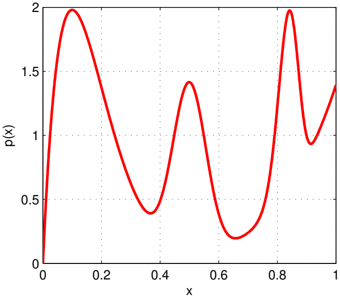
$$e_T = \sigma_1 \sqrt{-\log t^2}, \; t = e^{-\frac{1}{2}\left(\frac{e_T}{\sigma_1}\right)^2} \quad \text{e.g. } e_T = 4\sigma_1 \;\to\; t \approx 3.4 \cdot 10^{-4} \tag{30}$$

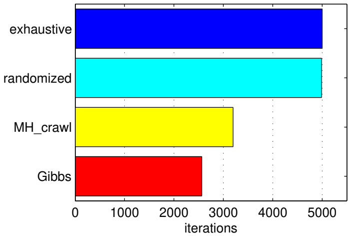
The full optimization problem (25) uses (29):

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

- 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 $\mathbf{F}$
- the 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?





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

- given a toy probability distribution $p(x)$ at left — $\theta = x$, 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
```

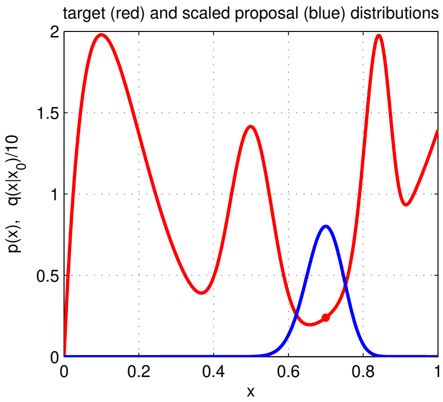
   - equally slow algorithm
   - fast to implement

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

- simpler (unimodal) distributions result in faster convergence

How To Generate Random Samples from a Complex Distribution?



- red: probability density function $\pi(x)$ of the toy distribution on the unit interval — target distribution

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

$$\mathrm{Be}(x; \alpha, \beta) = \frac{1}{\mathrm{B}(\alpha, \beta)} \cdot x^{\alpha-1}(1-x)^{\beta-1}, \quad \alpha, \beta \geq 0$$

- 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' proposal distribution $q(x \mid x_0)$, given the previous sample x_0 (blue)

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

- 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: carry information about $\boldsymbol{\theta}$ e.g. $C = \theta = x$ in →124, C - s-tuple on →115

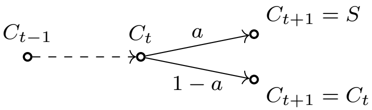
Goal: Generate a sequence of random samples $\{C_t\}$ from target distribution $\pi(C)$

Idea: Setup a Markov chain with a suitable transition probability to generate the sequence

Sampling procedure

1. given current configuration C_t, propose (draw a random) configuration sample S from $q(S \mid C_t)$
   q may use some information from C_t (Hastings)
2. compute acceptance probability — the redistribution filter; note 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. accept S with probability a
   a) draw a random number u from unit-interval uniform distribution $\mathrm{U}_{0,1}$
   b) if $u \le a$ then $C_{t+1} := S$ else $C_{t+1} := C_t$

'Programming' an MH sampler

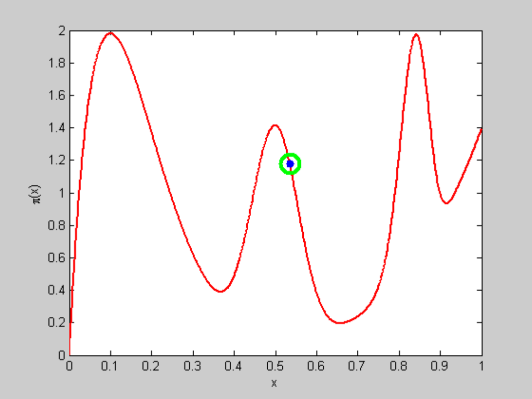
1. design a proposal distribution (mixture) q and a sampler from q
2. express functions $q(C_t \mid S)$ and $q(S \mid C_t)$ as proper distributions — not always simple

Finding the mode

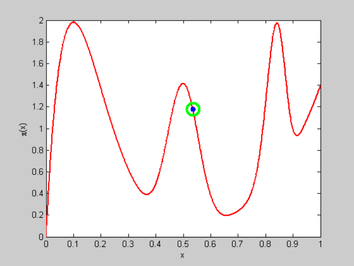
- remember the best sample — fast implementation but must wait long to hit the mode
- use simulated annealing — very slow
- use the sampler as an explorer and do local optimization from the accepted sample — a trade-off between speed and accuracy

an optimal algorithm does not use just the best sample: a Stochastic EM Algorithm (e.g. SAEM)

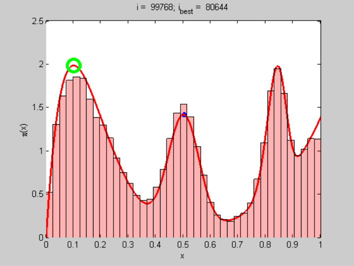
MH Sampling Demo



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



initial sample



final distribution of visited states

- blue point: current sample
- green circle: best sample so far
- histogram: current distribution of visited states
- the vicinity of modes are the most often visited states

quality $= \pi(x)$

Demo Source Code (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
```

```
%% 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
```

▶Stripping MH Down To Get RANSAC [Fischler & Bolles 1981]

- when we are interested in the best config only. . . and we need fast data exploration. . .
- . . . then Steps 2–4 below make no difference when waiting for the best sample configuration:

From sampling to RANSACing

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~~

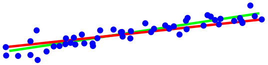
$$a = \min\left\{1,\ \frac{\pi(S)}{\pi(C_t)} \cdot \frac{q(C_t \mid S)}{q(S \mid C_t)}\right\}$$

3. ~~draw a random number u from unit-interval uniform distribution $U_{0,1}$~~
4. ~~if $u \le a$ then $C_{t+1} := S$ else $C_{t+1} := C_t$~~
5. if $\pi(S) > \pi(C_{\text{best}})$ then remember $C_{\text{best}} := S$

- this is (almost) the 'stupid' Method 2 from →123 but(!) the data-driven sampling via higher-order primitives
- it has a good overall exploration but slow convergence in the vicinity of a mode where C_t could serve as an attractor
- getting a good accuracy configuration might take very long this way
- (possibly robust) 'local optimization' necessary for reasonable performance
- unlike the full sampler, it cannot use the past generated configurations to estimate any parameters

The Elements of a Data-Driven MH Sampler

data-driven = proposals $q(S \mid C_t)$ are derived from data ⇒ parameter distribution follows the **empirical distribution** of the s-tuples of primitives. The parameter proposal is done via the minimal problem solver.

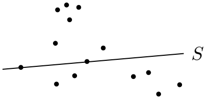


- pairs of points define line distribution $p(\mathbf{n} \mid X)$ (left)
- random correspondence 7-tuples define epipolar geometry distribution $p(\mathbf{F} \mid M)$

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 then be over a discrete 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 — $(\mathbf{x}^1, \mathbf{x}^2) \mapsto \mathbf{n}$
- fundamental matrix $\mathbf{F}$ from seven matches — $\{(\mathbf{x}_i^1, \mathbf{x}_i^2)\}_{i=1:7} \mapsto \mathbf{F}$
- homography $\mathbf{H}$ from four matches, etc — $\{(\mathbf{x}_i^1, \mathbf{x}_i^2)\}_{i=1:4} \mapsto \mathbf{H}$

cont'd

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 $\mathbf{F}$ (29)
  - robustified point distance from the line $\boldsymbol{\theta} = \mathbf{n}$
  - robustified Sampson error for $\boldsymbol{\theta} = \mathbf{F}$, etc
- posterior likelihood $p(E, D \mid \boldsymbol{\theta}(C))p(\boldsymbol{\theta}(C))$ can be used — MAPSAC ($\pi(S)$ includes the prior)

5. **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'

6. (optional) hard **inlier/outlier discrimination** by the threshold (30)

$$\hat{V}(e_{ij}) < e_T, \qquad e_T = \sigma_1 \sqrt{-\log t^2}$$

7. **local optimization** from promising proposals

- can use the hard inliers or just the robust error (29) — more expensive but more stable
- cannot be used to replace C_t — it would violate 'detailed balance' required for the MH scheme

8. **stopping** based on the probability of proposing an all-inlier configuration →117

Harnessing The Full Power of MH Sampler

By marginalization in (25) 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 $C = M$ — $\mathbf{F}$ computable from M
  - explicit labeling m_{ij} can be done by, e.g. sampling from

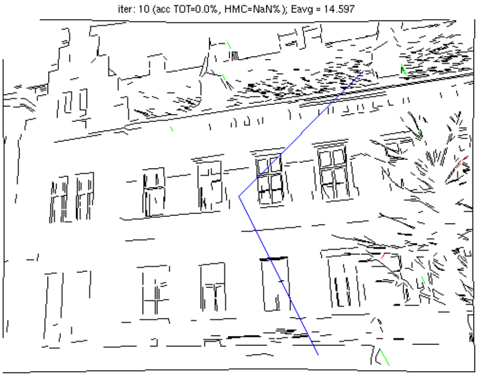
    $$q(m_{ij} \mid \mathbf{F}) \sim \big((1 - P_0)\, p_1(e_{ij} \mid \mathbf{F}),\ P_0\, p_0(e_{ij} \mid \mathbf{F})\big)$$

    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 the sequence $\{C_i\}$
  - local optimization can then use explicit inliers and $p(m_{ij})$
- error can be estimated for the elements of $\mathbf{F}$ from the sequence $\{C_i\}$ — 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 models (epipolar geometries, homographies, ...) — by reversible jump MCMC

  if there are multiple models explaning data, RANSAC will return one of them randomly

Example: MH Sampling for a More Complex Problem

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



click for video

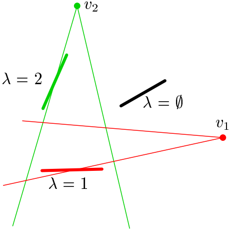
simplifications

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

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

- primitives = line segments
- latent variables
  1. each line has a vanishing point label $\lambda_i \in \{\emptyset, 1, 2\}$, $\emptyset$ = outlier
  2. 'mother line' parameters θ_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 $s = 2$
- Gibbs sampling for λ_i

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



- blue lines point away from the vanishing points
- proposal acceptance: 20%
- ca. 150 iterations to a good solution

Module VI

3D Structure and Camera Motion

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.

▶Reconstructing Camera System by Gluing Camera Triples

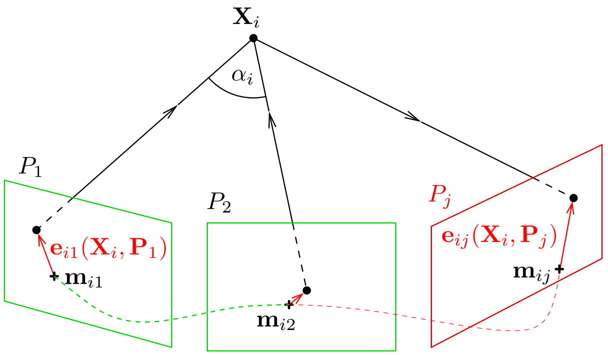
Given: Calibration matrices $\mathbf{K}_j$ and tentative correspondences per camera triples.

Initialization

1. initialize camera cluster $\mathcal{C}$ with a pair P_1, P_2
2. find essential matrix $\mathbf{E}_{12}$ and matches M_{12} by the 5-point algorithm →89
3. select scale $s_{1,2} = 1$
4. construct camera pair

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

5. triangulate $\{X_i\}$ per match from M_{12} →108
6. initialize point cloud $\mathcal{X}$ 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 not 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 bundle adjustment on $\mathcal{X}$ and $\mathcal{C}$ coming next →142

▶The Projective Reconstruction Theorem

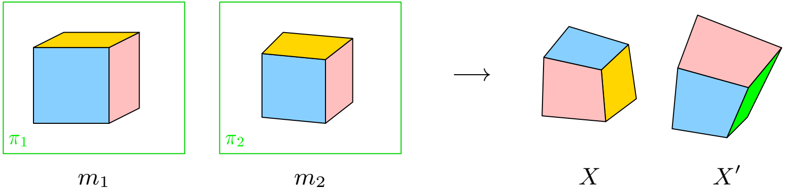
- We can run an analogical procedure when the cameras remain uncalibrated. But:

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

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

- 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 homography $\mathbf{H}$

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

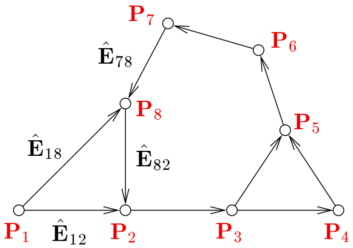


- when cameras are internally calibrated ($\mathbf{K}_j$ known) then $\mathbf{H}$ is restricted to a similarity since it must preserve the calibrations $\mathbf{K}_j$ [H&Z, Secs. 10.2, 10.3], [Longuet-Higgins 1981]
  (translation, rotation, scale) →137 for an indirect proof

▶Reconstructing Camera System from Pairs (Correspondence-Free)

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 $\mathbf{P}_i$, $i = 1, \dots, k$

→82 and →154 on representing $\mathbf{E}$



We construct calibrated camera pairs $\hat{\mathbf{P}}_{ij} \in \mathbb{R}^{6,4}$ see (19)

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

- 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 $\hat{\mathbf{P}}_{ij}\mathbf{H}_{ij} = \mathbf{P}_{ij}$ — $\mathbf{H}_{ij} \in \mathrm{SIM}(3)$

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hat{\mathbf{R}}_{ij} & \hat{\mathbf{t}}_{ij} \end{bmatrix}}_{\in \mathbb{R}^{6,4}} \underbrace{\begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\top & s_{ij} \end{bmatrix}}_{\mathbf{H}_{ij} \in \mathbb{R}^{4,4}} \overset{!}{=} \underbrace{\begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{R}_j & \mathbf{t}_j \end{bmatrix}}_{\in \mathbb{R}^{6,4}} \tag{31}$$

- (31) is a system of $24p$ eqs. in $7p + 6k$ unknowns — $24 = 6 \cdot 4$, $7p \sim (\mathbf{t}_{ij}, \mathbf{R}_{ij}, s_{ij})$, $6k \sim (\mathbf{R}_i, \mathbf{t}_i)$
- each $\hat{\mathbf{P}}_i = (\mathbf{R}_i, \mathbf{t}_i)$ appears on the RHS as many times as is the degree of node $\mathbf{P}_i$ — eg. P_5 3×

►cont'd

Eq. (31) implies $\begin{bmatrix}\mathbf{R}_{ij}\\ \hat{\mathbf{R}}_{ij}\mathbf{R}_{ij}\end{bmatrix} = \begin{bmatrix}\mathbf{R}_i\\ \mathbf{R}_j\end{bmatrix}$ and $\begin{bmatrix}\mathbf{t}_{ij}\\ \hat{\mathbf{R}}_{ij}\mathbf{t}_{ij} + s_{ij}\hat{\mathbf{t}}_{ij}\end{bmatrix} = \begin{bmatrix}\mathbf{t}_i\\ \mathbf{t}_j\end{bmatrix}$

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

$$\hat{\mathbf{R}}_{ij}\mathbf{R}_i = \mathbf{R}_j, \qquad \hat{\mathbf{R}}_{ij}\mathbf{t}_i + s_{ij}\hat{\mathbf{t}}_{ij} = \mathbf{t}_j, \qquad s_{ij} > 0 \tag{32}$$

- 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

$$\mathbf{R}_1 = \mathbf{I}, \qquad \sum_{i=1}^{k}\mathbf{t}_i = \mathbf{0}, \qquad \frac{1}{p}\sum_{i,j} s_{ij} = 1 \tag{33}$$

- rotation equations are decoupled from translation equations
- in principle, s_{ij} could correct the sign of $\hat{\mathbf{t}}_{ij}$ from essential matrix decomposition →82
  but $\mathbf{R}_i$ cannot correct the α sign in $\hat{\mathbf{R}}_{ij}$ ⇒ therefore make sure all points are in front of cameras and constrain $s_{ij} > 0$; →84

\+ pairwise correspondences are sufficient

− suitable for well-distributed cameras only (dome-like configurations) otherwise intractable or numerically unstable

Finding The Rotation Component in Eq. (32)

1. **Poor Man's Algorithm:**
   a) create a Minimum Spanning Tree of $\mathcal{G}$ from →136
   b) propagate rotations from $\mathbf{R}_1 = \mathbf{I}$ via $\hat{\mathbf{R}}_{ij}\mathbf{R}_i = \mathbf{R}_j$ from (32)

2. **Rich Man's Algorithm:**
   Consider $\hat{\mathbf{R}}_{ij}\mathbf{R}_i = \mathbf{R}_j$, $(i,j) \in E(\mathcal{G})$, where $\mathbf{R}$ are a 3×3 rotation matrices
   Errors per columns $c = 1, 2, 3$ of $\mathbf{R}_j$:

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

   Solve

$$\arg\min \sum_{(i,j)\in E(\mathcal{G})} \sum_{c=1}^{3} (\mathbf{e}_{ij}^c)^\top \mathbf{e}_{ij}^c \quad \text{s.t.} \quad (\mathbf{r}_i^k)^\top(\mathbf{r}_j^l) = \begin{cases} 1 & i = j \wedge k = l \\ 0 & i \neq j \wedge k = l \\ 0 & i = j \wedge k \neq l \end{cases}$$

   this is a quadratic programming problem

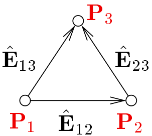
3. **SVD-Lover's Algorithm:**
   Ignore the constraints and project the solution onto rotation matrices see next

Per columns $c = 1, 2, 3$ of $\mathbf{R}_j$:

$$\hat{\mathbf{R}}_{ij}\mathbf{r}_i^c - \mathbf{r}_j^c = \mathbf{0}, \qquad \text{for all } i,\ j \tag{34}$$

- fix c and denote $\mathbf{r}^c = \left[\mathbf{r}_1^c, \mathbf{r}_2^c, \dots, \mathbf{r}_k^c\right]^\top$ c-th columns of all rotation matrices stacked; $\mathbf{r}^c \in \mathbb{R}^{3k}$
- then (34) becomes $\mathbf{D}\,\mathbf{r}^c = \mathbf{0}$ $\quad \mathbf{D} \in \mathbb{R}^{3p,3k}$
- $3p$ equations for $3k$ unknowns $\to p \geq k$ $\quad$ in a 1-connected graph we have to fix $\mathbf{r}_1^c = [1, 0, 0]$

Ex: $(k = p = 3)$



$\rightarrow$

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

$\rightarrow$

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

- must hold for any c

Idea: $\quad$ [Martinec & Pajdla CVPR 2007]

1. find the space of all $\mathbf{r}^c \in \mathbb{R}^{3k}$ that solve (34) $\quad$ $\mathbf{D}$ is sparse, use `[V,E] = eigs(D'*D,3,0);` (Matlab)
2. choose 3 unit orthogonal vectors in this space $\quad$ 3 smallest eigenvectors
3. find closest rotation matrices per cam. using SVD $\quad$ 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

Finding The Translation Component in Eq. (32)

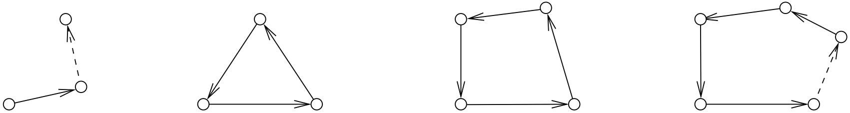
From (32) and (33): $0 < d \le 3$ – rank of camera center set, p – #pairs, k – #cameras

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

- in rank d: $\underbrace{d \cdot p}_{\text{(a)}} + \underbrace{d}_{\text{(b)}} + \underbrace{1}_{\text{(c)}}$ indep. eqns for $\underbrace{d \cdot k}_{\mathbf{t}_i} + \underbrace{p}_{s_{ij}}$ unknowns $\rightarrow p \ge \frac{d(k-1)-1}{d-1} \stackrel{\text{def}}{=} Q(d,k)$

Ex: Chains, circuits construction of $\mathbf{t}_i$ from sticks of known orientation $\hat{\mathbf{t}}_{ij}$ and unknown length s_{ij} up to overall scale?

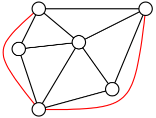
$p = k - 1$ | $k = p = 3$ | $k = p = 4$ | $k = p > 4$



$k \le 2$ for any d | $3 \ge d \ge 2$: non-collinear ok | $3 \ge d \ge 3$: non-planar ok | $3 \ge d \ge k-1$: impossible

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

maximal planar triangulated graph example:



Linear equations in (32) and (33) can be rewritten to

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

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

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

- 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 the initial estimates of all parameters

Given:

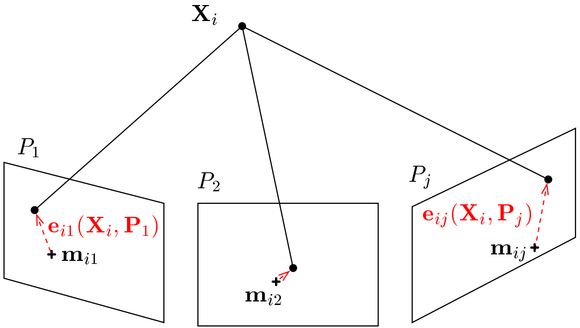
1. set of 3D points $\{\mathbf{X}_i\}_{i=1}^p$
2. set of cameras $\{\mathbf{P}_j\}_{j=1}^c$
3. correspondence & fixed tentative projections $\mathbf{m}_{ij}$

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 projection error $\mathbf{e}_{ij}(\mathbf{X}_i, \mathbf{P}_j) = \mathbf{x}_i - \mathbf{m}_i$ 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 constructed by marginalization over v_{ij}, as in the Robust Matching Model →120

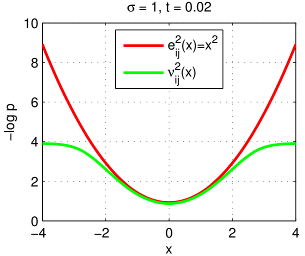
$$p(\{\mathbf{e}\} \mid \{\mathbf{P}, \mathbf{X}\}) = \prod_{\text{pts}:i=1}^{p} \prod_{\text{cams}:j=1}^{c} \Big((1 - P_0) p_1(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j) + P_0\, p_0(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j) \Big)$$

the marginalized negative log-density is (→121)

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

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

$$(\mathbf{L}_{ij})_l = \frac{\partial \nu_{ij}}{\partial \theta_l} = \underbrace{\frac{1}{1 + t\, e^{e_{ij}^2(\boldsymbol{\theta})/(2\sigma_1^2)}}}_{\text{small for } e_{ij} \gg \sigma_1} \cdot \frac{1}{\nu_{ij}(\theta)} \cdot \frac{1}{4\sigma_1^2} \cdot \frac{\partial e_{ij}^2(\theta)}{\partial \theta_l} \qquad (35)$$



but the LM method stays the same as before →110–111

- outliers (wrong v_{ij}): almost no impact on $\mathbf{d}_s$ in normal equations because the red term in (35) 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

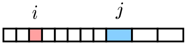
We have $q = 3p + 11k$ parameters: $\boldsymbol{\theta} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p; \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_k)$ points, cameras

We will use a multi-index $r = 1, \ldots, z$, $z = p \cdot k$. Then r correspond to point-cam pairs (i, j)

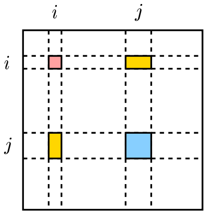
$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{r=1}^{z} \nu_r^2(\boldsymbol{\theta}), \qquad \boldsymbol{\theta}^{s+1} := \boldsymbol{\theta}^s + \mathbf{d}_s, \qquad -\sum_{r=1}^{z} \mathbf{L}_r^\top \nu_r(\boldsymbol{\theta}^s) = \left(\sum_{r=1}^{z} \mathbf{L}_r^\top \mathbf{L}_r +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

The block-form of $\mathbf{L}_r$ in Levenberg-Marquardt (→110) is zero except in columns i and j:

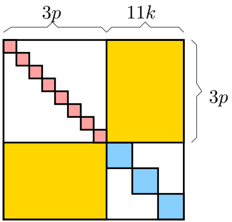
r-th error term is $\nu_r^2 = \rho(e_{ij}^2(\mathbf{X}_i, \mathbf{P}_j))$

$\mathbf{L}_r =$ 

$r = (i, j)$ blocks:
- ■: $\mathbf{X}_i$, 1×3
- ■: $\mathbf{P}_j$, 1×11

$\mathbf{L}_r^\top \mathbf{L}_r =$ 

blocks:
- ■: $\mathbf{X}_i - \mathbf{X}_i$, 3×3
- ■: $\mathbf{X}_i - \mathbf{P}_j$, 3×11
- ■: $\mathbf{P}_j - \mathbf{P}_j$, 11×11

$\sum_{r=1}^{z} \mathbf{L}_r^\top \mathbf{L}_r =$ 

- "points-first-then-cameras" parameterization scheme

▶Choleski Decomposition for B. A.

