OPPA European Social Fund
Prague & EU: We invest in your future.

OPPA European Social Fund
Prague & EU: We invest in your future.

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.felk.cvut.cz/doku.php/courses/a4m33tdv/

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

rev. December 18, 2012

Open Informatics Master's Course

Part II

Perspective Camera

1. Basic Entities: Points, Lines
2. Homography: Mapping Acting on Points and Lines
3. Canonical Perspective Camera
4. Changing the Outer and Inner Reference Frames
5. Projection Matrix Decomposition
6. Anatomy of Linear Perspective Camera
7. Vanishing Points and Lines
8. Real Camera with Radial Distortion

covered by

[H&Z] Secs: 2.1, 2.2, 3.1, 6.1, 6.2, 8.6, 2.5, 7.4, 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 $\mathbf{x} \in \mathbb{R}^{n,1}$
- 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 $\mathbf{Q} \in \mathbb{R}^{m,n}$

▶Image Line

line in the plane $a\,u + b\,v + c = 0$

corresponds to (homogeneous) vector $\underline{\mathbf{n}} \simeq (a,\, b,\, c)$

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

- 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
- standard representation 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 $\mathbf{1} = 1$
- naming convention: a special entity is the **Ideal Line** (line at infinity)

$$\underline{\mathbf{n}}_\infty \simeq (0, 0, 1)$$

- I may sometimes worngly use $=$ instead of $\simeq$, help me chase the mistakes down

▶Image Point

Point $\mathbf{m} = (u, v)$ is incident on the line $\underline{\mathbf{n}} = (a, b, c)$ iff — this works both ways!

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

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

point is also represented by a homogeneous vector $\underline{\mathbf{m}} \simeq (u, v, \mathbf{1})$

and the equivalence class for $\lambda \in \mathbb{R}, \lambda \neq 0$ is $(m_1, m_2, m_3) = \lambda\,\underline{\mathbf{m}} \simeq \underline{\mathbf{m}}$

- standard representation 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 components have a similar magnitude, $f \sim$ image diagonal

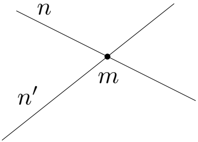
  use $\mathbf{1} = 1$ unless you know what you are doing;
  all entities participating in a formula must be expressed in the same units
- naming convention: **Ideal Point** (point at infinity) $\underline{\mathbf{m}}_\infty \simeq (m_1, m_2, 0)$

  a proper member of $\mathbb{P}^2$
- all such points lie on the ideal line $\underline{\mathbf{n}}_\infty \simeq (0, 0, 1)$, ie. $\underline{\mathbf{m}}_\infty^\top \underline{\mathbf{n}}_\infty = 0$

▶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,

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

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 at 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 the ideal line $\underline{\mathbf{n}}_\infty$
- line at infinity represents a set of directions in plane

▶Homography

Projective space $\mathbb{P}^2$: Vector space of dimension 3 excluding the zero vector, $\mathbb{R}^3 \setminus (0,0,0)$

but including 'points at infinity' and the 'line at infinity'

Collineation: Let x_1, x_2, x_3 be collinear points in $\mathbb{P}^2$. Bijection (1:1, onto) $h\colon \mathbb{P}^2 \mapsto \mathbb{P}^2$ is a collineation iff $h(x_1)$, $h(x_2)$, $h(x_3)$ are collinear.

i.e.

- collinear image points are mapped to collinear image points — lines are mapped to lines
- concurrent image lines are mapped to concurrent image lines — bijection!

  concurrent = intersecting at the same point
- point-line incidence is preserved

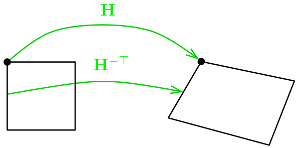
- a mapping $h : \mathbb{P}^2 \to \mathbb{P}^2$ is a collineation iff there exists a non-singular 3×3 matrix $\mathbf{H}$ such that

$$h(x) \simeq \mathbf{H}\,\underline{\mathbf{x}} \quad \text{for all } \underline{\mathbf{x}} \in \mathbb{P}^2$$

- homogeneous matrix representant: $\det \mathbf{H} = 1$
- collineations form a group isomorphic to $SO(3)$

  group of 3×3 matrices with unit determinant and with matrix multiplication
- in this course we will use the term **homography** but mean collineation

▶Mapping Points and Lines by Homography



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

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

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

1. collineation has 8 DOF; it is given by 4 correspondences (points, lines) in a general position
2. extending pixel coordinates to homogeneous coordinates $\underline{\mathbf{m}} = (u, v, \mathbf{1})$
3. mapping by homography, eg. $\underline{\mathbf{m}}' = \mathbf{H}\,\underline{\mathbf{m}}$
4. conversion of the result $\underline{\mathbf{m}}' = (m_1', m_2', m_3')$ to canonical coordinates (pixels):

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

5. can use the unity for the homogeneous coordinate on one side of the equation only!

Elementary Decomposition of a Homography

Unique decompositions: $\mathbf{A} = \mathbf{A}_S\,\mathbf{A}_A\,\mathbf{A}_P \quad (= \mathbf{A}'_P\,\mathbf{A}'_A\,\mathbf{A}'_S)$

$$\mathbf{A}_S = \begin{bmatrix} s\,\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \qquad \text{similarity}$$

$$\mathbf{A}_A = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \qquad \text{special affine}$$

$$\mathbf{A}_P = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^\top & w \end{bmatrix} \qquad \text{special projective}$$

$\mathbf{K}$ – upper triangular matrix with positive diagonal entries
$\mathbf{R}$ – orthogonal, $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$, $\det \mathbf{R} = 1$
$s, w \in \mathbb{R}$, $s > 0$, $w \neq 0$

$$\mathbf{A} = \begin{bmatrix} s\mathbf{R}\mathbf{K} + \mathbf{t}\,\mathbf{v}^\top & w\,\mathbf{t} \\ \mathbf{v}^\top & w \end{bmatrix}$$

- must use 'skinny' QR decomposition, which is unique [Golub & van Loan 1996, Sec. 5.2.6]
- $\mathbf{A}_S$, $\mathbf{A}_A$, $\mathbf{A}_P$ are collineation subgroups
  (eg. $\mathbf{K} = \mathbf{K}_1\mathbf{K}_2$, $\mathbf{K}^{-1}$, $\mathbf{I}$ are all upper triangular with unit determinant, associativity holds)

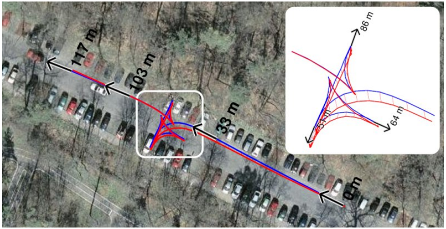
Homography Subgroups

group	DOF	matrix	invariant properties
projective	8	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$	incidence, concurrency, colinearity, cross-ratio, convex hull, order of contact (intersection, tangency, inflection), tangent discontinuities and cusps.
affine	6	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	all above plus: parallelism, ratio of areas, ratio of lengths on parallel lines, linear combinations of vectors (e.g. midpoints), line at infinity $\underline{\mathbf{n}}_\infty$ (not pointwise)
similarity	4	$\begin{bmatrix} s\cos\phi & s\sin\phi & t_x \\ -s\sin\phi & s\cos\phi & t_y \\ 0 & 0 & 1 \end{bmatrix}$	all above plus: ratio of lengths, angle, the circular points $I = (1, i, 0)$, $J = (1, -i, 0)$.
Euclidean	3	$\begin{bmatrix} \cos\phi & \sin\phi & t_x \\ -\sin\phi & \cos\phi & t_y \\ 0 & 0 & 1 \end{bmatrix}$	all above plus: length, area

Some Homographic Tasters

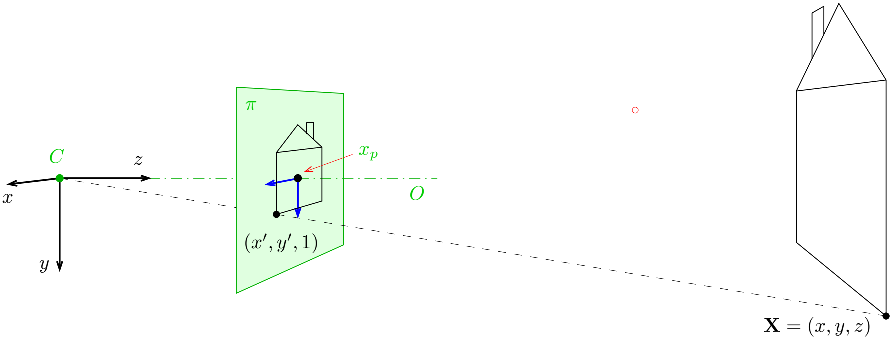
Rectification of camera rotation: Slides 63 (geometry), 120 (homography estimation)

Homographic Mouse for Visual Odometry: Slide TBD

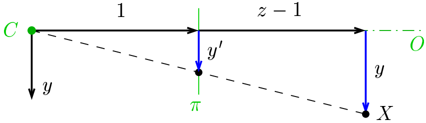


illustrations courtesy of AMSL Racing Team, Meiji University and LIBVISO: Library for VISual Odometry

▶Canonical Perspective Camera (Pinhole Camera, Camera Obscura)



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



projected point in the natural image coordinate system:

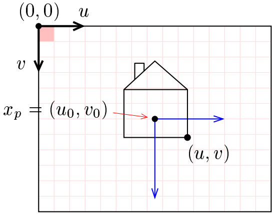
$$\frac{y'}{1} = y' = \frac{y}{1+z-1} = \frac{y}{z}, \qquad x' = \frac{x}{z}$$

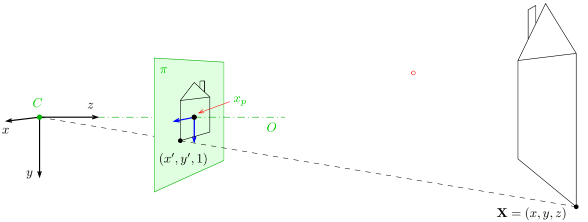
►Natural and Canonical Image Coordinate Systems

projected point **in canonical camera**

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix}^\top = \begin{bmatrix} \frac{x}{z}, & \frac{y}{z}, & 1 \end{bmatrix}^\top = \frac{1}{z} \begin{bmatrix} x, & y, & z \end{bmatrix}^\top \simeq \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{P}_0} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{P}_0 \, \underline{\mathbf{X}}$$

projected point **in scanned image** notice the chimney!





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

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

- 'calibration' matrix $\mathbf{K}$ transforms canonical camera $\mathbf{P}_0$ to standard projective camera $\mathbf{P}$

►Computing with Perspective Camera Projection Matrix

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

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

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

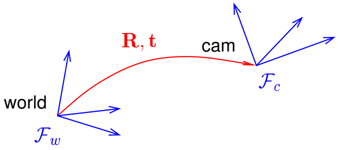
(u_0, v_0) – principal point in pixels

Perspective Camera:

1. dimension reduction — since $\mathbf{P} \in \mathbb{R}^{3,4}$
2. nonlinear unit change $\mathbf{1} \mapsto \mathbf{1} \cdot z/f$ — since $\underline{\mathbf{m}} \simeq (x, y, z/f)$
   for convenience we use $P_{11} = P_{22} = f$ rather than $P_{33} = 1/f$ and the u_0, v_0 in relative units
3. $m_3 = 0$ represents points at infinity in image plane π $(z = 0)$

▶Changing The Outer (World) Reference Frame

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



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

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

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

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

$\mathbf{P}_0$ selects the first 3 rows of $\mathbf{T}$ and discards the last row

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

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

  $\mathbf{C}$ – camera position in the world reference frame — $\mathbf{t} = -\mathbf{R}\mathbf{C}$

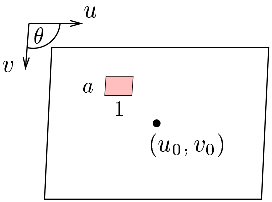
  $\mathbf{r}_3^\top$ – camera axis in the world reference frame — third row of $\mathbf{R}$: $\mathbf{r}_3 = \mathbf{R}^{-1}[0, 0, 1]^\top$

- we can save some conversion and computation by noting that $\mathbf{K}\mathbf{R}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix}\underline{\mathbf{X}} = \mathbf{K}\mathbf{R}(\mathbf{X} - \mathbf{C})$

►Changing the Inner (Image) Reference Frame

The general form of calibration matrix $\mathbf{K}$ includes

- digitization raster skew angle θ
- pixel aspect ratio a



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

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

⊛ H1; 2pt: Verify this $\mathbf{K}$; hints: $u'\mathbf{e}_{u'} + v'\mathbf{e}_{v'} = u\mathbf{e}_u + v\mathbf{e}_v$, boldface are basis vectors, $\mathbf{K}$ maps from an orthogonal system to a skewed system $[w'u', w'v', w']^\top = \mathbf{K}[u, v, 1]^\top$; first skew then sampling then shift by u_0, v_0 deadline LD+2 wk

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

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

a recipe for filling $\mathbf{P}$

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

▶Projection Matrix Decomposition

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

$\mathbf{Q} \in \mathbb{R}^{3,3}$ full rank (if finite perspective cam.)
$\mathbf{K} \in \mathbb{R}^{3,3}$ upper triangular with positive diagonal entries
$\mathbf{R} \in \mathbb{R}^{3,3}$ rotation: $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$ and $\det \mathbf{R} = +1$

1. $\mathbf{C} = -\mathbf{Q}^{-1}\mathbf{q}$ see next
2. RQ decomposition of $\mathbf{Q} = \mathbf{K}\mathbf{R}$ using three Givens rotations [H&Z, p. 579]

$$\mathbf{K} = \mathbf{Q}\underbrace{\mathbf{R}_{32}\mathbf{R}_{31}\mathbf{R}_{21}}_{\mathbf{R}^{-1}}$$

3. $\mathbf{t} = -\mathbf{R}\mathbf{C}$

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

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

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

- RQ decomposition nonuniqueness: $\mathbf{K}\mathbf{R} = \mathbf{K}\mathbf{T}^{-1}\mathbf{T}\mathbf{R}$, 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
 'skinny' RQ decomposition
- care must be taken to avoid overflow, see [Golub & van Loan 1996, sec. 5.2]

RQ Decomposition Step

```
Q = Array[q, {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
s1 = Solve[{Q1[[3]][[2]] == 0, c^2 + s^2 == 1}, {c, s}];
s1 = s1[[2]]
Q1 /. s1 // Simplify // MatrixForm
```

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

$$\left\{c \to \frac{q[3,3]}{\sqrt{q[3,2]^2 + q[3,3]^2}},\; s \to \frac{q[3,2]}{\sqrt{q[3,2]^2 + q[3,3]^2}}\right\}$$

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

▶Center of Projection

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

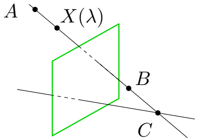
Theorem

Let there be $\underline{\mathbf{B}} \neq \mathbf{0}$ s.t. $\mathbf{P}\,\underline{\mathbf{B}} = \mathbf{0}$. Then $\underline{\mathbf{B}}$ is equal to the projection center $\underline{\mathbf{C}}$ (in world coordinate frame).

Proof.

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

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



2. it images to

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

- the whole line images to a single point ⇒ it must pass through the optical center of $\mathbf{P}$
- this holds for all choices of A ⇒ the only common point of the lines is the C, i.e. $\underline{\mathbf{B}} \simeq \underline{\mathbf{C}}$

□

Hence

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

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

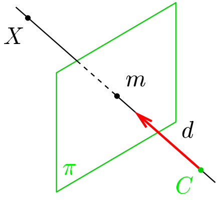
▶Optical Ray

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

1. consider line ($\mathbf{d}$ line direction vector, $\lambda \in \mathbb{R}$)

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

2. the image of point X is



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

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

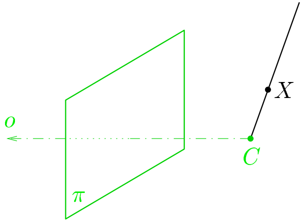
- optical ray line corresponding to image point m is

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

- optical ray may be represented by a point at infinity $(\mathbf{d}, 0)$

▶Optical Axis

Optical axis: The line through C that is perpendicular to image plane π



1. a line parallel to π images to line at infinity in π:

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

2. point X in parallel to π iff $\mathbf{q}_3^\top \mathbf{X} + q_{34} = 0$
3. this is a plane with $\pm\mathbf{q}_3$ as the normal vector
4. optical axis direction: substitution $\mathbf{P} \mapsto \lambda\mathbf{P}$ must not change the direction
5. we select (assuming $\det(\mathbf{R}) > 0$)

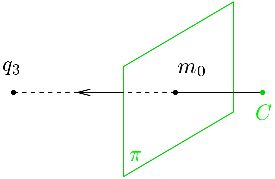
$$\mathbf{o} = \det(\mathbf{Q})\, \mathbf{q}_3$$

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

▶Principal Point

Principal point: The intersection of image plane and the optical axis

1. we take point at infinity on the optical axis that must project to principal point m_0



2. then

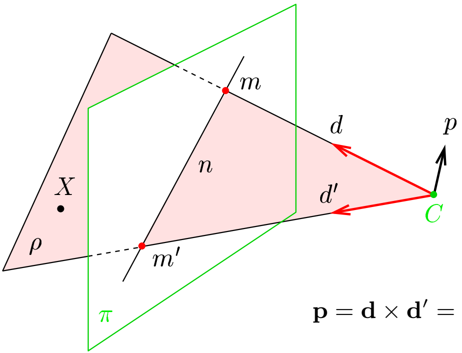
$$\underline{\mathbf{m}}_0 \simeq [\mathbf{Q} \quad \mathbf{q}] \begin{bmatrix} \mathbf{q}_3 \\ 0 \end{bmatrix} = \mathbf{Q}\,\mathbf{q}_3$$

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

- principal point is also the center of radial distortion (see Slide 50)

▶Optical Plane

A spatial plane with normal p passing through optical center C and a given image line n.



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

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

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

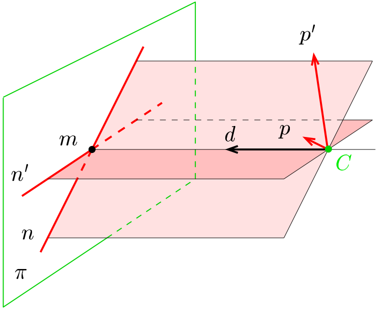
- note the factoring-out of $\mathbf{Q}$!

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

see Slide 28

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

Cross-Check: Optical Ray as Optical Plane Intersection



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

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

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

▶Summary: Optical Center, Ray, Axis, Plane

General finite camera

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

$\underline{\mathbf{C}} \simeq \mathrm{rnull}(\mathbf{P})$	optical center (world coords.)
$\mathbf{d} = \mathbf{Q}^{-1}\,\underline{\mathbf{m}}$	optical ray direction (world coords.)
$\det(\mathbf{Q})\,\mathbf{q}_3$	outward optical axis (world coords.)
$\mathbf{Q}\,\mathbf{q}_3$	principal point (in image plane)
$\boldsymbol{\rho} = \mathbf{P}^\top\,\underline{\mathbf{n}}$	optical plane (world coords.)
$\mathbf{K} = \begin{bmatrix} f & -f\cot\theta & u_0 \\ 0 & f/(a\sin\theta) & v_0 \\ 0 & 0 & 1\end{bmatrix}$	camera (calibration) matrix (f, u_0, v_0 in pixels)
$\mathbf{R}$	camera rotation matrix (cam coords.)
$\mathbf{t}$	camera translation vector (cam coords.)

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

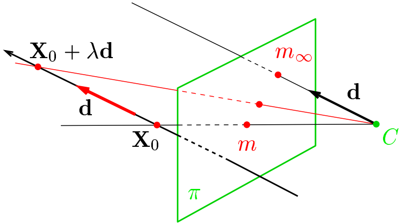
How far is the engine?

distance between sleepers 0.806m but we cannot count them, resolution is too low

We will review some life-saving theory. . .

▶Vanishing Point

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



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

⊛ P1; 1pt: Derive or prove

- V.P. is independent on line position, it depends on its orientation only
  all parallel lines have the same V.P.
- the image of the V.P. of a spatial line with direction vector $\mathbf{d}$ is $\underline{\mathbf{m}} = \mathbf{Q}\,\mathbf{d}$
- V.P. m corresponds to spatial direction $\mathbf{d} = \mathbf{Q}^{-1}\underline{\mathbf{m}}$ optical ray through m
- V.P. is the image of a point at infinity on any line, not just the optical ray as on Slide 33

Some Vanishing Point Applications

where is the sun?

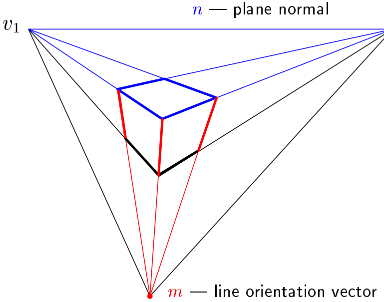
what is the wind direction?
(must have video)

fly above the lane,
at constant altitude!

▶Vanishing Line

Vanishing line: 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

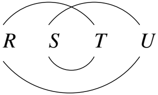


- V.L. n corresponds to space plane of normal vector $\mathbf{p} = \mathbf{Q}^\top \underline{\mathbf{n}}$
- a space plane of normal vector $\mathbf{p}$ has a V.L. represented by $\underline{\mathbf{n}} = \mathbf{Q}^{-\top} \mathbf{p}$.

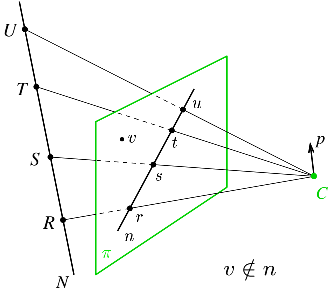
►Cross Ratio

Four collinear space points R, S, T, U define cross-ratio

$$[RSTU] = \frac{|RT|}{|RU|}\,\frac{|SU|}{|ST|}$$



a mnemonic



$|RT|$ – signed distance from R to T
(w.r.t. a fixed line orientation)

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

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

Corollaries:

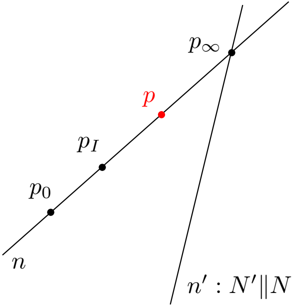
- cross ratio is invariant under collineations (homographies) $\underline{\mathbf{x}}' \simeq \mathbf{H}\underline{\mathbf{x}}$ — plug $\mathbf{H}\underline{\mathbf{x}}$ in (1)
- cross ratio is invariant under perspective projection: $[RSTU] = [r\,s\,t\,u]$

- 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

▶1D Projective Coordinates

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

$$[P] = [P_\infty\, P_0\, P_I\, P] = [p_\infty\, p_0\, p_I\, p] = \frac{|p_\infty\, p_I|}{|p_0\, p_I|}\,\frac{|p_0\, p|}{|p_\infty\, p|}$$



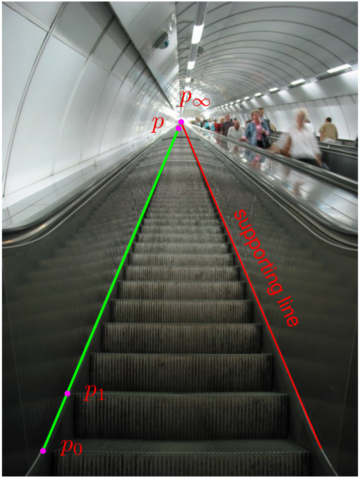
P_0 – the origin	$[P_0] = 0$
P_I – the unit point	$[P_I] = 1$
P_∞ – the supporting point	$[P_\infty] = \pm\infty$

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

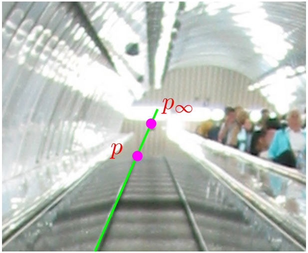
Applications

- Given the image of a line N, the origin, the unit point, and the vanishing point, then the Euclidean coordinate of any point $P \in N$ can be determined → see Slide 45
- Finding v.p. of a line through a regular object → see Slide 46

Application: Counting Steps



- Namesti Miru underground station in Prague

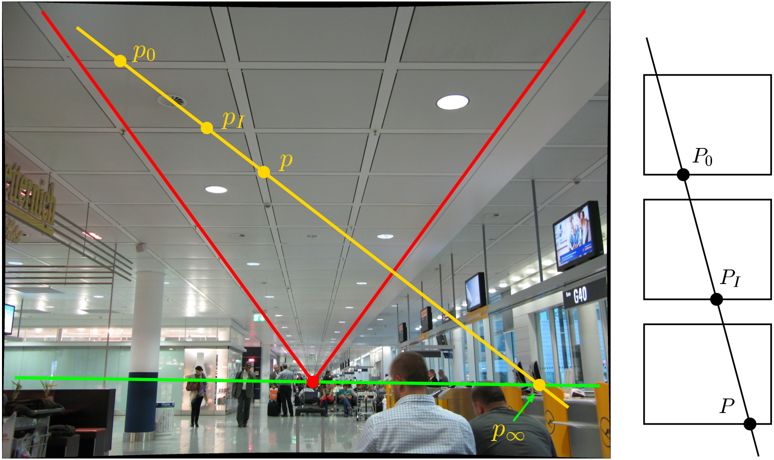


detail around the vanishing point

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

4Mpx camera

Application: Finding the Horizon from Repetitions



in 3D: $|P_0 P| = 2|P_0 P_I|$ then [H&Z, p. 218]⊛ P1; 1pt: How high is the camera above the floor?