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

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

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

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

1. decompose $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ — `L = chol(A);` transforms the problem to $\mathbf{L}\underbrace{\mathbf{L}^\top\mathbf{x}}_{\mathbf{c}} = \mathbf{b}$
2. solve for $\mathbf{x}$ in two passes:

$$\mathbf{L}\,\mathbf{c} = \mathbf{b} \qquad \mathbf{c}_i := \mathbf{L}_{ii}^{-1}\Big(\mathbf{b}_i - \sum_{j<i} \mathbf{L}_{ij}\mathbf{c}_j\Big)$$

forward substitution, $i = 1, \ldots, q$ (params)

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

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 $\mathbf{A}$ and diagonal pivoting for semi-definite $\mathbf{A}$ — see above; [Triggs et al. 1999]
- λ 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 (kalibrační invariance)

1. The external frame is not fixed: See the Projective Reconstruction Theorem →135

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

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

- $\mathbf{P}$ is 12 numbers for 11 parameters
- we may represent $\mathbf{P}$ in decomposed form $\mathbf{K}$, $\mathbf{R}$, $\mathbf{t}$ $5 + 3 + 3 = 11$
- but $\mathbf{R}$ is 9 numbers representing the 3 parameters of rotation

If ignored, then

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

Solutions

1. fixing the external frame (e.g. a selected camera frame) explicitly or by constraints
2. fixing the scale (e.g. $s_{1,2} = 1$)

3a. either imposing constraints on projective entities
   - cameras, e.g. $\mathbf{P}_{3,4} = 1$ this excludes affine cameras
   - points, e.g. $(\underline{\mathbf{X}}_i)_4 = 1$ or $\|\underline{\mathbf{X}}_i\|^2 = 1$ the 2nd: can represent points at infinity

3b. or using minimal representations
   - points in their Cartesian representation $\mathbf{X}_i$ but finite points may be an unrealistic model
   - rotation matrices can be represented by (the exponential of) skew-symmetric matrices →152

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 reduced parameters $\hat{\theta}$ and replication matrix $\mathbf{T}$:

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

then $\mathbf{L}_r$ in LM changes to $\mathbf{L}_r\mathbf{T}$ and everything else stays the same →110

$$\mathbf{T} = \begin{array}{c|cccc} & \hat{\theta}_1 & \hat{\theta}_2 & \hat{\theta}_3 & \hat{\theta}_4 \\ \hline \theta_1 & 1 & & & \\ \theta_2 & & 1 & & \\ \theta_3 & & & & \\ \theta_4 & & & & 1 \\ \theta_5 & & & & 1 \end{array} \qquad \mathbf{t} = \begin{bmatrix} \\ \\ 1 \\ \\ \\ \end{bmatrix}$$

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 θ_j corresponding to all-zero rows of $\mathbf{T}$ — fixed θ

- constraining projective entities →152–154
- more complex constraints tend to make normal equations dense
- implementing constraints is safer than reparameterization, it 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 Parameterization and Motion Representation

- for any square matrix we define

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

- some properties:

$$\operatorname{expm}(x) = e^x, \quad x \in \mathbb{R}, \quad \operatorname{expm}\mathbf{0} = \mathbf{I}, \quad \operatorname{expm}(-\mathbf{A}) = \big(\operatorname{expm}\mathbf{A}\big)^{-1},$$

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

$\operatorname{expm}(\mathbf{A}^\top) = (\operatorname{expm}\mathbf{A})^\top$ hence if $\mathbf{A}$ is skew symmetric then $\operatorname{expm}\mathbf{A}$ is orthogonal:

$$\big(\operatorname{expm}(\mathbf{A})\big)^\top = \operatorname{expm}(\mathbf{A}^\top) = \operatorname{expm}(-\mathbf{A}) = \big(\operatorname{expm}(\mathbf{A})\big)^{-1}$$

$$\det\big(\operatorname{expm}\mathbf{A}\big) = e^{\operatorname{tr}\mathbf{A}}$$

Some consequences

- traceless matrices ($\operatorname{tr}\mathbf{A} = 0$) map to unit-determinant matrices $\Rightarrow$ we can represent homogeneous matrices
- skew-symmetric matrices map to orthogonal matrices $\Rightarrow$ we can represent rotations
- matrix exponential provides the exponential map from the powerful (matrix) Lie group theory

Lie Groups Useful in 3D Vision

group		matrix	represent
special linear	$\mathrm{SL}(3,\mathbb{R})$	real 3×3, unit determinant $\mathbf{H}$	2D homography
special linear	$\mathrm{SL}(4,\mathbb{R})$	real 4×4, unit determinant $\mathbf{H}$	3D homography
special orthogonal	$\mathrm{SO}(3)$	real 3×3 orthogonal $\mathbf{R}$	3D rotation
special Euclidean	$\mathrm{SE}(3)$	4×4 $\begin{bmatrix}\mathbf{R} & \mathbf{t}\\ \mathbf{0} & 1\end{bmatrix}$, $\mathbf{R}\in\mathrm{SO}(3)$, $\mathbf{t}\in\mathbb{R}^3$	3D rigid motion
similarity	$\mathrm{Sim}(3)$	4×4 $\begin{bmatrix}\mathbf{R} & \mathbf{t}\\ \mathbf{0} & s^{-1}\end{bmatrix}$, $s\in\mathbb{R}\setminus 0$	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 a have a closed-form formula for the exponential and for its principal inverse
- Jacobians are also readily available for $\mathrm{SO}(3)$, $\mathrm{SE}(3)$ [Solà 2020]

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 $\operatorname{tr} \mathbf{Z} = 0$

$$\mathbf{R} = \operatorname{expm}\,[\boldsymbol{\phi}]_\times, \quad \boldsymbol{\phi} = (\phi_1,\, \phi_2,\, \phi_3) = \varphi\,\mathbf{e}_\varphi \in \mathbb{R}^3, \quad 0 \le \varphi < \pi, \quad \|\mathbf{e}_\varphi\| = 1$$

- 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

$$\mathbf{R} = \operatorname{expm}\,[\boldsymbol{\phi}]_\times = \sum_{n=0}^{\infty} \frac{[\boldsymbol{\phi}]_\times^n}{n!} = \overset{\circledast 1}{\cdots} = \mathbf{I} + \frac{\sin\varphi}{\varphi}\,[\boldsymbol{\phi}]_\times + \frac{1-\cos\varphi}{\varphi^2}\,[\boldsymbol{\phi}]_\times^2$$

- (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 φ in radians
- finite limits for $\varphi \to 0$ exist: $\sin(\varphi)/\varphi \to 1$, $(1-\cos\varphi)/\varphi^2 \to 1/2$

$$\mathbf{M} = \operatorname{expm}\,[\boldsymbol{\nu}]_\wedge\,, \qquad \boldsymbol{\nu} \in \mathbb{R}^6$$

- $\mathrm{SE}(3)$ group element — 4 × 4 matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad \text{s.t.} \quad \mathbf{R} \in \mathrm{SO}(3),\ \mathbf{t} \in \mathbb{R}^3$$

- $\mathfrak{se}(3)$ algebra element — 4 × 4 matrix; $\wedge = \times$ in $\mathrm{SO}(3)$

$$[\boldsymbol{\nu}]_\wedge = \begin{bmatrix} [\boldsymbol{\phi}]_\times & \boldsymbol{\rho} \\ \mathbf{0} & 0 \end{bmatrix} \quad \text{s.t.} \quad \boldsymbol{\phi} \in \mathbb{R}^3,\ \varphi = \|\boldsymbol{\phi}\| < \pi,\ \boldsymbol{\rho} \in \mathbb{R}^3$$

- exponential map in closed form

$$\mathbf{R} = \operatorname{expm}\,[\boldsymbol{\phi}]_\times\,, \qquad \mathbf{t} = \operatorname{dexpm}([\boldsymbol{\phi}]_\times)\,\boldsymbol{\rho}$$

$$\operatorname{dexpm}([\boldsymbol{\phi}]_\times) = \sum_{n=0}^{\infty} \frac{[\boldsymbol{\phi}]_\times^n}{(n+1)!} = \mathbf{I} + \frac{1-\cos\varphi}{\varphi^2}[\boldsymbol{\phi}]_\times + \frac{\varphi - \sin\varphi}{\varphi^3}[\boldsymbol{\phi}]_\times^2$$

$$\operatorname{dexpm}^{-1}([\boldsymbol{\phi}]_\times) = \mathbf{I} - \frac{1}{2}[\boldsymbol{\phi}]_\times + \frac{1}{\varphi^2}\left(1 - \frac{\varphi}{2}\cot\frac{\varphi}{2}\right)[\boldsymbol{\phi}]_\times^2$$

- dexpm: differential of the exponential in $\mathrm{SO}(3)$
- (principal) logarithm via a similar trick as in $\mathrm{SO}(3)$
- finite limits exist: $(\varphi - \sin\varphi)/\varphi^3 \to 1/6$
- this form is preferred to $\mathrm{SO}(3) \times \mathbb{R}^3$

▶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 pose via $\mathrm{SO}(3) \times \mathbb{R}^3$ or $\mathrm{SE}(3)$

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix} = \begin{bmatrix}\mathbf{K} & \mathbf{0}\end{bmatrix} \mathbf{M}, \qquad 5 + 3 + 3 = 11 \text{ DOF} \qquad \mathbf{M} \in \mathrm{SE}(3)$$

- $\mathrm{Sim}(3)$ useful for SfM without scale
  - closed-form formulae still exist but they are a bit too messy [Eade(2017)]
- 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.

E. Eade. Lie groups for 2D and 3D transformations. On-line at http://www.ethaneade.org/, May 2017.

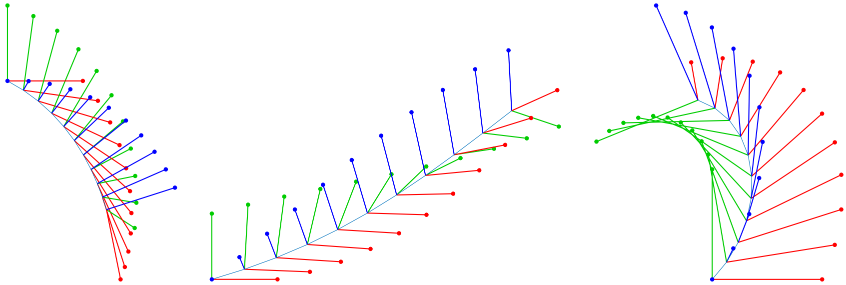
Motion Interpolation

- let G be a Lie group like $\mathrm{SO}(3)$, $\mathrm{SE}(3)$
- let $\mathbf{M} \in G$ be motion from time $t = 0$ to time $t = 1$
- then the motion from $t = 0$ to t is interpolated as

$$\mathbf{M}(t) = \exp\big(t \log(\mathbf{M})\big), \qquad t \in [0, 1]$$

- the trajectory is constant-speed,
- and the speed is $\log(\mathbf{M})$

Examples in $\mathrm{SE}(3)$:



- Integration formula — the motion is along the geodesic (shortest-distance curve)

$$\lim_{n\to\infty}\prod_{i=1}^{n}\exp\left(\frac{1}{n}\log(\mathbf{M})\right)=\mathbf{M}$$

- hat and vee functions:
  - $[\mathbf{a}]_\wedge$ maps vector $\mathbf{a}\in\mathbb{R}^d$ to algebra $\mathfrak{g}$ element (matrix)
  - $(\mathbf{B})_\vee$ maps algebra element $\mathbf{B}\in\mathfrak{g}$ to vector element, $\left([\mathbf{a}]_\wedge\right)_\vee=\mathbf{a}$

- the Log function is a composition of log and vee, $\mathrm{Log}:G\to\mathbb{R}^d$, $\mathrm{Log}(\mathbf{M})=(\log(\mathbf{M}))_\vee$ — $G\to\mathfrak{g}\to\mathbb{R}^d$
- then: left/right difference — $\mathbf{Y}\overset{\leftarrow}{\ominus}\mathbf{X}\in\mathbb{R}^d$

$$\mathbf{Y}\overset{\leftarrow}{\ominus}\mathbf{X}=\mathrm{Log}(\mathbf{Y}\mathbf{X}^{-1}),\quad \mathbf{Y}\overset{\rightarrow}{\ominus}\mathbf{X}=\mathrm{Log}(\mathbf{X}^{-1}\mathbf{Y})$$

- skew-symmetry

$$\mathbf{Y}\overset{\leftarrow}{\ominus}\mathbf{X}=-(\mathbf{X}\overset{\leftarrow}{\ominus}\mathbf{Y}),\quad \mathbf{Y}\overset{\rightarrow}{\ominus}\mathbf{X}=-(\mathbf{X}\overset{\rightarrow}{\ominus}\mathbf{Y})$$

- left/right distance

$$\overleftarrow{d}(\mathbf{X},\mathbf{Y})=\|\mathbf{Y}\overset{\leftarrow}{\ominus}\mathbf{X}\|,\quad \overrightarrow{d}(\mathbf{X},\mathbf{Y})=\|\mathbf{Y}\overset{\rightarrow}{\ominus}\mathbf{X}\|$$

- not equal but both are non-negative, symmetric — + additional properties, e.g. left/right invariance,...

Module VII

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.

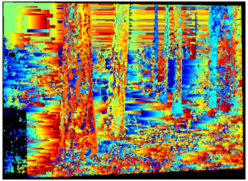
M. Pollefeys, R. Koch, and L. V. Gool. A simple and efficient rectification method for general motion. In *Proc Int Conf on Computer Vision*, vol. 1:496–501, 1999.

Stereovision = Getting Relative Distances Per Pixel given the Epipolar Geometry

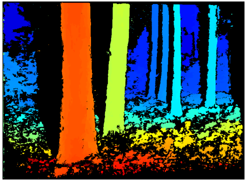
The success of a model-free stereo matching algorithm is unlikely:

WTA Matching:

For every left-image pixel find the most similar right-image pixel along the corresponding epipolar line. [Marroquin 83]



disparity map from WTA



a good disparity map

- monocular vision already gives a rough 3D sketch because we understand the scene
- pixelwise independent matching without any problem understanding is difficult
- matching can benefit from a geometric simplification of the problem: epipolar rectification

▶Linear Epipolar Rectification for Easier Correspondence Search

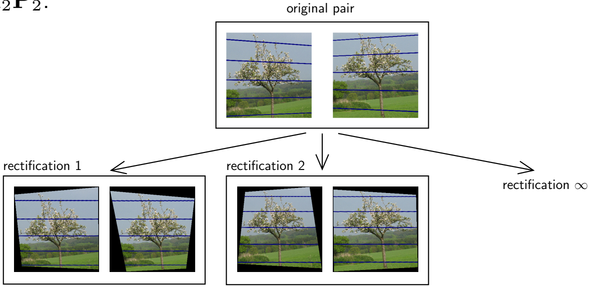
Obs:

- epipoles and epipolars are elements of $\mathbb{P}^2$, they may be mapped by homographies
- if we map epipoles to infinity, epipolars become parallel
- we then rotate them to become horizontal
- we then scale the images to make corresponding epipolars 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 epipolars to horizontal lines 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 $\mathbf{F} \mapsto \mathbf{H}_2^{-\top}\mathbf{F}\mathbf{H}_1^{-1}$ or the cameras $\mathbf{P}_1 \mapsto \mathbf{H}_1\mathbf{P}_1$, $\mathbf{P}_2 \mapsto \mathbf{H}_2\mathbf{P}_2$.

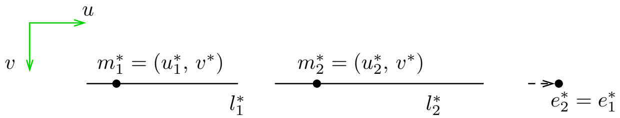


▶Rectification Homographies

Assumption: Cameras $(\mathbf{P}_1, \mathbf{P}_2)$ are rectified by a homography pair $(\mathbf{H}_1, \mathbf{H}_2)$:

$$\mathbf{P}_i^* \simeq \mathbf{H}_i \mathbf{P}_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{q}_i \end{bmatrix} = \mathbf{H}_i \mathbf{K}_i \mathbf{R}_i \begin{bmatrix} \mathbf{I} & -\mathbf{C}_i \end{bmatrix}, \quad i = 1, 2$$

rectified entities: $\mathbf{F}^*$, l_1^*, l_2^*, etc:



- the rectified location difference $d = u_1^* - u_2^*$ is called disparity

corresponding epipolar lines must be:

1. parallel to image rows $\Rightarrow$ epipoles become $e_1^* = e_2^* = (1, 0, 0)$
2. equivalent $l_2^* = l_1^*$: $\quad \underline{\mathbf{l}}_1^* \simeq \underline{\mathbf{e}}_1^* \times \underline{\mathbf{m}}_1 = [\underline{\mathbf{e}}_1^*]_\times \underline{\mathbf{m}}_1 \simeq \underline{\mathbf{l}}_2^* \simeq \mathbf{F}^* \underline{\mathbf{m}}_1 \quad \Rightarrow \quad \mathbf{F}^* = [\underline{\mathbf{e}}_1^*]_\times$

- therefore the canonical fundamental matrix is

$$\mathbf{F}^* \simeq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

A two-step rectification procedure

1. find some pair of primitive rectification homographies $\hat{\mathbf{H}}_1$, $\hat{\mathbf{H}}_2$
2. upgrade to a pair of optimal rectification homographies while preserving $\mathbf{F}^*$

▶Primitive Rectification

Goal: Given fundamental matrix $\mathbf{F}$, derive some easy-to-obtain 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 $\mathbf{D} = \mathrm{diag}(1, d^2, 0)$, $\quad 1 \geq d^2 > 0$
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 →160)

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -d & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & d & 0 \end{bmatrix}$$

3. Then

$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^\top = \underbrace{\mathbf{U}\mathbf{A}^\top}_{\hat{\mathbf{H}}_2^\top} \mathbf{F}^* \underbrace{\mathbf{B}\mathbf{V}^\top}_{\hat{\mathbf{H}}_1} = \hat{\mathbf{H}}_2^\top \mathbf{F}^* \hat{\mathbf{H}}_1 \qquad \hat{\mathbf{H}}_1, \hat{\mathbf{H}}_2 \text{ orthogonal}$$

and the primitive rectification homographies are

$$\hat{\mathbf{H}}_2 = \mathbf{A}\mathbf{U}^\top, \qquad \hat{\mathbf{H}}_1 = \mathbf{B}\mathbf{V}^\top$$

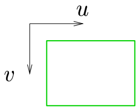
⊛ P1; 1pt: derive some other admissible $\mathbf{A}$, $\mathbf{B}$

- **Hence:** Rectification homographies do exist →160
- there are other primitive rectification homographies, these suggested are just easy to obtain

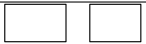
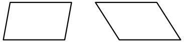
▶The Set of All Rectification Homographies

Proposition 1 Homographies $\mathbf{A}_1$ and $\mathbf{A}_2$ are rectification-preserving if the images stay rectified, i.e. if $\mathbf{A}_2^{-\top}\,\mathbf{F}^*\,\mathbf{A}_1^{-1} \simeq \mathbf{F}^*$, which gives

$$\mathbf{A}_1 = \begin{bmatrix} l_1 & l_2 & l_3 \\ 0 & s_v & t_v \\ 0 & q & 1 \end{bmatrix}, \qquad \mathbf{A}_2 = \begin{bmatrix} r_1 & r_2 & r_3 \\ 0 & s_v & t_v \\ 0 & q & 1 \end{bmatrix}, \tag{36}$$



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 9 free parameters.

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, where $\mathbf{A}_1$, $\mathbf{A}_2$ are as in (36).

Proposition 2 Pairs of rectification-preserving homographies $(\mathbf{A}_1, \mathbf{A}_2)$ form a group, with group operation (composition) $(\mathbf{A}'_1, \mathbf{A}'_2) \circ (\mathbf{A}_1, \mathbf{A}_2) = (\mathbf{A}'_1 \mathbf{A}_1, \mathbf{A}'_2 \mathbf{A}_2)$.

Proof:

- closure by Proposition 1
- associativity by matrix multiplication
- identity belongs to the set
- inverse element belongs to the set by $\mathbf{A}_2^\top \mathbf{F}^* \mathbf{A}_1 \simeq \mathbf{F}^* \Leftrightarrow \mathbf{F}^* \simeq \mathbf{A}_2^{-\top} \mathbf{F}^* \mathbf{A}_1^{-1}$

▶Primitive Rectification Suffices for Calibrated Cameras

Obs: calibrated cameras: $d = 1 \Rightarrow \hat{\mathbf{H}}_1$, $\hat{\mathbf{H}}_2$ (→161) are orthonormal

1. determine primitive rectification homographies $(\hat{\mathbf{H}}_1, \hat{\mathbf{H}}_2)$ from the essential matrix
2. choose a suitable common calibration matrix $\mathbf{K}$, e.g. from $\mathbf{K}_1$, $\mathbf{K}_2$:

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad f = \frac{1}{2}(f^1 + f^2), \quad u_0 = \frac{1}{2}(u_0^1 + u_0^2), \quad \text{etc.}$$

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 (→165) 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 $\mathbf{F}$

$$\mathbf{H}_1\mathbf{P}_1 = \mathbf{K}\hat{\mathbf{H}}_1\mathbf{K}_1^{-1}\mathbf{P}_1 = \mathbf{K}\underbrace{\mathbf{B}\mathbf{V}^\top\mathbf{R}_1}_{\mathbf{R}^*}\begin{bmatrix} \mathbf{I} & -\mathbf{C}_1 \end{bmatrix} = \mathbf{K}\mathbf{R}^*\begin{bmatrix} \mathbf{I} & -\mathbf{C}_1 \end{bmatrix}$$

$\mathbf{A}$, $\mathbf{B}$ from →161

$$\mathbf{H}_2\mathbf{P}_2 = \mathbf{K}\hat{\mathbf{H}}_2\mathbf{K}_2^{-1}\mathbf{P}_2 = \mathbf{K}\underbrace{\mathbf{A}\mathbf{U}^\top\mathbf{R}_2}_{\mathbf{R}^*}\begin{bmatrix} \mathbf{I} & -\mathbf{C}_2 \end{bmatrix} = \mathbf{K}\mathbf{R}^*\begin{bmatrix} \mathbf{I} & -\mathbf{C}_2 \end{bmatrix}$$

- 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 (15)
- Note that points at infinity project by $\mathbf{K}\mathbf{R}^*$ in both cameras ⇒ they have zero disparity (→168), hence. . .