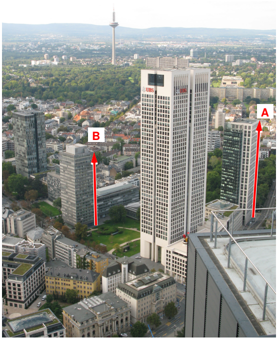
$$[P_\infty P_0 P_I P] = \frac{|P_0 P|}{|P_0 P_I|} = 2 \quad \Rightarrow \quad |p_\infty p_0| = \frac{|p_0 p_I| \cdot |p_0 p|}{|p_0 p| - 2|p_0 p_I|}$$

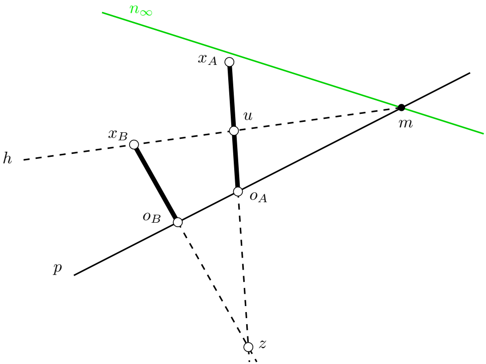
- could be applied to counting steps (Slide 45)

Homework Problem

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

- expected: conceptual solution
- deadline: +2 weeks

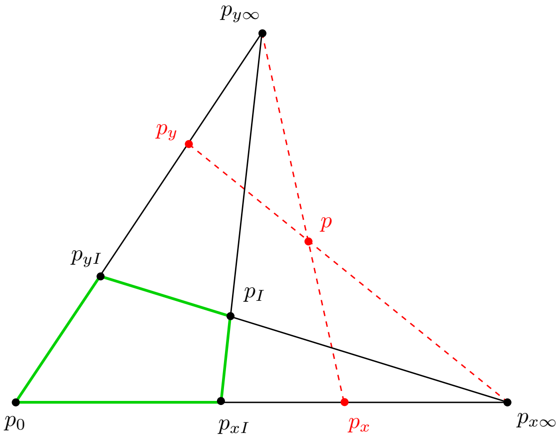




Hints

1. what are the properties of line h connecting the top of Buiding B with the point m at which the horizon is intersected with the line p joining the foots of both buildings? [1 point]
2. how do we actually get the horizon n_∞? [1 point] (we do not see it directly, there are hills there)
3. what tool measures the length? [formula = 1 point]

2D Projective Coordinates



$$[P_x] = [P_{x\infty}\ P_0\ P_{xI}\ P_x]$$

$$[P_y] = [P_{y\infty}\ P_0\ P_{yI}\ P_y]$$

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 see the calibrating object (vanishing points)

▶Real Camera with Radial Distortion

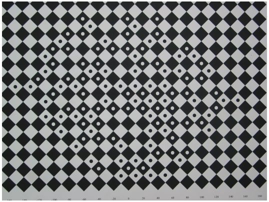
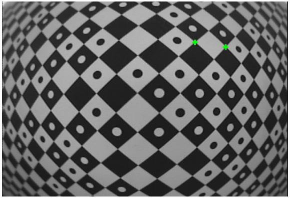


image with no radial distortion



an extreme case of radial distortion

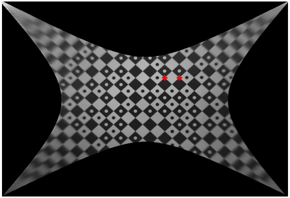
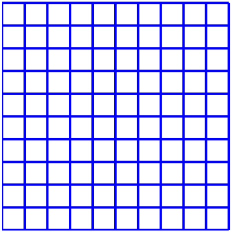
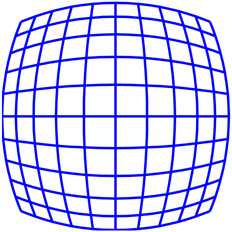


image undistorted by division model

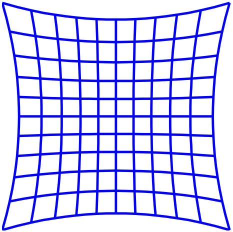
distortion types



none ($\lambda = 0$)

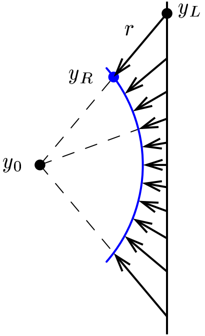


barrel ($\lambda = 0.3$)



pincushion ($\lambda = -0.3$)

►The Radial Distortion Mapping

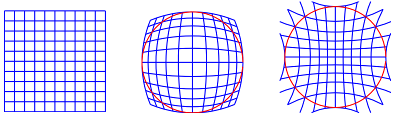


y_0 – center of radial distortion (usually principal point)

y_L – linearly projected point

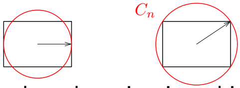
y_R – radially distorted point

- radial distortion r maps y_L to y_R along the radial direction
- magnitude of the transfer depends on the radius $\|y_L - y_0\|$ 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 does not scale

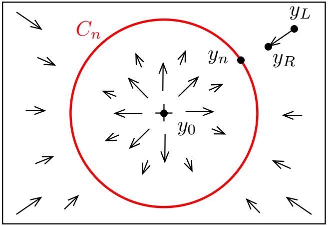
distortion	inside C_n	outside C_n
barrel	expanding	contracting
pincushion	contracting	expanding



in barrel   in pincushion

- choose boundary point that preserves all image content within the same image size

▶Radial Distortion Models



- 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 are often interested in $\mathbf{z}_L = r^{-1}(\mathbf{z}_R)$

- $\mathbf{y}_n$ – a no-distortion point on C_n: $r(\mathbf{y}_n) = \mathbf{y}_n$
- $\mathbf{z}_n = \mathbf{y}_n - \mathbf{y}_0$

Division Model

single parameter $-1 \leq \lambda < 1$, has an analytic inverse, models even some fish-eye lenses

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

$\lambda > 0$ – barrel distortion, $\lambda < 0$ – pincushion distortion

Polynomial Model

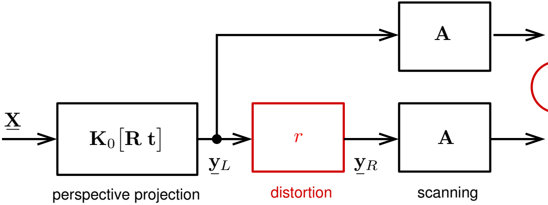
better fit for $n \geq 3$, no analytic inverse, may loose monotonicity, hard to calibrate

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

e.g. $k_i \geq 0$ – barrel distortion, $k_i \leq 0$ – pincusion distortion, $i = 1, \ldots, n$

Zernike polynomials R_i^0 are a better choice: $R_2^0(\rho) = 2\rho^2 - 1$, $R_4^0(\rho) = 6\rho^4 - 6\rho^2 + 1$, $R_6^0(\rho) = \cdots$

▶Real and Linear Camera Models



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

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

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

$$\mathbf{K}_0 = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ '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 (here, it includes conversion from/to homogeneous representation!)

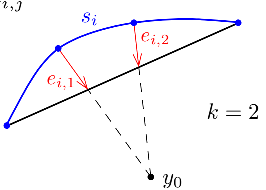
Notes

- 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: $\lambda,\ u_0,\ v_0,\ s,\ a$

1. detect a set of straight line segment images $\{s_i\}_{i=1}^n$ from a calibration target
2. select a suitable set of k measurement points per segment — how to select k?
3. define invariant radial transfer error per measurement point $e_{i,j}$ and per segment $e_i^2 = \sum_{j=1}^{k-2} e_{i,j}^2$ — invariant to rotation, translation



4. minimize total radial transfer error: $$\arg \min_{\lambda,\, u_0,\, v_0,\, s,\, a} \sum_{i=1}^{n} e_i^2$$

- line segments from real-world images requires segmentation to inliers/outliers

  inliers = lines that are straight in reality

- marginalisation over the hidden label gives a 'robust' error, e.g.

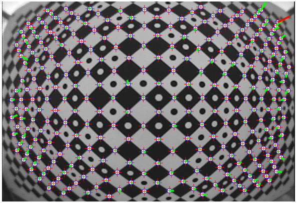
$$\varepsilon_i^2 = -\log\left(e^{-\frac{e_i^2}{2\sigma^2}} + t\right), \qquad t > 0$$

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

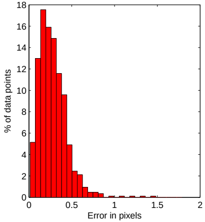
Example Calibrations

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

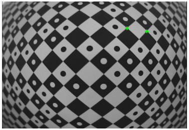
Camera 0, im. 6: Reprojection errors (16x)

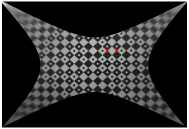


Camera 0: Error histogram.



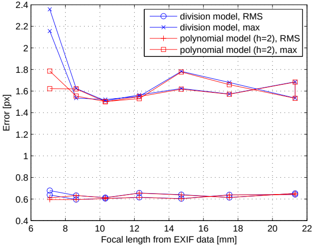
Cam	0
RMS [px]	0.33
max [px]	1.97
f [px]	94.59
a [-]	1.0951
u_0 [px]	243.26
v_0 [px]	353.37
(poly) k_1	0.8256
k_2	-0.2261
k_3	1.2524



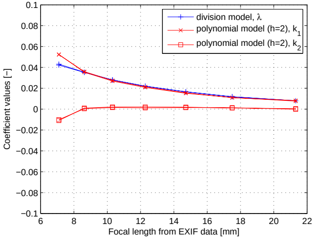


4 Mpix consumer camera

Calibration errors



Radial distortion coefficient values



- radial distortion is slightly dependend on focal length

Part III

Computing with a Single Camera

9 Calibration: Internal Camera Parameters from Vanishing Points and Lines

10 Resectioning: Projection Matrix from 6 Known Points

11 Exterior Orientation: Camera Rotation and Translation from 3 Known Points

covered by

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

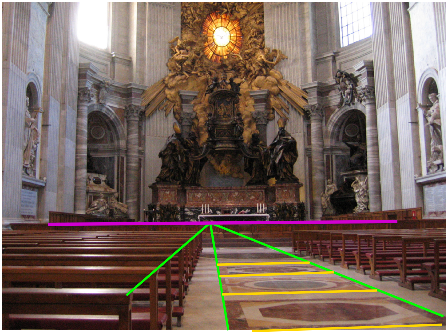
[2] Fischler, M.A. and Bolles, R.C . Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24(6):381–395, 1981

[3] [Golub & van Loan 1996, Sec. 2.5]

Obtaining Vanishing Points and Lines

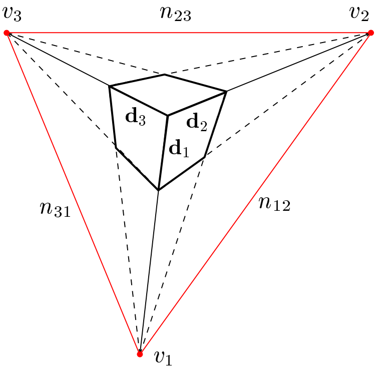
- orthogonal pairs can be collected from more images by camera rotation

- vanishing line can be obtained without vanishing points (see Slide 46)



▶Camera Calibration from Vanishing Points and Lines

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



$$\mathbf{d}_i = \mathbf{Q}^{-1}\underline{\mathbf{v}}_i, \qquad i = 1,2,3 \qquad \text{Slide 33}$$

$$\mathbf{p}_{ij} = \mathbf{Q}^{\top}\underline{\mathbf{n}}_{ij}, \qquad i,j = 1,2,3,\ i \neq j \qquad \text{Slide 36}$$

Constraints

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

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

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

▶cont'd

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

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

  λ can be eliminated from (4)

  we will come to solving overdetermined homogeneous equations later → Slide **??**
- we need at least 5 constraints for full $\mathbf{K}$
- 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 a set of quadratic equations for the parameters in $\mathbf{K}$

```
In[1]:= K = {{f, s, u[0]}, {0, a * f, v[0]}, {0, 0, 1}};
        K // MatrixForm
```

Out[2]//MatrixForm=

$$\begin{pmatrix} f & s & u[0] \\ 0 & a\,f & v[0] \\ 0 & 0 & 1 \end{pmatrix}$$

```
In[4]:= ω = Inverse[K.Transpose[K]] * Det[K]^2;
        ω // Simplify // MatrixForm
```

Out[5]//MatrixForm=

$$\begin{pmatrix} a^2 f^2 & -a f s & a f (-a f u[0] + s v[0]) \\ -a f s & f^2 + s^2 & a f s u[0] - (f^2 + s^2) v[0] \\ a f (-a f u[0] + s v[0]) & a f s u[0] - (f^2 + s^2) v[0] & a^2 f^2 (f^2 + u[0]^2) - 2 a f s u[0] v[0] + (f^2 + s^2) v[0]^2 \end{pmatrix}$$

```
In[8]:= ω / f^2 /. s → 0 // Simplify // MatrixForm
```

Out[8]//MatrixForm=

$$\begin{pmatrix} a^2 & 0 & -a^2 u[0] \\ 0 & 1 & -v[0] \\ -a^2 u[0] & -v[0] & a^2 (f^2 + u[0]^2) + v[0]^2 \end{pmatrix}$$

```
In[10]:= ω /. {u[0] → 0, v[0] → 0} // MatrixForm
```

Out[10]//MatrixForm=

$$\begin{pmatrix} a^2 f^2 & -a f s & 0 \\ -a f s & f^2 + s^2 & 0 \\ 0 & 0 & a^2 f^4 \end{pmatrix}$$

```
In[17]:= ω / f^2 /. {a → 1, s → 0} // Simplify // MatrixForm
```

Out[17]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & -u[0] \\ 0 & 1 & -v[0] \\ -u[0] & -v[0] & f^2 + u[0]^2 + v[0]^2 \end{pmatrix}$$

Examples

Ex 1:

Assuming known $m_0 = (u_0, v_0)$, two finite orthogonal vanishing points suffice to get f

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

$$f^2 = \left|(\mathbf{v}_1 - \mathbf{m}_0)^\top (\mathbf{v}_2 - \mathbf{m}_0)\right|$$

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

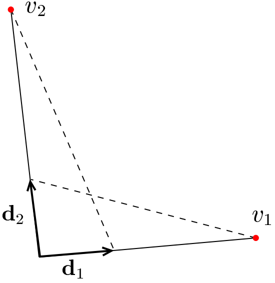
$$\boldsymbol{\omega} = \frac{1}{f^2}\begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ -u_0 & -v_0 & f^2 + u_0^2 + v_0^2 \end{bmatrix}$$

- 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 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.



- coordinate system choice, e.g.:

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

- we know that

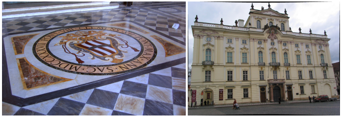
$$\mathbf{d}_i \simeq \mathbf{Q}^{-1}\underline{\mathbf{v}}_i = (\mathbf{K}\mathbf{R})^{-1}\underline{\mathbf{v}}_i = \mathbf{R}^{-1}\underbrace{\mathbf{K}^{-1}\underline{\mathbf{v}}_i}_{\underline{\mathbf{w}}_i}.$$

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

- then $\underline{\mathbf{w}}_i / \|\underline{\mathbf{w}}_i\|$ is the i–th column $\mathbf{r}_i$ of $\mathbf{R}$
- the third column is orthogonal:
  $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$

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

some suitable scenes



Application: Planar Rectification

Principle: Rotate camera parallel to the plane of interest.

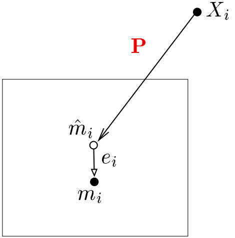
$$\underline{\mathbf{m}} \simeq \mathbf{K}\mathbf{R}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix}\underline{\mathbf{X}} \qquad \underline{\mathbf{m}}' \simeq \mathbf{K}\begin{bmatrix}\mathbf{I} & -\mathbf{C}\end{bmatrix}\underline{\mathbf{X}}$$

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

- $\mathbf{H}$ is the rectifying homography
- both $\mathbf{K}$ and $\mathbf{R}$ can be calibrated from two finite vanishing points
- not possible when one (or both) of them are infinite

▶Camera Resectioning

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



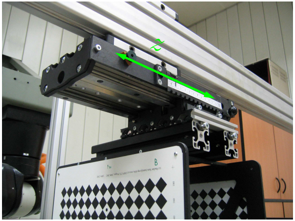
X_i is considered exact

m_i is a measurement

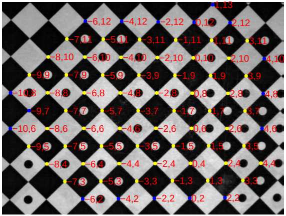
$e_i^2 = \|\mathbf{m}_i - \hat{\mathbf{m}}_i\|^2$

where $\underline{\hat{\mathbf{m}}}_i \simeq \mathbf{P}\underline{\mathbf{X}}_i$

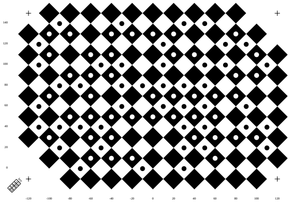
projection error



calibration target with translation stage



automatic calibration point detection



calibration chart

▶The Minimal Problem for Resectioning

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

$$\lambda_i \underline{\mathbf{m}}_i = \mathbf{P}\underline{\mathbf{X}}_i, \qquad \mathbf{P} = \begin{bmatrix} \mathbf{q}_1^\top & q_{14} \\ \mathbf{q}_2^\top & q_{24} \\ \mathbf{q}_3^\top & q_{34} \end{bmatrix} \qquad \begin{array}{l} \underline{\mathbf{X}}_i = (x_i, y_i, z_i, 1), \quad i = 1, 2, \ldots, k, \ k = 6 \\ \underline{\mathbf{m}}_i = (u_i, v_i, 1), \quad \lambda_i \in \mathbb{R}, \ \lambda_i \neq 0 \end{array}$$

easy to modify for infinite points X_i

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

eliminating λ gives: $(\mathbf{q}_3^\top \mathbf{X}_i + q_{34}) u_i = \mathbf{q}_1^\top \mathbf{X}_i + q_{14}, \quad (\mathbf{q}_3^\top \mathbf{X}_i + q_{34}) v_i = \mathbf{q}_2^\top \mathbf{X}_i + q_{24}$

Then

$$\mathbf{A}\,\mathbf{q} = \begin{bmatrix} \mathbf{X}_1^\top & 1 & \mathbf{0}^\top & 0 & -u_1 \mathbf{X}_1^\top & -u_1 \\ \mathbf{0}^\top & 0 & \mathbf{X}_1^\top & 1 & -v_1 \mathbf{X}_1^\top & -v_1 \\ \vdots & & & & & \vdots \\ \mathbf{X}_k^\top & 1 & \mathbf{0}^\top & 0 & -u_k \mathbf{X}_k^\top & -u_k \\ \mathbf{0}^\top & 0 & \mathbf{X}_k^\top & 1 & -v_k \mathbf{X}_k^\top & -v_k \end{bmatrix} \cdot \begin{bmatrix} \mathbf{q}_1 \\ q_{14} \\ \mathbf{q}_2 \\ q_{24} \\ \mathbf{q}_3 \\ q_{34} \end{bmatrix} = \mathbf{0} \tag{8}$$

- 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 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 data?

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. normalize $\mathbf{q}_i$ to $\hat{\mathbf{q}}_i = \mathbf{q}_i / q_{12}$ — this assumes finite camera with $P_{3,3} = 1$
3. from all n vectors $\hat{\mathbf{q}}_i$ collected in Step 1d compute

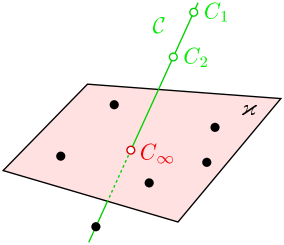
$$\mathbf{q} = \frac{1}{n} \sum_{i=1}^{n} \hat{\mathbf{q}}_i, \qquad \operatorname{var}[\mathbf{q}] = \frac{n-1}{n} \operatorname{diag} \sum_{i-1}^{n} (\hat{\mathbf{q}}_i - \mathbf{q})(\hat{\mathbf{q}}_i - \mathbf{q})^\top$$

- have a solution + an error estimate, per individual elements of $\mathbf{P}$
- at least 5 points must be in a general position — see Slide 67
- large error indicates near degeneracy
- computation not efficient with $k > 6$ points, needs $\binom{2k}{11}$ draws, e.g. $k = 7 \to 364$ draws
- one needs $k \geq 7$ for the full covariance matrix
- better error estimation method: decompose $\mathbf{P}_i$ to $\mathbf{K}_i$, $\mathbf{R}_i$, $\mathbf{t}_i$ (Slide 30), represent $\mathbf{R}_i$ with 3 parameters (e.g. Euler angles, or in Cayley representation, see Slide 136) and compute the errors for the parameters

▶Degenerate (Critical) Configurations for Resectioning

Let $\mathcal{X} = \{X_i;\ i = 1, \ldots\}$ be a set of points and $\mathbf{P}_1 \not\simeq \mathbf{P}_2$ be two regular (rank-3) cameras. Then two configurations $(\mathbf{P}_1, \mathcal{X})$ and $(\mathbf{P}_2, \mathcal{X})$ are image-equivalent if

$$\mathbf{P}_1 \underline{\mathbf{X}}_i \simeq \mathbf{P}_2 \underline{\mathbf{X}}_i \quad \text{for all} \quad X_i \in \mathcal{X}$$



Case 4

- if all calibration points $X_i \in \mathcal{X}$ lie on a plane $\varkappa$ the camera resectioning 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

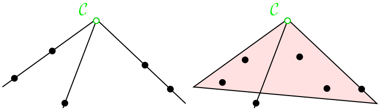
  this also means we cannot resect if all X_i are infinite
- by adding points $X_i \in \mathcal{X}$ to $\mathcal{C}$ we gain nothing
- there are additional image-equivalent configurations, see next

see proof sketch in the notes or in [H&Z, Sec. 22.1.2]

Note that if $\mathbf{Q}$, $\mathbf{T}$ are suitable non-singular homographies then $\mathbf{P}_1 \simeq \mathbf{Q}\mathbf{P}_0\mathbf{T}$, where $\mathbf{P}_0$ is canonical and

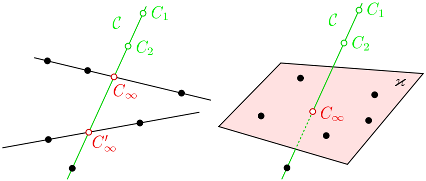
$$\mathbf{P}_0 \underbrace{\mathbf{T}\underline{\mathbf{X}}_i}_{\underline{\mathbf{Y}}_i} \simeq \mathbf{P}_2 \underbrace{\mathbf{T}\underline{\mathbf{X}}_i}_{\underline{\mathbf{Y}}_i} \quad \text{for all} \quad Y_i \in \mathcal{Y}$$

cont'd (all cases)



Case 5 Case 6

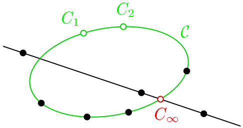
- cameras C_1, C_2 co-located at point $\mathcal{C}$
- points on three optical rays or one optical ray and one optical plane
- Case 5: we see 3 isolated point images
- Case 6: we see a line of points and an isolated point



Case 3 **Case 4**

- cameras lie on a line $\mathcal{C} \setminus \{C_\infty, C'_\infty\}$
- points lie on $\mathcal{C}$ and
  1. on two lines meeting $\mathcal{C}$ at C_∞, C'_∞
  2. or on a plane meeting $\mathcal{C}$ at C_∞
- Case 3: we see 2 lines of points

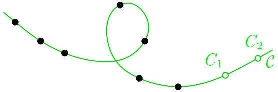
Case 2



- cameras lie on a planar conic $\mathcal{C} \setminus \{C_\infty\}$

  not necessarily an ellipse
- points lie on $\mathcal{C}$ and an additional line meeting the conic at C_∞
- Case 2: we see 2 lines of points

Case 1



- cameras and points all lie on a twisted cubic $\mathcal{C}$
- Case 1: we see a conic

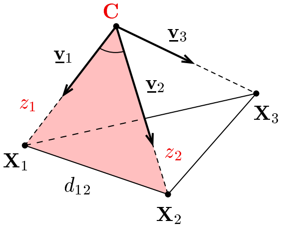
▶Three-Point Exterior Orientation Problem (P3P)

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

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

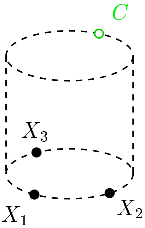
configuration w/o rotation



1. Transform $\underline{\mathbf{v}}_i \stackrel{\text{def}}{=} \mathbf{K}^{-1}\underline{\mathbf{m}}_i$. Then
$$\lambda_i \underline{\mathbf{v}}_i = \mathbf{R}\,(\mathbf{X}_i - \mathbf{C}). \tag{9}$$
2. Eliminate $\mathbf{R}$ by taking rotation preserves length: $\|\mathbf{R}\mathbf{x}\| = \|\mathbf{x}\|$
$$|\lambda_i| \cdot \|\underline{\mathbf{v}}_i\| = \|\mathbf{X}_i - \mathbf{C}\| \stackrel{\text{def}}{=} z_i \tag{10}$$
3. Consider only angles among $\underline{\mathbf{v}}_i$ and apply Cosine Law per triangle $(\mathbf{C}, \mathbf{X}_i, \mathbf{X}_j)$ $i, j = 1, 2, 3,\ i \neq j$
$$d_{ij}^2 = z_i^2 + z_j^2 - 2\, z_i\, z_j\, c_{ij},$$
$$z_i = \|\mathbf{X}_i - \mathbf{C}\|,\ d_{ij} = \|\mathbf{X}_j - \mathbf{X}_i\|,\ c_{ij} = \cos(\angle \underline{\mathbf{v}}_i\, \underline{\mathbf{v}}_j)$$
4. Solve system of 3 quadratic eqs in 3 unknowns z_i [Fischler & Bolles, 1981]
there may be no real root; there are up to 4 solutions that cannot be ignored
(verify on additional points)
5. Compute $\mathbf{C}$ by trilateration (3-sphere intersection) from $\mathbf{X}_i$ and z_i; then λ_i from (10) and $\mathbf{R}$ from (9)