▶Geometric Interpretation of Linear Rectification

What pair of physical cameras is compatible with $\mathbf{F}^*$?

- we know that $\mathbf{F} = (\mathbf{Q}_1 \mathbf{Q}_2^{-1})^\top [\underline{\mathbf{e}}_1]_\times$ →80
- we choose $\mathbf{Q}_1^* = \mathbf{K}_1^*$, $\mathbf{Q}_2^* = \mathbf{K}_2^* \mathbf{R}^*$; then

$$\mathbf{F}^* \simeq (\mathbf{Q}_1^* \mathbf{Q}_2^{*-1})^\top [\underline{\mathbf{e}}_1^*]_\times \overset{!}{\simeq} (\mathbf{K}_1^* \mathbf{R}^{*\top} \mathbf{K}_2^{*-1})^\top \mathbf{F}^*$$

- we look for $\mathbf{R}^*$, $\mathbf{K}_1^*$, $\mathbf{K}_2^*$ compatible with equations

$$(\mathbf{K}_1^* \mathbf{R}^{*\top} \mathbf{K}_2^{*-1})^\top \mathbf{F}^* = \lambda \mathbf{F}^*, \qquad \mathbf{R}^* \mathbf{R}^{*\top} = \mathbf{I}, \qquad \mathbf{K}_1^*, \mathbf{K}_2^* \text{ upper triangular}$$

- we also want $\mathbf{b}^*$ from $\underline{\mathbf{e}}_1^* \simeq \mathbf{P}_1^* \underline{\mathbf{C}}_2^* = \mathbf{K}_1^* \mathbf{b}^*$ $\quad$ $\mathbf{b}^*$ in camera-1 frame
- result after equations reduction:

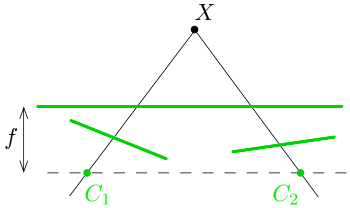
$$\mathbf{R}^* = \mathbf{I}, \quad \mathbf{b}^* = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{K}_1^* = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}_2^* = \begin{bmatrix} k_{21} & k_{22} & k_{23} \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{37}$$

- rectified cameras are in canonical relative pose $\quad$ not rotated, canonical baseline
- rectified calibration matrices can differ in the first row only
- if $\mathbf{K}_1^* = \mathbf{K}_2^*$, the rectified pair is called the standard stereo pair and we have the 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

►Summary & Remarks: Linear Rectification



…It follows: Standard rectification homographies reproject onto a common image plane parallel to the baseline

- rectification is done with a pair of homographies (one per image) →159
  - ⇒ projection centers of rectified cameras 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 →165
  - ⇒ rectified image projection planes are coplanar
- equal rectified calibration matrices give standard rectification →165
  - ⇒ rectified image projection planes are equal
- primitive rectification is already standard in calibrated cameras →164

- 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 (37), from $\mathbf{K}_1 \underline{\mathbf{X}}_\infty \simeq \mathbf{K}_2 \underline{\mathbf{X}}_\infty$

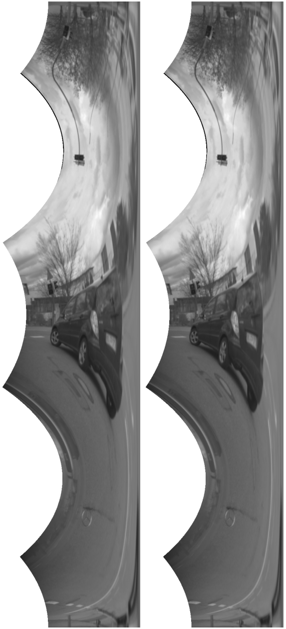
Optimal choice for the free parameters in $\mathbf{H}_{1,2}$

- by minimization of residual image distortion, eg. [Gluckman & Nayar 2001]

$$\mathbf{A}_i^* = \arg\min_{\mathbf{A}_i} \iint_\Omega \left(\det J\big((A_i \circ H_i)(\mathbf{x})\big) - 1\right)^2 d\mathbf{x}\,, \quad i = 1,2$$

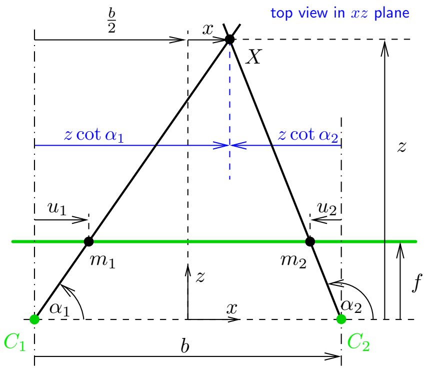
- by minimization of image information loss [Matoušek, ICIG 2004]
- non-linear rectification — suitable for forward motion
  non-parametric: [Pollefeys et al. 1999]
  analytic: [Geyer & Daniilidis 2003]

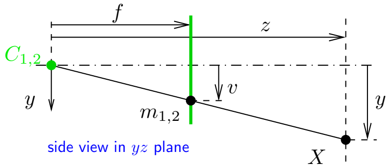
forward egomotion



rectified images, Pollefeys' method

►Trivializing Epipolar Geometry: Binocular Disparity in a Standard Stereo Pair





- Assumptions: single image line, standard camera pair

$$b = z\cot\alpha_1 - z\cot\alpha_2 \qquad b = \frac{b}{2} + x - z\cot\alpha_2$$
$$u_1 = f\cot\alpha_1 \qquad u_2 = f\cot\alpha_2$$

- eliminate α_1, α_2 and obtain:
  $X = (x, y, z)$ from **disparity** $d = u_1 - u_2$:

$$z = \frac{b\,f}{d}, \quad x = \frac{b}{d}\,\frac{u_1+u_2}{2}, \quad y = \frac{b\,v}{d}$$

f, d, u, v in pixels, b, x, y, z in meters

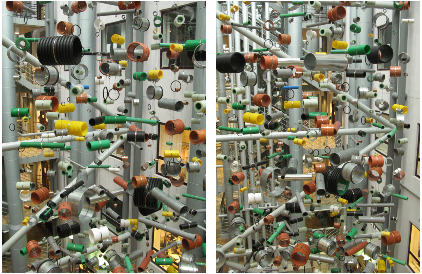
Observations

- 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, hence reduces the error

How Difficult Is Stereo?



Centrum för teknikstudier at Malmö Högskola, Sweden

The Vyšehrad Fortress, Prague

- top: easy interpretation from even a single image
- bottom left: we have no help from image interpretation
- bottom right: ambiguous interpretation due to a combination of missing texture and occlusion

A Summary of Our Observations and an Outlook

1. simple matching algorithms do not work
   - the success of a model-free stereo matching is unlikely →158
   - without scene recognition or use high-level constraints the problem seems difficult
2. stereopsis requires image interpretation in sufficiently complex scenes — or another-modality measurement

we have a tradeoff: model strength ↔ universality

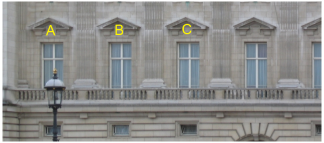
Outlook:

1. represent the occlusion constraint: — correspondences are not independent due to occlusions
   - disparity in rectified images
   - uniqueness as an occlusion constraint
2. represent piecewise continuity — the weakest of interpretations; piecewise: object boundaries
   - ordering as a weak continuity model
3. use a consistent framework
   - finding the most probable solution (MAP)

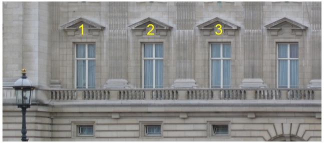
Structural Ambiguity in Stereovision

- 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)

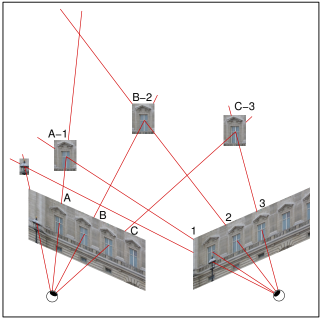


left image

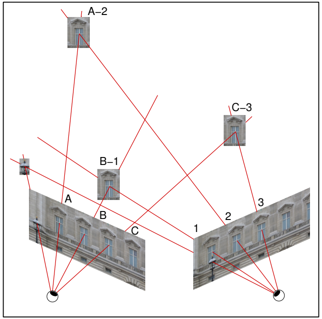


right image

- Illustration of the problem
- Keypoints: Window detections



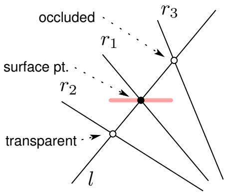
matching/interpretation 1

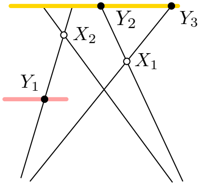


interpretation 2

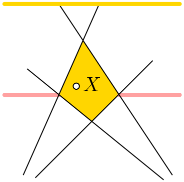
- Repetitive keypoints $\Rightarrow$ non-unique matching
- Cameras are not canonical; constant-depth surface is not a plane

▶Understanding Basic Occlusion Types





half occlusion



mutual occlusion

mutual occlusion in the wall hole

- 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) and (l, r_2) are <u>excluded</u> by (l, r_1)

- in half-occlusion, every 3D point such as X_1 or X_2 is excluded by a binocularly visible surface point such as Y_1, Y_2, Y_3

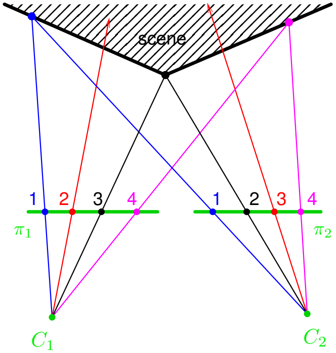
  $\Rightarrow$ decisions on correspondences <u>are not independent</u>

- in mutual occlusion this is no longer the case: any X in the yellow zone above is <u>not excluded</u>

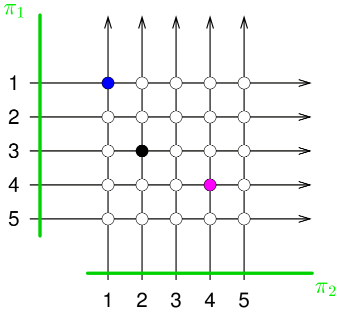
  $\Rightarrow$ decisions inside the zone <u>are independent</u> on the rest

▶Matching Table

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

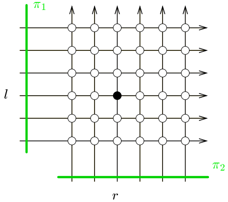
- rows and columns represent optical rays
- nodes: possible correspondence pairs
- full nodes: matches
- numerical values associated with nodes: descriptor similarities

see next

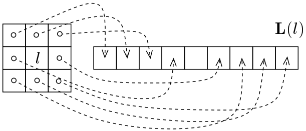
►Constructing An Image Similarity Cost

- 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 ↦ 'a set of random-variable samples':



- a simple block (dis-)similarity is $\mathrm{SAD}(l, r) = \|\mathbf{L}(l) - \mathbf{R}(r)\|_1$ L_1 metric (sum of absolute differences; smaller is better)
- a scaled-descriptor (dis-)similarity is $\mathrm{sim}(l, r) = \dfrac{\|\mathbf{L}(l) - \mathbf{R}(r)\|^2}{\sigma_I^2(l, r)}$ smaller is better
- σ_I^2 – the difference scale; 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

$$\mathrm{sim}(l, r) = 1 - \underbrace{\frac{2\,\mathrm{cov}\big(\mathbf{L}(l), \mathbf{R}(r)\big)}{\mathrm{var}\big(\mathbf{L}(l)\big) + \mathrm{var}\big(\mathbf{R}(r)\big)}}_{\rho(\mathbf{L}(l), \mathbf{R}(r))} \tag{38}$$

$\mathrm{var}(\cdot)$, $\mathrm{cov}(\cdot)$ is sample (co-)variance, not invariant to scale difference

- ρ – MNCC – Moravec's Normalized Cross-Correlation similarity   bigger is better   [Moravec 1977]

$$\rho^2 \in [0, 1], \qquad \mathrm{sign}\, \rho \sim \text{'phase'}$$

- another successful (dis-)similarity is the Hamming Distance over the Census Transform   related to local binary patterns

Census Transform (CT)

- CT: Per-pixel binarization, given reference value (e.g the window center)
- For a grayscale image:

$$\varkappa(x_{ij};\, r) = \begin{cases} 0 & x_{ij} \leq r \\ 1 & x_{ij} > r \end{cases}$$

189	235	181
217	185	228
231	61	254

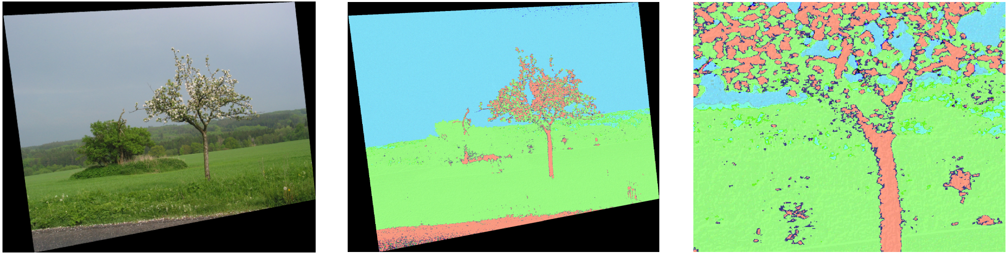
input window

$\xrightarrow{\varkappa(\cdot;\,185)}$

1	1	0
1	0	1
1	0	1

bits

= 483 (new value for the central pixel)



input image

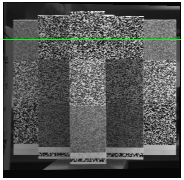
$\varkappa$: RGB CT, $3 \times 2 = 6$-bit per pixel, 3×3 window $=$ 48 bit/px

- preserves sharp boundaries
- may or may not use windowing (cost aggregation)

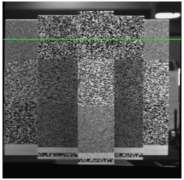
How A Scene Looks in The Filled-In Matching Table

object

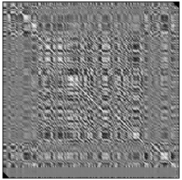
left image



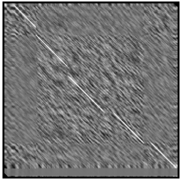
right image



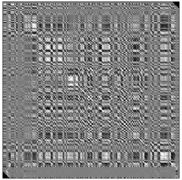
5×5 window

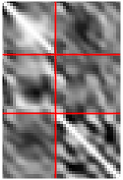


11×11 window

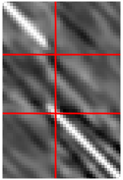


3×3 window

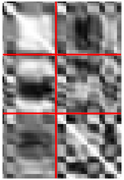




a good tradeoff



occlusion artefacts



undiscriminable

- MNCC ρ used ($\alpha = 1.5$, $\beta = 1$) →182
- high-similarity structures correspond to scene objects

Things to notice:

constant disparity

- a diagonal in the matching table
- zero disparity is the main diagonal

  assuming standard stereopair

depth discontinuity

- horizontal or vertical jump in matching table

large image window

- similarity values have better discriminability
- worse occlusion localization

repeated texture

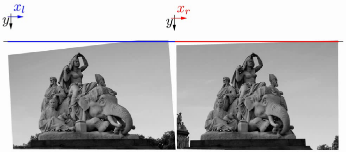
- horizontal and vertical block repetition

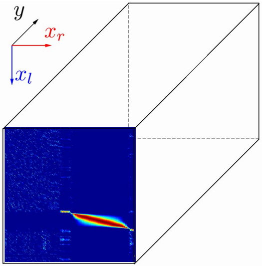
Image Point Descriptors And Their Similarity

Descriptors: Image points are tagged by their (viewpoint-invariant) physical properties:

- texture window [Moravec 77]
- Census Transform [Zabih & Woodfill 94]
- a descriptor like DAISY [Tola et al. 2010]
- learned descriptors
- reflectance profile under a moving illuminant
- (pixelwise) photometric ratios [Wolff & Angelopoulou 93-94]
- dual photometric stereo [Ikeuchi 87]
- (pixelwise) polarization signature
- ...

- similar points are more likely to match
- image similarity values for all 'match candidates' give the 3D matching table — also called: 'disparity volume'



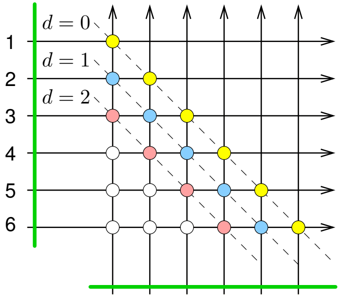


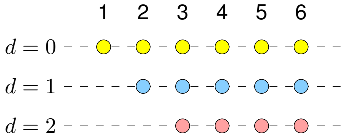
click for video

▶Marroquin's Winner Take All (WTA) Matching Algorithm

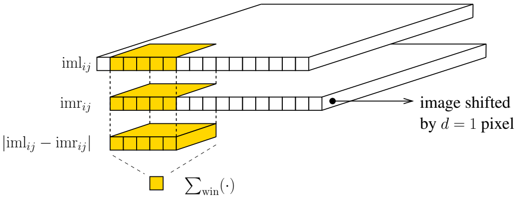
Alg: Per left-image pixel: The most SAD-similar pixel along the right epipolar line →174

1. select disparity range — this is a critical weak point
2. represent the matching table diagonals in a compact form





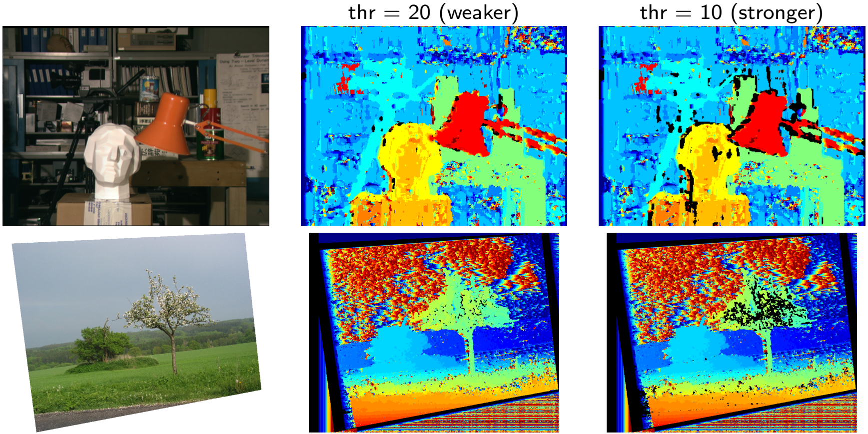
3. use the 'image sliding & cost aggregation algorithm'
4. take the maximum over disparities d
5. threshold results by the maximal allowed SAD dissimilarity (or minimal MNCC similarity)



A Matlab Code for WTA

```
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
```

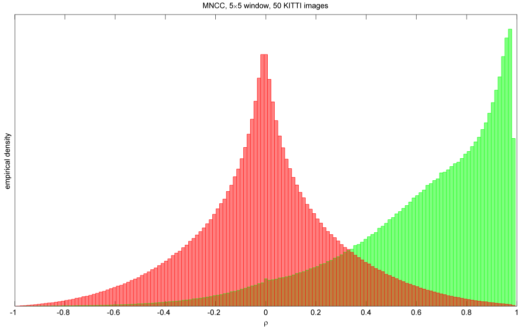
WTA: Some Results



- 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 →172)

▶A Principled Approach to Similarity

Empirical Distribution of MNCC ρ for Matches (green) and Non-Matches (red)



- histograms of ρ computed from 5×5 correlation window

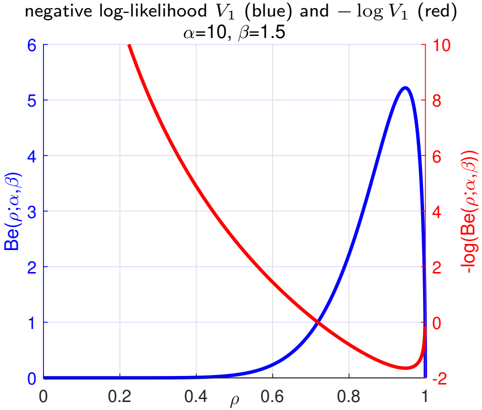
ρ: bigger is better

- 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 ρ
- matches (green) may have arbitrarily low ρ
- $\rho = 1$ is improbable for matches

Match Likelihood



- ρ is just a normalized measurement
- we need a probability distribution on $[0, 1]$
  e.g. the histogram or the Beta distribution:

$$p_1(\rho) = \frac{1}{B(\alpha, \beta)} |\rho|^{\alpha-1}(1 - |\rho|)^{\beta-1}$$

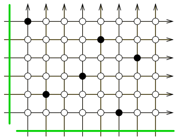
- note that uniform distribution is obtained for $\alpha = \beta = 1$
- when $\alpha = 2$ and $\beta = 1$ then $p_1(\cdot) = 2|\rho|$

- 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

$$V_1\big(\rho(l, r)\big) = -\log p_1\big(\rho(l, r)\big) \qquad \text{smaller is better} \tag{39}$$

- we should also define similarity (and negative log-likelihood $V_0(\rho(l, r))$) for non-matches

▶A Principled Approach to Matching: Formulating 'What We Want'



- given matching M in table T, what is the likelihood of observed data D?
- data – all cost pairs (V_0, V_1) in the matching table T
- matches – pairs $p_i = (l_i, r_i) \in M \subset T, \quad i = 1, \ldots, n$
- 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) constant number of variables in T

$$V(D \mid M, T) = \sum_{p \in M} V_1(D \mid p) + \sum_{p \in T \setminus M} V_0(D \mid p)$$

$V_1(D \mid p)$ – negative log-probability of data D at matched pixel p (39)
$V_0(D \mid p)$ – ditto at unmatched pixel p →181 and →182

- matching problem

$$M^* = \arg \min_{M \in \mathcal{M}(T)} V(D \mid M, T)$$

$\mathcal{M}(T)$ – the set of all matchings in table T

- symmetric: formulated over pairs, invariant to left ↔ right image swap unlike in WTA

▸(cont'd) Log-Likelihood Ratio

- we need to reduce the matching to a standard polynomial-complexity problem

1. convert the matching cost to an 'easier' sum

$$V(D \mid M, T) = \sum_{p \in M} V_1(D \mid p) + \sum_{p \in T \setminus M} V_0(D \mid p) \quad + \overbrace{\sum_{p \in M} V_0(D \mid p) - \sum_{p \in M} V_0(D \mid p)}^{0}$$

$$= \underbrace{\sum_{p \in M} \Big(V_1(D \mid p) - V_0(D \mid p) \Big)}_{-L(D \mid p)} + \underbrace{\sum_{p \in T \setminus M} V_0(D \mid p) + \sum_{p \in M} V_0(D \mid p)}_{\sum_{p \in T} V_0(D \mid p) \,=\, \mathrm{const}}$$

2. hence

$$\arg \min_{M \in \mathcal{M}(T)} V(D \mid M) = \arg \max_{M \in \mathcal{M}(T)} \sum_{p \in M} L(D \mid p) \tag{40}$$

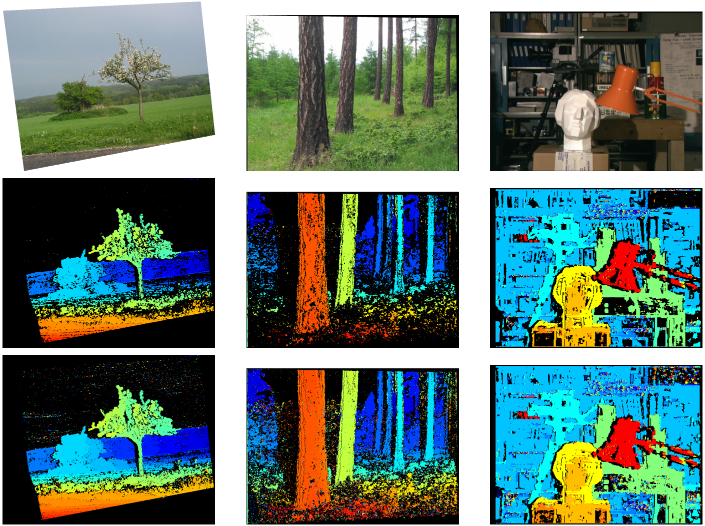
$L(D \mid p)$ – logarithm of matched-to-unmatched likelihood ratio (bigger is better)

why this way: we want to use maximum-likelihood on the entire T

3. (40) is max-cost matching (maximum assignment) for the maximum-likelihood (ML) matching problem
   - use the Hungarian (Munkres) algorithm and threshold the result with τ: $L(D \mid p) > \tau \geq 0$

   or approximate the problem by sacrificing symmetry and accuracy to speed and use dynamic programming

Some Results for the Maximum-Likelihood (ML) Matching



left image

3% / 61%

4.3% / 76%

- unlike the WTA we can efficiently control the density/accuracy tradeoff with τ — black = no match
- middle row: threshold τ for $L(D \mid p)$ set to achieve error rate of 3% (and 61% density results)
- bottom row: threshold τ set to achieve density of 76% (and 4.3% error rate results)

▶Basic Stereoscopic Matching Models

- 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)

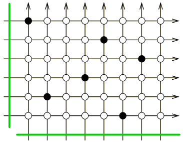
1. Uniqueness: Every image point matches at most once
   - excludes semi-transparent objects
   - used in the ML matching algorithm (but not in the WTA algorithm)

2. Monotonicity: Matched pixel ordering is preserved →189
   - for all $(i,j) \in M, (k,l) \in M, \quad k > i \Rightarrow l > j$

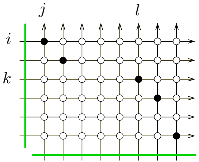
     Notation: $(i,j) \in M$ or $j = M(i)$ – left-image pixel i matches right-image pixel j
   - excludes thin objects close to the cameras
   - used in 3-Label Dynamic Programming (3LDP) [SP]

3. Coherence: 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]

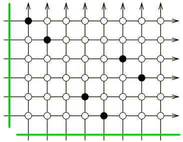
4. (Piecewise) binocular continuity: The scene images continuously w/o self-occlusions
   - disparities do not differ much in neighboring pixels (except at object boundaries)
   - full binocular continuity too strong, except in some applications
   - piecewise binocular continuity is combined with monotonicity in 3LDP



incoherent



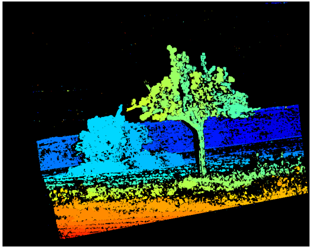
monotonic coherent



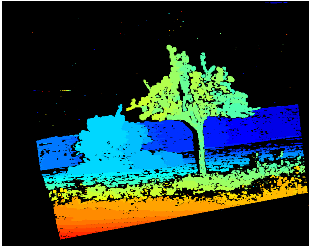
non-monotonic coherent

left image

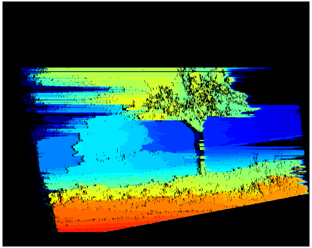
right image



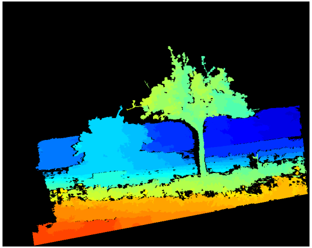
ML →184



3LDP with ordering
[SP]



naïve DP
[Cox et al. 1992]

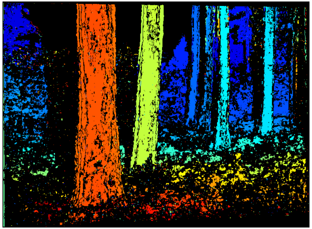


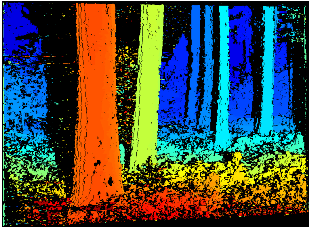
Stable Segmented 3LDP
[Aksoy et al. PRRS 2008]

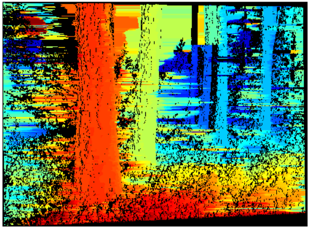
- 3LDP parameters α_i, V_e learned on Middlebury stereo data http://vision.middlebury.edu/stereo/

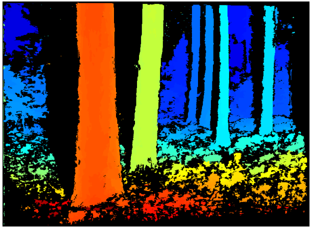

left image


right image


ML →184

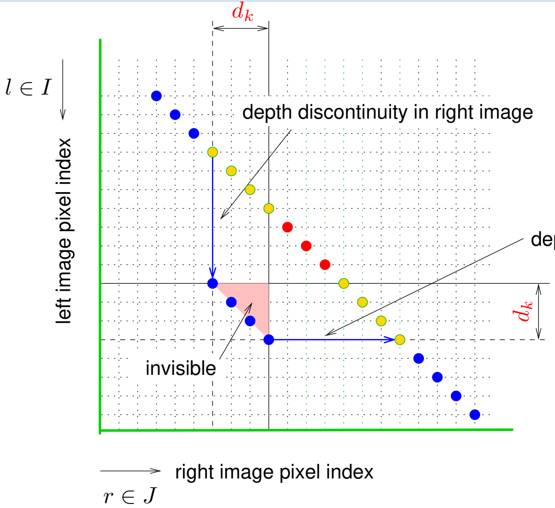

3LDP w/ordering [SP]


naïve DP


Stable Segmented 3LDP

- naïve DP: no mutual occlusion model, ignores symmetry, has no similarity distribution model, ignores $T \setminus M$
- but even 3LDP has errors in mutually occluded region
- Stable Segmented 3LDP: few errors in mutually occluded region since it uses a coherence model

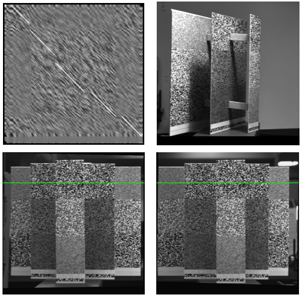
Binocular Discontinuities in Matching Table



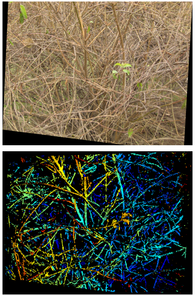
- binocularly visible foreground points
- binocularly visible background pts violating ordering
- monocularly visible points (half-occluded in the other cam)

d_k critical disparity

- this leads to the concept of 'forbidden zone'

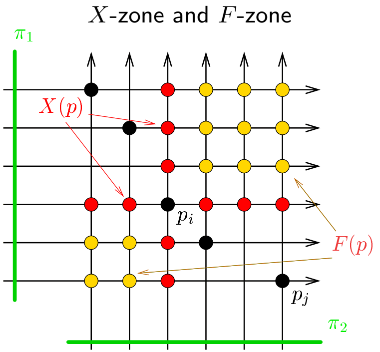


ordering holds



no ordering; Alg: GCS

Formally: Uniqueness and Ordering in Matching Table T



$p_j \notin X(p_i), \quad p_j \notin F(p_i)$

- **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 maximal monotonic matchings? (e.g. 27 for $n = 4$; hard!)

- uniqueness constraint is a basic occlusion model
- ordering constraint is a weak continuity model — and partly also an occlusion model
- monotonic matchings can be found by **dynamic programming**

Algorithm Comparison

Marroquin's Winner-Take-All (WTA →178)

- the ur-algorithm — very weak model
- dense disparity map
- $O(N^3)$ algorithm, simple but it rarely works

Maximum Likelihood Matching (ML →184)

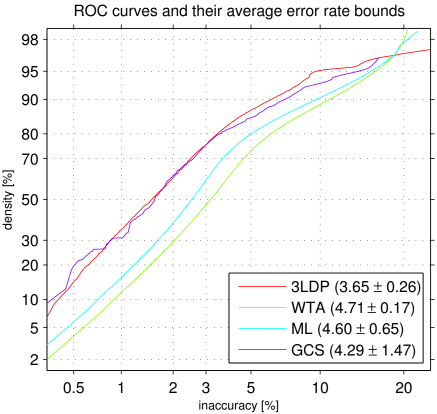
- semi-dense disparity map
- many small isolated errors
- models basic occlusion
- $O(N^3 \log(NV))$ algorithm — max-flow by cost scaling

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

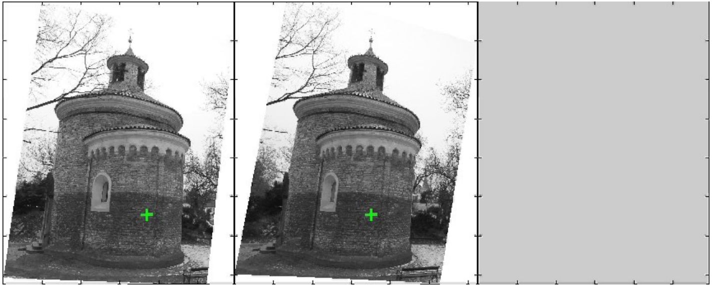


- ROC-like curve captures the density/accuracy tradeoff
- numbers: AUC (smaller is better)
- GCS is the one used in the exercises
- more algorithms at http://vision.middlebury.edu/stereo/ (good luck!)

GCS: Growing Correspondence Seeds

Alg: [Cech & Sara, BenCOS 2007]

1. Grow seed correspondences until they violate uniqueness severely — by a X-zone test
2. Select final unique matches by match competition in the X/FX-zone — by the stable matching algorithm



click for video

- explores only the "promising" regions in disparity space
- does not need "good" seeds because the competition revises them
- requires good-accuracy epipolar rectification — as all the algorithms mentioned do

Module IX

Additional Topics

9.1 Real Camera with Radial Distortion

covered by

[H&Z] Sec 7.4

Real Camera with Radial Distortion

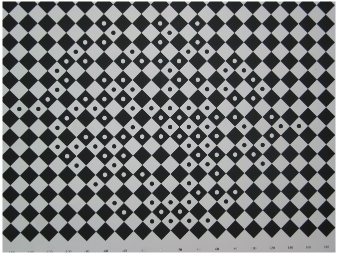
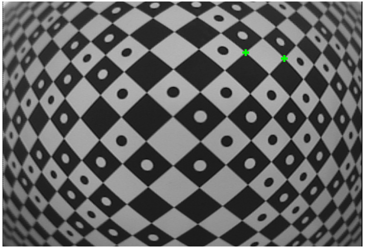


image with no radial distortion



an extreme case of barrel radial distortion

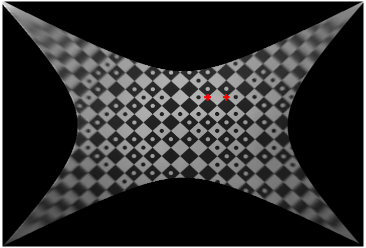
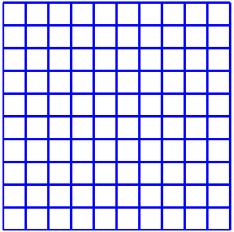
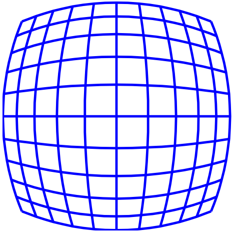


image undistorted by division model

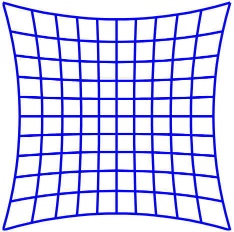
distortion types



none ($\lambda = 0$)

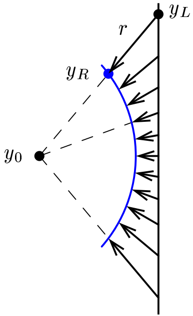


barrel ($\lambda = 0.3$)



pincushion ($\lambda = -0.3$)

The Radial Distortion Mapping



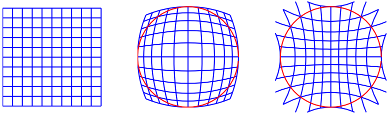
- everything is happening in the image plane

  y_0 – center of radial distortion (usually the principal point)

  y_L – linearly projected point (unknown)

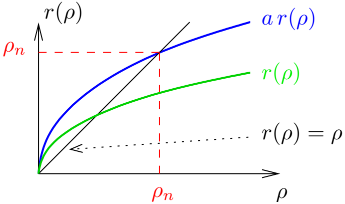
  y_R – radially distorted point (known)

- radial distortion r maps y_L to y_R along the radial direction
- magnitude of the transfer depends on the radius $\rho = \|y_R - y_0\|$ only

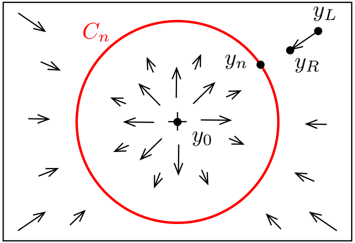


- circles centered at y_0 map to centered circles, lines incident on y_0 map on themselves
- the mapping $r()$ can be scaled to $a\, r()$ so that a particular circle C_n of radius ρ_n does not scale

distortion	inside C_n	outside C_n
barrel	expanding	contracting
pincushion	contracting	expanding

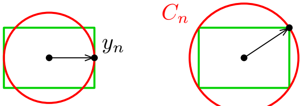


Radial Distortion Models



barrel distortion
arrows represent $\mathbf{z}_R - r(\mathbf{z}_L)$

- let $\mathbf{z} = \mathbf{y} - \mathbf{y}_0$ non-homogeneous
- we have $\mathbf{z}_R = r(\mathbf{z}_L)$ $\mathbf{z}_L$ – linear, $\mathbf{z}_R$ – distorted
- but we are often interested in $\mathbf{y}_L = r^{-1}(\mathbf{y}_R)$
- $\mathbf{y}_n$ – a no-distortion point on C_n: $r(\mathbf{y}_n) = \mathbf{y}_n$
- $\mathbf{z}_n = \mathbf{y}_n - \mathbf{y}_0$
- $\mathbf{y}_n$: a boundary point that preserves image content within the image size



in pincushion in barrel

Division Model

$$\mathbf{z}_L = \frac{1-\lambda}{1-\lambda\frac{\|\mathbf{z}_R\|^2}{\|\mathbf{z}_n\|^2}}\,\mathbf{z}_R \quad \text{and} \quad \mathbf{z}_R = \frac{\hat{\mathbf{z}}}{1+\sqrt{1+\lambda\frac{\|\hat{\mathbf{z}}\|^2}{\|\mathbf{z}_n\|^2}}}, \quad \text{where } \hat{\mathbf{z}} = \frac{2\,\mathbf{z}_L}{1-\lambda}$$

- single parameter $-1 \le \lambda < 1$: $\lambda > 0$ – barrel distortion, $\lambda < 0$ – pincushion distortion
- has a closed-form inverse
- models even some fish-eye lenses

Polynomial Model

$$\mathbf{z}_L = \frac{D(\mathbf{z}_R; \mathbf{z}_n, \mathbf{k})}{1 + \sum_{i=1}^{n} k_i}\, \mathbf{z}_R\,,$$

$$D(\mathbf{z}_R; \mathbf{z}_n, \mathbf{k}) = 1 + k_1 \rho^2 + k_2 \rho^4 + \cdots + k_n \rho^{2n}, \quad \rho = \frac{\|\mathbf{z}_R\|}{\|\mathbf{z}_n\|}, \quad \mathbf{k} = (k_{1:n})$$

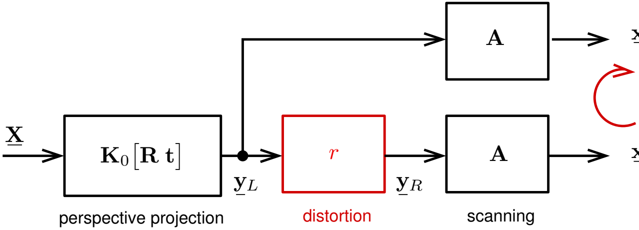
- e.g. $k_i \geq 0$ – barrel distortion, $k_i \leq 0$ – pincusion distortion, $i = 1, \ldots, n$
- typically $n = 3$
- no closed-form inverse
- may loose monotonicity without requiring equal signs in all k_i — the undistorted image may then fold over itself
- hard to calibrate — higher coefficients tend to dominate

- Zernike orthogonal polynomials R_i^0 are a better choice

$$R_2^0(\rho) = 2\rho^2 - 1, \quad R_4^0(\rho) = 6\rho^4 - 6\rho^2 + 1, \quad R_6^0(\rho) = 20\rho^6 - 30\rho^4 + 12\rho^2 - 1, \cdots$$

- then $D(\mathbf{z}_R; \mathbf{z}_n, \mathbf{k}) = 1 + k_1\, R_2^0(\rho) + k_2\, R_4^0(\rho) + \cdots + k_n\, R_{2n}^0(\rho)$
- must know the field of view of the lens in the image plane; since ρ must satisfy $0 \leq \rho \leq 1$
- coefficients k_i will typically decrease in magnitude with increasing i — unlike in the plain polynomial model

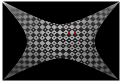
Real and Linear Camera Models

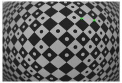


$\underline{\mathbf{x}}_L \simeq \mathbf{A}\mathbf{K}_0 \left[\mathbf{R}\ \mathbf{t}\right] \underline{\mathbf{X}}$ (linear camera)

undistortion: $\underline{\mathbf{x}}_L = \mathbf{A}\, r^{-1}(\mathbf{A}^{-1} \underline{\mathbf{x}}_R)$

$\underline{\mathbf{x}}_R \simeq \mathbf{A}\, r\Big(\mathbf{K}_0 \left[\mathbf{R}\ \mathbf{t}\right] \underline{\mathbf{X}}\Big)$ (real camera)





$$\mathbf{K}_0 = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ 'ideal' calibration matrix $$\mathbf{A}\mathbf{K}_0 = \begin{bmatrix} f & s\,f & u_0 \\ 0 & a\,f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & s & u_0 \\ 0 & a & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$ everything affecting radial distortion — center, skew, aspect ratio

r radial distortion function — including the conversion from/to homogeneous representation!