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

Degenerate (Critical) Configurations for Exterior Orientation



unstable solution

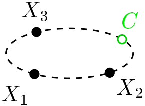
- center of projection C located on the orthogonal circular cylinder with base circumscribing the three points X_i

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)

unstable: a small change of X_i results in a large change of C

can be detected by error propagation



no solution

1. C cocyclic with (X_1, X_2, X_3)

- additional critical configurations depend on the method to solve the quadratic equations

[Haralick et al. IJCV 1994]

▶Populating A Little ZOO of Minimal Geometric Problems in CV

problem	given	unknown	slide
resectioning	6 world–img correspondences $\left\{(X_i,\, m_i)\right\}_{i=1}^{6}$	$\mathbf{P}$	65
exterior orientation	$\mathbf{K}$, 3 world–img correspondences $\left\{(X_i,\, m_i)\right\}_{i=1}^{3}$	$\mathbf{R}$, $\mathbf{C}$	69

- resectioning 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
- more problems to come

Part IV

Computing with a Camera Pair

12 Camera Motions Inducing Epipolar Geometry

13 Estimating Fundamental Matrix from 7 Correspondences

14 Estimating Essential Matrix from 5 Correspondences

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

covered by

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

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

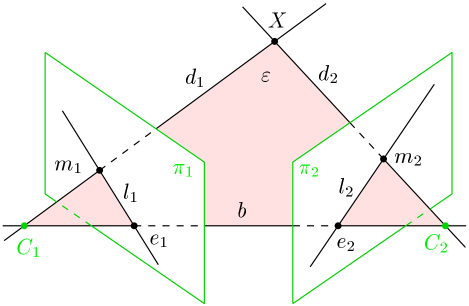
additional references

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

▶Geometric Model of a Camera Pair

Epipolar geometry:

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



two-camera setup

Description

- <u>baseline</u> b joins projection centers C_1, C_2

$$\mathbf{b} = \mathbf{C}_2 - \mathbf{C}_1$$

- <u>epipole</u> $e_i \in \pi_i$ is the image of C_j:

$$\underline{\mathbf{e}}_1 \simeq \mathbf{P}_1 \underline{\mathbf{C}}_2, \quad \underline{\mathbf{e}}_2 \simeq \mathbf{P}_2 \underline{\mathbf{C}}_1$$

- $l_i \in \pi_i$ is the image of <u>epipolar plane</u>

$$\varepsilon = (C_2, X, C_1)$$

- l_j is the <u>epipolar line</u> in image π_j induced by m_i in image π_i

Epipolar constraint: d_2, b, d_1 are coplanar

a necessary condition, see also Slide 83

►Cross Products and Maps by Antisymmetric 3×3 Matrices

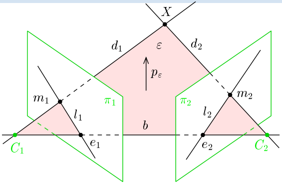
- There is an equivalence $\mathbf{b} \times \mathbf{m} = [\mathbf{b}]_\times \mathbf{m}$, where $[\mathbf{b}]_\times$ is a 3×3 antisymmetric matrix

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

Some properties

1. $[\mathbf{b}]_\times^\top = -[\mathbf{b}]_\times$ — the general antisymmetry property
2. $\|[\mathbf{b}]_\times\|_F = \sqrt{2}\,\|\mathbf{b}\|$ — Frobenius norm ($\|\mathbf{A}\|_F^2 = \sum_{i,j} |a_{ij}|^2$)
3. $[\mathbf{b}]_\times \mathbf{b} = \mathbf{0}$
4. $\operatorname{rank}[\mathbf{b}]_\times = 2$ iff $\|\mathbf{b}\| > 0$ — check minors of $[\mathbf{b}]_\times$
5. if $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$ then $[\mathbf{R}\mathbf{b}]_\times = \mathbf{R}\,[\mathbf{b}]_\times \mathbf{R}^\top$
6. $[\mathbf{B}\mathbf{z}]_\times \simeq \mathbf{B}^{-\top}[\mathbf{z}]_\times \mathbf{B}^{-1}$ — in general, $[\mathbf{A}^{-1}\mathbf{t}]_\times \cdot \det \mathbf{A} = \mathbf{A}^\top [\mathbf{t}]_\times \mathbf{A}$
7. if $\mathbf{R}_b$ is rotation about $\mathbf{b}$ then $[\mathbf{R}_b \mathbf{b}]_\times = [\mathbf{b}]_\times$

▶Expressing Epipolar Constraint Algebraically



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

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

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

remember: $\mathbf{C} = -\mathbf{Q}^{-1}\mathbf{q} = -\mathbf{R}^\top \mathbf{t}$ (Slides 30 and 32)

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

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

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

$$\underline{\mathbf{e}}_1 \simeq \mathbf{Q}_1 \mathbf{C}_2 + \mathbf{q}_1 = \mathbf{Q}_1 \mathbf{C}_2 - \mathbf{Q}_1 \mathbf{C}_1 = \mathbf{K}_1 \mathbf{R}_1 \mathbf{b}$$

$$\mathbf{F} = \mathbf{Q}_2^{-\top}\mathbf{Q}_1^\top [\underline{\mathbf{e}}_1]_\times = \mathbf{Q}_2^{-\top}\mathbf{Q}_1^\top [\mathbf{K}_1 \mathbf{R}_1 \mathbf{b}]_\times = \overset{\circledast\, 1}{\cdots} = \mathbf{K}_2^{-\top} \underbrace{[-\mathbf{t}_{21}]_\times \mathbf{R}_{21}}_{\text{essential matrix } \mathbf{E}} \mathbf{K}_1^{-1}$$ Slide 74

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

Epipole is the Image of the Other Camera

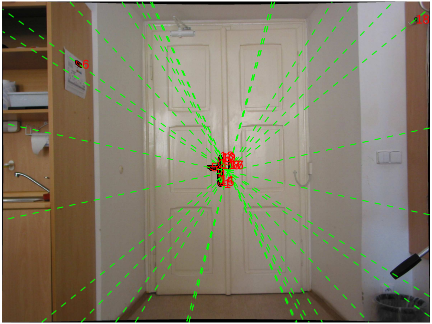


image 1

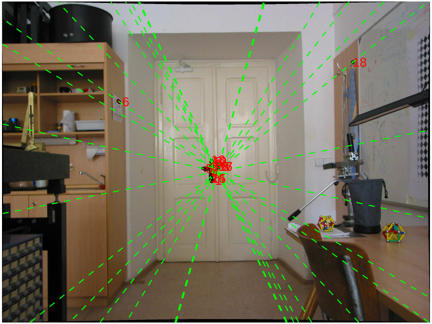
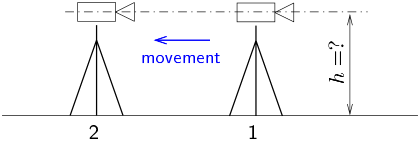
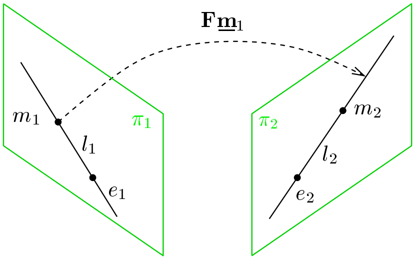


image 2

Camera moved horizontally: How high is it above floor?



►A Summary of Epipolar Constraint



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

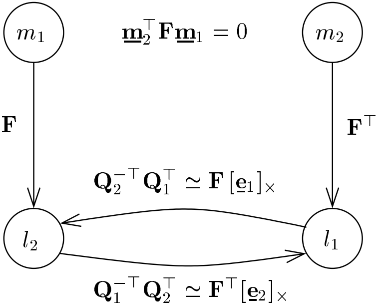
$$\mathbf{F} \simeq \mathbf{K}_2^{-\top} \mathbf{E}\, \mathbf{K}_1^{-1}$$

$$\mathbf{E} \simeq [-\mathbf{t}_{21}]_\times \mathbf{R}_{21} = [\mathbf{R}_2 \mathbf{b}]_\times \mathbf{R}_{21} = \mathbf{R}_{21} [\mathbf{R}_1 \mathbf{b}]_\times$$

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

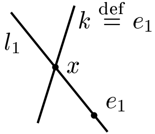
- $\mathbf{E}$ captures the relative pose [Longuet-Higgins 1981]
- the translation length $\mathbf{t}_{21}$ is lost

$\mathbf{E}$ is homogeneous



proof of $\underline{\mathbf{l}}_2 \simeq \mathbf{F}\, [\underline{\mathbf{e}}_1]_\times\, \underline{\mathbf{l}}_1$: line/point transmutation

$$\underline{\mathbf{l}}_2 \simeq \mathbf{F}\, \underline{\mathbf{x}} \simeq \mathbf{F}(\underline{\mathbf{k}} \times \underline{\mathbf{l}}_1) = \mathbf{F}\, [\underline{\mathbf{k}}]_\times\, \underline{\mathbf{l}}_1 = \mathbf{F}\, [\underline{\mathbf{e}}_1]_\times\, \underline{\mathbf{l}}_1$$



$x \neq e_1$, $e_1 \notin k$: $\underline{\mathbf{k}}^\top \underline{\mathbf{e}}_1 = \|\underline{\mathbf{e}}_1\|^2 \neq 0$

▶The Representation Theorem for Essential Matrices

Let $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ s.t. $\mathbf{D} = \mathrm{diag}(1,1,0)$ then $\mathbf{E} \simeq [\mathbf{u}_3]_\times \mathbf{R}$, where $\mathbf{R}$ is orthogonal   nonunique!

Proof.

We introduce $\mathbf{W} = \begin{bmatrix} 0 & \alpha & 0 \\ -\alpha & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ s.t. $|\alpha| = 1$ (rotation by $\pm 90°$). Then

$$\mathbf{U}\mathbf{D}\mathbf{V}^\top = \underbrace{\mathbf{U}\mathbf{D}\mathbf{W}^\top}\mathbf{W}\mathbf{V}^\top = \underbrace{\mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & -\alpha & 0 \\ \alpha & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{U}\begin{bmatrix} 0 & -\alpha & 0 \\ \alpha & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{U}\begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix}_\times = \alpha[\mathbf{u}_3]_\times \mathbf{U}} \mathbf{W}\mathbf{V}^\top = \alpha[\mathbf{u}_3]_\times \underbrace{\mathbf{U}\mathbf{W}\mathbf{V}^\top}_{\mathbf{R}}$$

→Slide 74

□

- we needed rotation $\mathbf{W}$ s.t. $\mathbf{D}\mathbf{W}^\top$ is antisymmetric, the choice is unique up to sign α

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 $\mathbf{D} \simeq \mathrm{diag}(1,1,0)$.

Proof.

Direct implication above. Converse: Let $\mathbf{U}\mathbf{D}\mathbf{V}^\top$ be an SVD with $\mathbf{D} = \mathrm{diag}(1,1,0)$. Then

$$\mathbf{U}\mathbf{D}\mathbf{V}^\top = \underbrace{\mathbf{U}\mathbf{D}\mathbf{W}^\top}_{[\mathbf{u}_3]_\times \mathbf{U}}\mathbf{W}\mathbf{V}^\top$$

□

▶Essential Matrix Decomposition

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

1. compute SVD of $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ s.t. $\mathbf{D} = \operatorname{diag}(1,1,0)$
2. if $\det \mathbf{U} < 0$ transform it to $-\mathbf{U}$, do the same for $\mathbf{V}$ — the overall sign is dropped
3. compute

$$\mathbf{R}_{21} = \mathbf{U} \underbrace{\begin{bmatrix} 0 & \alpha & 0 \\ -\alpha & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{W}} \mathbf{V}^\top, \quad \mathbf{t}_{21} = -\beta\, \mathbf{u}_3, \qquad |\alpha| = 1, \quad \beta \neq 0 \qquad (11)$$

Notes

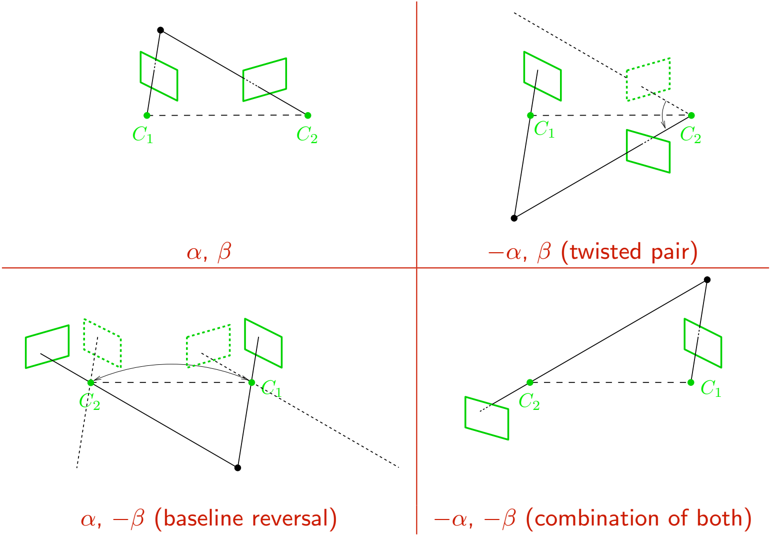
- $\mathbf{t}_{21}$ recoverable up to scale β and direction $\operatorname{sign} \beta$
- the result for $\mathbf{R}_{21}$ is unique up to $\alpha = \pm 1$ — despite non-uniqueness of SVD
- change of sign in $\mathbf{W}$ rotates the solution by 180° about $\mathbf{t}$

  $\mathbf{R}_1 = \mathbf{U}\mathbf{W}\mathbf{V}^\top$, $\mathbf{R}_2 = \mathbf{U}\mathbf{W}^\top\mathbf{V}^\top \Rightarrow \mathbf{T} = \mathbf{R}_2\mathbf{R}_1^\top = \cdots = \mathbf{U}\operatorname{diag}(-1,-1,1)\mathbf{U}^\top$ which is a rotation by 180° about $\mathbf{u}_3 = \mathbf{t}_{21}$:

$$\mathbf{U}\operatorname{diag}(-1,-1,1)\mathbf{U}^\top \mathbf{u}_3 = \mathbf{U} \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{u}_3$$

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

▶Four Solutions to Essential Matrix Decomposition



- chirality constraint: all 3D points are in front of both cameras
- this singles-out the upper left case [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 $k = 7$ correspondences, estimate f. m. $\mathbf{F}$.

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

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

Solution:

$$\mathbf{D} = \begin{bmatrix} u_1^1 u_1^2 & u_1^1 v_1^2 & u_1^1 & u_1^2 v_1^1 & v_1^1 v_1^2 & v_1^1 & u_1^2 & v_1^2 & 1 \\ u_2^1 u_2^2 & u_2^1 v_2^2 & u_2^1 & u_2^2 v_2^1 & v_2^1 v_2^2 & v_2^1 & u_2^2 & v_2^2 & 1 \\ u_3^1 u_3^2 & u_3^1 v_3^2 & u_3^1 & u_3^2 v_3^1 & v_3^1 v_3^2 & v_3^1 & u_3^2 & v_3^2 & 1 \\ \vdots & & & & & & & & \vdots \\ u_k^1 u_k^2 & u_k^1 v_k^2 & u_k^1 & u_k^2 v_k^1 & v_k^1 v_k^2 & v_k^1 & u_k^2 & v_k^2 & 1 \end{bmatrix} \mathbf{D} \in \mathbb{R}^{k,9}$$

$$\mathbf{D}\mathbf{f} = \mathbf{0}, \qquad \mathbf{f} = \begin{bmatrix} f_{11} & f_{21} & f_{31} & \ldots & f_{33} \end{bmatrix}^\top, \qquad \mathbf{f} \in \mathbb{R}^9,$$

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

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

  3. get up to 3 fundamental matrices $\mathbf{F} = \alpha_i \mathbf{F}_1 + (1 - \alpha_i)\mathbf{F}_2$

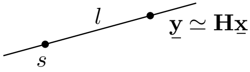
- the result may depend on image transformations
- normalization improves conditioning → Slide 88
- this gives a good starting point for the full algorithm → Slide 104
- dealing with mismatches need not be a part of the 7-point algorithm → Slide 105

▶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 $C_1 = C_2$: $\quad \mathbf{H} = \mathbf{K}_2 \mathbf{R}_{21} \mathbf{K}_1^{-1}$
   b. camera moves but all 3D points lie in a plane $(\mathbf{n}, d)$: $\quad \mathbf{H} = \mathbf{K}_2 (\mathbf{R}_{21} - \mathbf{t}_{21} \mathbf{n}^\top / d) \mathbf{K}_1^{-1}$

   - epipolar geometry is not defined
   - we do get an $\mathbf{F}$ from the 7-point algorithm but it is of the form of $\mathbf{F} = [\mathbf{s}]_\times \mathbf{H}$ with $\mathbf{s}$ arbitrary (nonzero)



- correspondence $x \leftrightarrow y$
- y is the image of x: $\underline{\mathbf{y}} \simeq \mathbf{H} \underline{\mathbf{x}}$
- this can be written as $y \in l$, $\quad \underline{\mathbf{l}} \simeq \underline{\mathbf{s}} \times \mathbf{H} \underline{\mathbf{x}}$ arbitrary $\mathbf{s}$

$$0 = \underline{\mathbf{y}}^\top (\underline{\mathbf{s}} \times \mathbf{H} \underline{\mathbf{x}}) = \underline{\mathbf{y}}^\top [\underline{\mathbf{s}}]_\times \mathbf{H} \underline{\mathbf{x}}$$

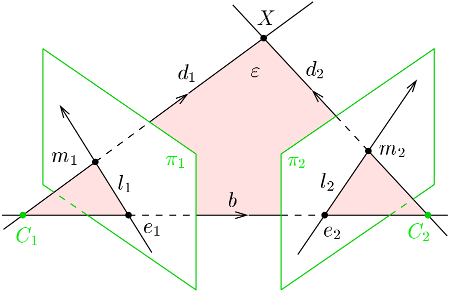
2. both camera centers and all 3D points lie on a ruled quadric
   hyperboloid of one sheet, cones, cylinders, two planes
   - there are 3 solutions for $\mathbf{F}$

notes

- estimation of $\mathbf{E}$ can deal with planes: $[\mathbf{s}]_\times \mathbf{H} = [\mathbf{s}]_\times (\mathbf{R}_{21} - \mathbf{t}_{21} \mathbf{n}^\top / d)$ has equal eigenvalues iff $\mathbf{s} = \mathbf{t}_{21}$, the decomposition works (nonunique, as before) ⊛ 1pt for a proof
- a complete treatment with additional degenerate configurations in [H&Z, sec. 22.2]
- a stronger epipolar constraint could reject some configurations

A Note on Oriented Epipolar Constraint

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



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

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

- 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 Slide 105 — [Chum et al. 2004]
- an even more tight constraint: scene points in front of both cameras — expensive

this is called chirality constraint

▶Five-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. $\mathbf{R}$ – 3DOF, $\mathbf{t}$ – we can recover 2DOF only, in total 5 DOF $\rightarrow$ we need 3 constraints on $\mathbf{E}$
2. real $\mathbf{F} \in \mathbb{R}^{3,3}$ is a fundamental matrix iff $\det \mathbf{F} = 0$
3. fundamental matrix is essential iff its two non-zero eigenvalues are equal

This gives an equation system:

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

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

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

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

- when all 3D points lie on a plane: at most 2 solutions (twisted-pair)
  can be disambiguated in 3 views
  or by chirality constraint (Slide 80) unless all 3D points are closer to one camera
- 6-point problem for unknown f [Kukelova et al. BMVC 2008]
- resources at `http://cmp.felk.cvut.cz/minimal/5_pt_relative.php`

►The Triangulation Problem

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

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

Linear triangulation method

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

Gives

$$\mathbf{D}\underline{\mathbf{X}} = \mathbf{0}, \qquad \mathbf{D} = \begin{bmatrix} u^1\, (\mathbf{p}_3^1)^\top - (\mathbf{p}_1^1)^\top \\ v^1\, (\mathbf{p}_3^1)^\top - (\mathbf{p}_2^1)^\top \\ u^2\, (\mathbf{p}_3^2)^\top - (\mathbf{p}_1^2)^\top \\ v^2\, (\mathbf{p}_3^2)^\top - (\mathbf{p}_2^2)^\top \end{bmatrix}, \qquad \mathbf{D} \in \mathbb{R}^{4,4}, \quad \underline{\mathbf{X}} \in \mathbb{R}^4 \tag{12}$$

- back-projected rays will generally not intersect due to image error, see next
- using Jack-knife (Slide 66) not recommended — sensitive to small error
- we will use SVD (Slide 86)
- 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 $\underline{\mathbf{X}} \mapsto \mathbf{H}^{-1}\underline{\mathbf{X}}$
- the homogeneous form in (12) can represent points 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 $\mathbf{D}$, then

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

- we write the SVD of $\mathbf{Q}$ as $\mathbf{Q} = \sum_{j=1}^{4} \sigma_j^2\, \mathbf{u}_j \mathbf{u}_j^\top$, in which [Golub & van Loan 1996, Sec. 2.5]

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

- then

$$\underline{\mathbf{X}} = \arg\min_{\mathbf{q},\|\mathbf{q}\|=1} \mathbf{q}^\top \mathbf{Q}\,\mathbf{q} = \mathbf{u}_4, \qquad \mathbf{q}^\top \mathbf{Q}\,\mathbf{q} = \sum_{j=1}^{4} \sigma_j^2\, \mathbf{q}^\top \mathbf{u}_j\, \mathbf{u}_j^\top \mathbf{q} = \sum_{j=1}^{4} \sigma_j^2\, (\mathbf{u}_j^\top \mathbf{q})^2$$

we have a sum of non-negative elements $0 \leq (\mathbf{u}_j^\top \mathbf{q})^2 \leq 1$, let $\mathbf{q} = \mathbf{u}_4 + \bar{\mathbf{q}}$ s.t. $\bar{\mathbf{q}} \perp \mathbf{u}_4$, then

$$\mathbf{q}^\top \mathbf{Q}\,\mathbf{q} = \sigma_4^2 + \sum_{j=1}^{3} \sigma_j^2\, (\mathbf{u}_j^\top \bar{\mathbf{q}})^2 \geq \sigma_4^2$$

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

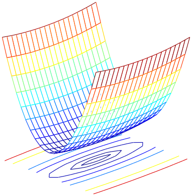
Matlab code for the least-squares solver:

```
[U,O,V] = svd(D);
X = V(:,end);
q = O(3,3)/O(4,4);
```

⊛ P1; 2pt: Why did we decompose $\mathbf{D}$ and not $\mathbf{Q} = \mathbf{D}^\top\mathbf{D}$? Could we use QR decomposition instead of SVD?

▶Numerical Conditioning

- The equation $\mathbf{D}\underline{\mathbf{X}} = \mathbf{0}$ in (12) 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 $\mathbf{S} \in \mathbb{R}^{4,4}$

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

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

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

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

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

- when SVD is used in camera resectioning, conditioning is essential for success   → Slide 65

Algebraic Error vs Reprojection Error

- algebraic residual error: from SVD → Slide 87

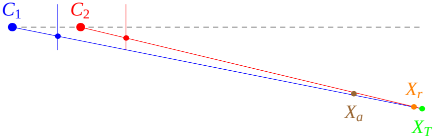
$$\varepsilon^2 = \sigma_4^2 = \sum_{c=1}^{2} \left[\left(u^c (\mathbf{p}_3^c)^\top \underline{\mathbf{X}} - (\mathbf{p}_1^c)^\top \underline{\mathbf{X}} \right)^2 + \left(v^c (\mathbf{p}_3^c)^\top \underline{\mathbf{X}} - (\mathbf{p}_2^c)^\top \underline{\mathbf{X}} \right)^2 \right]$$

- reprojection error

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

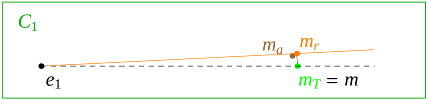
- algebraic error zero ⇒ reprojection error zero $\sigma_4 = 0 \Rightarrow$ non-trivial null space
- epipolar constraint satisfied ⇒ equivalent results
- in general: minimizing algebraic error cheap but it gives inferior results
- minimizing reprojection error expensive but it gives good results
- the gold standard method – deferred to Slide 100

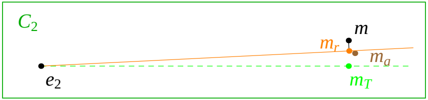
Ex:



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

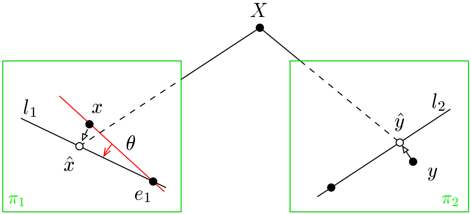
X_T – noiseless ground truth position
X_r – reprojection error minimizer
X_a – algebraic error minimizer
m – measurement (m_T with noise in v^2)





Optimal Triangulation for the Geeks

- detected image points x, y do not satisfy epipolar geometry exactly
- as a result optical rays do not intersect in space, we must correct the image points to $\hat{x}$, $\hat{y}$ first



1. given epipolar line l_1 and l_2, $\underline{\mathbf{l}}_2 \simeq \mathbf{F}[\underline{\mathbf{e}}_1]_\times \underline{\mathbf{l}}_1$ the $\hat{x}$, $\hat{y}$ are the closest points on l_1, l_2
2. parameterize all possible l_1 by θ
   - find θ after translating $\underline{\mathbf{x}}$, $\underline{\mathbf{y}}$ to $(0, 0, 1)$, rotating the epipoles to $(1, 0, f_1)$, $(1, 0, f_2)$, and parameterising $\underline{\mathbf{l}}_1 = (0, \theta, 1) \times (1, 0, f_1)$
3. minimise the error

$$\theta^* = \arg\min_\theta d^2\big(x, l_1(\theta)\big) + d^2\big(y, l_2(\theta)\big)$$

the problem reduces to 6-th degree polynomial root finding, see [H&Z, Sec 12.5.2]

4. compute $\hat{x}$, $\hat{y}$ and triangulate using the linear method on Slide 85

- the midpoint of the common perpendicular to both optical rays gives about 50% greater error in 3D
- a fully optimal procedure requires error re-definition in order to get the most probable $\hat{x}$, $\hat{y}$

▸We Have Added to The ZOO

Continuation from Slide 71

problem	given	unknown	slide
resectioning	6 world–img correspondences $\{(X_i, m_i)\}_{i=1}^{6}$	$\mathbf{P}$	65
exterior orientation	$\mathbf{K}$, 3 world–img correspondences $\{(X_i, m_i)\}_{i=1}^{3}$	$\mathbf{R}$, $\mathbf{C}$	69
fundamental matrix	7 img–img correspondences $\{(m_i, m'_i)\}_{i=1}^{7}$	$\mathbf{F}$	81
relative orientation	$\mathbf{K}$, 5 img–img correspondences $\{(m_i, m'_i)\}_{i=1}^{5}$	$\mathbf{R}$, $\mathbf{t}$	84
triangulation	1 img–img correspondence (m_i, m'_i)	X	85

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

calibrated problems

- have fewer degenerate configurations
- can do with fewer points (good for geometry proposal generators → Slide 113)

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

Part V

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.