- assumption: the principal point and the center of radial distortion coincide
- f included in $\mathbf{K}_0$ to make radial distortion independent of focal length
- $\mathbf{A}$ makes radial lens distortion an elliptic image distortion
- it suffices in practice that r^{-1} is an analytic function (r need not be)

Calibrating Radial Distortion

- radial distortion calibration includes at least 5 parameters: $\theta = (\lambda,\, u_0,\, v_0,\, s,\, a)$
- we may asume ORUA: $s = 0,\ a = 1$

Alg:

1. detect a set of straight line segment images $\{s_i\}_{i=1}^{n}$ from a calibration target   checkerboard patterns have many
2. select a suitable set of k measurement points per segment   checkerboard: given, in general: how to select k?
3. define (rotation/translation-) invariant radial transfer error per measurement point $e_{i,j}$ in segment i:

$$e_i^2(\theta) = \sum_{j=1}^{k-2} e_{i,j}^2(\theta) \qquad \text{eg. line fit residual (closed form)}$$

4. minimize total radial transfer error <u>while preserving y_n</u>   to avoid collapse to a point

$$\arg \min_{\theta=(\lambda,\, u_0,\, v_0,\, s,\, a)} \sum_{i=1}^{n} e_i^2(\theta) \qquad \text{s.t.} \quad y_n \text{ preserved}$$

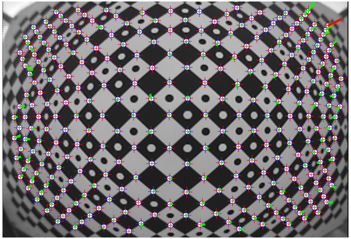
- line segments from real-world images requires segmentation to inliers/outliers   inliers = lines that are straight in reality
- marginalisation over the hidden inlier/outlier label gives a 'robust' error, e.g.   → Slide 121

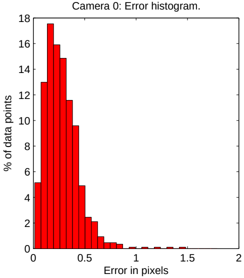
$$\varepsilon_i^2 = -\log\left(e^{-\frac{e_i^2}{2\sigma^2}} + t\right), \qquad t > 0$$

- direct optimization usually suffices but in general such optimization is unstable

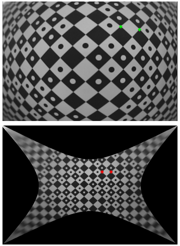
Example Calibrations

Low-resolution (VGA) wide field-of-view (130°) camera

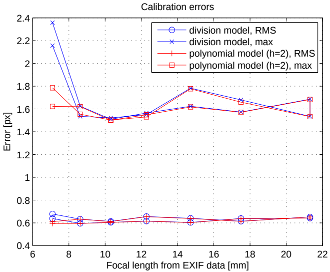


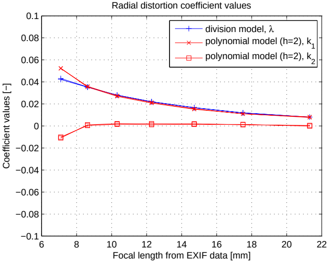


Cam	0
RMS [px]	0.33
max [px]	1.97
f [px]	94.59
a [-]	1.0951
u_0 [px]	243.26
v_0 [px]	353.37
(poly) k_1	$+0.8256$
k_2	-0.2261
k_3	$+1.2524$



4 Mpix consumer camera with a zoom





- polynomial model suffices for greater focal lengths
- above: alternating signs and similar-magnitude coefficients k_i are a sign of a low efficiency of the plain polynomial model
- below: radial distortion is slightly dependend on focal length

A Summary of This Course Highlights

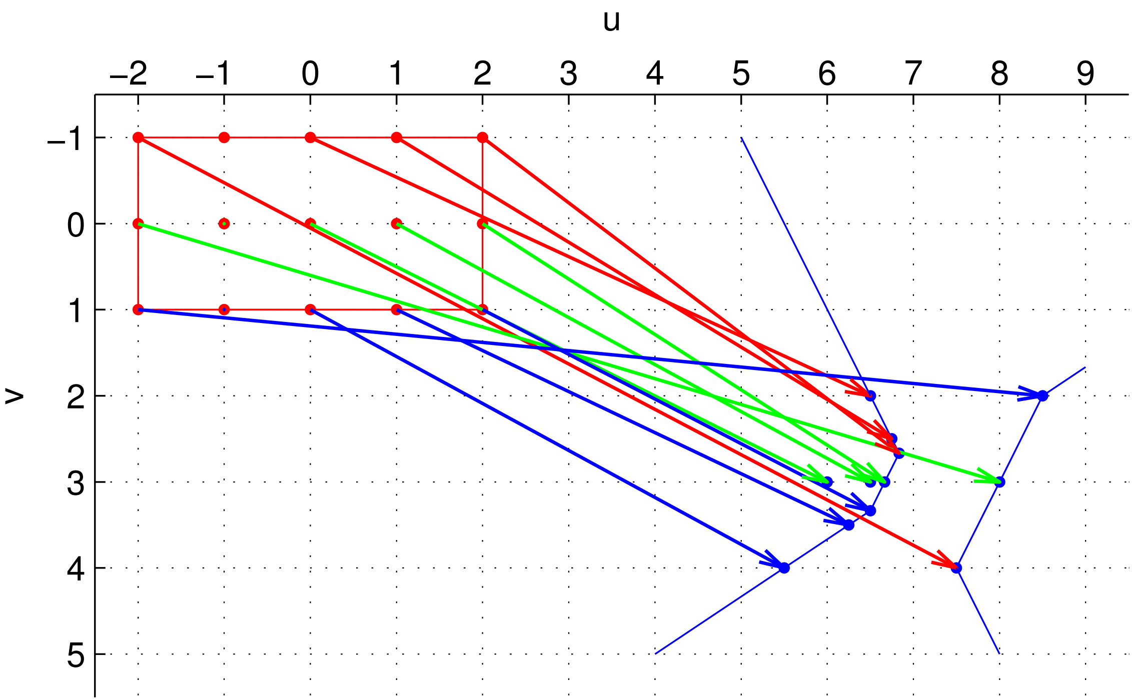
- homography as a two-image model
- epipolar geometry as a two-image model
- core algorithms for 3D vision:
  - simple intrinsic calibration methods
  - 6-pt alg for camera resection and 3-pt alg for exterior orientation (calibrated resection)
  - 7-pt alg for fundamental matrix, 5-pt alg for essential matrix
  - essential matrix decomposition to rotation and translation
  - efficient accurate triangulation
  - robust matching by RANSAC sampling
  - camera system reconstruction
  - efficient bundle adjustment
  - stereoscopic matching basics
- statistical robustness as a way to work with partially unknown information

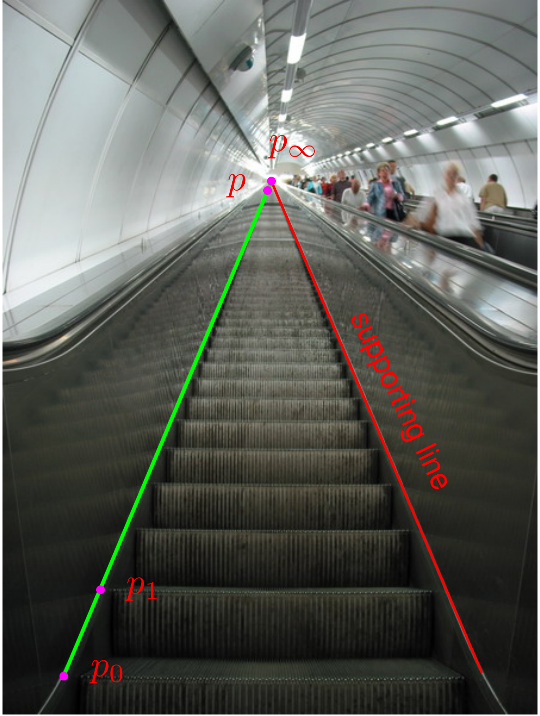
What can we do with these tools?

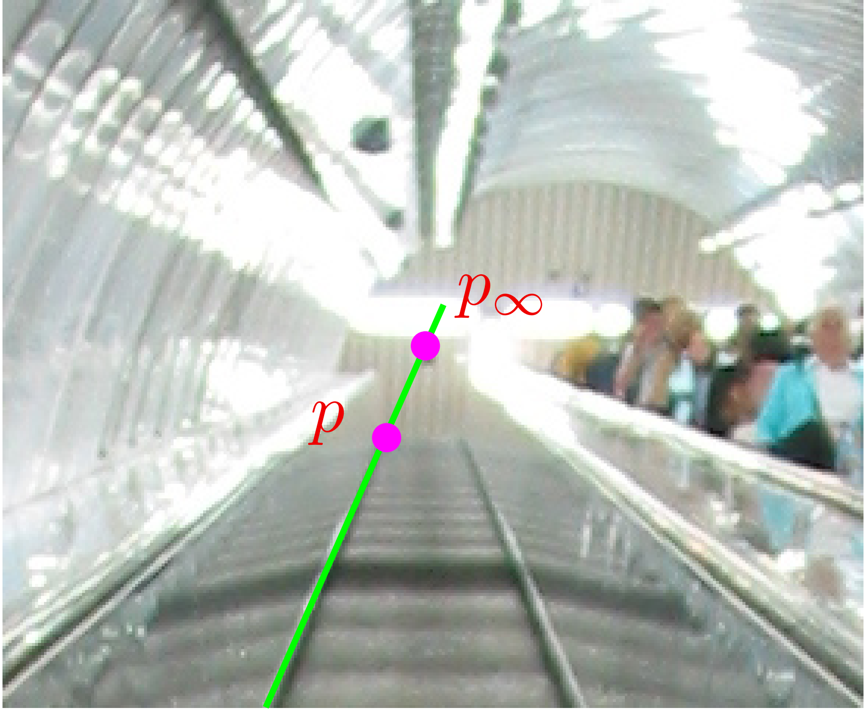
- perspective image rectification
- 3D scene reconstruction
- motion capture
- visual odometry
- robotic self-localization and mapping (SLAM) for navigation and motion planning

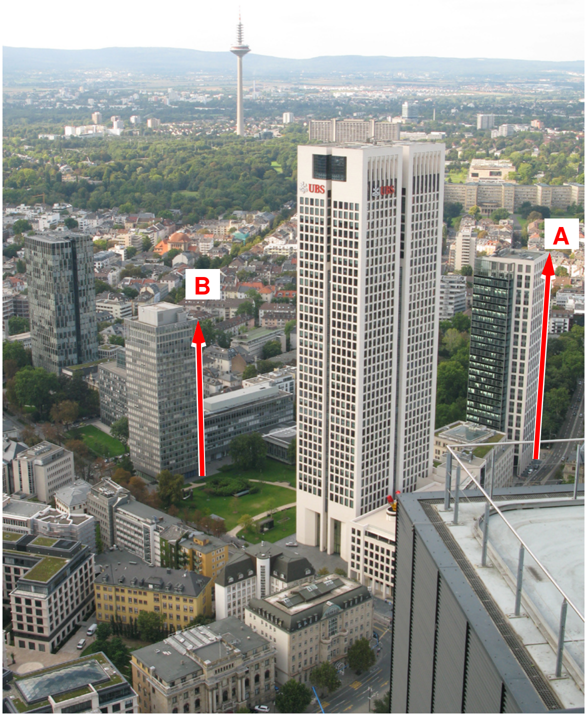
we did not cover 3D aggregation in scene maps

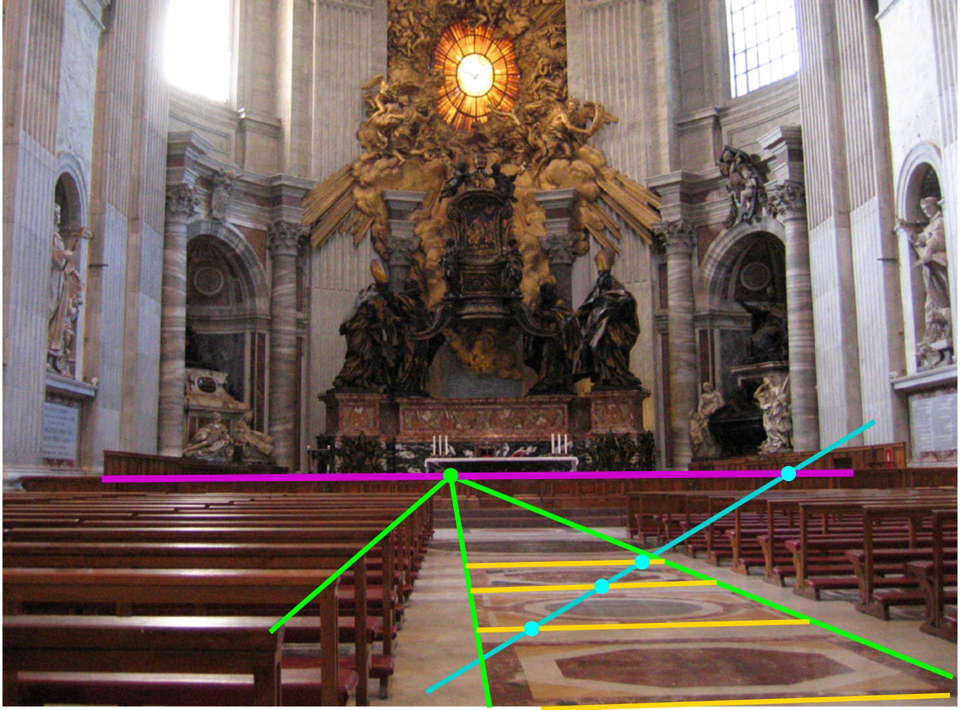
Thank You


u
−2
−1
0
1
2
3
4
5
6
7
8
9
v
−1
0
1
2
3
4
5


p_∞
p
supporting line
p_1
p_0

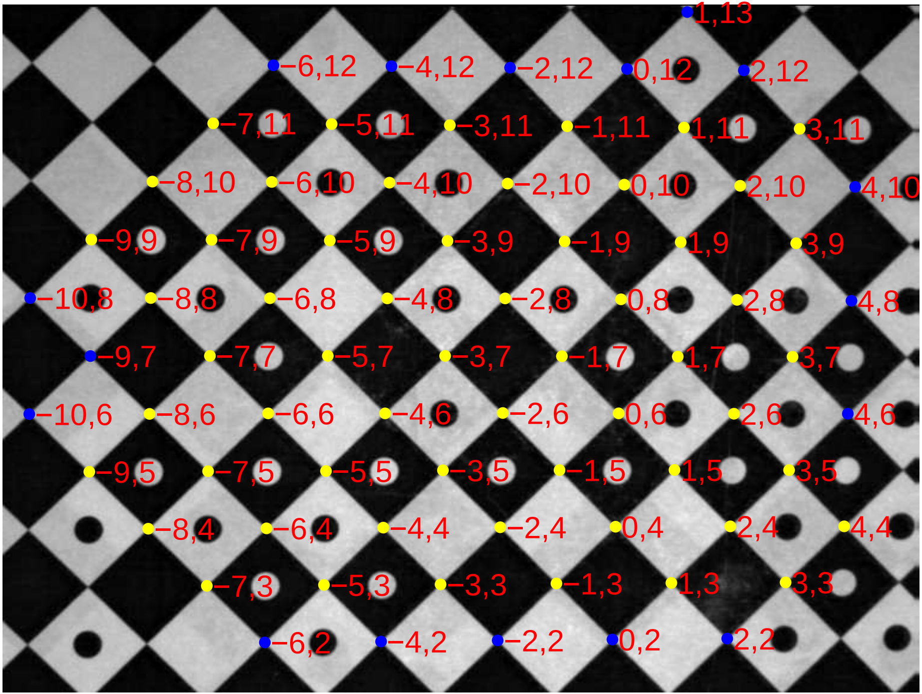

p_∞
p

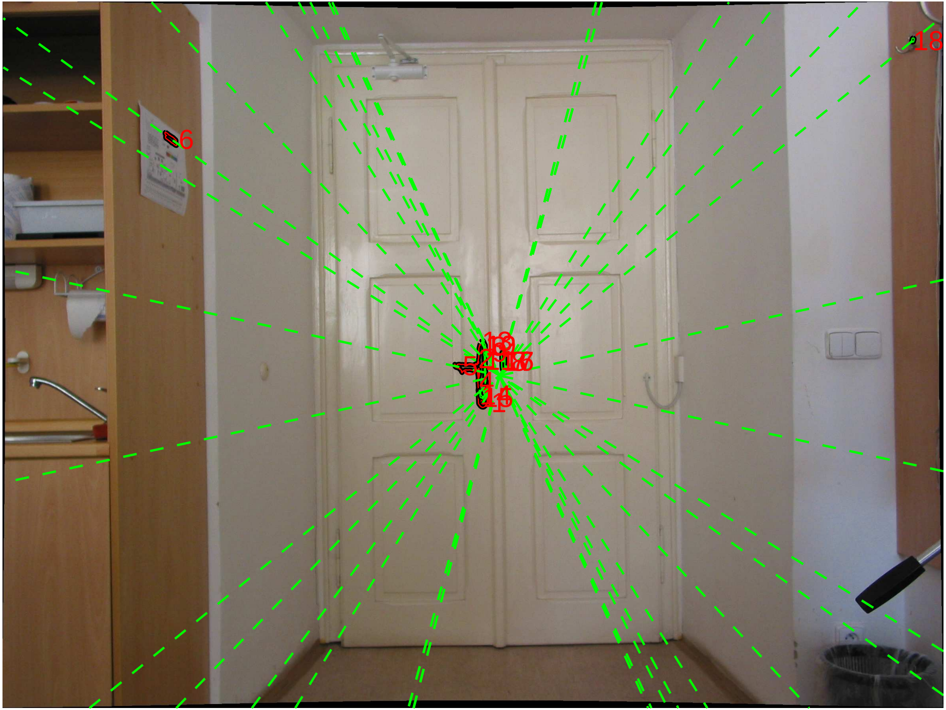

UBS
UBS
A
B



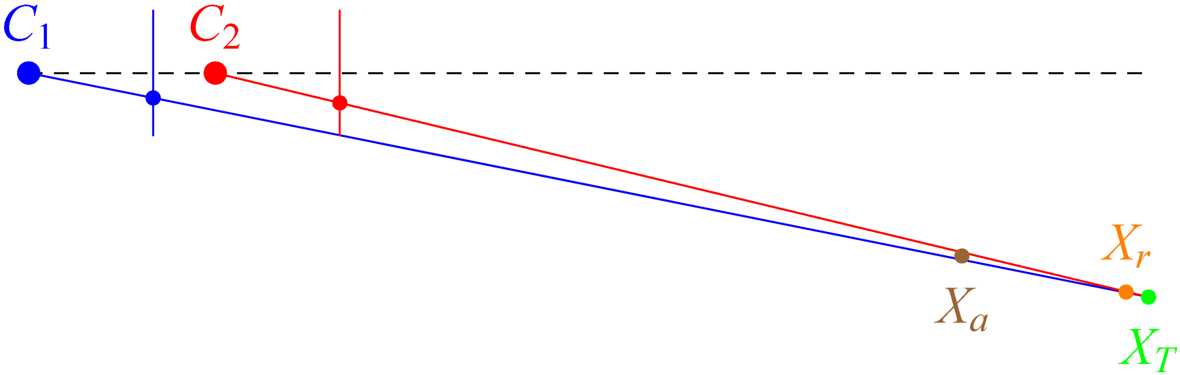

CLEMENS · XII · PONT · MAX · ANN · SAC · MDCCXXXVII · PONT · VIIII

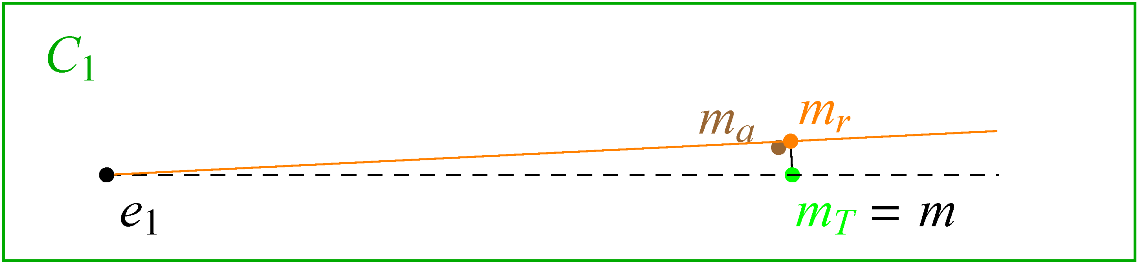

BOOKS
FILMS
TICKETS

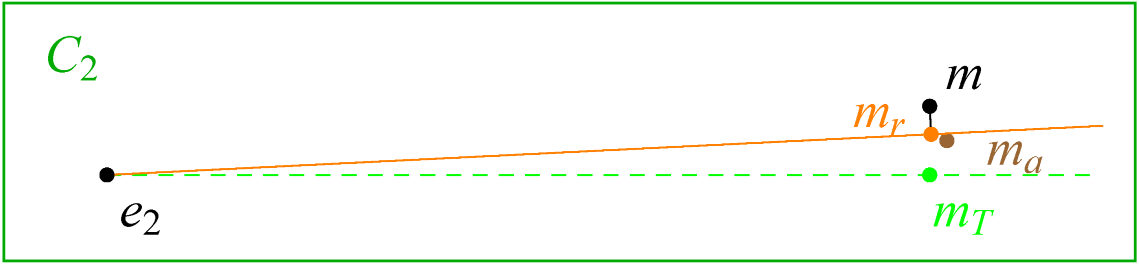

1,13
−6,12
−4,12
−2,12
0,12
2,12
−7,11
−5,11
−3,11
−1,11
1,11
3,11
−8,10
−6,10
−4,10
−2,10
0,10
2,10
4,10
−9,9
−7,9
−5,9
−3,9
−1,9
1,9
3,9
−10,8
−8,8
−6,8
−4,8
−2,8
0,8
2,8
4,8
−9,7
−7,7
−5,7
−3,7
−1,7
1,7
3,7
−10,6
−8,6
−6,6
−4,6
−2,6
0,6
2,6
4,6
−9,5
−7,5
−5,5
−3,5
−1,5
1,5
3,5
−8,4
−6,4
−4,4
−2,4
0,4
2,4
4,4
−7,3
−5,3
−3,3
−1,3
1,3
3,3
−6,2
−4,2
−2,2
0,2
2,2