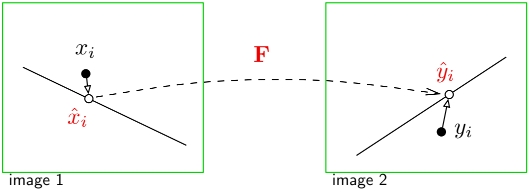
O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *Proc DAGM*, LNCS 2781:236–243. Springer-Verlag, 2003.

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.

▶The Concept of Error for Epipolar Geometry

Problem: Given at least 8 corresponding points $x_i \leftrightarrow y_j$ in a general position, estimate the most likely (or most probable) fundamental matrix $\mathbf{F}$.

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



- detected points x_i, y_i; the correspondence set is $S = \big\{(x_i,\, y_i)\big\}_{i=1}^{k}$
- corrected points $\hat{x}_i$, $\hat{y}_i$; the set is $\hat{S} = \big\{(\hat{x}_i,\, \hat{y}_i)\big\}_{i=1}^{k}$
- corrected points 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
- ok, but we need to choose a definite error function for optimization that is tractable

- the solution for calibrated cameras (unknown $\mathbf{E}$) is essentially the same and is not mentioned here explicitly

►cont'd

- Let $V(\cdot)$ be a positive semi-definite 'energy function'
- e.g., per correspondence,

$$V_i(x_i, y_i \mid \hat{x}_i, \hat{y}_i, \mathbf{F}) = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 + \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2 \tag{13}$$

- the total (negative) log-likelihood (of all data) then is

$$L(S \mid \hat{S}, \mathbf{F}) = \sum_{i=1}^{k} V_i(x_i, y_i \mid \hat{x}_i, \hat{y}_i, \mathbf{F})$$

- and the optimization problem is

$$(\hat{S}^*, \mathbf{F}^*) = \arg \min_{\substack{\mathbf{F} \\ \operatorname{rank} \mathbf{F} = 2}} \min_{\substack{\hat{S} \\ \underline{\hat{\mathbf{y}}}_i^\top \mathbf{F} \, \underline{\hat{\mathbf{x}}}_i = 0}} \sum_{i=1}^{k} V_i(x_i, y_i \mid \hat{x}_i, \hat{y}_i, \mathbf{F}) \tag{14}$$

we mention 3 approaches

1. direct optimization of 'geometric error' over all variables $\hat{S}$, $\mathbf{F}$ Slide 95
2. approximate minimization of $L(S \mid \hat{S}, \mathbf{F})$ over $\hat{S}$ followed by minimization over $\mathbf{F}$ Slide 96
3. marginalization of $L(S, \hat{S} \mid \mathbf{F})$ over $\hat{S}$ followed by minimization over $\mathbf{F}$

Method 1: Geometric Error Optimization

- 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
- equivalent projection matrices are see [H&Z,Sec. 9.5] for complete characterization

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

⊛ H3; 2pt: Verify that $\mathbf{F}$ is a f.m. of $\mathbf{P}_1$, $\mathbf{P}_2$, for instance that $\mathbf{F} \simeq \mathbf{Q}_2^{-\top} \mathbf{Q}_1^\top [\mathbf{e}_1]_\times$

1. compute $\mathbf{F}^{(0)}$ by the 7-point algorithm → Slide 81; construct camera $\mathbf{P}_2^{(0)}$ from $\mathbf{F}^{(0)}$
2. triangulate 3D points $\hat{X}_i^{(0)}$ from correspondences (x_i, y_i) for all $i = 1, \ldots, k$ → Slide 85
3. express the energy function as reprojection error

$$W_i(x_i, y_i \mid \hat{X}_i, \mathbf{P}_2) = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 + \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2 \quad \text{where} \quad \hat{\underline{\mathbf{x}}}_i \simeq \mathbf{P}_1 \hat{\underline{\mathbf{X}}}_i,\ \hat{\underline{\mathbf{y}}}_i \simeq \mathbf{P}_2 \hat{\underline{\mathbf{X}}}_i$$

4. starting from $\mathbf{P}_2^{(0)}$, $\hat{X}^{(0)}$ minimize

$$(\hat{X}^*, \mathbf{P}_2^*) = \arg\min_{\mathbf{P}_2, \hat{X}} \sum_{i=1}^{k} W_i(x_i, y_i \mid \hat{X}_i, \mathbf{P}_2)$$

5. compute $\mathbf{F}$ from $\mathbf{P}_1$, $\mathbf{P}_2^*$

- $3k + 12$ parameters to be found: latent: $\hat{\mathbf{X}}_i$, for all i (correspondences!), non-latent: $\mathbf{P}_2$
- minimal representation: $3k + 7$ parameters, $\mathbf{P}_2 = \mathbf{P}_2(\mathbf{F})$ → Slide 138
- there are pitfalls; this is essentially bundle adjustment; we will return to this later Slide 131

▶Method 2: First-Order Error Approximation

An elegant method for solving problems like (14):

- we will get rid of the latent parameters [H&Z, p. 287], [Sampson 1982]
- we will recycle the algebraic error $\boldsymbol{\varepsilon} = \underline{\mathbf{y}}^\top \mathbf{F}\, \underline{\mathbf{x}}$ from Slide 81

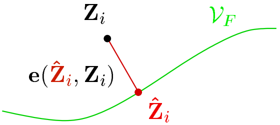
Observations:

- correspondences $\hat{x}_i \leftrightarrow \hat{y}_i$ 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 $\hat{\mathbf{Z}} = (\hat{u}^1,\, \hat{v}^1,\, \hat{u}^2,\, \hat{v}^2)$ consistent with $\mathbf{F}$
- let $\hat{\mathbf{Z}}_i$ be the closest point on $\mathcal{V}_F$ to measurement $\mathbf{Z}_i$, then (see (13))

$$\|\mathbf{Z}_i - \hat{\mathbf{Z}}_i\|^2 = (u_i^1 - \hat{u}_i^1)^2 + (v_i^1 - \hat{v}_i^1)^2 + (u_i^2 - \hat{u}_i^2)^2 + (v_i^2 - \hat{v}_i^2)^2 =$$

$$= V_i(x_i, y_i \mid \hat{x}_i, \hat{y}_i, \mathbf{F}) \overset{\text{def}}{=} \|\mathbf{e}(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)\|^2$$

which is what we needed in (14)



$\mathbf{Z}_i = (u^1, v^1, u^2, v^2)$ – measurement

algebraic error: $\boldsymbol{\varepsilon}(\hat{\mathbf{Z}}_i) \overset{\text{def}}{=} \underline{\hat{\mathbf{y}}}_i^\top \mathbf{F}\, \underline{\hat{\mathbf{x}}}_i$ $(= \mathbf{0})$

Sampson's idea: Linearize $\boldsymbol{\varepsilon}(\hat{\mathbf{Z}}_i)$ (with hat!) at $\mathbf{Z}_i$ (no hat!) and estimate $\mathbf{e}(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)$ with it

▶Sampson's Idea

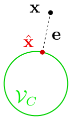
Linearize $\boldsymbol{\varepsilon}(\hat{\mathbf{Z}}_i)$ at $\mathbf{Z}_i$ per correspondence and estimate $\mathbf{e}(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)$ with it

have: $\boldsymbol{\varepsilon}(\mathbf{Z}_i)$, want: $\mathbf{e}(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)$

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

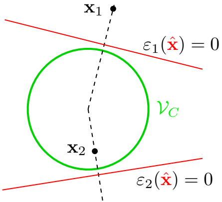
Illustration on circle fitting

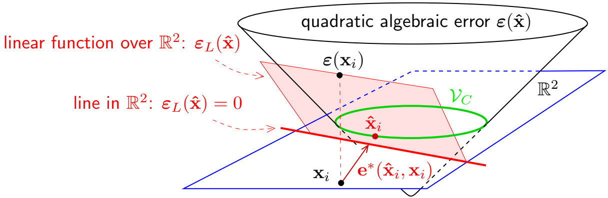
We are estimating distance from point $\mathbf{x}$ to circle $\mathcal{V}_C$ of radius r in canonical position. The circle is $\boldsymbol{\varepsilon}(\mathbf{x}) = \|\mathbf{x}\|^2 - r^2 = 0$. Then



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

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





▶Sampson Error Approximation

In general, the Taylor expansion is

$$\boldsymbol{\varepsilon}(\mathbf{Z}_i) + \underbrace{\frac{\partial \boldsymbol{\varepsilon}(\mathbf{Z}_i)}{\partial \mathbf{Z}_i}}_{\mathbf{J}_i(\mathbf{Z}_i)} \underbrace{(\hat{\mathbf{Z}}_i - \mathbf{Z}_i)}_{\mathbf{e}(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)} = \underbrace{\boldsymbol{\varepsilon}(\mathbf{Z}_i)}_{\boldsymbol{\varepsilon}_i \in \mathbb{R}^n} + \underbrace{\mathbf{J}(\mathbf{Z}_i)}_{\mathbf{J}_i \in \mathbb{R}^{n,d}} \underbrace{\mathbf{e}(\hat{\mathbf{Z}}_i, \mathbf{Z}_i)}_{\mathbf{e}_i \in \mathbb{R}^d} \stackrel{!}{=} 0$$

to find $\hat{\mathbf{Z}}_i$ closest to $\mathbf{Z}_i$, we estimate $\mathbf{e}_i$ from $\boldsymbol{\varepsilon}_i$ by minimizing — per correspondence $\mathbf{X}_i$

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

which gives a closed-form solution — ⊛ P1; 1pt: derive $\mathbf{e}_i^*$

$$\mathbf{e}_i^* = -\mathbf{J}_i^\top (\mathbf{J}_i \mathbf{J}_i^\top)^{-1} \boldsymbol{\varepsilon}_i$$

$$\|\mathbf{e}_i^*\|^2 = \boldsymbol{\varepsilon}_i^\top (\mathbf{J}_i \mathbf{J}_i^\top)^{-1} \boldsymbol{\varepsilon}_i$$

- note that $\mathbf{J}_i$ is not invertible!
- we often do not need $\hat{\mathbf{Z}}_i$, just the squared distance $\|\mathbf{e}_i\|^2$ — exception: triangulation → Slide 100
- the unknown parameters $\mathbf{F}$ are inside: $\mathbf{e}_i = \mathbf{e}_i(\mathbf{F})$, $\boldsymbol{\varepsilon}_i = \boldsymbol{\varepsilon}_i(\mathbf{F})$, $\mathbf{J}_i = \mathbf{J}_i(\mathbf{F})$

▶Sampson Error: Result for Fundamental Matrix Estimation

The fundamental matrix estimation problem becomes

$$\mathbf{F}^* = \arg \min_{\mathbf{F}, \operatorname{rank} \mathbf{F}=2} \sum_{i=1}^{k} e_i^2(\mathbf{F})$$

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

Sampson

$\varepsilon_i = \underline{\mathbf{y}}_i^\top \mathbf{F} \, \underline{\mathbf{x}}_i$	$\varepsilon_i \in \mathbb{R}$	scalar algebraic error from Slide 81
$\mathbf{J}_i = \left[\frac{\partial \varepsilon_i}{\partial u_i^1}, \frac{\partial \varepsilon_i}{\partial v_i^1}, \frac{\partial \varepsilon_i}{\partial u_i^2}, \frac{\partial \varepsilon_i}{\partial v_i^2} \right]$	$\mathbf{J}_i \in \mathbb{R}^{1,4}$	derivatives over point coords.
$e_i^2(\mathbf{F}) = \frac{\varepsilon_i^2}{\lVert \mathbf{J}_i \rVert^2}$	$e_i \in \mathbb{R}$	Sampson error

$$\mathbf{J}_i = \left[(\mathbf{F}_1)^\top \underline{\mathbf{y}}_i, \ (\mathbf{F}_2)^\top \underline{\mathbf{y}}_i, \ (\mathbf{F}^1)^\top \underline{\mathbf{x}}_i, \ (\mathbf{F}^2)^\top \underline{\mathbf{x}}_i \right] \qquad e_i^2(\mathbf{F}) = \frac{(\underline{\mathbf{y}}_i^\top \mathbf{F} \underline{\mathbf{x}}_i)^2}{\lVert \mathbf{S}\mathbf{F}\underline{\mathbf{x}}_i \rVert^2 + \lVert \mathbf{S}\mathbf{F}^\top \underline{\mathbf{y}}_i \rVert^2}$$

- Sampson correction 'normalizes' the algebraic error
- automatically copes with multiplicative factors $\mathbf{F} \mapsto \lambda \mathbf{F}$
- actual optimization not yet covered → Slide 103

▶Back to Triangulation: The Golden Standard Method

We are given $\mathbf{P}_1$, $\mathbf{P}_2$ and a single correspondence $x \leftrightarrow y$ and we look for 3D point $\mathbf{X}$ projecting to x and y. → Slide 85

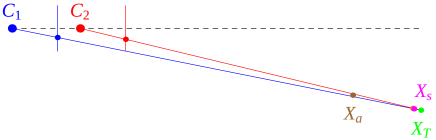
Idea:

1. compute $\mathbf{F}$ from $\mathbf{P}_1$, $\mathbf{P}_2$, e.g. $\mathbf{F} = (\mathbf{Q}_1\mathbf{Q}_2^{-1})^\top[\mathbf{q}_1 - (\mathbf{Q}_1\mathbf{Q}_2^{-1})\mathbf{q}_2]_\times$
2. correct measurement by the linear estimate of the correction vector → Slide 98

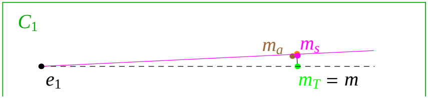
$$\begin{bmatrix}\hat{u}^1\\ \hat{v}^1\\ \hat{u}^2\\ \hat{v}^2\end{bmatrix} \approx \begin{bmatrix}u^1\\ v^1\\ u^2\\ v^2\end{bmatrix} - \frac{\varepsilon}{\|\mathbf{J}\|^2}\mathbf{J}^\top = \begin{bmatrix}u^1\\ v^1\\ u^2\\ v^2\end{bmatrix} - \frac{\underline{\mathbf{y}}^\top\mathbf{F}\underline{\mathbf{x}}}{\|\mathbf{S}\mathbf{F}\underline{\mathbf{x}}\|^2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

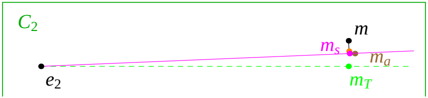
3. use the SVD algorithm with numerical conditioning → Slide 86

Ex (cont'd from Slide 89):



- X_T – noiseless ground truth position
- ● – reprojection error minimizer
- X_s – Sampson-corrected algebraic error minimizer
- X_a – algebraic error minimizer
- m – measurement (m_T with noise in v^2)





Levenberg-Marquardt (LM) Iterative Estimation

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 $s = 0, 1, 2, \ldots$

$$\boldsymbol{\theta}^{s+1} := \boldsymbol{\theta}^s + \mathbf{d}_s\,, \qquad \text{where} \qquad \mathbf{d}_s = \arg\min_{\mathbf{d}} \sum_{i=1}^{k} \|\mathbf{e}_i(\boldsymbol{\theta}^s + \mathbf{d})\|^2 \tag{15}$$

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

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

typically a long matrix

Then the solution to Problem (15) is a set of normal eqs

$$-\underbrace{\sum_{i=1}^{k} \mathbf{L}_i^\top \mathbf{e}_i(\boldsymbol{\theta}^s)}_{\mathbf{e}\in\mathbb{R}^{q,1}} = \underbrace{\left(\sum_{i=1}^{k} \mathbf{L}_i^\top \mathbf{L}_i\right)}_{\mathbf{L}\in\mathbb{R}^{q,q}} \mathbf{d}_s, \tag{16}$$

- $\mathbf{d}_s$ can be solved for by Gaussian elimination using Choleski decomposition of $\mathbf{L}$
  $\mathbf{L}$ symmetric $\Rightarrow$ use Choleski, almost $2\times$ faster than Gauss-Seidel, see bundle adjustment slide 134
- 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:

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

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 $\lambda := \lambda/10$, $s := s + 1$
3. otherwise set $\lambda := 10\lambda$ and recompute $\mathbf{d}_s$

- sometimes different constants are needed for the 10 and 10^{-3}
- note that $\mathbf{L}_i \in \mathbb{R}^{m,q}$ (long matrix) but each contribution $\mathbf{L}_i^\top \mathbf{L}_i$ is a square singular $q \times q$ matrix (always singular for $k < q$)
- error can be made robust to outliers, see the trick on Slide 106
- 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]
- λ helps avoid the consequences of gauge freedom $\to$ Slide 136

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

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

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

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

- $\mathbf{L}_i$ is a 3×3 matrix, must be reshaped to dimension-9 vector
- $\underline{\mathbf{x}}_i$ and $\underline{\mathbf{y}}_i$ in Sampson error are normalized to unit homogeneous coordinate
- reinforce $\operatorname{rank} \mathbf{F} = 2$ after each LM update to stay in the fundamental matrix manifold and $\|\mathbf{F}\| = 1$ to avoid gauge freedom (by SVD, see Slide 104)
- LM linearization could be done by numerical differentiation (small dimension)

►Local Optimization for Fundamental Matrix Estimation

Given a set $\{(x_i, y_i)\}_{i=1}^{k}$ of $k > 7$ inlier correspondences, compute an efficient estimate for fundamental matrix $\mathbf{F}$.

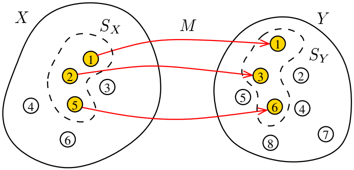
1. Find the conditioned (→ Slide 88) 7-point $\mathbf{F}_0$ (→ Slide 81) from a suitable 7-tuple
2. Improve the $\mathbf{F}_0^*$ using the LM optimization (→ Slides 101–102) and the Sampson error (→ Slide 103) on all inliers, reinforce rank-2, unit-norm $\mathbf{F}_k^*$ after each LM iteration using SVD

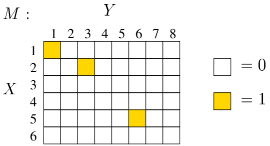
- if there are no wrong matches (outliers), this gives a local optimum
- contamination of (inlier) correspondences by outliers may wreak havoc with this algorithm
- the full problem involves finding the inliers!
- in addition, we need a mechanism for jumping out of local minima (and exploring the space of all fundamental matrices)

►The Full Problem of Matching and Fundamental Matrix Estimation

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

1. inliers $S_X \subseteq X$, $S_Y \subseteq Y$
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 geometric error $\sum_{ij} e_{ij}^2(\mathbf{F})$ is small — note a slight change in notation: e_{ij}
5. inlier-outlier and outlier-outlier matches are improbable





$$(M^*, \mathbf{F}^*) = \arg\max_{M,\mathbf{F}} p(M, \mathbf{F} \mid X, Y, D) \qquad (17)$$

- probabilistic model: an efficient language for task formulation
- the (17) is a p.d.f. for all the involved variables — (there is a constant number of 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 Marginalization

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

$$\mathbf{F}^* = \arg\max_{\mathbf{F}} p(\mathbf{F} \mid X, Y, D) \tag{18}$$

by marginalization of $p(M, \mathbf{F} \mid X, Y, D)$ over M this simplification changes the problem!

$$p(M, \mathbf{F} \mid X, Y, D) \simeq p(M, \mathbf{F}, X, Y, D) = p(X, Y, D, M \mid \mathbf{F}) \cdot p(\mathbf{F})$$

assuming correspondence-wise independence:

$$p(X, Y, D, M \mid \mathbf{F}) = \prod_{i=1}^{m} \prod_{j=1}^{n} p(x_i, y_j, D, m_{ij} \mid \mathbf{F}) \overset{\text{def}}{=} \prod_{i=1}^{m} \prod_{j=1}^{n} p_e(e_{ij}, d_{ij}, m_{ij} \mid \mathbf{F})$$

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

Marginalization:

$$\sum_{m_{11} \in \{0,1\}} \sum_{m_{12}} \cdots \sum_{m_{mn}} p(X, Y, D, M \mid \mathbf{F}) = \sum_{m_{11}} \sum_{m_{12}} \cdots \sum_{m_{mn}} \prod_{i=1}^{m} \prod_{j=1}^{n} p_e(e_{ij}, d_{ij}, m_{ij} \mid \mathbf{F}) =$$

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

Robust Matching Model (cont'd)

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

$$= \underbrace{p_e(e_{ij}, d_{ij} \mid m_{ij} = 1, \mathbf{F})}_{p_1(e_{ij}, d_{ij} \mid \mathbf{F})} \cdot \underbrace{p(m_{ij} = 1 \mid \mathbf{F})}_{1-\alpha_0} + \underbrace{p_e(e_{ij}, d_{ij} \mid m_{ij} = 0, \mathbf{F})}_{p_0(e_{ij}, d_{ij} \mid \mathbf{F})} \cdot \underbrace{p(m_{ij} = 0 \mid \mathbf{F})}_{\alpha_0} =$$

$$= (1 - \alpha_0)\, p_1(e_{ij}, d_{ij} \mid \mathbf{F}) + \alpha_0\, p_0(e_{ij}, d_{ij} \mid \mathbf{F}) \qquad (19)$$

- the $p_0(e_{ij}, d_{ij} \mid \mathbf{F}) \approx \text{const}$ is a penalty for 'missing a correspondence' but it should be a p.d.f. (cannot be a constant) (see Slide 108 for a simplification)

$$\alpha_0 \to 1, \quad p_0 \to 0 \quad \text{so that} \quad \frac{\alpha_0}{1-\alpha_0}\, p_0 \approx \text{const}$$

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

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

- we choose, eg.

$$p_1(e_{ij} \mid \mathbf{F}) = \frac{1}{T_e(\sigma_1, \mathbf{F})}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1{}^2}}, \quad p_1(d_{ij}) = \frac{1}{T_d(\sigma_d, \dim \mathbf{d})}\, e^{-\frac{\|\mathbf{d}(x_i) - \mathbf{d}(y_j)\|^2}{2\sigma_d{}^2}} \qquad (20)$$

- σ_1, σ_d, α_0 are 'hyper-parameters'
- the form of $T(\sigma_1, \mathbf{F})$ depends on error definition
- we will continue with the result from (19)

▶Simplified Robust Energy (Error) Function

- assuming the choice of p_1 as in (20), we are simplifying

$$p(X,Y,D \mid \mathbf{F}) = \prod_{i=1}^{m}\prod_{j=1}^{n}\Big[(1-\alpha_0)\, p_1(e_{ij}, d_{ij} \mid \mathbf{F}) + \alpha_0\, p_0(e_{ij}, d_{ij} \mid \mathbf{F})\Big] \tag{21}$$

- we define 'energy' as: $V(x) = -\log p(x)$ this helps simplify the formulas
- for simplicity, we omit d_{ij}
- we choose $\sigma_0 \gg \sigma_1$ and the missed-correspondence penalty function as

$$p_0(e_{ij} \mid \mathbf{F}) = \frac{1}{T_e(\sigma_0, \mathbf{F})}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_0^2}}$$

- then

$$V(X,Y,D \mid \mathbf{F}) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left[\underbrace{-\log\frac{1-\alpha_0}{T_e(\sigma_1,\mathbf{F})}}_{\Delta(\mathbf{F})} - \log\Bigg(e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + \underbrace{\frac{\alpha_0}{1-\alpha_0}\,\frac{T_e(\sigma_1,\mathbf{F})}{T_e(\sigma_0,\mathbf{F})}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_0^2}}}_{t\,\approx\,\mathrm{const}}\Bigg)\right]$$

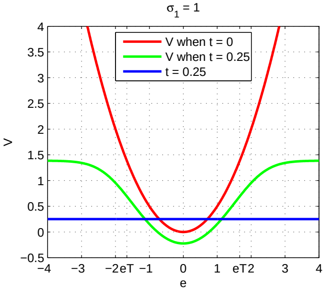
- by choosing representative of $\mathbf{F}$ such that $\Delta(\mathbf{F}) = \mathrm{const}$, we get

$$V(X,Y,D \mid \mathbf{F}) = m\,n\,\Delta + \sum_{i=1}^{m}\sum_{j=1}^{n}\underbrace{-\log\Big(e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + t\Big)}_{\hat{V}(e_{ij})} \tag{22}$$

note that m, n are fixed

▶The Action of the Robust Matching Model on Data

Example for $\hat{V}(e)$ from (22):



red – the usual (non-robust) error when $t = 0$

blue – the rejected correspondence penalty t

green – 'robust energy' (22)

- if the error of a correspondence exceeds a limit, it is ignored
- then $\hat{V}(e) = \text{const}$ and we essentially count outliers in (22)
- t controls the 'turn-off' point
- the inlier/outlier threshold is e_T is the error for which $(1-\alpha_0)\,p_1(e_T) = \alpha_0\,p_0(e_T)$: note that $t \approx 0$

$$e_T = \sigma_1\sqrt{-\log t^2} \tag{23}$$

The full optimization problem is (18):

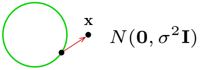
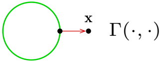
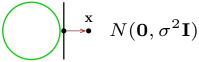
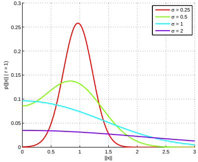
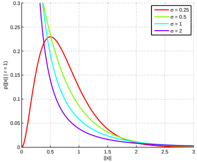
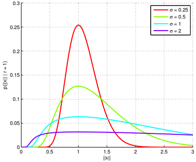
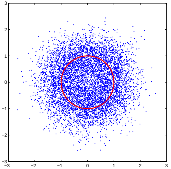
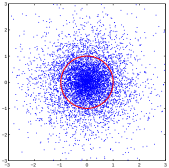
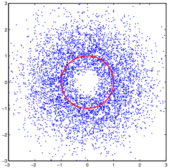
$$\mathbf{F}^* = \arg\max_{\mathbf{F}} p(\mathbf{F} \mid X, Y, D) = \arg\max_{\mathbf{F}} \frac{\overbrace{p(X,Y,D \mid \mathbf{F})}^{\text{likelihood}} \cdot \overbrace{p(\mathbf{F})}^{\text{prior}}}{\underbrace{p(X,Y,D)}_{\text{evidence}}} =$$

$$= \arg\min_{\mathbf{F}} \{V(X,Y,D \mid \mathbf{F}) + V(\mathbf{F})\}$$

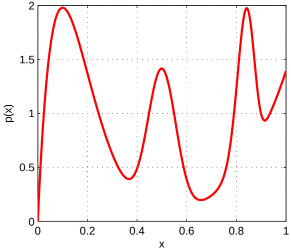
- typically we take $V(\mathbf{F}) = 0$ unless we need to stabilize a computation, e.g. when video camera moves smoothly (on a high-mass vehicle) and we have a prediction for $\mathbf{F}$
- evidence is not needed unless we want to compare different models

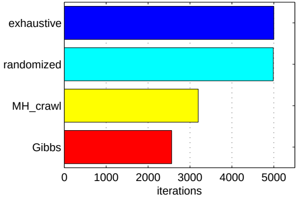
Discussion: On The Art of Probabilistic Model Design...