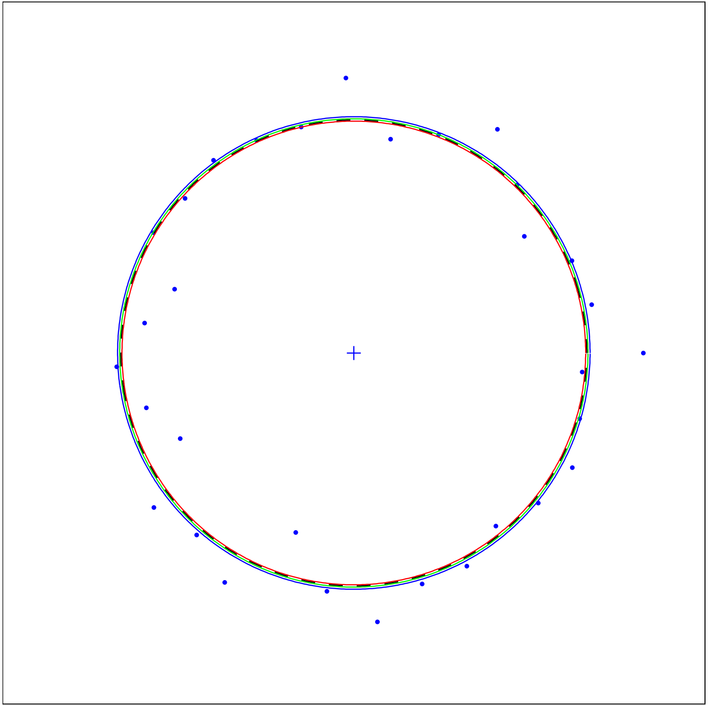
18
6

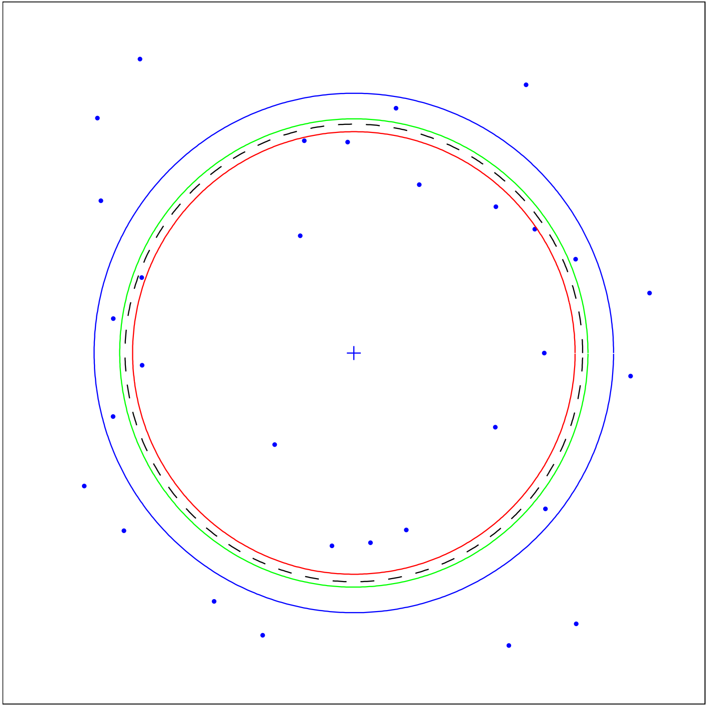

C_1
C_2
X_r
X_a
X_T

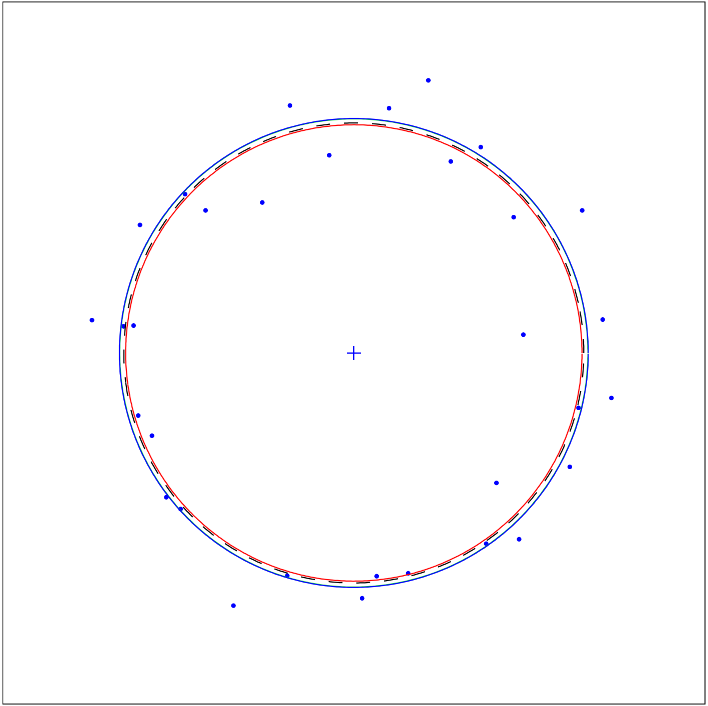

C_1
m_a
m_r
e_1
$m_T = m$

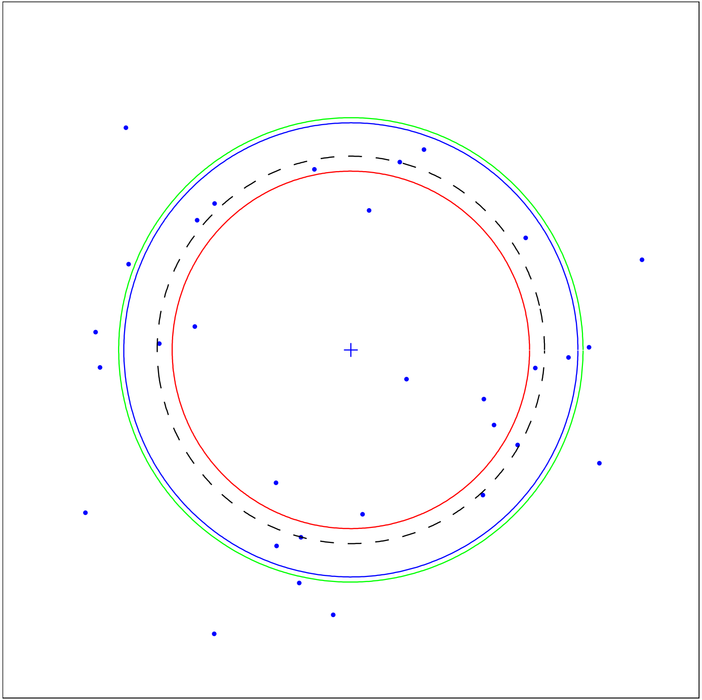


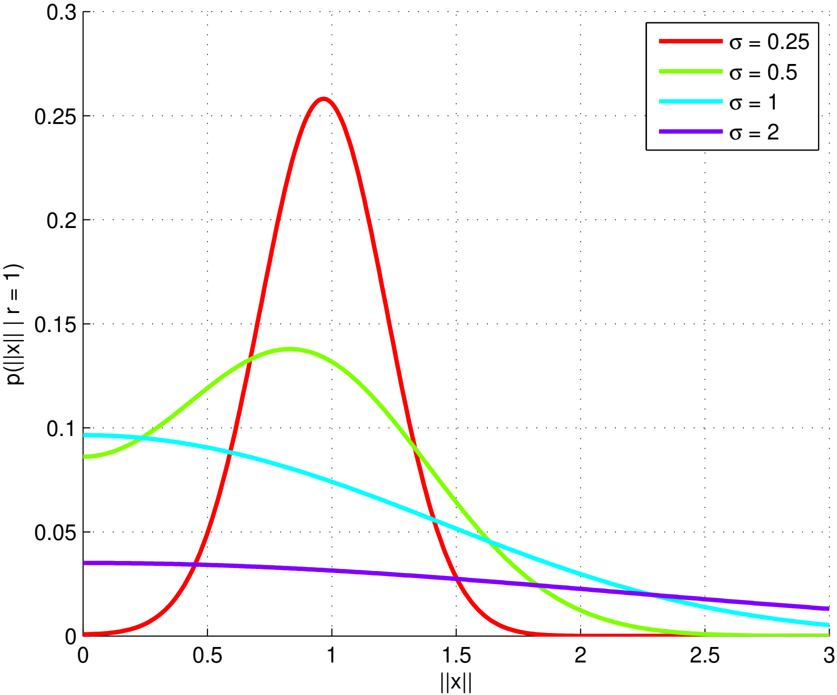
C_2
m
m_r
m_a
e_2
m_T



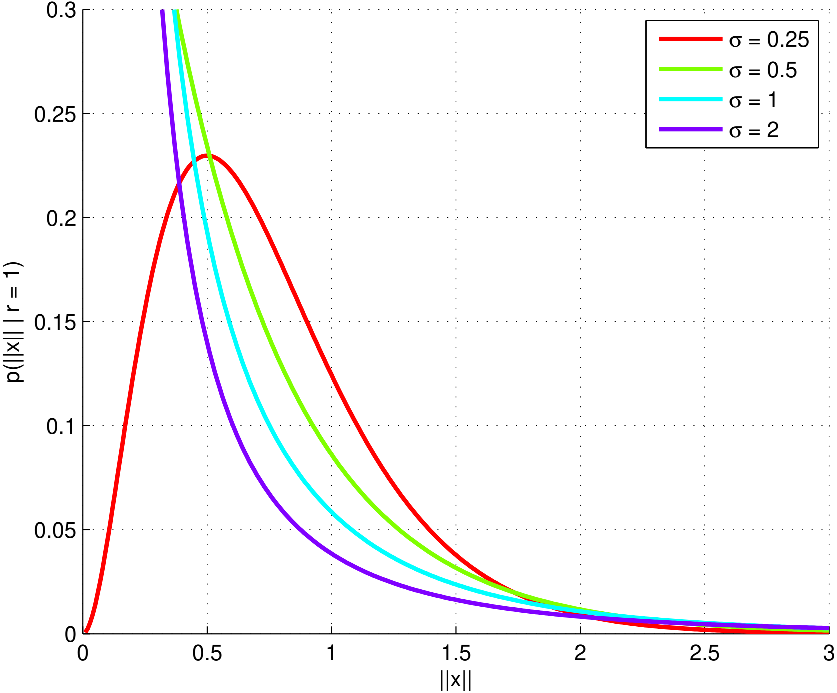




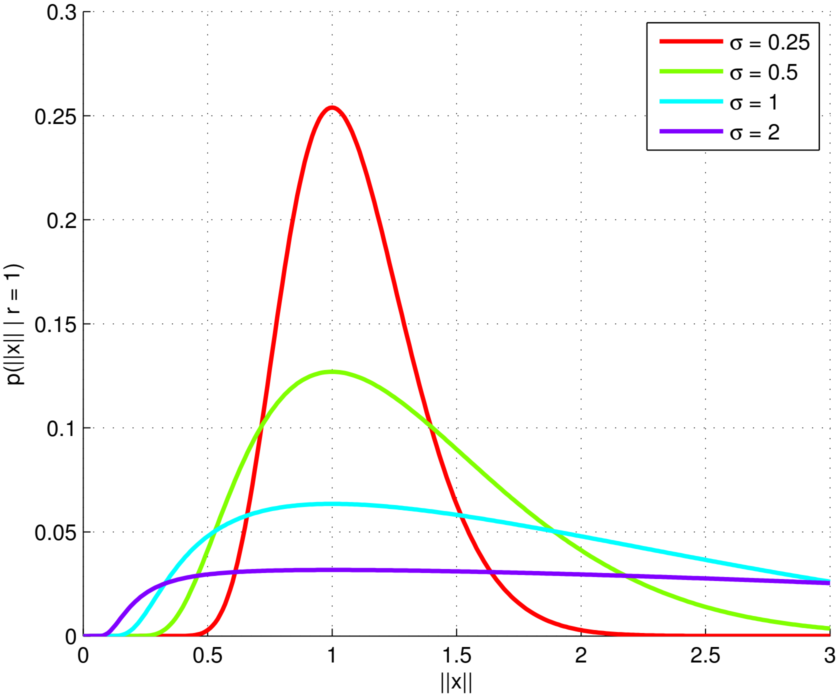




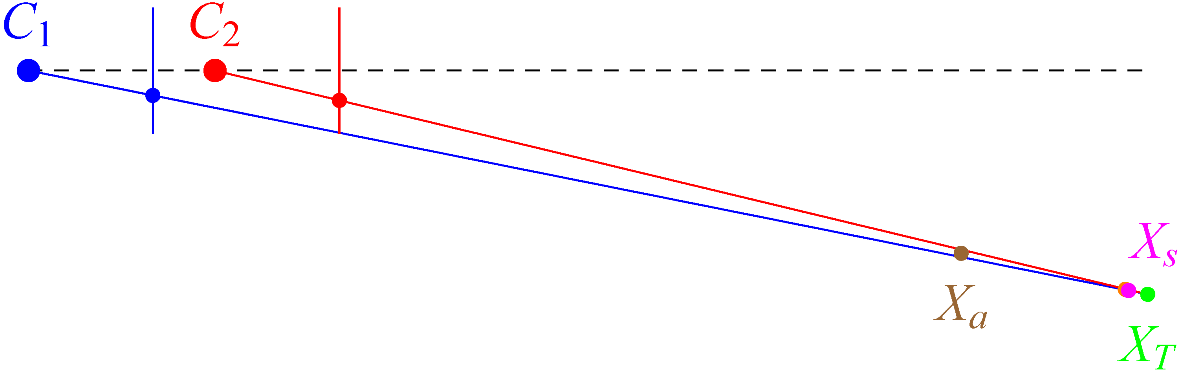
σ = 0.25
σ = 0.5
σ = 1
σ = 2
p(||x|| | r = 1)
||x||
0
0.05
0.1
0.15
0.2
0.25
0.3
0
0.5
1
1.5
2
2.5
3

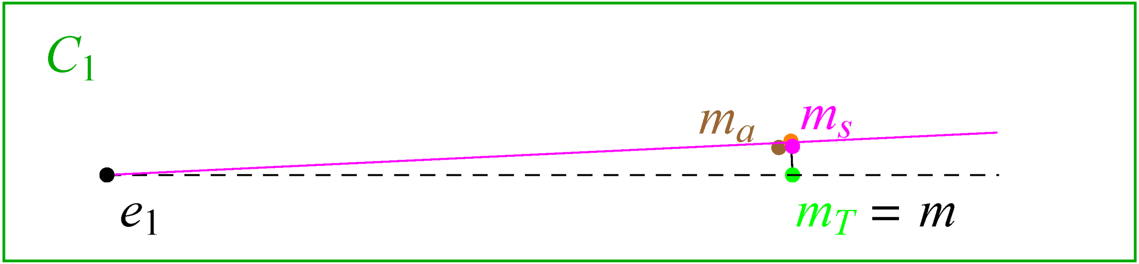


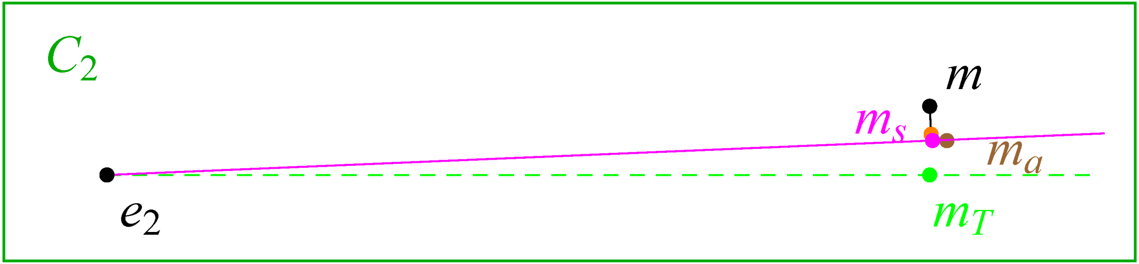
σ = 0.25
σ = 0.5
σ = 1
σ = 2
p(||x|| | r = 1)
||x||
0
0.05
0.1
0.15
0.2
0.25
0.3
0
0.5
1
1.5
2
2.5
3

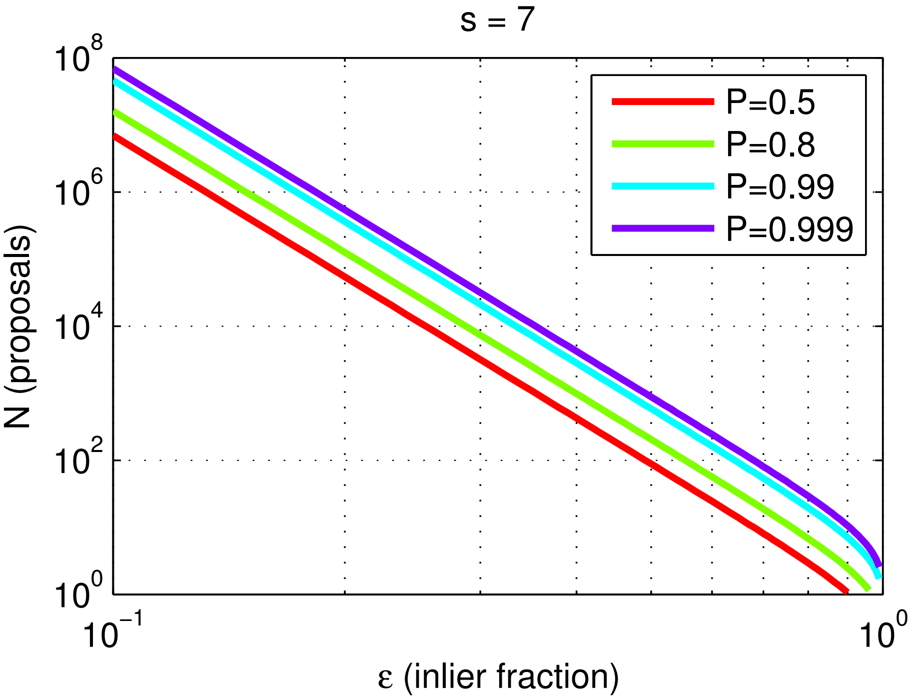


σ = 0.25
σ = 0.5
σ = 1
σ = 2
p(||x|| | r = 1)
||x||
0
0.05
0.1
0.15
0.2
0.25
0.3
0
0.5
1
1.5
2
2.5
3

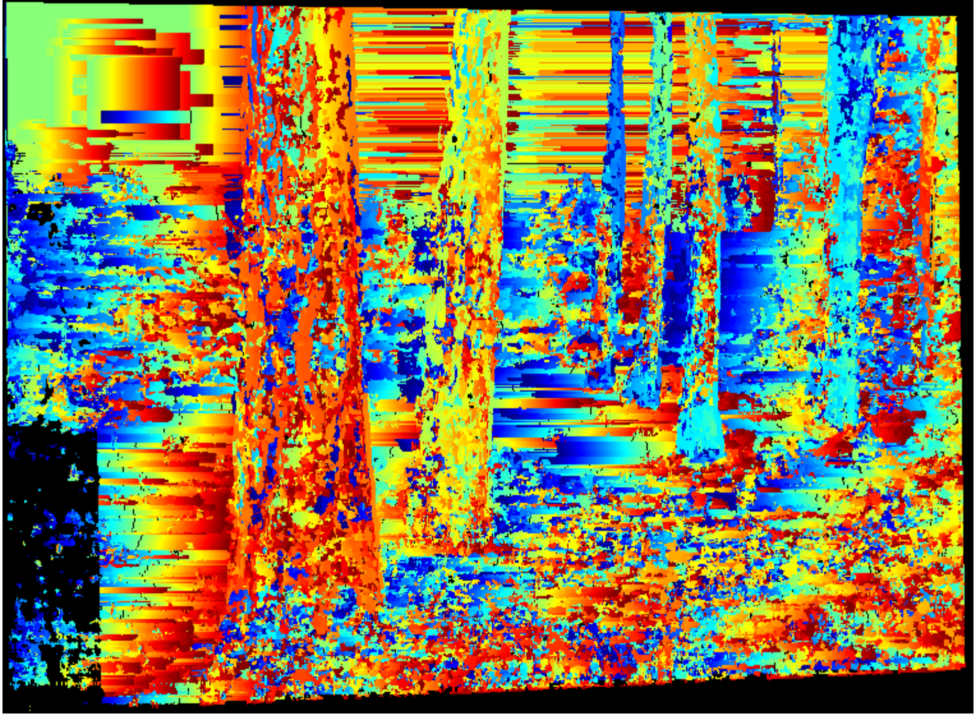

C_1
C_2
X_s
X_a
X_T

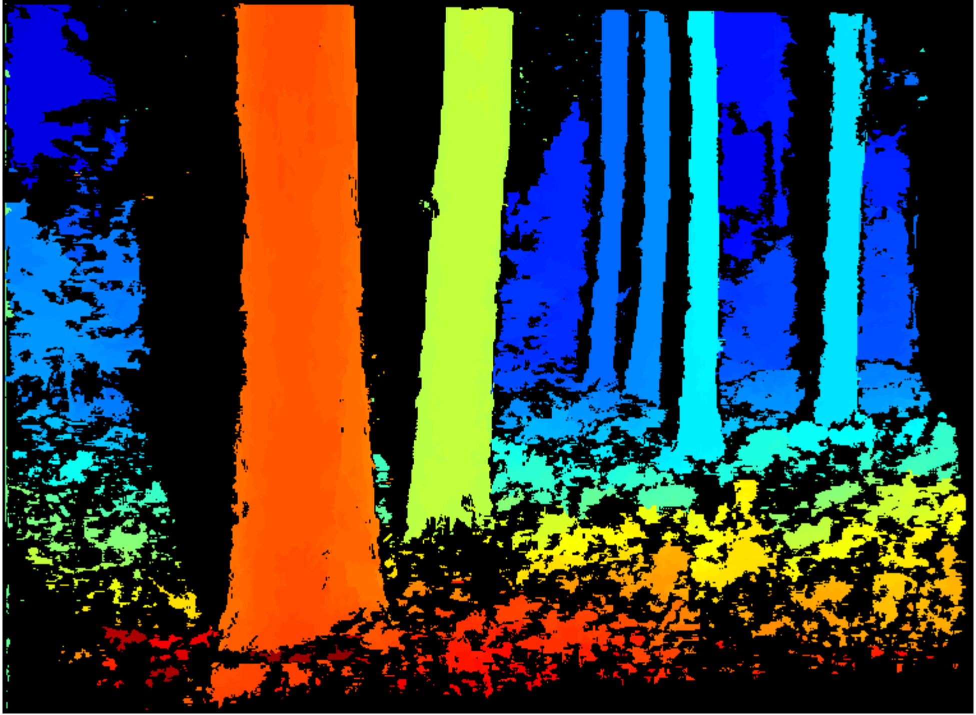

C_1
m_a
m_s
e_1
$m_T = m$

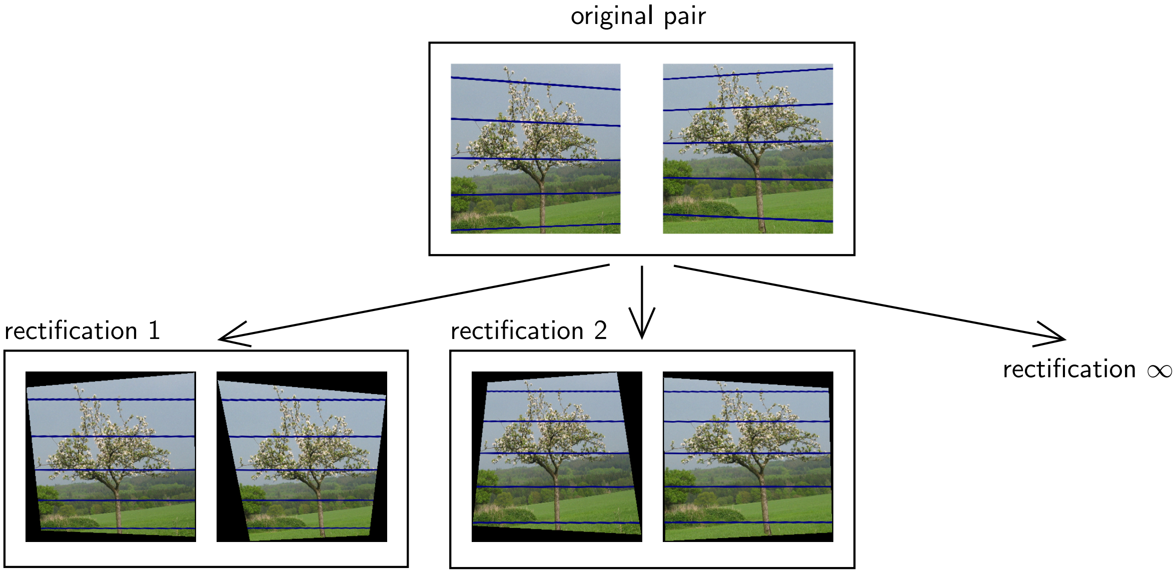

C_2
m
m_s
m_a
e_2
m_T

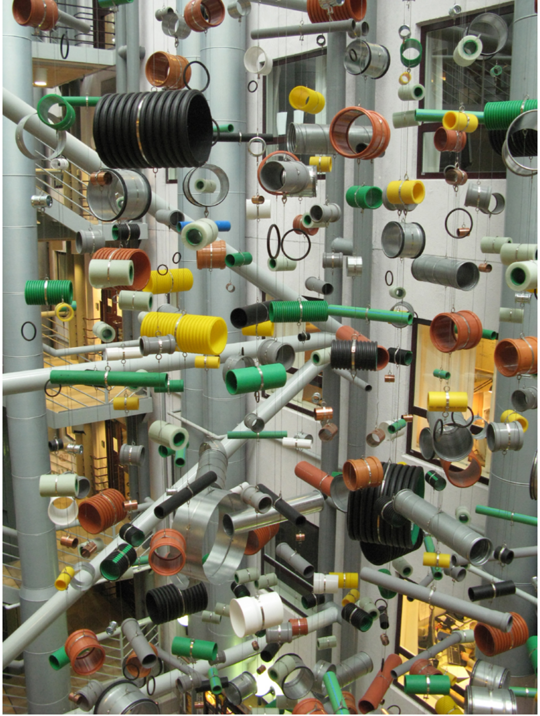


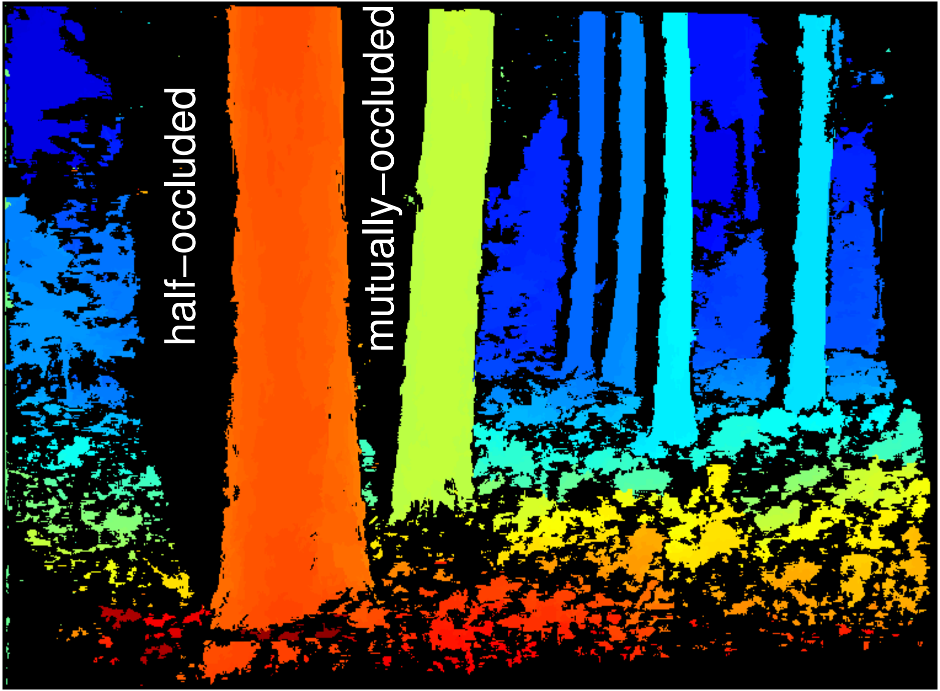
s = 7
P=0.5
P=0.8
P=0.99
P=0.999
N (proposals)
ε (inlier fraction)
10^8
10^6
10^4
10^2
10^0
10^−1
10^0