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

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

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





- averaged over 10^4 trials
- number of proposals before $|x - x_{\text{true}}| \leq \text{step}$
- uniform and Gibbs give the theoretical result

- consider the function $p(x)$ at left p.d.f. on $[0, 1]$, mode at 0.1
- consider several methods:
  1. exhaustive search

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

     • slow algorithm (definite quantization); faster variants exist • fast to implement
  2. randomized search with uniform sampling

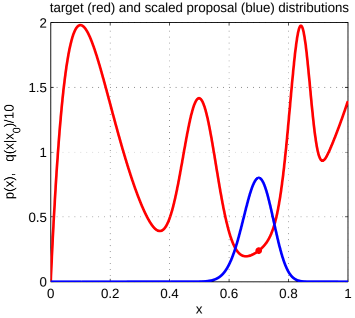
     ```
     x = rand(1);
     if p(x) > bestp
      bestx = x; bestp = p(x);
     end
     ```

     • slow algorithm but better convergence • fast to implement • how to stop it?
  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))
  4. Metropolis-Hastings sampling

     • almost as fast (with care) • not so fast to implement • rarely infeasible • RANSAC belongs here

How To Generate Random Samples from a Complex Distribution?



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

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

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

- note we can generate samples from this $p(x)$ — how?

- suppose we cannot sample from $p(x)$ but we can sample from some 'simple' distribution, given the last sample x_0 (blue) — proposal distribution

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

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

▶Metropolis-Hastings (MH) Sampling

C – configuration (of all variable values) Here $C = \mathbf{F}$ and $p(C) = p(\mathbf{F} \mid X, Y, D)$

Goal: Generate a sequence of random samples $\{C_i\}$ from $p(C)$

- setup a Markov chain with a suitable transition probability function so that it generates the sequence

Sampling procedure

1. given C_i, generate random sample S from $q(S \mid C_i)$
   q may use some information from C_i (Hastings)
2. compute acceptance ratio the evidence term drops out
$$a = \frac{p(S)}{p(C_i)} \cdot \frac{q(C_i \mid S)}{q(S \mid C_i)}$$
3. generate random number u from unit-interval uniform distribution $\mathrm{U}_{0,1}$
4. if $u < a$ then $C_{i+1} := S$ else $C_{i+1} := C_i$

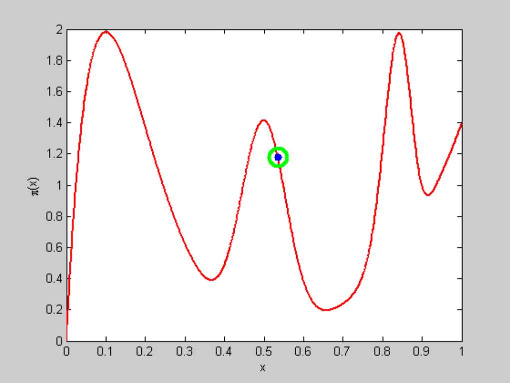
'Programing' an MH sampler

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

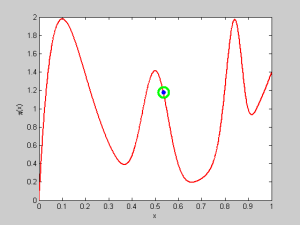
Finding the mode

- remember the best sample fast implementation but must wait long to hit the mode
- use simulated annealing very slow
- start local optimization from the best sample good trade-off between speed and accuracy

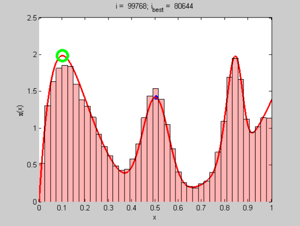
MH Sampling Demo



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



initial sample



final distribution of visited states

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

```
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 < 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
```

▶From MH Sampling to RANSAC

- configuration = k-tuple of inlier correspondences

  the minimization will be over a discrete set of epipolar geometries proposable from 7-tuples
- data-driven proposals q:
  1. select k-tuple from data independently and uniformly $q(S) = \binom{mn}{k}^{-1}$
  2. solve the minimal geometric problem $\mapsto$ geometry proposal (e.g. $\mathbf{F}$ from $k = 7$)
- independent sampling $a = \frac{p(S')}{p(S_i)} \cdot \frac{q(S_i)}{q(S')}$
  1. q uniform, then $a = \frac{p(S')}{p(S_i)}$ — MAPSAC ($p(S)$ includes the prior)
  2. q dependent on descriptor similarity — PROSAC (similar pairs are proposed more often)

LO-MAPSAC

1. generate random sample S_b from $q(S)$
2. set initial $N := \binom{mn}{k}$
3. repeat N-times:
   a. generate random sample S' from $q(S)$
   b. if $p(S') > p(S_b)$ then
      i. $S_b := S'$
      ii. threshold-out inliers — using e_T from (23)
      iii. start local optimization from S_b and update S_b with the result
      iv. re-estimate N from inlier counts — using the standard formula for RANSAC termination, see Slide 117
4. output S_b

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

▶Stopping RANSAC

Principle: what is the number of proposals N that are needed to hit an all-inlier sample?

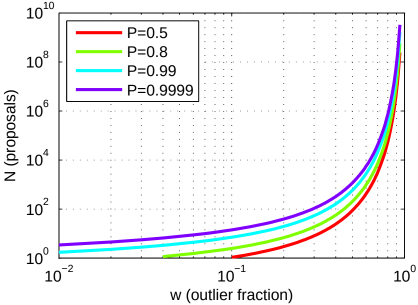
$$N \geq \frac{\log(1-P)}{\log(1-(1-w)^s)}$$

- $(1-w)^s$ – proposal does not contain an outlier
- $1-(1-w)^s$ – proposal contains at least one outlier
- $1-P =$ all proposals contained an outlier $= (1-(1-w)^s)^N$

P – probability that at least one sample is all-inlier
w – the fraction of outliers among tentative correspondences
s – sample size (7 in 7-point algorithm)

N for $s = 7$

w	P = 0.8	P = 0.99
0.5	205	590
0.8	$1.3 \cdot 10^5$	$3.5 \cdot 10^5$
0.9	$1.6 \cdot 10^7$	$4.6 \cdot 10^7$

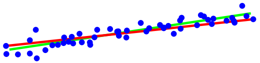


- N can be re-estimated using the current estimate for w (if there is LO, then after LO)
  the quasi-posterior estimate for w is the average over all samples generated so far
- for $w \to 1$ we gain nothing over the standard MH-sampler stoppig criterion

►The Difference between RANSAC and a General MH Sampler

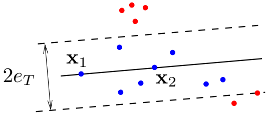
RANSAC = five ideas: [Fischler & Bolles 1981]

1. proposal distribution is given by the <u>empirical distribution</u> of data sample:



- pairs of points define line distribution from $p(\mathbf{n} \mid X)$ (left)
- random correspondence tuples drawn uniformly propose samples of $\mathbf{F}$ from a data-driven distribution $q(\mathbf{F} \mid X, Y)$

2. stopping based on the probability of mode-hitting → Slide 117
3. standard RANSAC replaces probability maximization with <u>consensus maximization</u>



the e_T is the inlier/outlier threshold from (23)

4. when counting inliers, do not work with all m_{ij} but with a set of <u>tentative correspondences</u> that form a matching, e.g. selected by stable matching:
   a. find a pair m_{ij} of greatest $p_1(d_{ij})$ and remember it
   b. remove row i and column j from the matching table (needs some bookkeeping and reindexing)
   c. repeat Steps a–c until the table is empty
   d. return the remembered set
5. each time a new best sample occurs, start <u>local optimization</u> from inliers

or LO weighted by posterior $p(m_{ij})$ [Chum et al. 2003]

LM optimization with Sampson error (and re-weighting)

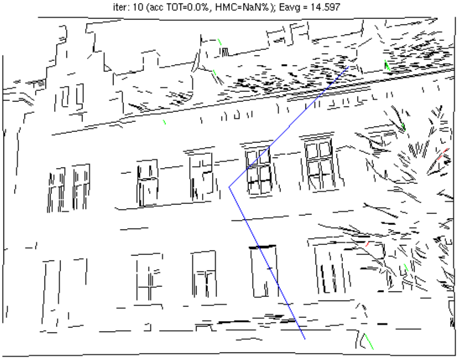
Example Matching Results for the 7-point Algorithm with RANSAC

input images — interest points (ca. 3600) — tentative corresp. (416) — matching (340) notice wrong matches

- the minimization os over a discrete set of epipolar geometries proposable from 7-tuples

Example: MH Sampling for a More Complex Problem

Task: Find two vanishing points from line segments detected in input image.

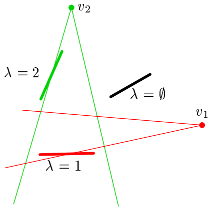


video

Model

- principal point known, square pixel
- explicit variables
  1. two unknown vanishing points v_1, v_2
- latent variables
  1. each line has a vanishing point label $\lambda_i \in \{\emptyset, 1, 2\}$, $\emptyset$ represents an outlier
  2. 'mother lines' passing through vanishing points

$$\arg\min_{v_1, v_2, \Lambda, L} V(v_1, v_2, \Lambda, L \mid S)$$



simplifications

- vanishing points restricted to the set of all pairwise segment intersections
- mother lines fixed by segment centroid

Beyond RANSAC

Note that by simplification in (18) on Slide 106 we have lost constraints on M (eg. uniqueness). One can choose a better model when not marginalizing:

$$p(M, \mathbf{F}, X, Y, D) = \underbrace{p(X, Y \mid M, \mathbf{F})}_{\text{geometric error}} \cdot \underbrace{p(D \mid M)}_{\text{similarity}} \cdot \underbrace{p(M)}_{\text{constraints}} \cdot \underbrace{p(\mathbf{F})}_{\text{prior}}$$

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

In the MH scheme

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

    $$q(m_{ij} \mid \mathbf{F}) \sim \big((1-\alpha_0)\, p_1(e_{ij} \mid \mathbf{F}),\ \alpha_0\, p_0(e_{ij} \mid \mathbf{F})\big)$$

    when $p(M)$ uniform then always accepted, $a = 1$ ⊛ derive
  - additional proposals from $q(\mathbf{F} \mid M)$ are possible, with explicit inliers Hybrid Monte Carlo
  - we can compute the posterior probability of each match $p(m_{ij})$ by histogramming m_{ij} over $\{S_i\}$
  - local optimization can then use explicit inliers and $p(m_{ij})$
- error can be estimated for elements of $\mathbf{F}$ from $\{S_i\}$ does not work in RANSAC!
- large error indicates problem degeneracy this is not directly available in RANSAC
- good conditioning is not a requirement we work with the entire distribution $p(\mathbf{F})$
- one can find the most probable number of epipolar geometries by reversible jump MCMC (homographies or other models)

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

Part 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.

▶Constructing Cameras from the Fundamental Matrix

Given $\mathbf{F}$, construct some cameras $\mathbf{P}_1$, $\mathbf{P}_2$ such that $\mathbf{F}$ is their fundamental matrix.

Solution See [H&Z, p. 256]

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{P}_2 = \begin{bmatrix} [\underline{\mathbf{e}}_2]_\times \mathbf{F} + \underline{\mathbf{e}}_2 \underline{\mathbf{v}}^\top & \lambda \underline{\mathbf{e}}_2 \end{bmatrix}$$

where

- $\underline{\mathbf{v}}$ is any 3-vector, e.g. $\underline{\mathbf{v}} = \underline{\mathbf{e}}_1$ to make the camera finite
- $\lambda \neq 0$ is a scalar,
- $\underline{\mathbf{e}}_2 = \mathrm{null}(\mathbf{F}^\top)$, i.e. $\underline{\mathbf{e}}_2^\top \mathbf{F} = 0$

Proof

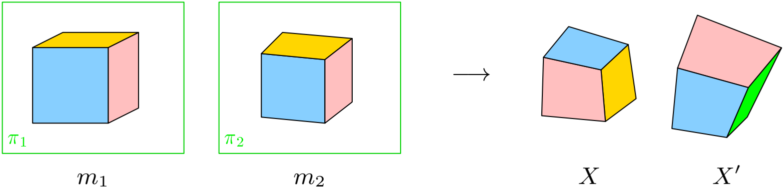
1. $\mathbf{S}$ is antisymmetric iff $\mathbf{x}^\top \mathbf{S} \mathbf{x} = 0$ for all $\mathbf{x}$ look-up the proof!
2. we have $\underline{\mathbf{x}} \simeq \mathbf{P} \underline{\mathbf{X}}$
3. a non-zero $\mathbf{F}$ is a f.m. iff $\mathbf{P}_2^\top \mathbf{F} \mathbf{P}_1$ is antisymmetric
4. if $\mathbf{P}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$ and $\mathbf{P}_2 = \begin{bmatrix} \mathbf{S}\mathbf{F} & \underline{\mathbf{e}}_2 \end{bmatrix}$ then $\mathbf{F}$ corresponds to $(\mathbf{P}_1, \mathbf{P}_2)$ by Step 3
5. we can write $\mathbf{S} = [\mathbf{s}]_\times$
6. a suitable choice is $\mathbf{s} = \underline{\mathbf{e}}_2$ [Luong96]
7. for the full the class including $\underline{\mathbf{v}}$, see [H&Z, Sec. 9.5]

▶The Projective Reconstruction Theorem

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

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

- when $\mathbf{P}_i$ and $\underline{\mathbf{X}}$ are both determined from correspondences (including calibrations $\mathbf{K}_i$), they are given up to a common 3D homography $\mathbf{H}$
  (translation, rotation, scale, shear, pure perspectivity)

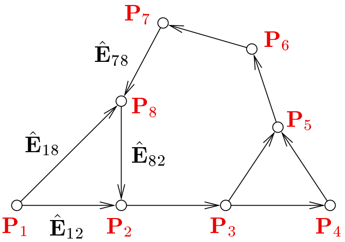


- when cameras are internally calibrated ($\mathbf{K}_i$ known) then $\mathbf{H}$ is restricted to a similarity since it must preserve the calibrations $\mathbf{K}_i$ [H&Z, Secs. 10.2, 10.3], [Longuet & Higgins 81]
  (translation, rotation, scale)

▶Reconstructing Camera Systems

Problem: Given a set of p decomposed pairwise essential matrices $\hat{\mathbf{E}}_{ij} = [\hat{\mathbf{t}}_{ij}]_\times \hat{\mathbf{R}}_{ij}$ and calibration matrices $\mathbf{K}_i$ reconstruct the camera system $\mathbf{P}_i$, $i = 1, \ldots, k$

→ Slides 78 and 138 on representing $\mathbf{E}$



We construct camera pairs $\hat{\mathbf{P}}_{ij} \in \mathbb{R}^{6,4}$ → Slide 123

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

- singletons i, j correspond to vertices V — k vertices
- pairs ij correspond to graph edges E — p edges

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

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hat{\mathbf{R}}_{ij} & \hat{\mathbf{t}}_{ij} \end{bmatrix}}_{\mathbb{R}^{6,4}} \underbrace{\begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\top & s_{ij} \end{bmatrix}}_{\mathbf{H}_{ij} \in \mathbb{R}^{4,4}} \stackrel{!}{=} \underbrace{\begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{R}_j & \mathbf{t}_j \end{bmatrix}}_{\mathbb{R}^{6,4}} \tag{24}$$

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

►cont'd

Eq. (24) implies $\begin{bmatrix} \mathbf{R}_{ij} \\ \hat{\mathbf{R}}_{ij}\mathbf{R}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i \\ \mathbf{R}_j \end{bmatrix} \qquad \begin{bmatrix} \mathbf{t}_{ij} \\ \hat{\mathbf{R}}_{ij}\mathbf{t}_{ij} + s_{ij}\hat{\mathbf{t}}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_j \end{bmatrix}$

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

$$\hat{\mathbf{R}}_{ij}\mathbf{R}_i = \mathbf{R}_j, \qquad \hat{\mathbf{R}}_{ij}\mathbf{t}_i + s_{ij}\hat{\mathbf{t}}_{ij} = \mathbf{t}_j, \qquad s_{ij} > 0 \tag{25}$$

- 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 by e.g. selecting

$$\mathbf{R}_1 = \mathbf{I}, \qquad \sum_{i=1}^{k} \mathbf{t}_i = \mathbf{0}, \qquad \frac{1}{p}\sum_{i,j} s_{ij} = 1 \tag{26}$$

- rotation equations are decoupled from translation equations
- in principle, s_{ij} could correct the sign of $\hat{\mathbf{t}}_{ij}$ from essential matrix decomposition Slide 78
  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 $s_{ij} > 0$; see Slide 80

\+ pairwise correspondences are sufficient

− suitable for well-located cameras only (dome-like configurations)

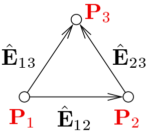
otherwise intractable or numerically unstable

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

$$\hat{\mathbf{R}}_{ij}\mathbf{r}_i^c - \mathbf{r}_j^c = \mathbf{0}, \qquad \text{for all } i,\ j \tag{27}$$

- fix c and denote $\mathbf{r}^c = \left[\mathbf{r}_1^c, \mathbf{r}_2^c, \ldots, \mathbf{r}_k^c\right]^\top$ c-th columns of all rotation matrices stacked; $\mathbf{r}^c \in \mathbb{R}^{3k}$
- then (27) becomes $\mathbf{D}\,\mathbf{r}^c = \mathbf{0}$ $\mathbf{D} \in \mathbb{R}^{3p,3k}$
- $3p$ equations for $3k$ unknowns $\rightarrow p \geq k$ in a 1-connected graph we have to fix $\mathbf{r}_1^c = [1, 0, 0]$

Ex: $(k = p = 3)$



$\rightarrow$

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

$\rightarrow$

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

- must hold for any c

Idea: [Martinec & Pajdla CVPR 2007]

1. find the space of all $\mathbf{r}^c \in \mathbb{R}^{3k}$ that solve (27) $\mathbf{D}$ is sparse, use `[V,E] = eigs(D'*D,3,0);` (Matlab)
2. choose 3 unit orthogonal vectors in this space 3 smallest eigenvectors
3. find closest rotation matrices per cam. using SVD because $\|\mathbf{r}^c\| = 1$ is necessary but insufficient
   $\mathbf{R}_i^* = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{R}_i = \mathbf{U}\mathbf{D}\mathbf{V}^\top$

- global world rotation is arbitrary

Finding The Translation Component in Eq. (25)

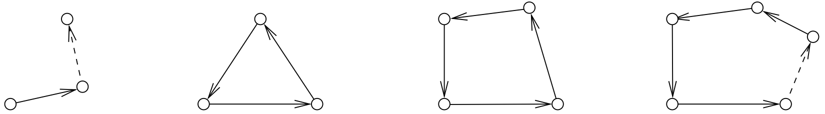
From eqs. (25) and (26): d – rank of camera center set p – No. of pairs, k – No. of cameras

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

- in rank d: $d \cdot p + d + 1$ equations for $d \cdot k + p$ unknowns $\rightarrow p \geq \frac{d(k-1)-1}{d-1}$

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

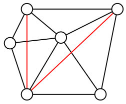
$p = k - 1$ | $k = p = 3$ | $k = p = 4$ | $k = p > 4$



$k \leq 2$ for any d | $d \geq 2$: non-collinear ok | $d \geq 3$: non-planar ok | $d \geq k - 1$: not possible

'

- rank is not sufficient for chains, trees, or when $d = 1$ (collinear cameras)
- 3-connectivity gives a sufficient rank for $d = 3$ (cams. in general pos. in 3D)
  - s-connected graph has $p \geq \lceil \frac{sk}{2} \rceil$ edges for $s \geq 2$, hence $p \geq \lceil \frac{3k}{2} \rceil \geq \frac{3k}{2} - 2$
- 4-connectivity gives a sufficient rank for any k for $d = 2$ (coplanar cams)
  - since $p \geq \lceil 2k \rceil \geq 2k - 3$
  - maximal planar tringulated graphs have $p = 3k - 6$ and give the rank for $k \geq 3$



cont'd

Linear equations in (25) and (26) can be rewritten to

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

for $d = 3$: $\mathbf{t} \in \mathbb{R}^{3k+p}$, $\mathbf{D} \in \mathbb{R}^{3p,3k+p}$ is sparse

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

- this is a quadratic programming problem (constraints!)

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

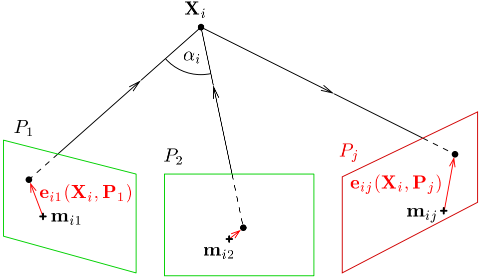
- but check the rank first!

▸Solving Eq. (25) by Stepwise Gluing

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

Initialization

1. initialize camera cluster $\mathcal{C}$ with P_1, P_2,
2. find essential matrix $\mathbf{E}_{12}$ and matches M_{12} by the 5-point algorithm Slide 84
3. construct camera pair
$$\mathbf{P}_1 = \mathbf{K}_1 \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}, \ \mathbf{P}_2 = \mathbf{K}_2 \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$$
4. compute 3D reconstruction $\{X_i\}$ per match from M_{12} Slide 90
5. 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 $\mathcal{X}_j$ Slide 69
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

▶Bundle Adjustment

Given:

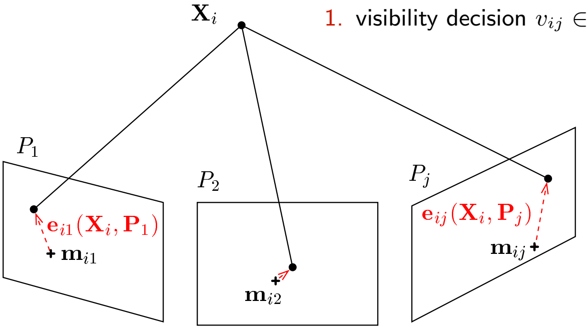
1. set of 3D points $\{\mathbf{X}_i\}_{i=1}^p$
2. set of cameras $\{\mathbf{P}_j\}_{j=1}^c$
3. fixed tentative projections $\mathbf{m}_{ij}$

Required:

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

Latent:

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



- for simplicity, $\mathbf{X}$, $\mathbf{m}$ are considered direct (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

Data likelihood is   constructed by marginalization, as in Robust Matching Model, Slide 107

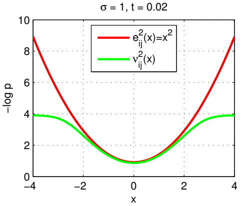
$$p(\{\mathbf{m}\} \mid \{\mathbf{P}\}) = \prod_{\text{pts}:i=1}^{p} \prod_{\text{cams}:j=1}^{c} \Big((1-\alpha_0)p_1(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j) + \alpha_0\, p_0(e_{ij} \mid \mathbf{X}_i, \mathbf{P}_j)\Big)$$

the simplified log-likelihood is (as on Slide 108)

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

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

$$(\mathbf{L}_{ij})_l = \frac{\partial \nu_{ij}}{\partial \theta_l} = \underbrace{\frac{1}{1 + t\, e^{e_{ij}^2(\theta)/(2\sigma_1^2)}}}_{\text{small for big } e_{ij}} \cdot \frac{1}{\nu_{ij}(\theta)} \cdot \frac{1}{4\sigma_1^2} \cdot \frac{\partial e_{ij}^2(\theta)}{\partial \theta_l} \quad (28)$$



but the LM method stays the same as on Slides 101–102

- outliers have virtually no impact on $\mathbf{d}_s$ in normal equations because of the red term in (28) that scales contributions to the sums down

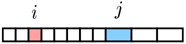
$$-\sum_{i,j} \mathbf{L}_{ij}^\top\, \nu_{ij}(\theta^s) = \Big(\sum_{i,j}^{k} \mathbf{L}_{ij}^\top \mathbf{L}_{ij}\Big)\mathbf{d}_s$$

►Sparsity in Bundle Adjustment

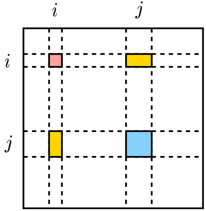
We have $q = 3p + 11c$ parameters: $\theta = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p; \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_c)$ points, cameras
We will use a running index $r = 1, \ldots, k$, $k = p \cdot c$. Then each r corresponds to some i, j

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

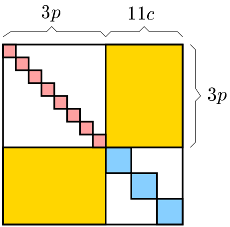
The block form of $\mathbf{L}_r$ in Levenberg-Marquardt (Slide 101) is zero except in columns i and j:
r-th error term is $\nu_r^2 = \rho(e_{ij}^2(\mathbf{X}_i, \mathbf{P}_j))$

$\mathbf{L}_r =$ 

blocks:
- pink: $\mathbf{X}_i, 1 \times 3$
- blue: $\mathbf{P}_j, 1 \times 11$

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

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

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

- "points first, then cameras" scheme
- standard bundle adjustment eliminates points and solves cameras, then back-substitutes

▶Choleski Decomposition for B. A.

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

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

This is a linear set of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, where

- $\mathbf{A}$ 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 and symmetric, $\mathbf{A}^{-1}$ is dense — direct matrix inversion is prohibitive

> Choleski: Every 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 $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ — transforms the problem to solving $\mathbf{L}\underbrace{\mathbf{L}^\top\mathbf{x}}_{\mathbf{c}} = \mathbf{b}$
2. solve for $\mathbf{x}$ in two passes:

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

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

- Choleski decomposition is fast (does not touch zero blocks)
  non-zero elements are $9p + 121c + 66pc \approx 3.4 \cdot 10^6$; 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 $\mathbf{A}$ [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 useful for arrowhead sparse matrices.

 [p,q] = size(A);
 if p ~= q, error 'Matrix must be 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 must be positive definite'; end
  L(i,i) = sqrt(a);
 end
end
```

▶Gauge Freedom

1. The external frame is not fixed: See Projective Reconstruction Theorem, Slide 124

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

2. Some representations are not minimal, e.g.
    - $\mathbf{P}$ is 12 numbers for 11 parameters
    - we may represent $\mathbf{P}$ in decomposed form $\mathbf{K}$, $\mathbf{R}$, $\mathbf{t}$
    - but $\mathbf{R}$ is 9 numbers representing the 3 parameters of rotation

As a result

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

Solutions

- fixing the external frame (e.g. a selected camera frame) explicitly or by constraints
- imposing constraints on projective entities
    - cameras, e.g. $\mathbf{P}_{3,4} = 1$ — this excludes affine cameras
    - points, e.g. $\|\underline{\mathbf{X}}_i\|^2 = 1$ — this way we can represent points at infinity
- using minimal representations
    - points in their Euclidean representation $\mathbf{X}_i$ — but finite points may be an unrealistic model
    - rotation matrix can be represented by Cayley transform see next

▶Minimal Representations for Rotation

- $\mathbf{o}$ – rotation axis, $\|\mathbf{o}\| = 1$, φ – rotation angle
- **wanted**: simple mapping to/from rotation matrices

1. Rodrigues' representation

$$\mathbf{R} = \mathbf{I} + \sin\varphi [\mathbf{o}]_\times + (1 - \cos\varphi)[\mathbf{o}]_\times^2$$

$$\sin\varphi\, [\mathbf{o}]_\times = \frac{1}{2}(\mathbf{R} - \mathbf{R}^\top), \quad \cos\varphi = \frac{1}{2}(\operatorname{tr}\mathbf{R} - 1)$$

   - hiding φ in the vector $\mathbf{o}$ as in $[\sin\varphi\, \mathbf{o}]_\times$ is not so easy
   - Cayley tried:

2. Cayley's representation; let $\mathbf{a} = \mathbf{o} \tan\frac{\varphi}{2}$, then

$$\mathbf{R} = (\mathbf{I} + [\mathbf{a}]_\times)(\mathbf{I} - [\mathbf{a}]_\times)^{-1}$$

$$[\mathbf{a}]_\times = (\mathbf{R} + \mathbf{I})^{-1}(\mathbf{R} - \mathbf{I})$$

$$\mathbf{a}_1 \circ \mathbf{a}_2 = \frac{\mathbf{a}_1 + \mathbf{a}_2 - \mathbf{a}_1 \times \mathbf{a}_2}{1 - \mathbf{a}_1^\top \mathbf{a}_2}$$

composition of rotations $\mathbf{R} = \mathbf{R}_1\mathbf{R}_2$

   - no trigonometric functions
   - cannot represent rotation by $180°$
   - explicit composition formula

3. exponential map $\mathbf{R} = \exp\,[\varphi\,\mathbf{o}]_\times$, inverse by Rodrigues' formula

Minimal Representations for Other Entities

1. with the help of rotation we can minimally represent
   - fundamental matrix

     $$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^\top, \quad \mathbf{D} = \operatorname{diag}(d, 1, 0), \quad \mathbf{U}, \mathbf{V} \text{ are rotations}, \quad 3 + 1 + 3 = 7 \text{ DOF}$$

   - essential matrix

     $$\mathbf{E} = [-\mathbf{t}]_\times \mathbf{R}, \quad \mathbf{R} \text{ is rotation}, \quad \|\mathbf{b}\| = 1, \qquad 3 + 2 = 5 \text{ DOF}$$

   - camera

     $$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}, \qquad 5 + 3 + 3 = 11 \text{ DOF}$$

2. homography can be represented via exponential map

$$\exp \mathbf{A} = \sum_{k=0}^{\infty} \frac{1}{k!} \mathbf{A}^k \qquad \text{note: } \mathbf{A}^0 = \mathbf{I}$$

some properties

$$\exp \mathbf{0} = \mathbf{I}, \quad \exp(-\mathbf{A}) = \left(\exp \mathbf{A}\right)^{-1}, \quad \exp(\mathbf{A} + \mathbf{B}) \neq \exp(\mathbf{A}) \exp(\mathbf{B})$$

$\exp(\mathbf{A}^\top) = (\exp \mathbf{A})^\top$ hence if $\mathbf{A}$ antisymmetric then $\exp \mathbf{A}$ orthogonal

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

$\det \exp \mathbf{A} = \exp(\operatorname{tr} \mathbf{A})$ a key to homography representation:

$$\mathbf{H} = \exp \mathbf{Z} \text{ such that } \operatorname{tr} \mathbf{Z} = 0, \text{ eg. } \mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & -(z_{11} + z_{22}) \end{bmatrix}, \quad 8 \text{ DOF}$$

▶Implementing Simple Constraints

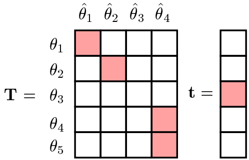
What for?

1. fixing external frame $\rightarrow \theta_i = \theta_i^0$ — 'trivial gauge'
2. representing additional knowledge $\rightarrow \theta_i = \theta_j$ — e.g. cameras share calibration matrix $\mathbf{K}$

We introduce reduced parameters $\hat{\theta}$:

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

Then $\mathbf{L}_r$ in LM changes to $\mathbf{L}_r\,\mathbf{T}$ and everything else stays the same



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 initialization by pseudoinverse $\theta^0 \mapsto \mathbf{T}^\dagger \theta^0$
- we need not compute derivatives for θ_j that correspond to all-zero rows $\mathbf{T}_j$ — fixed params

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

Part 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.

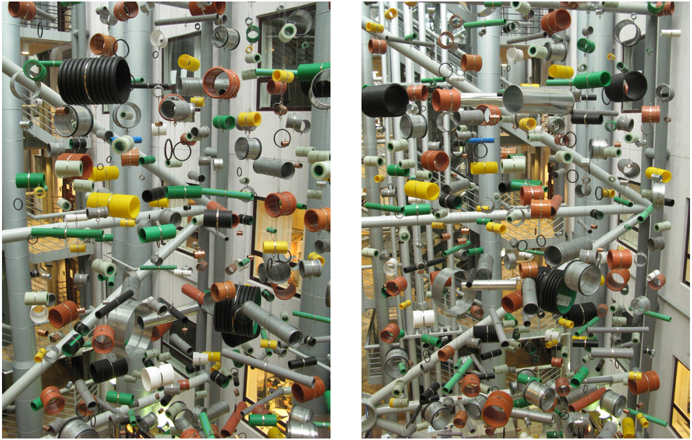
J. Gluckman and S. K. Nayar. Rectifying transformations that minimize resampling effects. In *Proc IEEE CS Conf on Computer Vision and Pattern Recognition*, vol. 1:111–117. 2001.

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.

What Are The Relative Distances?

- monocular vision already gives a rough 3D sketch because we understand the scene

What Are The Relative Distances?



Centrum för teknikstudier at Malmö Högskola, Sweden

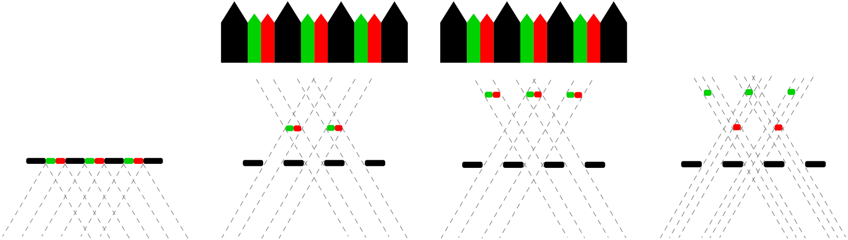
- we have no help from image interpretation here
- this is how difficult is low-level stereo we will attempt to solve

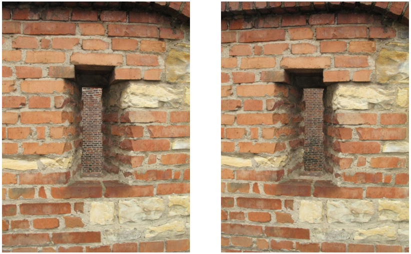
What Are The Relative Distances? (Why?)

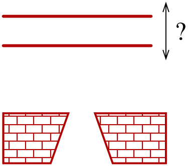
- a combination of lack of texture and occlusion $\longrightarrow$ ambiguous interpretation

Repetition: How Many Scenes Correspond to a Stereopair?

Consider the fence and the fortress worlds . . .

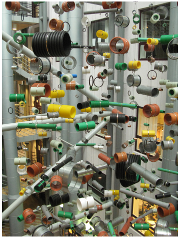






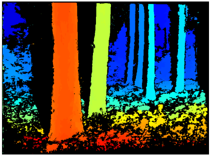
- lack of texture is a limiting case of repetition

How Difficult Is Stereo?

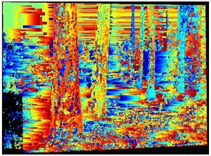


- when we do not recognize the scene and cannot use high-level constraints the problem seems difficult (right, less so in the center)
- most stereo matching algorithms do not require scene understanding prior to matching
- the success of a model-free stereo matching algorithm is unlikely:

left image



disparity map



disparity map from WTA

WTA Matching:

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

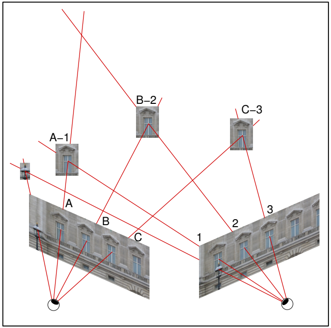
Why Model-Free Stereo Fails?

- lack of an occlusion model
- lack of a continuity model

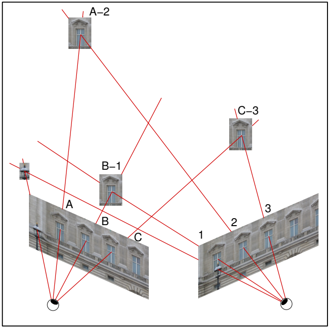
⇒ structural ambiguity

left image

right image

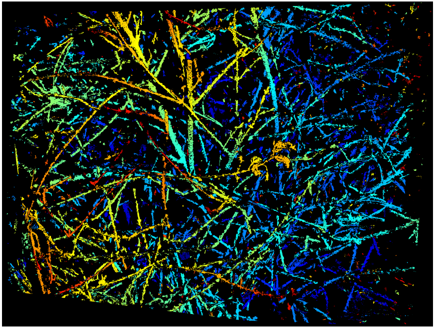


interpretation 1



interpretation 2

But What Kind of Continuity Model Applies Here?



- continuity alone is not a sufficient model
- occlusion model is more primal
- but occlusion model alone is insufficient, since it does not solve structural ambiguity

A Summary of Our Observations and an Outlook

- simple matching algorithms do not work
- decisions on matches are not independent due to occlusions

  occlusion constraint works along epipolars only
- occlusion model alone is insufficient — does not resolve the structural ambiguity
- a continuity model can resolve structural ambiguity

  but continuity is piecewise due to object boundaries
- in sufficiently complex scenes the only possibility is that stereopsis uses scene interpretation (or another-modality measurement)

Outlook:

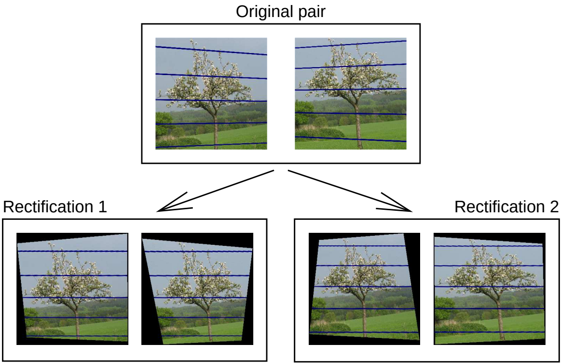
1. represent the occlusion constraint:
   - epipolar rectification
   - disparity
   - uniqueness as an occlusion constraint
2. represent piecewise continuity
   - ordering as a weak continuity model
3. use a consistent framework
   - looking for the most probable solution (MAP)

▶Epipolar Rectification

Problem: Given fundamental matrix $\mathbf{F}$ or camera matrices $\mathbf{P}_1$, $\mathbf{P}_2$, transform images so that epipolar lines become horizontal with the same row coordinate. The result is a standard stereo pair. for easier correspondence search

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 modify fundamental matrix $\mathbf{F} \mapsto \mathbf{H}_2^{-\top}\mathbf{F}\mathbf{H}_1^{-1}$ or cameras $\mathbf{P}_1 \mapsto \mathbf{H}_1\mathbf{P}_1$, $\mathbf{P}_2 \mapsto \mathbf{H}_2\mathbf{P}_2$.



- there is a 9-parameter family of rectification homographies for binocular rectification, see next

Rectification Example

Four cameras in general position

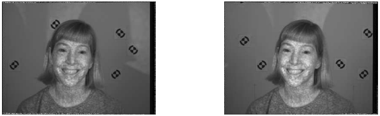

cam 1

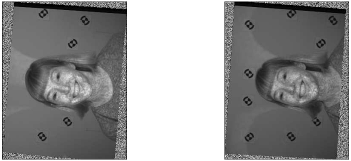

cam 2

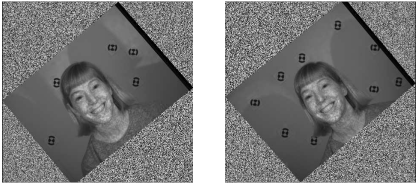

cam 3


cam 4

Rectified pairs


pair 1 – 2

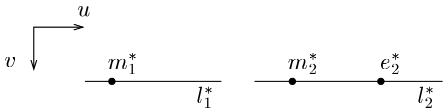

pair 2 – 4


pair 1 – 4

▶Rectification Homographies

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

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



rectified entities: $\mathbf{F}^*$, $\underline{\mathbf{l}}_2^*$, $\underline{\mathbf{l}}_1^*$, etc:

corresponding epipolar lines must be:

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

both conditions together give the rectified fundamental matrix

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

A two-step rectification procedure

1. Find some pair of <u>primitive rectification homographies</u> $\hat{\mathbf{H}}_1$, $\hat{\mathbf{H}}_2$
2. Upgrade them to a pair of <u>optimal rectification homographies</u> from the class preserving $\mathbf{F}^*$.

▶Geometric Interpretation of Linear Rectification

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

- we know that $\mathbf{F} = (\mathbf{Q}_1 \mathbf{Q}_2^{-1})^\top [\underline{\mathbf{e}}_1]_\times$ Slide 77
- we choose $\mathbf{Q}_1^* = \mathbf{K}_1^*$, $\mathbf{Q}_2^* = \mathbf{K}_2^* \mathbf{R}^*$; then

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

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

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

- we also want $\mathbf{b}^*$ from $\underline{\mathbf{e}}_1^* \simeq \mathbf{P}_1^* \underline{\mathbf{C}}_2^* = \mathbf{K}_1^* \mathbf{b}^*$ $\mathbf{b}^*$ in cam. 1 frame
- result:

$$\mathbf{R}^* = \mathbf{I}, \quad \mathbf{b}^* = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{K}_1^* = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}_2^* = \begin{bmatrix} k_{21} & k_{22} & k_{23} \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{29}$$

- rectified cameras are in canonical position with respect to each other

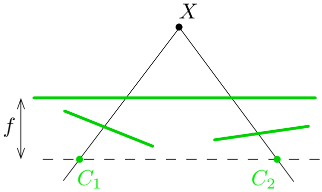
  not rotated, canonical baseline

- rectified calibration matrices can differ in the first row only
- when $\mathbf{K}_1^* = \mathbf{K}_2^*$ then the rectified pair is called the standard stereo pair and the homographies standard rectification homographies

▶cont'd

- rectification is a homography (per image)
  - ⇒ rectified camera centers are equal to the original ones
- standard rectified cameras are in canonical orientation
  - ⇒ rectified image projection planes are coplanar
- standard rectification guarantees equal rectified calibration matrices
  - ⇒ rectified image projection planes are equal

standard rectification homographies reproject onto a common image plane parallel to the baseline



Corollary

- the standard rectified stereo pair has vanishing disparity for 3D points at infinity
  - but known $\mathbf{F}$ alone does not give any constraints to obtain standard rectification homographies
  - for that we need either of these:
    1. projection matrices, or
    2. calibrated cameras, or
    3. a few points at infinity calibrating k_{1i}, k_{2i}, $i = 1, 2, 3$ in (29)

▶Primitive Rectification

Goal: Given fundamental matrix $\mathbf{F}$, derive some simple rectification homographies $\mathbf{H}_1$, $\mathbf{H}_2$

1. Let the SVD of $\mathbf{F}$ be $\mathbf{U}\mathbf{D}\mathbf{V}^\top = \mathbf{F}$, where $\mathbf{D} = \mathrm{diag}(1, d^2, 0)$, $\quad 1 \geq d^2 > 0$
2. decompose $\mathbf{D} = \mathbf{A}^\top \mathbf{F}^* \mathbf{B}$, where ($\mathbf{F}^*$ is given → Slide 151)

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

3. then

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

and the primitive rectification homographies are

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

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

- rectification homographies do exist
- there are other primitive rectification homographies, these suggested are just simple to obtain

►Primitive Rectification Suffices for Calibrated Cameras

Obs: calibrated cameras: $d = 1 \Rightarrow \hat{\mathbf{H}}_1$, $\hat{\mathbf{H}}_2$ are orthogonal

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.

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

3. the final rectification homographies are

$$\mathbf{H}_1 = \mathbf{K}\hat{\mathbf{H}}_1, \quad \mathbf{H}_2 = \mathbf{K}\hat{\mathbf{H}}_2$$

- we got a standard camera pair and non-negative disparity

$$\mathbf{P}_i^+ \stackrel{\text{def}}{=} \mathbf{K}_i^{-1}\mathbf{P}_i = \mathbf{R}_i \begin{bmatrix} \mathbf{I} & -\mathbf{C}_i \end{bmatrix}, \quad i = 1, 2$$

note we started from $\mathbf{E}$, not $\mathbf{F}$

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

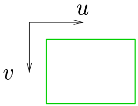
$$\mathbf{H}_2\mathbf{P}_2^+ = \mathbf{K}\hat{\mathbf{H}}_2\mathbf{P}_2^+ = \mathbf{K}\underbrace{\mathbf{A}\mathbf{U}^\top\mathbf{R}_2}_{\mathbf{R}^*}\begin{bmatrix} \mathbf{I} & -\mathbf{C}_2 \end{bmatrix} = \mathbf{K}\mathbf{R}^*\begin{bmatrix} \mathbf{I} & -\mathbf{C}_2 \end{bmatrix}$$

- one can prove that $\mathbf{B}\mathbf{V}^\top\mathbf{R}_1 = \mathbf{A}\mathbf{U}^\top\mathbf{R}_2$ with the help of (11)
- points at infinity project to $\mathbf{K}\mathbf{R}^*$ in both images $\Rightarrow$ they have zero disparity

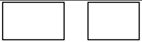
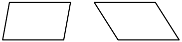
►The Degrees of Freedom in Epipolar Rectification

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

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



where $s \neq 0,\ u_0,\ l_1,\ l_2 \neq 0,\ l_3,\ r_1,\ r_2 \neq 0,\ r_3,\ q$ are 9 free parameters.

general	transformation		canonical	type
$l_1,\ r_1$	horizontal scales		$l_1 = r_1$	algebraic
$l_2,\ r_2$	horizontal skews		$l_2 = r_2$	algebraic
$l_3,\ r_3$	horizontal shifts		$l_3 = r_3$	algebraic
q	common special projective			geometric
s_v	common vertical scale			geometric
t_v	common vertical shift			algebraic
9 DoF			$9 - 3 = 6$ DoF	

- q is 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 $\mathbf{A}_2\mathbf{K}_2^* = \hat{\mathbf{K}}_2\mathbf{G}$

The Rectification Group

Corollary for Proposition 1 Let $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$ be (primitive or other) rectification homographies. Then $\mathbf{H}_1 = \mathbf{A}_1\bar{\mathbf{H}}_1, \quad \mathbf{H}_2 = \mathbf{A}_2\bar{\mathbf{H}}_2$ are also rectification homographies.

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

Proof:

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

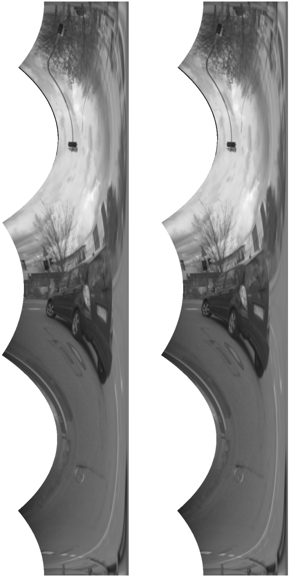
Optimal choice for the free parameters

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

$$\mathbf{A}_1^* = \arg\min_{\mathbf{A}_1} \iint_{\Omega} \left(\det J(\mathbf{A}_1 \hat{\mathbf{H}}_1 \underline{\mathbf{x}}) - 1\right)^2 d\mathbf{x}$$

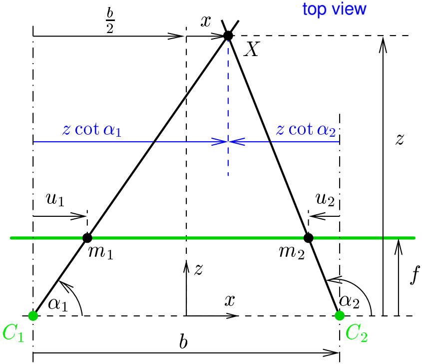
- by minimization of image information loss [Matoušek, ICIG 2004]
- non-linear rectification suitable for forward motion [Pollefeys et al. 1999], [Geyer & Daniilidis 2003]

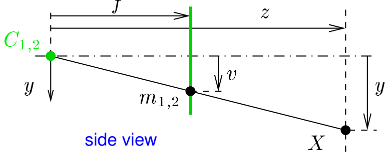
forward egomotion



rectified images, Pollefeys' method

►Binocular Disparity in Standard Stereo Pair





- Assumptions: single image line, standard camera pair

$$b = z \cot \alpha_1 - z \cot \alpha_2$$

$$u_1 = f \cot \alpha_1 \qquad u_2 = f \cot \alpha_2$$

$$b = \frac{b}{2} + x - z \cot \alpha_2$$

$X = (x, z)$ 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

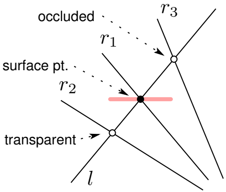
Observations

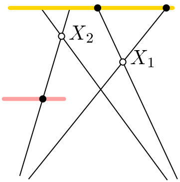
- constant disparity surface is a frontoparallel plane
- distant points have small disparity
- relative error in z is large for small disparity

$$\frac{1}{z}\,\frac{dz}{dd} = -\frac{1}{d}$$

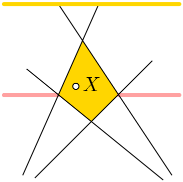
- increasing baseline increases disparity and reduces the error

▶Understanding Basic Occlusion Types





half occlusion

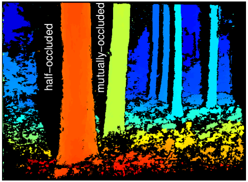


mutual occlusion

- surface point at the intersection of rays l and r_1 occludes a world point at the intersection (l, r_3) and implies the world point (l, r_2) is transparent, therefore

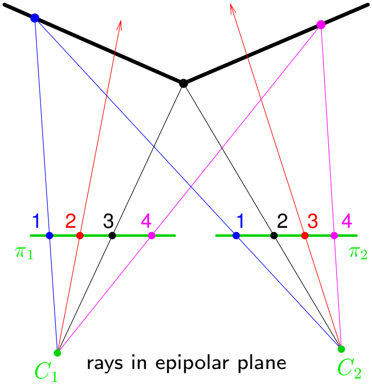
(l, r_3) and (l, r_2) are excluded by (l, r_1)

- in half-occlusion, every world point such as X_1 or X_2 is excluded by a binocularly visible surface point
  $\Rightarrow$ decisions on correspondences are not independent
- in mutual occlusion this is no longer the case: any X in the yellow zone is not excluded
  $\Rightarrow$ decisions in the zone are independent on the rest

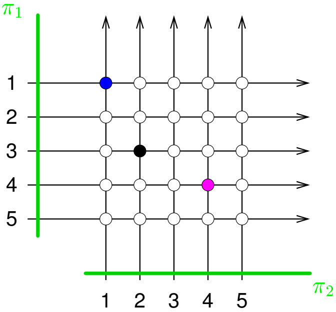


▶Matching Table

Based on 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: correspondences
- numerical values associated with nodes: descriptor similarities

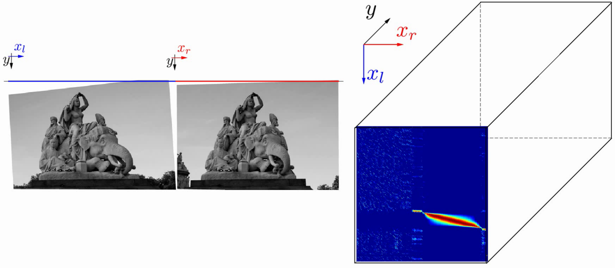
see next

Image Point Descriptors And Their Similarity

Descriptors: Tag image points by their (viewpoint-invariant) physical properties:

- texture window [Moravec 77]
- reflectance profile under a moving illuminant
- photometric ratios [Wolff & Angelopoulou 93-94]
- dual photometric stereo [Ikeuchi 87]
- polarization signature
- ...

- similar points are more likely to match
- we will compute image similarity for all 'match candidates' and get the matching table

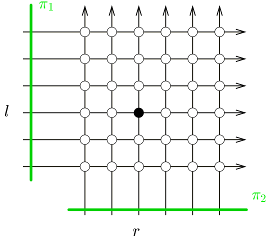


video

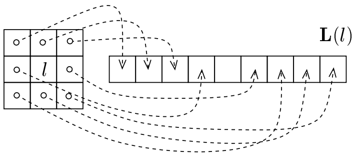
▶Constructing A Suitable Image Similarity

- 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:



in the left image:



- a natural descriptor similarity is $\operatorname{sim}(l, r) = \frac{\|\mathbf{L}(l) - \mathbf{R}(r)\|^2}{\sigma_I^2(l, r)}$
- σ_I^2 – the difference scale; a suitable (plug-in) estimate is $\frac{1}{2}\left[s^2\big(\mathbf{L}(l)\big) + s^2\big(\mathbf{R}(r)\big)\right]$, giving

$$\operatorname{sim}(l, r) = 1 - \underbrace{\frac{2\, s\big(\mathbf{L}(l), \mathbf{R}(r)\big)}{s^2\big(\mathbf{L}(l)\big) + s^2\big(\mathbf{R}(r)\big)}}_{\rho\left(\mathbf{L}(l), \mathbf{R}(r)\right)} \qquad (30)$$

$s^2(\cdot)$ is sample (co-)variance

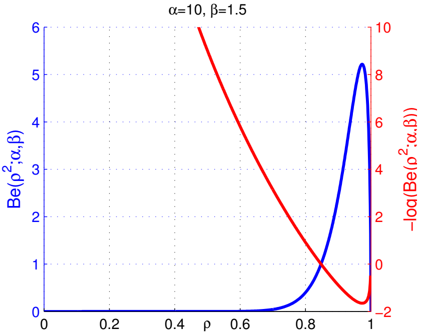
- ρ – MNCC – Moravec's Normalized Cross-Correlation [Moravec 1977]

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

- we choose some probability distribution on $[0,1]$, e.g. Beta distribution

$$p_1\big(\mathrm{sim}(l,r)\big) = \frac{1}{B(\alpha,\beta)}\,\rho^{2(\alpha-1)}(1-\rho^2)^{\beta-1}$$

- note that uniform distribution is obtained for $\alpha=\beta=1$

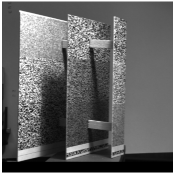


- the mode is at $\sqrt{\frac{\alpha-1}{\alpha+\beta-2}} \approx 0.9733$ for $\alpha=10,\ \beta=1.5$
- if we chose $\beta=1$ then the mode was at $\rho=1$
- perfect similarity is 'suspicious' (depends on expected camera noise level)
- from now on we will work with

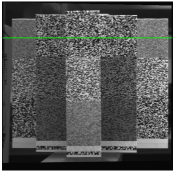
$$V_1\big(\mathrm{sim}(l,r)\big) = -\log p_1\big(\mathrm{sim}(l,r)\big) \qquad (31)$$

How A Scene Looks in The Filled-In Similarity Table

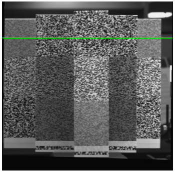
scene



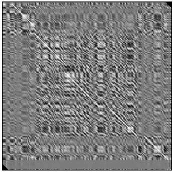
left image



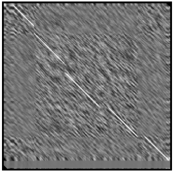
right image



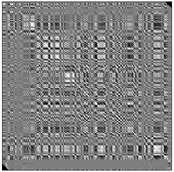
5×5 window

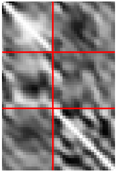


11×11 window

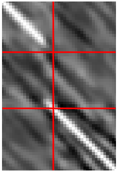


3×3 window

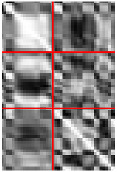




a good tradeoff



occlusion artefacts



undiscrimiable

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

constant disparity

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

depth discontinuity

- horizontal or vertical jump in correlation table

large image window

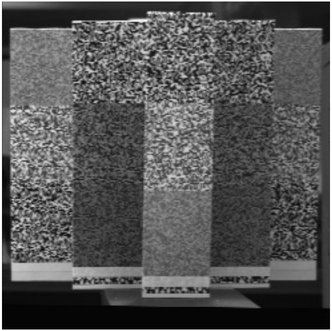
- better correlation
- worse occlusion localization

see next

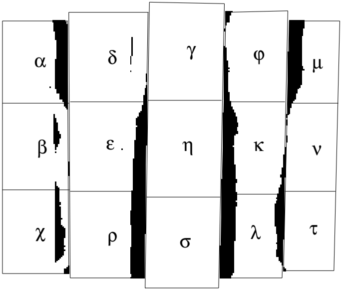
repeated texture

- horizontal and vertical block repetition

Note: Errors at Occlusion Boundaries for Large Windows



NCC, Disparity Error



- this used really large window of 25×25 px
- errors depend on the relative contrast across the occlusion boundary
- the direction of 'overlow' depends on the combination of texture contrast and edge contrast
- solutions:
  1. small windows (5×5 typically suffices)
  2. eg. 'guided filtering' methods for computing image similarity [Hosni 2011]

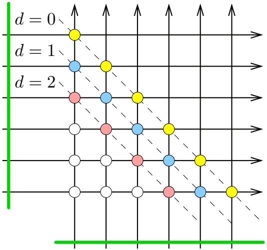
▶Marroquin's Winner Take All (WTA) Matching Algorithm

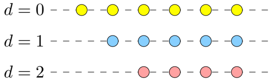
1. per left-image pixel: find the most similar right-image pixel

   $$\mathrm{SAD}(l, r) = \|\mathbf{L}(l) - \mathbf{R}(r)\|_1$$

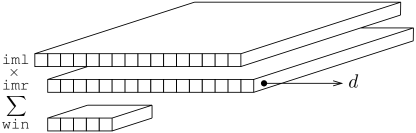
   L_1 norm instead of the L_2 norm in (30); unnormalized

2. represent the dissimilarity table diagonals in a compact form





3. use the 'image sliding aggregation algorithm'

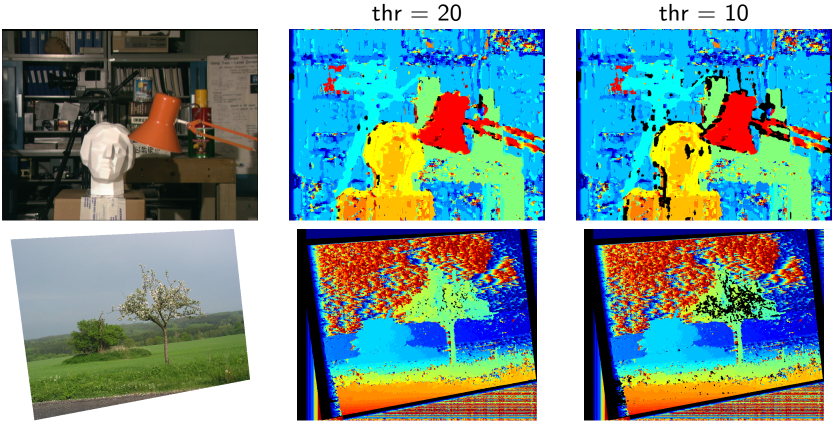


4. threshold results by maximal allowed dissimilarity

The 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)

 % the size of each local patch; it is N=(2r+1)^2 except for boundary pixels
 N = boxing(ones(size(iml)), winsize);

 % computing dissimilarity per pixel (unscaled SAD)
 for d = 0:disparityRange                               % cycle over all disparities
  slice = abs(imr(:,1:end-d) - iml(:,d+1:end));  % pixelwise dissimilarity
  V(:,d+1:end,d+1) = boxing(slice, winsize)./N;  % window aggregation
 end

 % collect winners, threshold, and output disparity map
 [cmap,dmap] = min(V,[],3);
 dmap(cmap > thr) = NaN;    % mask-out high dissimilarity pixels
end

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
```