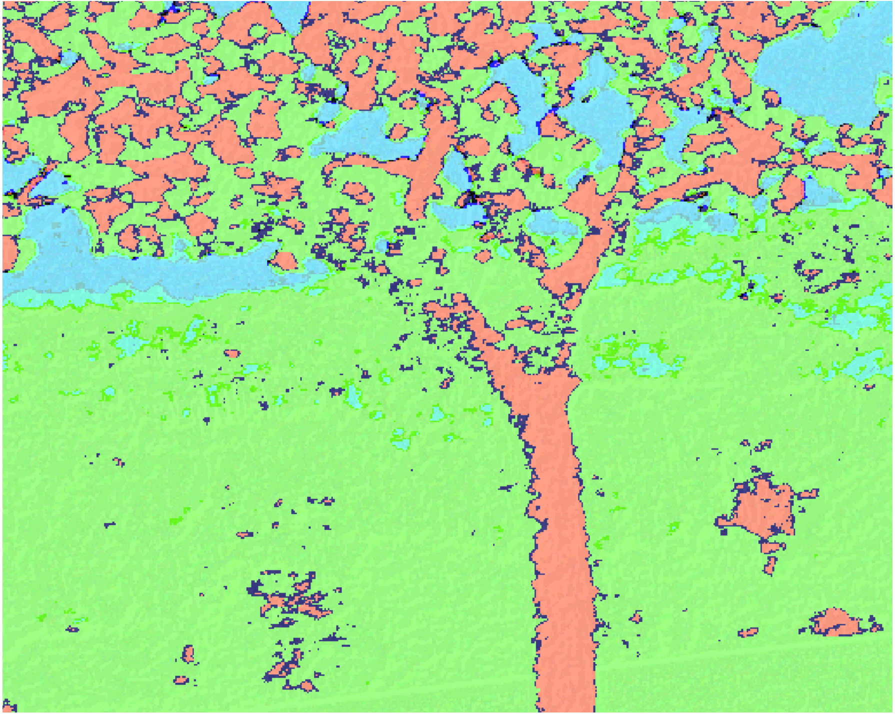


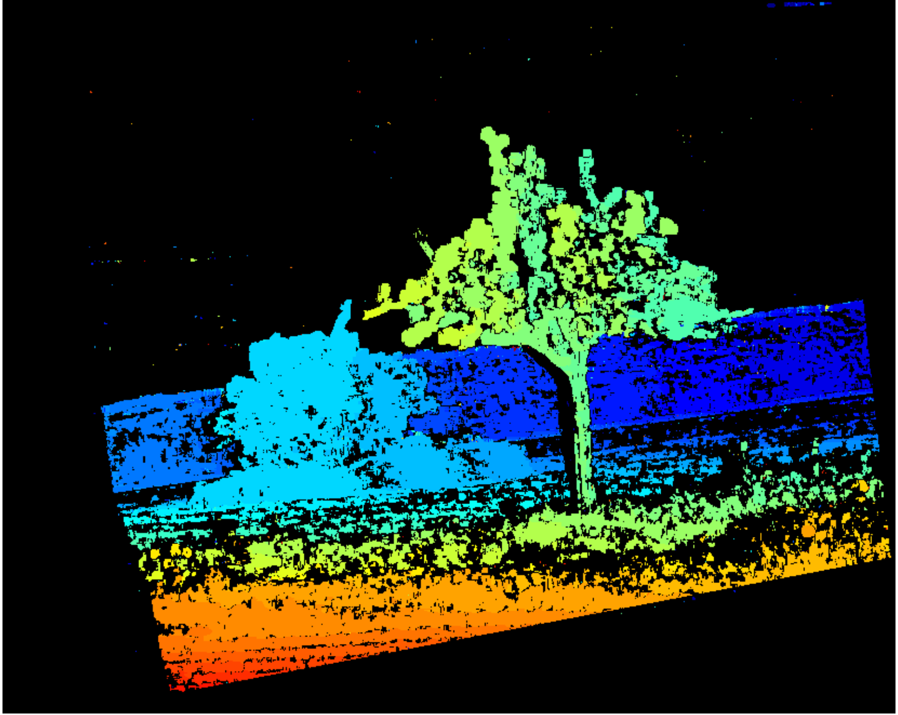


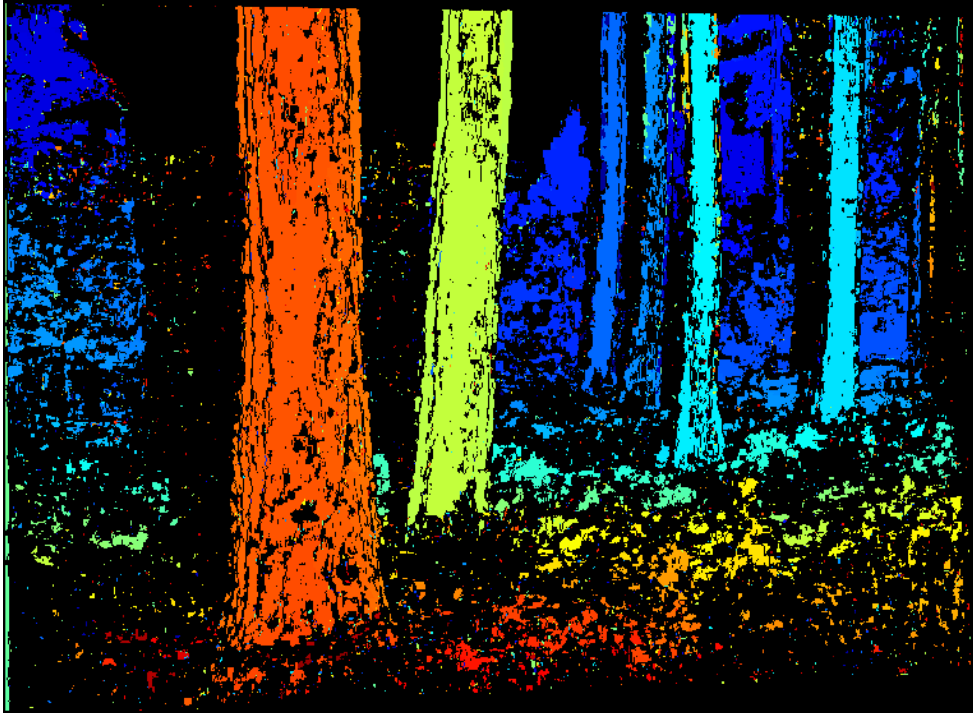

original pair
rectification 1
rectification 2
rectification ∞

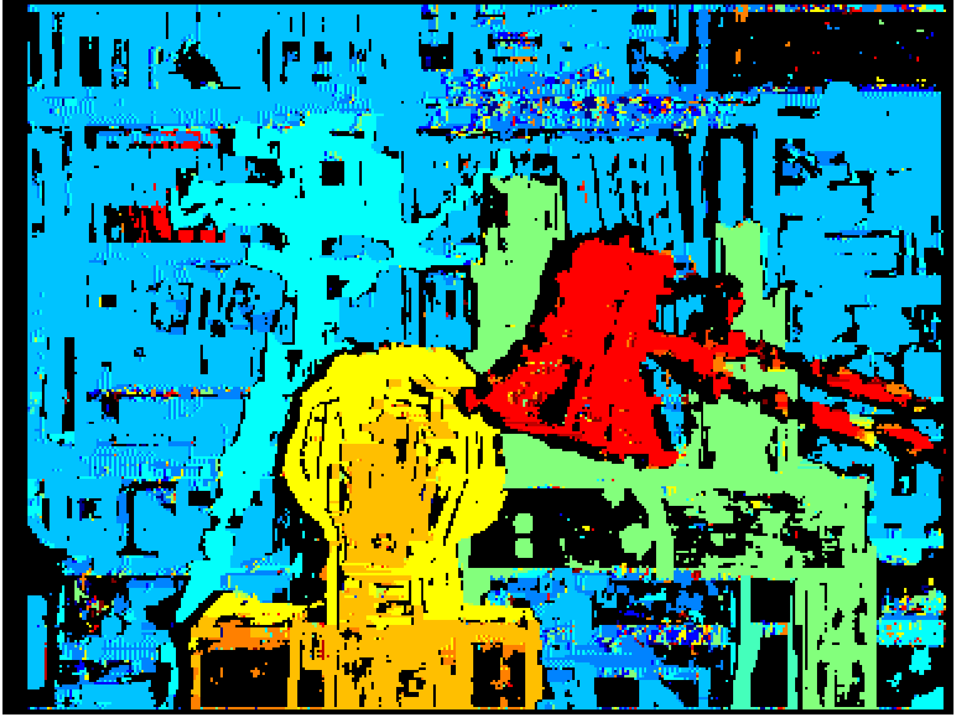


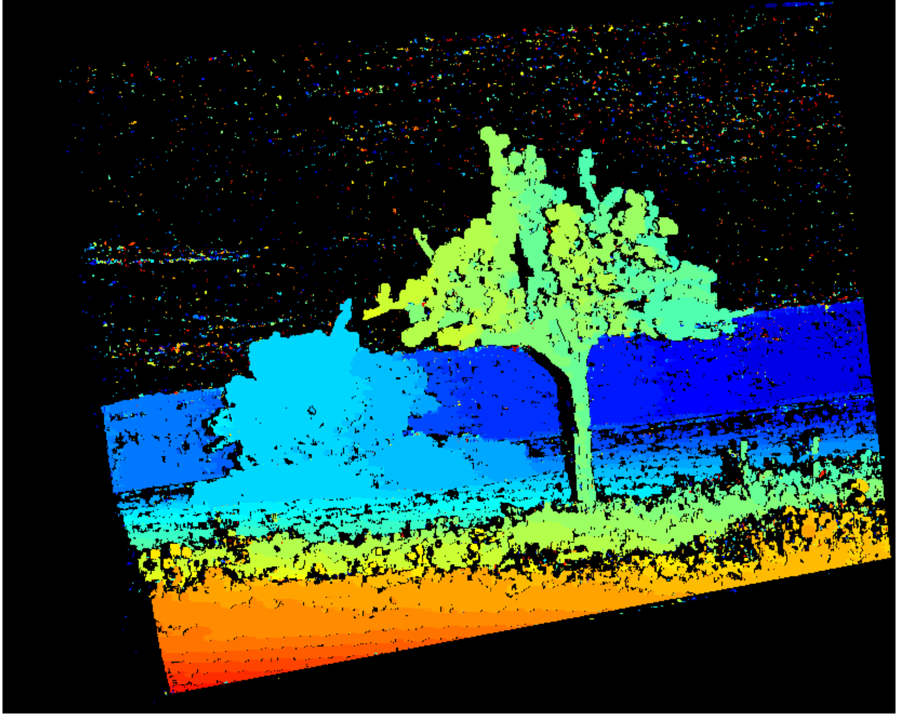

half–occluded
mutually–occluded

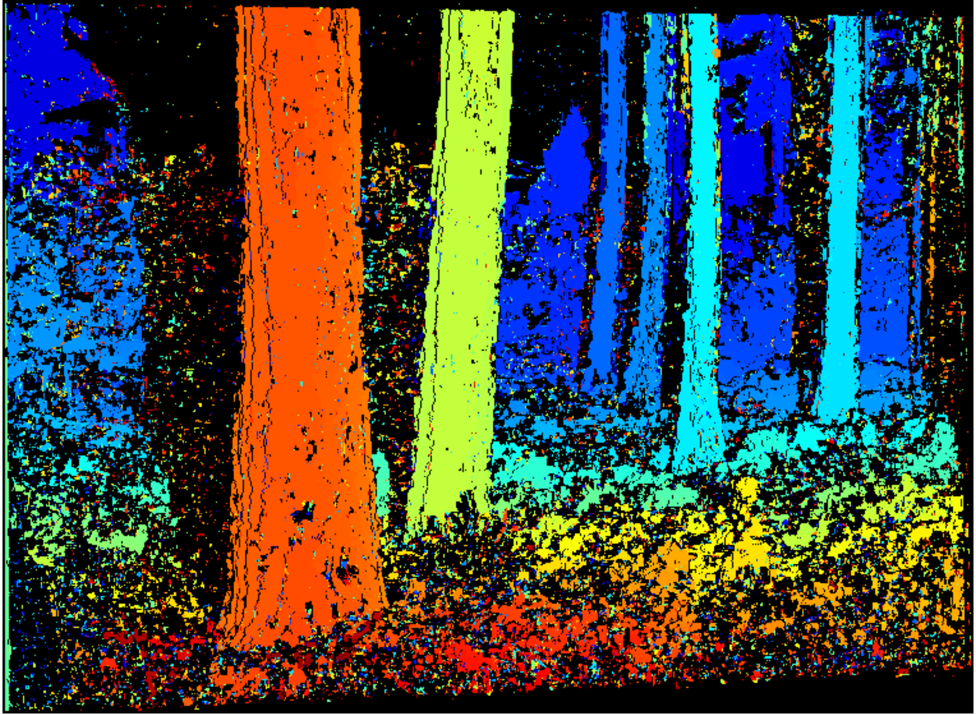


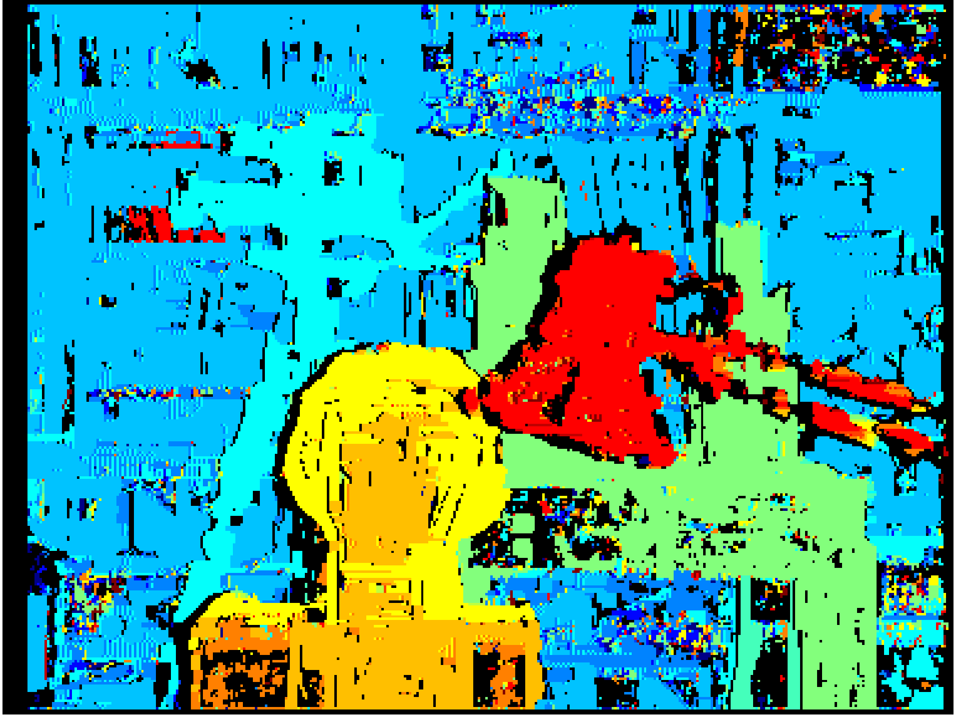


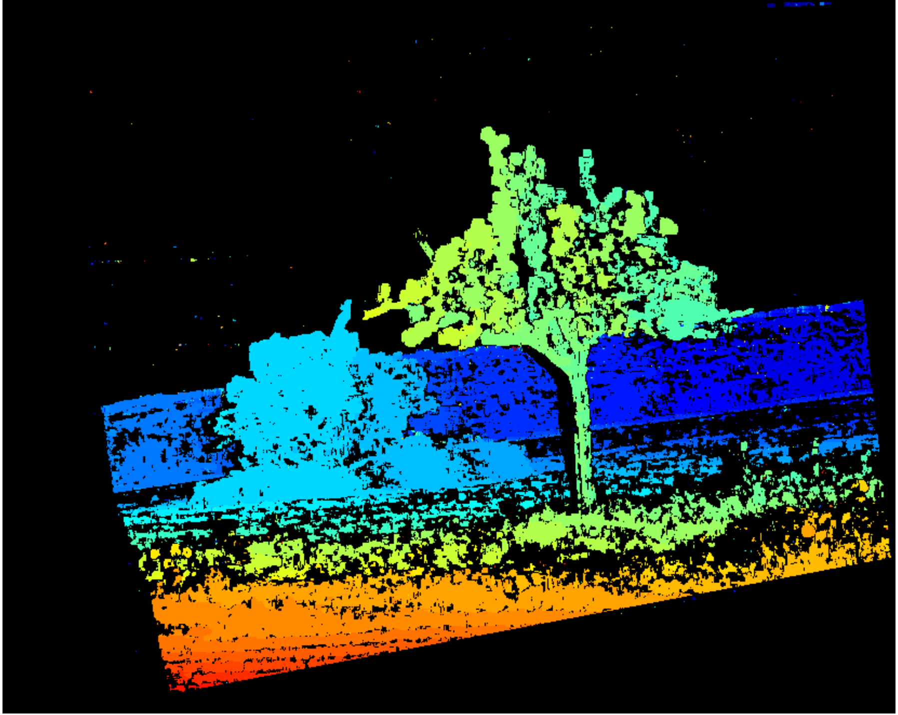


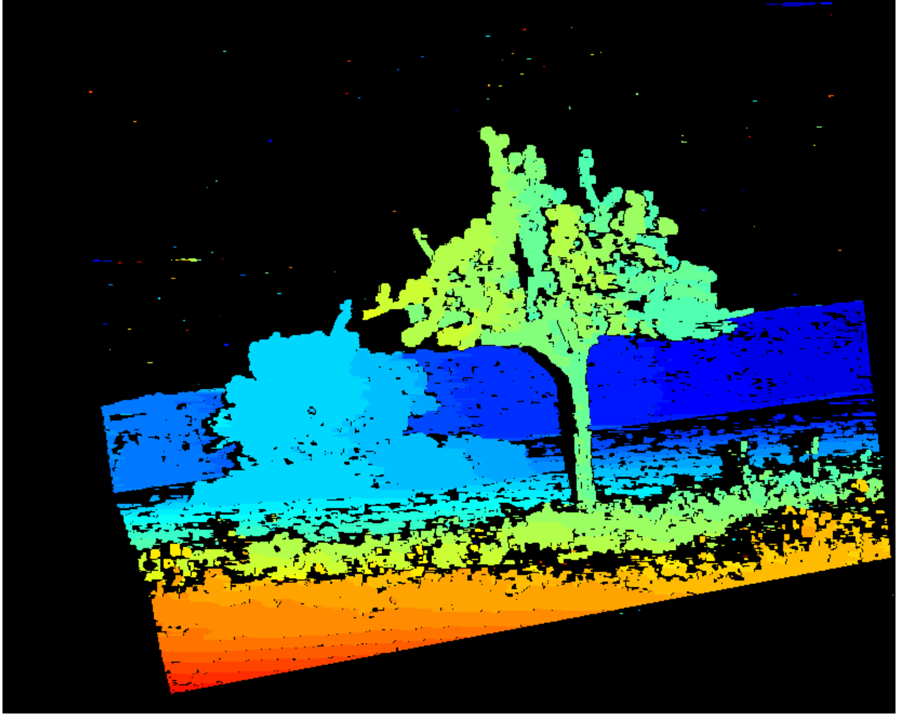


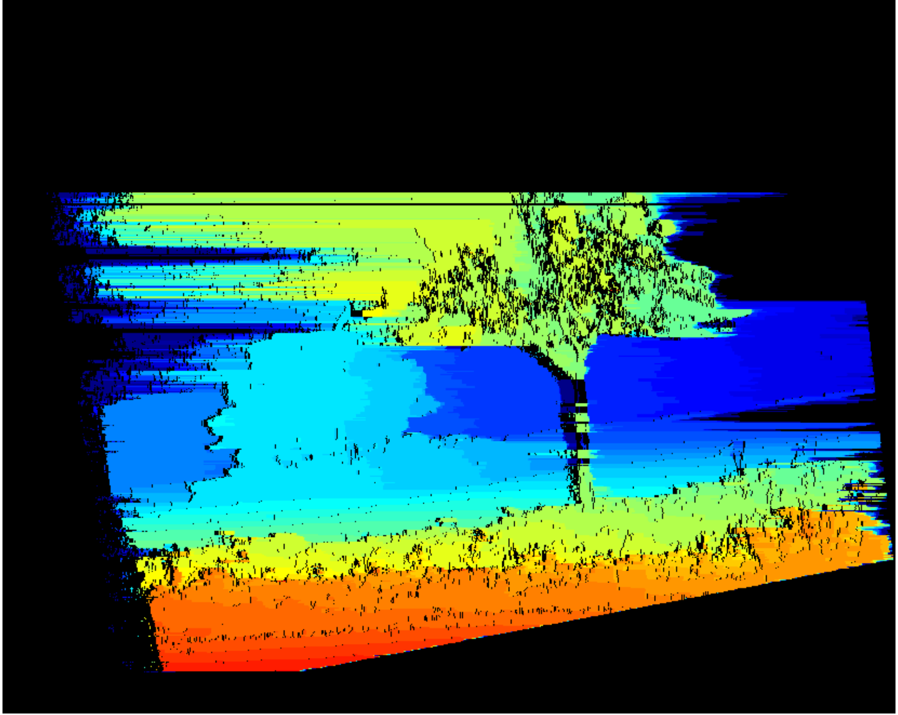


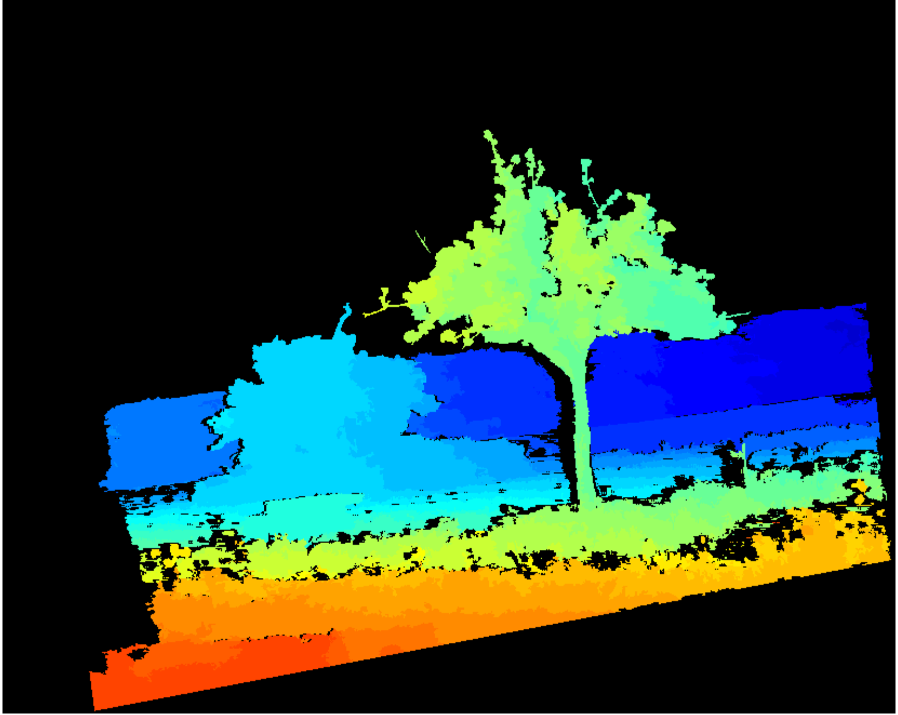


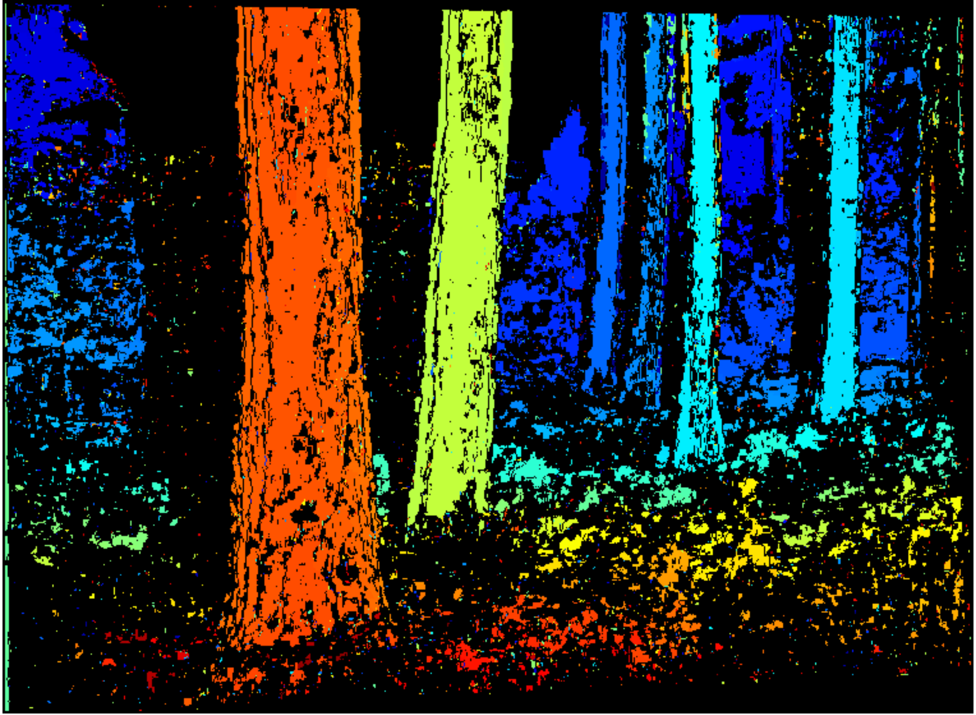


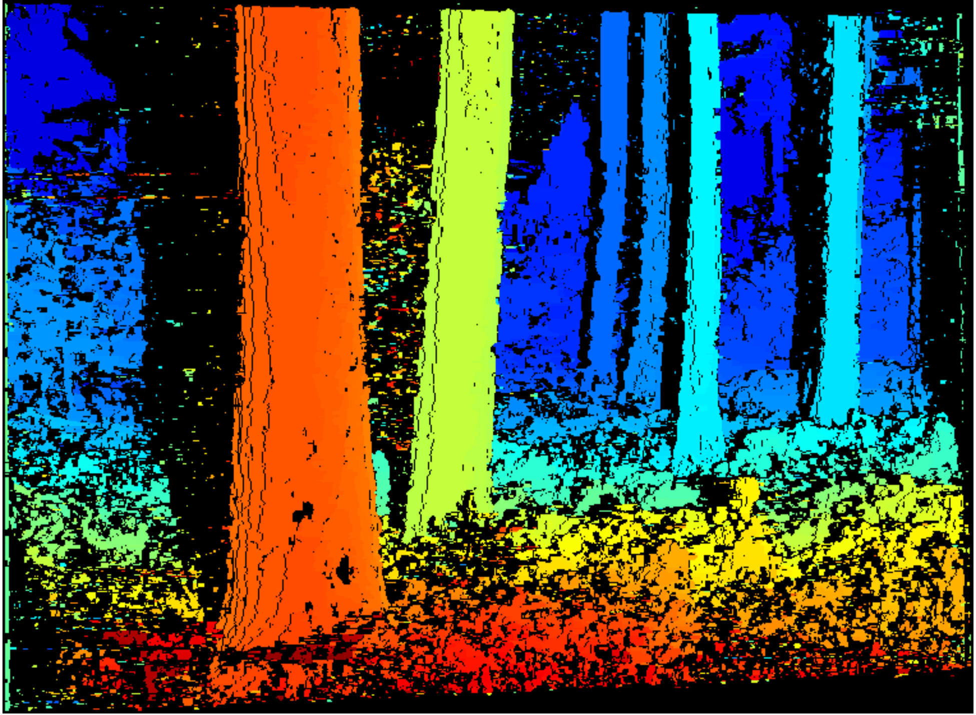


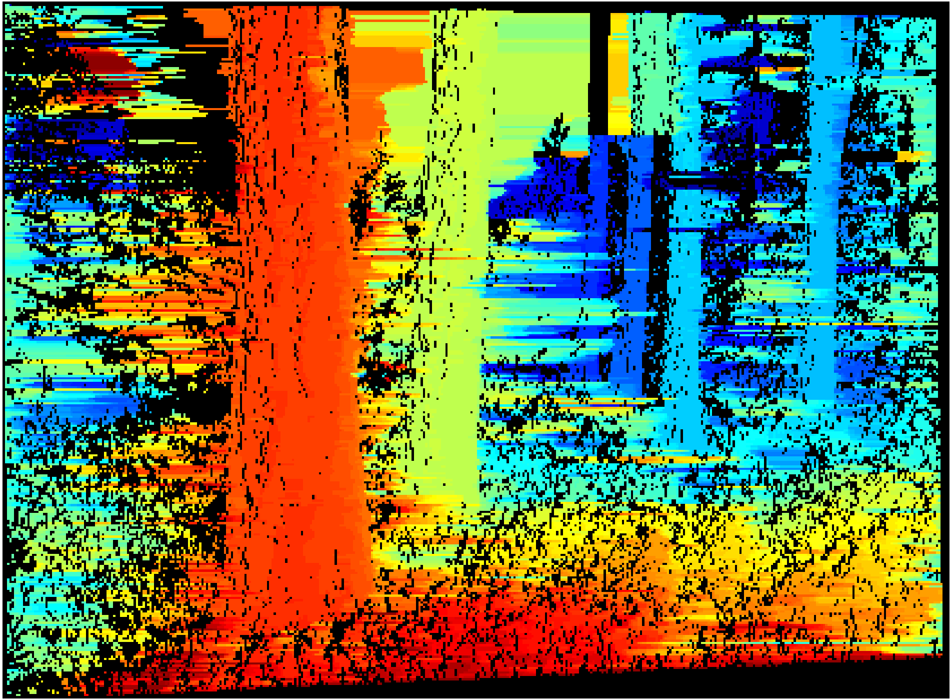


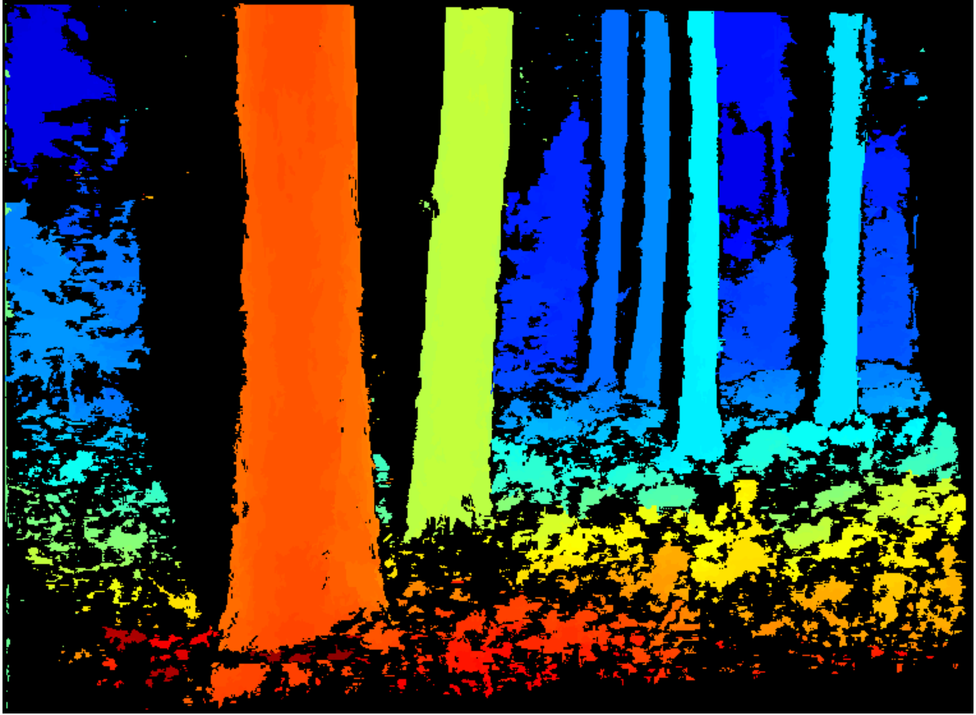


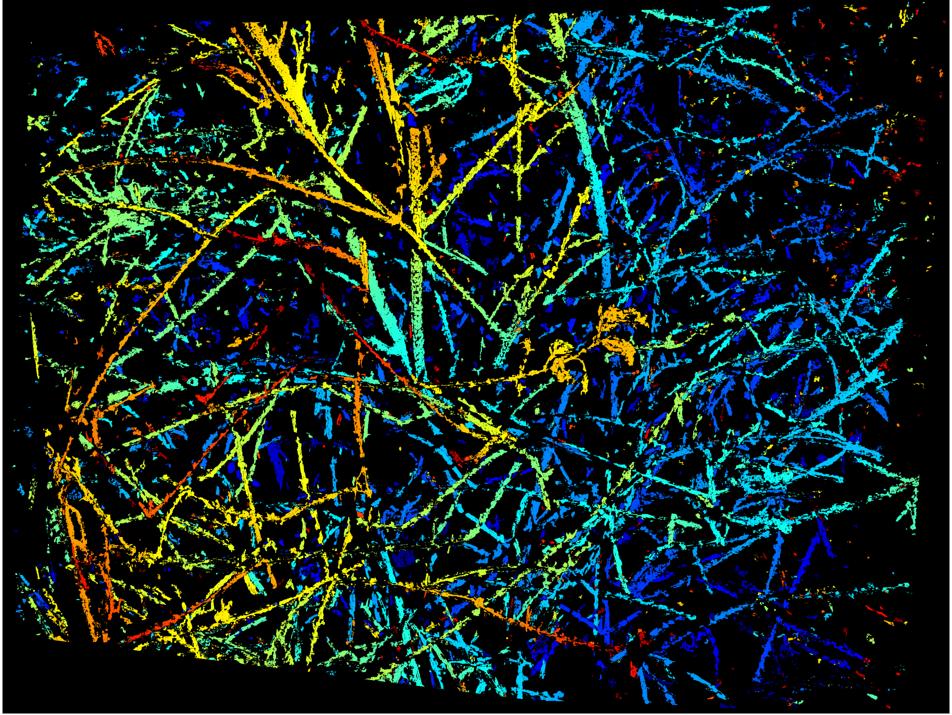


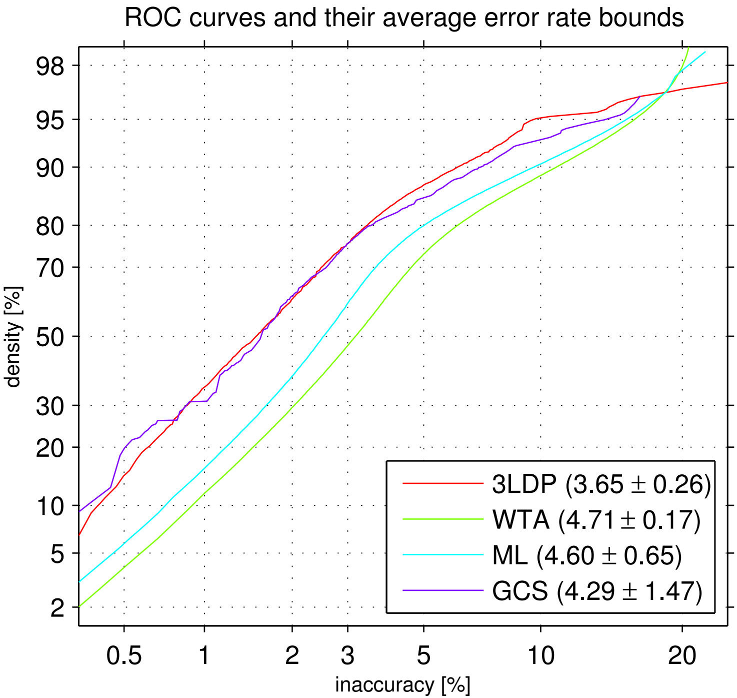




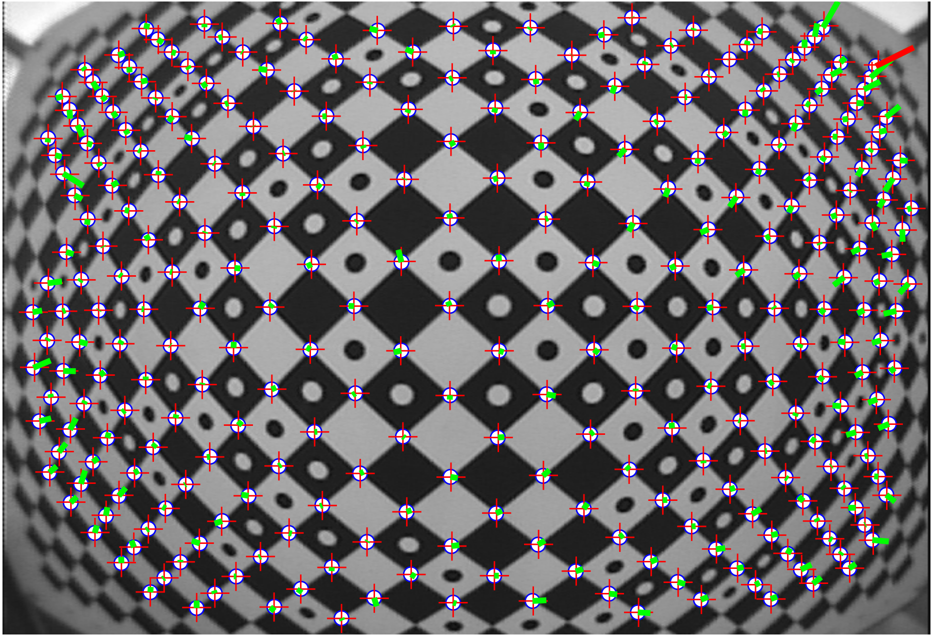