WTA: Some Results



- results are 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

▶Negative Log-Likelihood of Observed Images

- given matching M what is the likelihood of observed data D?
- we need the ability 'not to match'
- matches are pairs $p_i = (l_i, r_i), \quad i = 1, \dots, n$
- we will mask-out some matches by a binary label $\lambda \in \{\mathrm{e}, \mathrm{m}\}$ — excluded, matched
- labeled matching is a set

$$M = \Big\{ \big(p_1, \lambda(p_1)\big), \big(p_2, \lambda(p_2)\big), \dots, \big(p_n, \lambda(p_n)\big) \Big\}$$

p_i are matching table pairs; there are no more than n in the table T

The negative log-likelihood is then — the likelihood of data D given labeled matching M

$$V(D \mid M) = \sum_{p_i \in M} V\big(D(p_i) \mid \lambda(p_i)\big)$$

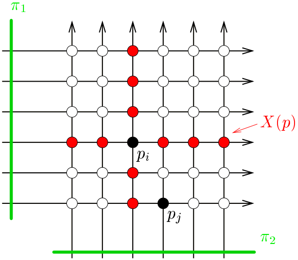
Our choice:

$$V\big(D(p_i) \mid \lambda(p_i) = \mathrm{e}\big) = V_{\mathrm{e}}$$ penalty for unexplained data, $V_{\mathrm{e}} \geq 0$

$$V\big(D(p_i) \mid \lambda(p_i) = \mathrm{m}\big) = V_1\big(D(l, r)\big)$$ probability of match $p_i = (l, r)$ from (31)

- the $V\big(D(p_i) \mid \lambda(p_i) = \mathrm{e}\big)$ could also be a non-uniform distribution but the extra effort does not pay off

▶Maximum Likelihood (ML) Matching



Uniqueness constraint: Each point in the left image matches at most once and vice versa.

A node set of T that follows the uniqueness constraint is called matching in graph theory

A set of pairs $M = \{p_i\}_{i=1}^n$, $p_i \in T$ is a matching iff
$$\forall p_i, p_j \in M, i \neq j : \; p_j \notin X(p_i).$$

The $X(p)$ is called the X-zone of p and it defines dependencies

- ML matching will observe the uniqueness constraint only
- epipolar lines are independent wrt uniqueness constraint
- we can solve the problem per image lines i independently:

⊛ H4; 2pt: How many are there: (1) binary partitionings of T, (2) maximal matchings in T; prove the results.

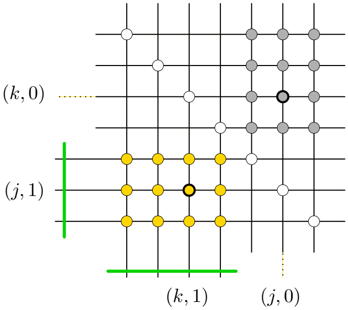
$$M^* = \arg\min_{M \in \mathcal{M}} \sum_{p \in M} V\big(D(p) \mid \lambda(p)\big) = \arg\min_{M \in \mathcal{M}} \Bigg(\underbrace{|M|_{\mathrm{e}} \cdot V_{\mathrm{e}}}_{\text{unexplained pixels}} + \underbrace{\sum_{p \in M:\; \lambda(p)=\mathrm{m}} V\big(D(p) \mid \lambda(p) = \mathrm{m}\big)}_{\text{matching likelihood proper}} \Bigg)$$

$\mathcal{M}$ – set of all perfect labeled matchings, $|M|_{\mathrm{e}}$ – number of pairs with $\lambda = \mathrm{e}$ in M, $|M|_{\mathrm{e}} \leq n$
perfect = every table row (column) contains exactly 1 match

- the total number of individual terms in the sum is n (which is fixed)

▶'Programming' The ML Matching Algorithm

- we restrict ourselves to a single (rectified) image line and reduce the problem to min-cost perfect matching
- extend every matching table pair $p \in T$, $p = (j, k)$ to 4 combinations $\big((j, s_j), (k, s_k)\big)$, $s_j \in \{0, 1\}$ and $s_k \in \{0, 1\}$ selects/rejects pixels for matching   unlike λ selecting matches
- binary label $m_{jk} = 1$ then means that (j, s_j) matches (k, s_k)



● $V_{jk} = V(D(j,k) \mid \lambda_{jk} = \mathrm{m})$ ● $V_{jk} = 0$

○ $V_{jk} = \frac{1}{2} V_{\mathrm{e}}$ + $V_{jk} = \infty$

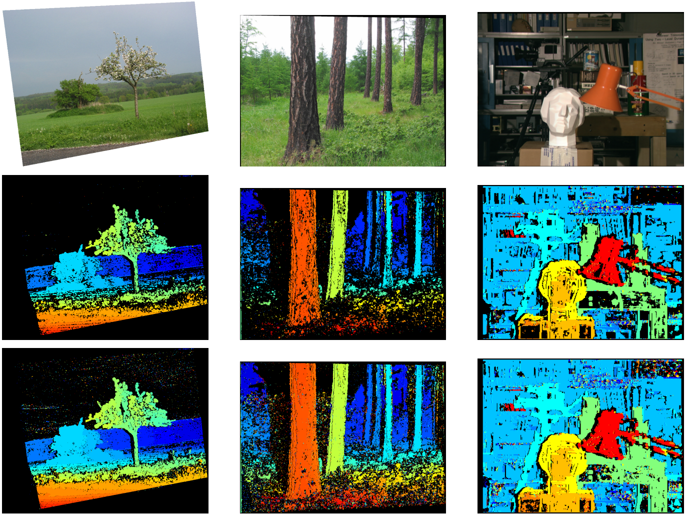
- each $(j, 1)$ either matches some $(k, 1)$ or it 'matches' $(j, 0)$
- each $(k, 1)$ either matches some $(j, 1)$ or $(k, 0)$
- if M is maximal in the yellow quadrant then there will be n auxiliary 'matches' in the gray quadrant
- otherwise every empty line in the yellow quadrant induces an empty column in the quadrant, the cost is $2 \cdot \frac{1}{2} V_{\mathrm{e}} = V_{\mathrm{e}}$

- our problem becomes minimum-cost perfect matching in an $(m + n) \times (m + n)$ table

$$M^+ = \arg\min_{M} \sum_{j,k} V_{jk} \cdot m_{jk}, \quad \sum_{k} m_{jk} = 1 \text{ for every } j, \quad \sum_{j} m_{jk} = 1 \text{ for every } k$$

- we collect our matches M^* in the yellow quadrant

Some Results for the ML Matching



- unlike the WTA we can efficiently control the density/accuracy tradeoff
- middle row: V_{e} set to error rate of 3% (and 61% density is achieved)   holes are black
- bottom row: V_{e} set to density of 76% (and 4.3% error rate is achieved)

Some Notes on ML Matching

- an algorithm for maximum weighted bipartite matching can be used as well, with $V \mapsto -V$
- maximum weighted bipartite matching = maximum weighted assignment problem

by eg. Hungarian Algorithm

Idea?: This looks simpler: Run matching with $V_\mathrm{e} = 0$ and then threshold the result to remove bad matches.

Ex: $V_\mathrm{e} = 8$

thresholding

8	3	**9**
10	**6**	9
7	1	8

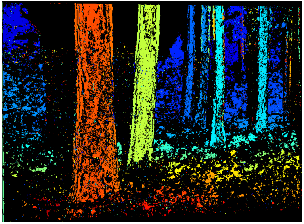
$V = 9 + 2 \cdot 8 = 25$

our ML matching

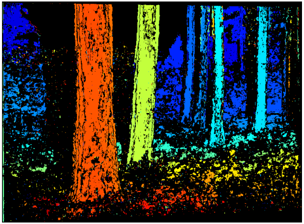
8	3	**9**
10	6	9
7	1	8

$V = 9 + 10 + 8 = 27$

- our matching gives a better cost, also greater cardinality (density)
- the idea was not good!



thresholding



our ML

A Stronger Model Needed

- notice many small isolated errors in the ML matching
- we need a continuity model
- does human stereopsis teach us something?

Potential models for M

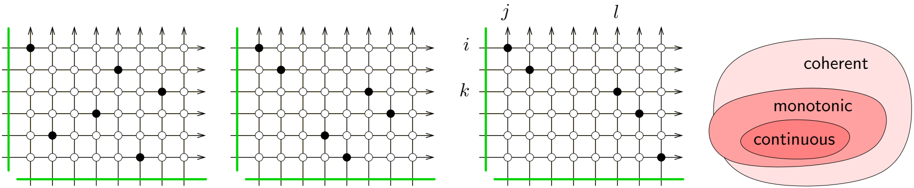
1. Monotonicity (ie. ordering preserved):

$$\text{For all } (i,j) \in M, (k,l) \in M, \quad k > i \Rightarrow l > j$$

Notation: $(i,j) \in M$ or $j = M(i)$ – left-image pixel i matches right-image pixel j.

2. Coherence [Prazdny 85]

"the world is made of objects each occupying a well defined 3D volume"

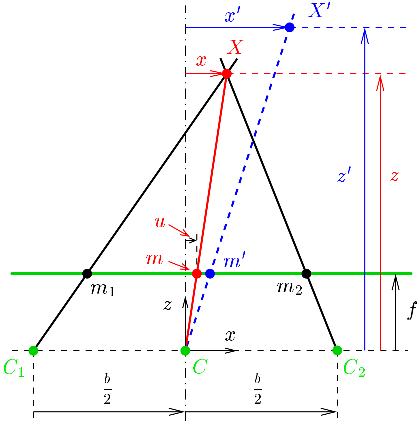


non-monotonic incoherent | non-monotonic coherent | monotonic coherent | model 'strength'

▶An Auxiliary Construct: Cyclopean Camera

Cyclopean coordinate u from the psychophysiology of vision [Julesz 1971]

$$\text{new: } u = f\,\frac{x}{z}, \quad \text{known: } d = f\,\frac{b}{z}, \quad x = \frac{b}{d}\,\frac{u_1+u_2}{2} \quad \Rightarrow \quad u = \frac{u_1+u_2}{2}$$



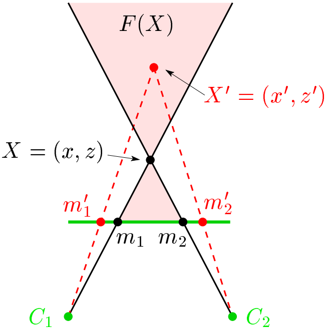
Disparity gradient [Pollard, Mayhew, Frisby 1985]

$$DG = \frac{|d-d'|}{|u-u'|} = \frac{\left|bf\left(\frac{1}{z}-\frac{1}{z'}\right)\right|}{\left|f\left(\frac{x}{z}-\frac{x'}{z'}\right)\right|} = \\ = b\,\frac{|z'-z|}{|xz'-x'z|}$$

- human stereovision fails to perceive a continuous surface when disparity gradient exceeds a limit

►Forbidden Zone and The Ordering Constraint

Forbidden zone $F(X)$: $DG > k$ with boundary $b\,(z' - z) = \pm k\,(xz' - x'z)$

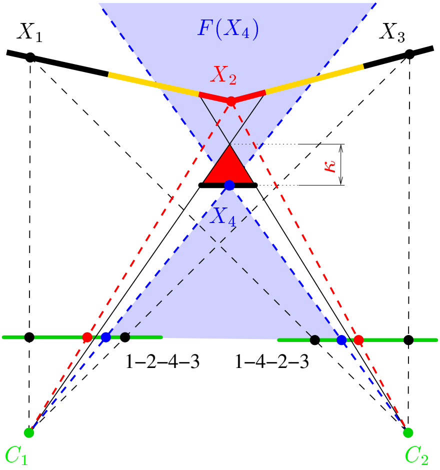


- boundary: a pair of lines in the $x - z$ plane

  a degenerate conic
- point $x = x'$, $z = z'$ lies on the boundary
- coincides with optical rays for $k = 2$
- small k means wide F

- disparity gradient limit is exceeded when $X' \in F(X)$
- symmetry: $X' \in F(X) \Leftrightarrow X \in F(X')$
- **Obs:** X' and X swap their <u>order</u> in the other image when $X' \in F(X)$ $k = 2$
- real scenes often preserve ordering
- thin and close objects violate ordering see next

Ordering and Critical Distance κ



- object (thick):
  - black – binocularly visible
  - yellow – half-occluded
  - red – ordering violated wrt foreground
- solid red zone of depth κ:
  - spatial points visible in neither camera
  - bounded by the foreground object

> Ordering is violated iff both X_i, X_j s.t. $X_i \in F(X_j)$ are visible in both cameras.

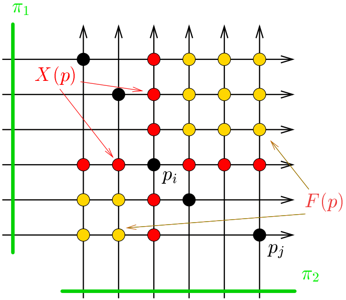
eg. X_2, X_4

- ordering is preserved in scenes where critical distances κ are not exceeded, ie. when 'the red background hides in the solid red zone'

> Thinner objects and/or wider baseline require flatter scenes to preserve ordering.

▶The X-zone and the F-zone in Matching Table T

- these are necessary and sufficient conditions for uniqueness and monotonicity



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

- **Uniqueness Constraint:**

A set of pairs $M = \{p_i\}_{i=1}^N$, $p_i \in T$ is a matching iff
$$\forall p_i, p_j \in M, i \neq j : \; p_j \notin X(p_i).$$

- **Ordering Constraint:**

Matching M is monotonic iff
$$\forall p_i, p_j \in M : \; p_j \notin F(p_i).$$

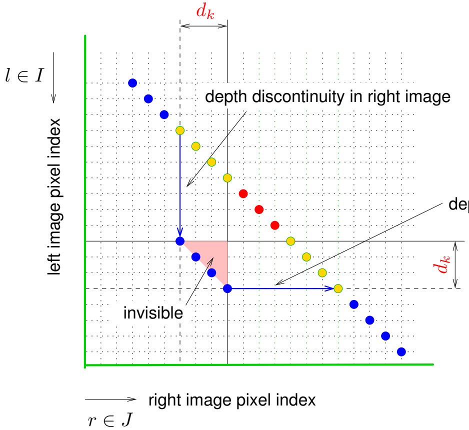
  - ordering constraint: matched points form a monotonic set in both images
  - ordering is a powerful constraint: monotonic matchings $O(4^N) \ll O(N!)$ all matchings in $N \times N$ table

⊛ 2: how many are there maximal monotonic matchings?