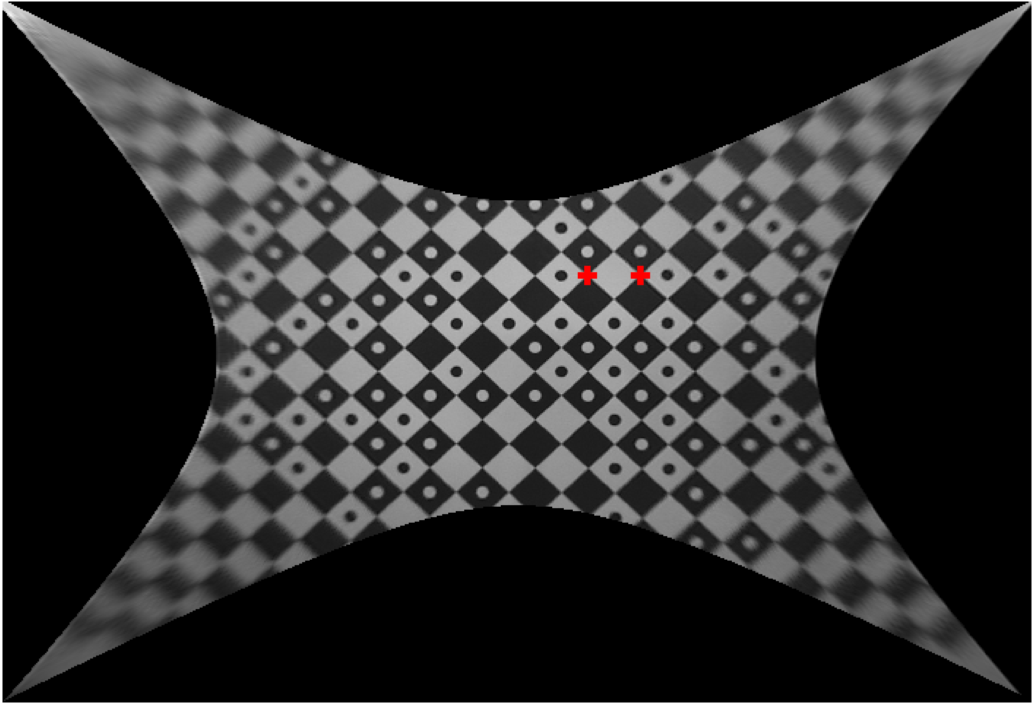


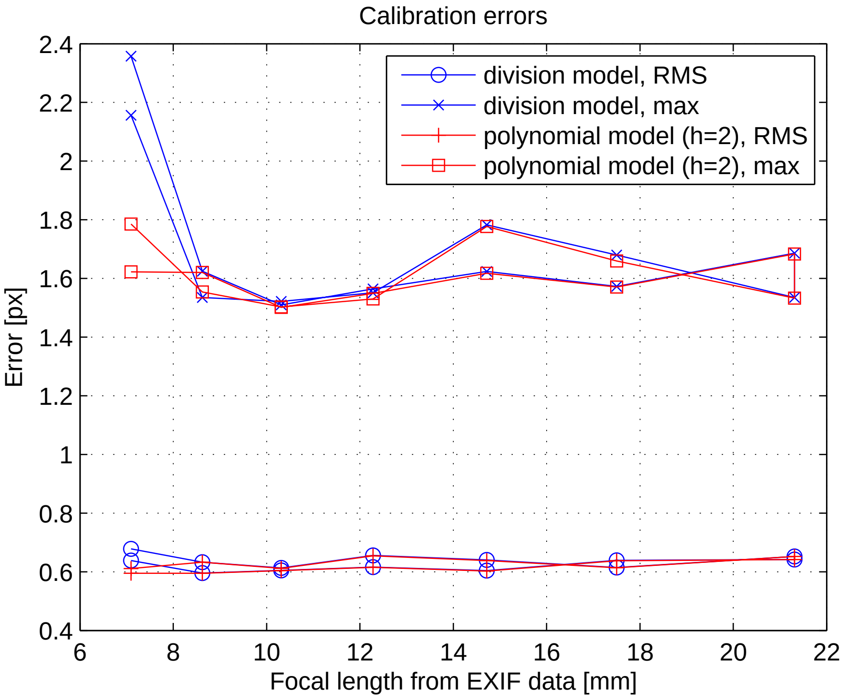


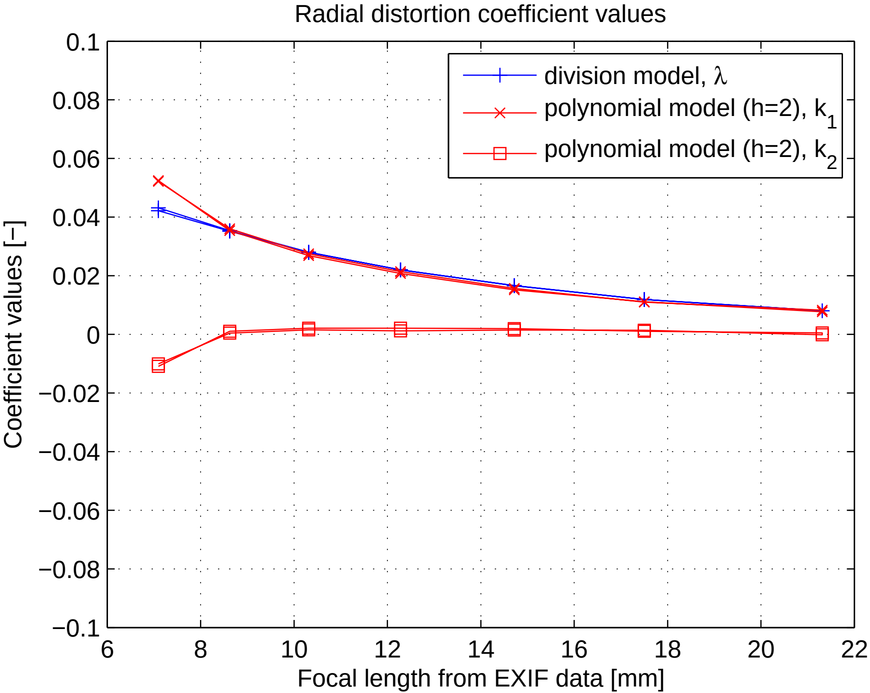

ROC curves and their average error rate bounds
98
95
90
80
70
50
30
20
10
5
2
density [%]
0.5
1
2
3
5
10
20
inaccuracy [%]
3LDP (3.65 ± 0.26)
WTA (4.71 ± 0.17)
ML (4.60 ± 0.65)
GCS (4.29 ± 1.47)

Camera 0, im. 6: Reprojection errors (16x)






Calibration errors
division model, RMS
division model, max
polynomial model (h=2), RMS
polynomial model (h=2), max
Error [px]
Focal length from EXIF data [mm]
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
6
8
10
12
14
16
18
20
22


Radial distortion coefficient values
division model, λ
polynomial model (h=2), k1
polynomial model (h=2), k2
Coefficient values [–]
Focal length from EXIF data [mm]
0.1
0.08
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
−0.08
−0.1
6
8
10
12
14
16
18
20
22