- uniqueness constraint is a basic occlusion model
- ordering constraint is a weak continuity model and partly also an occlusion model

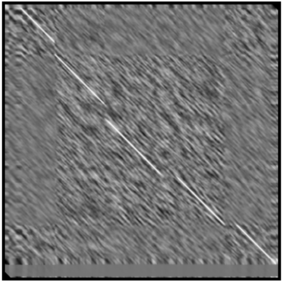
▶Understanding Matching Table

- this is essentially the picture from Slide 178



- binocularly visible foreground points
- binocularly visible background pts violating ordering
- monocularly visible points

d_k critical disparity



Bayesian Decision Task for Matching

Idea: $L(d, M)$ – decision cost (loss) d – our decision (matching) M – true correspondences

Choice: $L(d, M) : \begin{cases} \text{if } d = M & \text{then } L(d, M) = 0 \\ \text{if } d \neq M & \text{then } L(d, M) = 1 \end{cases}$ i.e. $L(d, M) = [d \neq M]$

Bayesian Loss

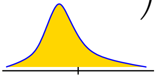
$$L(d \mid D) = \sum_{M \in \mathcal{M}} p(M \mid D)\, L(d, M)$$

$\mathcal{M}$ – the set of all matchings $D = \{I_L, I_R\}$ – data

Solution for the best decision d

$$d^* = \arg\min_d \sum_{M \in \mathcal{M}} p(M \mid D)\,(1 - [d = M]) = \arg\min_d \left(1 - \sum_{M \in \mathcal{M}} p(M \mid D)[d = M]\right) =$$

$$= \arg\max_d \sum_{M \in \mathcal{M}} p(M \mid D)\,[d = M] = \arg\max_M p(M \mid D) =$$



$$= \arg\min_M(-\log p(M \mid D)) \stackrel{\text{def}}{=} \arg\min_M V(M \mid D) = \arg\min_{M \in \mathcal{M}} \Big(\underbrace{V(D \mid M)}_{\text{likelihood}} + \underbrace{V(M)}_{\text{prior}}\Big)$$

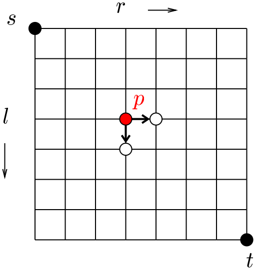
- this is Maximum Aposteriori Probability (MAP) estimate
  - other loss functions result in different solutions
  - our choice of $L(d, M)$ looks oversimple but it results in algorithmically tractable problems

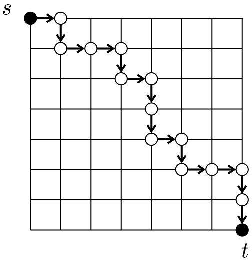
▶Constructing The Prior Model Term $V(M)$

- the prior $V(M)$ should capture
  1. uniqueness
  2. ordering
  3. coherence

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

- we need a suitable representation to encode $V(M)$
  - Every $p = (l, r)$ of the $|I| \times |J|$ matching table T (except for the last row and column) receives two succesors $(l+1, r)$ and $(l, r+1)$



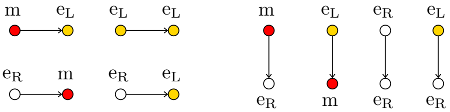


  - this gives an acyclic directed graph $\mathcal{G}$ — optimal paths in acyclic graphs are an easier problem
  - the set of s-t paths starting in s and ending in t will represent the set of matchings
  - all such s-t paths have **equal** length $n = |I| + |J| - 1$

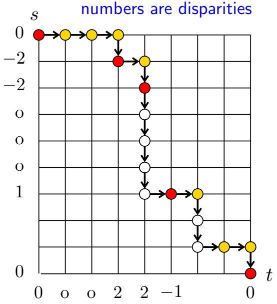
    all prospective matchings will have the same number of terms in $V(D \mid M)$ and in $V(M)$

Endowing s-t Paths with Useful Properties

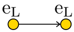
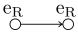
- introduce node labels $\Lambda = \{\mathrm{m}, \mathrm{e_L}, \mathrm{e_R}\}$ matched, left-excluded, right-excluded
- s-t path neighbors are allowed only some label combinations:



numbers are disparities



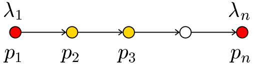
Observations

- no two neighbors have label m
- in each labeled s-t path there is at most one transition:
  1. $\mathrm{m} \to \mathrm{e_L}$ or $\mathrm{e_R} \to \mathrm{m}$ per matching table row,
  2. $\mathrm{m} \to \mathrm{e_R}$ or $\mathrm{e_L} \to \mathrm{m}$ per matching table column
- pairs labeled m on every s-t path satisfy uniqueness and ordering constraints
- transitions $\mathrm{e_L} \to \mathrm{e_R}$ or $\mathrm{e_R} \to \mathrm{e_L}$ along an s-t path allow skipping a contiguous segment in either or in both images this models half occlusion and mutual occlusion
- disparity change is the number of edges  or 

- a given monotonic matching can be traversed by one or more s-t paths

Labeled s-t paths

$$P = \big((p_1, \lambda_1), (p_2, \lambda_2), \dots, (p_n, \lambda_n)\big)$$

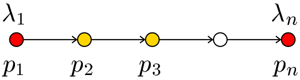


The Structure of The Prior Model $V(P)$ Gives a MC Recognition Problem

ideas:

- we choose energy of path P dependent on its labeling only
- we choose additive penalty per transition $\mathrm{e_L} \to \mathrm{e_L}$, $\mathrm{e_R} \to \mathrm{e_R}$, and $\mathrm{e_L} \to \mathrm{e_R}$, $\mathrm{e_R} \to \mathrm{e_L}$
- no penalty for $\mathrm{m} \to \mathrm{e_L}$, $\mathrm{m} \to \mathrm{e_R}$

Employing Markovianity



$$V(P) = V(\lambda_n, \lambda_{n-1}, \ldots, \lambda_1) = V(\lambda_n \mid \lambda_{n-1}, \ldots, \lambda_1) + V(\lambda_{n-1}, \ldots, \lambda_1) =$$

$$= V(\lambda_n \mid \lambda_{n-1}) + V(\lambda_{n-1}, \ldots, \lambda_1) = V(\lambda_1) + \sum_{i=2}^{n} V(\lambda_i \mid \lambda_{i-1})$$

The matching problem is then a decision over labeled s-t paths $P \in \mathcal{P}$:

$$P^* = \arg\min_{P \in \mathcal{P}} \left\{ V_{p_1}(D \mid \lambda_1) + V(\lambda_1) + \sum_{i=2}^{n} \Big[V_{p_i}(D \mid \lambda_i) + V(\lambda_i \mid \lambda_{i-1}) \Big] \right\} \tag{32}$$

- the data likelihood term $V_{p_i}(D \mid \lambda_i)$ is the same as in (31) on Slide 164
- note that one can add/subtract a fixed term from any of the functions V_p, V in (32)

A Choice of $V(\lambda_i \mid \lambda_{i-1})$

- A natural requirement: symmetry of probability $p(\lambda_i, \lambda_{i-1}) = e^{-V(\lambda_i,\, \lambda_{i-1})}$

$p(\lambda_i, \lambda_{i-1})$		λ_i: m	λ_i: e_L	λ_i: e_R
λ_{i-1}	m	0	$p(m, e)$	$p(m, e)$
	e_L	$p(m, e)$	$p(e, e)$	$p(e_L, e_R)$
	e_R	$p(m, e)$	$p(e_L, e_R)$	$p(e, e)$

3 DOF, 1 constraint $\Rightarrow$ 2 parameters

$$\alpha_1 = \frac{p(e_L, e_R)}{p(e, e)} \qquad 0 \le \alpha_1 \le 1$$

$$\alpha_2 = \frac{p(m, e)}{p(e, e)} \qquad 0 < \alpha_2 \le 1 + \alpha_1$$

- **Result** for $V(\lambda_i \mid \lambda_{i-1})$ (after subtracting common terms):

$V(\lambda_i \mid \lambda_{i-1})$		λ_i: m	λ_i: e_L	λ_i: e_R
λ_{i-1}	m	∞	0	0
	e_L	$\ln \frac{1+\alpha_1+\alpha_2}{2\,\alpha_2}$	$\ln \frac{1+\alpha_1+\alpha_2}{2}$	$\ln \frac{1+\alpha_1+\alpha_2}{2\,\alpha_1}$
	e_R	$\ln \frac{1+\alpha_1+\alpha_2}{2\,\alpha_2}$	$\ln \frac{1+\alpha_1+\alpha_2}{2\,\alpha_1}$	$\ln \frac{1+\alpha_1+\alpha_2}{2}$

by marginalization:

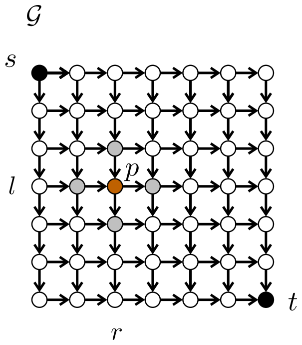
$$V(m) = \ln \frac{1+\alpha_1+\alpha_2}{2\,\alpha_2}$$

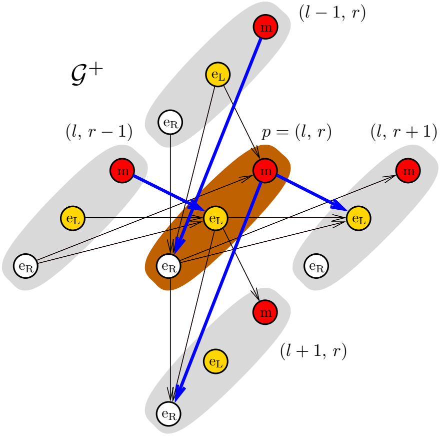
$$V(e_L) = V(e_R) = 0$$

parameters

- α_1 – likelihood of mutual occlusion ($\alpha_1 = 0$ forbids mutual occlusion)
- α_2 – likelihood of irregularity ($\alpha_2 \to 0$ helps suppress small objects and holes)
- α, β – similarity model parameters (see $V_1\big(D(l, r)\big)$ on Slide 164)
- V_e – penalty for disregarded data (see $V(D(p_i) \mid \lambda(p_i) = e)$ on Slide 170)

- given $\mathcal{G}$, construct directed graph $\mathcal{G}^+$
- triple of vertices per node of s-t path representing three hypotheses $\lambda(p)$ for $\lambda \in \Lambda$
- arcs have costs $V(\lambda_i \mid \lambda_{i-1})$, nodes have costs $V(D \mid \lambda_i)$
- orientation of $\mathcal{G}^+$ is inherited from the orientation of s-t paths
- we converted the shortest labeled-path problem to ordinary shortest path problem

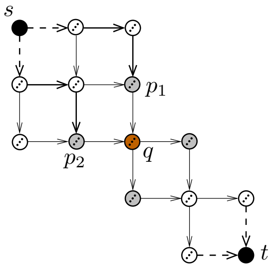




neighborhood of p; strong blue edges are of zero penalty

cont'd: Dynamic Programming on $\mathcal{G}^+$

- $\mathcal{G}^+$ is a topologically ordered directed graph
- we can use <u>dynamic programming</u> on $\mathcal{G}^+$



$$V^*_{s:q}(\lambda_q) = \min_{z \in \{p_1, p_2\}, \lambda_z \in \Lambda} \Big\{ V^*_{s:z}(\lambda_z) + V_z(D \mid \lambda_z) + V(\lambda_q \mid \lambda_z) \Big\}$$

$V^*_{s:q}(\lambda_q)$ – cost of min-path from s to label λ_q at node q

- complexity is $O(|I| \cdot |J|)$, ie. stereo matching on $N \times N$ images needs $O(N^3)$ time
- speedup by limiting the range in which the disparities $d = l - r$ are allowed to vary

Implementation of 3LDP in a few lines of code. . .

```
#define clamp(x, mi, ma) ((x) < (mi) ? (mi) : ((x) > (ma) ? (ma) : (x)))
#define MAXi(tab,j) clamp((j)+(tab).drange[1], (tab).beg[0], (tab).end[0])
#define MINi(tab,j) clamp((j)+(tab).drange[0], (tab).beg[0], (tab).end[0])

#define ARG_MIN2(Ca, La, C0, L0, C1, L1) if ((C0) < (C1)) { Ca = C0; La = L0; } else { Ca = C1; La = L1; }

#define ARG_MIN3(Ca, La, C0, L0, C1, L1, C2, L2) \
 if ( (C0) <= MIN(C1, C2) ) { Ca = C0; La = L0; } else if ( (C1) < MIN(C0, C2) ) { Ca = C1; La = L1; } else { Ca = C2; La = L2; }
```

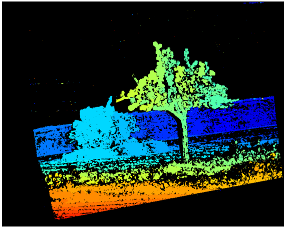
```
void DP3LForward(MatchingTableT tab) {

 int i = tab.beg[0]; int j = tab.beg[1];
 C_m[j][i-1] = C_m[j-1][i] = MAXDOUBLE;
 C_oL[j][i-1] = C_oR[j-1][i] = 0.0;
 C_oL[j-1][i] = C_oR[j][i-1] = -penalty[0];

 for(j = tab.beg[1]; j <= tab.end[1]; j++)
  for(i = MINi(tab,j); i <= MAXi(tab,j); i++) {

    ARG_MIN2(C_m[j][i], P_m[j][i],
             C_oR[j-1][i] + penalty[2], lbl_oR,
             C_oL[j][i-1] + penalty[2], lbl_oL);
    C_m[j][i] += 1.0 - tab.MNCC[j][i];

    ARG_MIN3(C_oL[j][i], P_oL[j][i], C_m[j-1][i], lbl_m,
             C_oL[j-1][i] + penalty[0], lbl_oL,
             C_oR[j-1][i] + penalty[1], lbl_oR);
    C_oL[j][i] += penalty[3];

    ARG_MIN3(C_oR[j][i], P_oR[j][i], C_m[j][i-1], lbl_m,
             C_oR[j][i-1] + penalty[0], lbl_oR,
             C_oL[j][i-1] + penalty[1], lbl_oL);
    C_oR[j][i] += penalty[3];
  }
}
```

```
void DP3LReverse(double *D, MatchingTableT tab) {

 int i,j; labelT La; double Ca;
 for(i=0; i<nl; i++) D[i] = nan; /* not-a-number */

 i = tab.end[0]; j = tab.end[1];
 ARG_MIN3(Ca, La, C_m[j][i], lbl_m,
          C_oL[j][i], lbl_oL, C_oR[j][i], lbl_oR);

 while (i >= tab.beg[0] && j >= tab.beg[1] && La > 0)
  switch (La) {
   case lbl_m: D[i] = i-j;
    switch (La = P_m[j][i]) {
     case lbl_oL: i--; break;
     case lbl_oR: j--; break;
     default: Error(...);
    } break;

   case lbl_oL: La = P_oL[j][i]; j--; break;
   case lbl_oR: La = P_oR[j][i]; i--; break;
   default: Error(...);
  }
}
```

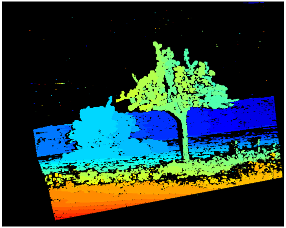
Some Results: AppleTree

left image

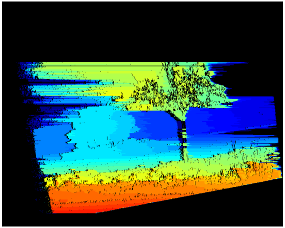
right image



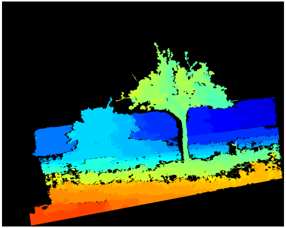
ML (slide 172)



3LDP (slide 186)



naïve DP [Cox et al. 1992]



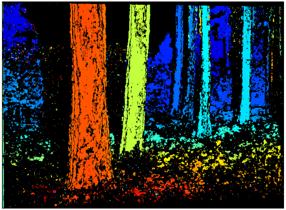
stable segmented 3LDP (see [SP])

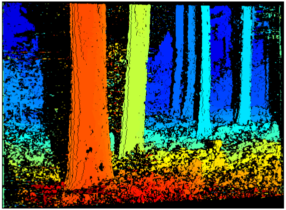
- 3LDP parameters α_i, V_{e} learned on Middlebury stereo data http://vision.middlebury.edu/stereo/

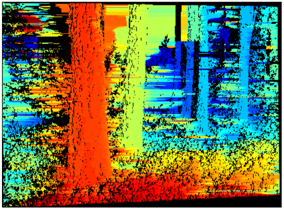
Some Results: Larch

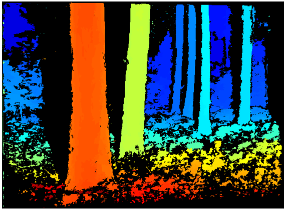

left image


right image


ML (slide 172)


3LDP (slide 186)


naïve DP


stable segmented 3LDP

- naïve DP does not model mutual occlusion
- but even 3LDP has errors in mutually occluded region
- stable segmented 3LDP has few errors in mutually occluded region since it uses a weak form of 'image understanding'

Algorithm Comparison

Winner-Take-All (WTA)

- the ur-algorithm [Marroquin 83] no model
- dense disparity map
- $O(N^3)$ algorithm, simple but it rarely works

Maximum Likelihood (ML)

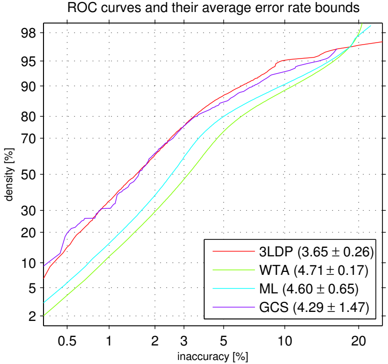
- semi-dense disparity map
- many small isolated errors
- models basic occlusion
- $O(N^3 \log(NV))$ algorithm max-flow by cost scaling

MAP with Min-Cost Labeled Path (3LDP)

- semi-dense disparity map
- models occlusion in flat, piecewise continuos scenes
- has 'illusions' if ordering does not hold
- $O(N^3)$ algorithm

Stable Segmented 3LDP

- better (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
- GCS is the one used in the exercises
- more algorithms at `http://vision.middlebury.edu/stereo/` (good luck!)

Part VIII

Shape from Reflectance

mostly covered by

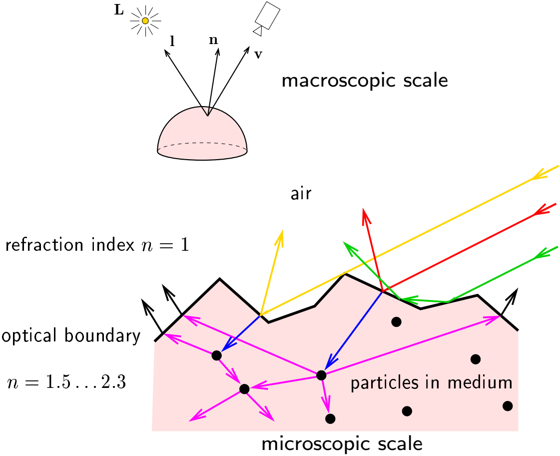
Forsyth, David A. and Ponce, Jean. *Computer Vision: A Modern Approach*. Prentice Hall 2003. Chap. 5

additional references

R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):439–451, July 1988.

P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. In *Proc Conf Computer Vision and Pattern Recognition*, pp. 1060–1066, 1997.

►Basic Surface Reflectance Mechanisms



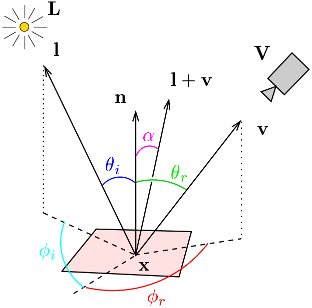
- reflection on (rough) optical boundary
- masking and shadowing
- interreflection
- refraction into the body
- subsurface scattering
- refraction into the air

▶Parametric Reflectance Models

Image intensity (measurement) at pixel m — given by surface reflectance function R

$$J(m) = \eta\, f_{i,r}(\theta_i, \phi_i; \theta_r, \phi_r) \cdot \underbrace{\frac{\Phi_e}{4\pi \|\mathbf{L}-\mathbf{x}\|^2}}_{\sigma}\, \mathbf{n}^\top \mathbf{l} = R(\mathbf{n}), \qquad \mathbf{l} = \frac{\mathbf{L}-\mathbf{x}}{\|\mathbf{L}-\mathbf{x}\|}$$

- η – sensor sensitivity — for simplicity, we select $\eta = 2\pi$
- $f_{i,r}()$ – bidirectional reflectance distribution function (BRDF)
  $[f_{i,r}()] = \mathrm{sr}^{-1}$ how much of irradiance in Wm^{-2} is redistributed per solid angle element
- $\mathbf{L}$ – point light source position
- Φ_e – radiant power of the light source, $[\Phi_e] = \mathrm{W}$
- $\mathbf{n}$ – surface normal
- σ – irradiance of a surfel orthogonal to incident light direction



pixel projected onto surface

Isotropic (Lambertian) reflection

[Lambert 1760]

no optical boundary

$$f_{i,r}(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{\rho}{2\pi}, \qquad \rho \text{ – albedo}$$

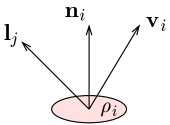
$$J(m) = \sigma \rho \cos\theta_i = \sigma \rho\, \mathbf{n}^\top \mathbf{l}$$

▶Photometric Stereo

Lambertian model (light $j \in \{1, 2, 3\}$, pixel $i \in \{1, \ldots, n\}$)

$$J_{ji} = (\sigma_j \, \mathbf{l}_j)^\top (\rho_i \, \mathbf{n}_i) = \mathbf{s}_j^\top \, \mathbf{b}_i$$

$\mathbf{b}_i$ – scaled normals, $\mathbf{s}_j$ – scaled lights



3 independent scaled lights and n scaled normals, one per pixel (in n pixels); can be stacked in matrices:

$$\begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \\ J_{31} & J_{32} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1^\top \mathbf{b}_1 & \mathbf{s}_1^\top \mathbf{b}_2 \\ \mathbf{s}_2^\top \mathbf{b}_1 & \mathbf{s}_2^\top \mathbf{b}_2 \\ \mathbf{s}_3^\top \mathbf{b}_1 & \mathbf{s}_3^\top \mathbf{b}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1^\top \\ \mathbf{s}_2^\top \\ \mathbf{s}_3^\top \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 \end{bmatrix}$$

$n = 2$ pixels, 3 lights

pixel indexing i:

1	2	3	4
5	6	7	8
9	10	11	12

in general, stacked per columns:

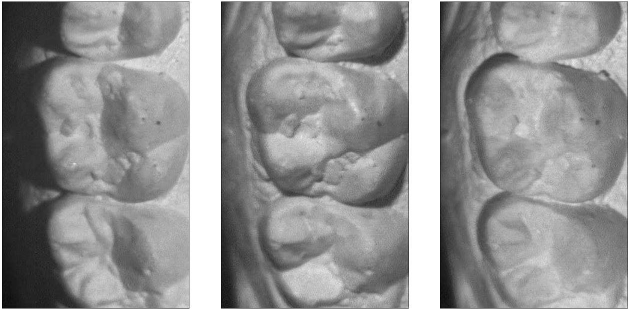
$$\mathbf{S} = [\,\mathbf{s}_1, \, \mathbf{s}_2, \, \mathbf{s}_3\,] \in \mathbb{R}^{3,3} \qquad \mathbf{B} = [\,\mathbf{b}_1, \, \mathbf{b}_2, \, \ldots, \, \mathbf{b}_n] \in \mathbb{R}^{3,n}$$

Solution to Photometric Stereo

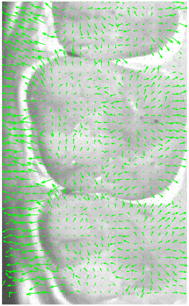
$$\mathbf{J} = \mathbf{S}^\top \mathbf{B} \quad \Rightarrow \quad \mathbf{B} = \mathbf{S}^{-\top} \mathbf{J} \qquad \mathbf{J} \in \mathbb{R}^{3,n}$$

$$\rho_i = \|\mathbf{b}_i\| \quad \text{albedo map}, \qquad \mathbf{n}_i = \frac{1}{\rho_i} \mathbf{b}_i \quad \text{needle map}$$

Photometric Stereo: Plaster Cast Example



input images (known lights)



needle & albedo maps

We have: 1. shape (surface normals), 2. intrinsic texture (albedo)

The shape can be represented as unit normal vectors $\mathbf{n}$ or as a gradient field (p, q):

$$\mathbf{n}(u,v) = \big(n_1(u,v),\, n_2(u,v),\, n_3(u,v)\big),$$

$$\frac{\partial z(u,v)}{\partial u} \stackrel{\text{def}}{=} z_u(u,v) = p(u,v) = \pm\frac{n_1(u,v)}{2n_3(u,v)^2 - 1},$$

$$\frac{\partial z(u,v)}{\partial v} \stackrel{\text{def}}{=} z_v(u,v) = q(u,v) = \pm\frac{n_2(u,v)}{2n_3(u,v)^2 - 1}$$

▶The Integration Algorithm of Frankot and Chellappa (FC)

Task: Given gradient fields $p(u,v)$, $q(u,v)$, find height function $z(u,v)$ such that z_u is close to p and z_v is close to q in the sense of a functional norm.

$$z^* = \arg\min_z Q(z), \qquad Q(z) = \iint |z_u(u,v) - p(u,v)|^2 + |z_v(u,v) - q(u,v)|^2 \, du\, dv$$

In the Fourier domain this can be written as $\mathcal{F}(z;\boldsymbol{\omega}) = \frac{1}{2\pi} \iint z(u,v) e^{-j(u\omega_u + v\omega_v)}\, du\, dv$

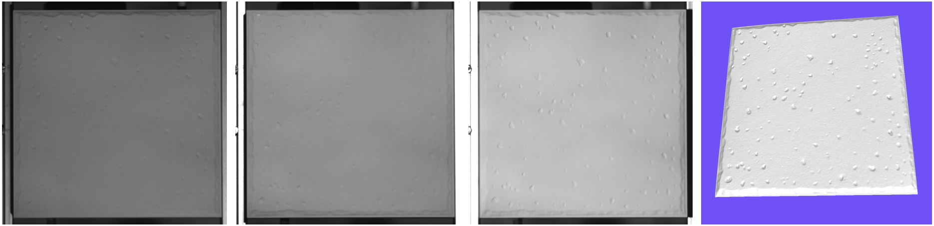
$$Q(z) = \iint \underbrace{|j\omega_u\, \mathcal{F}(z;\boldsymbol{\omega}) - \mathcal{F}(p;\boldsymbol{\omega})|^2 + |j\omega_v\, \mathcal{F}(z;\boldsymbol{\omega}) - \mathcal{F}(q;\boldsymbol{\omega})|^2}_{A(\mathcal{F}(z;\boldsymbol{\omega}))} \, d\boldsymbol{\omega}, \qquad \boldsymbol{\omega} = (\omega_u, \omega_v)$$

and its minimiser is from vanishing formal derivative of $A(\mathcal{F}(z;\boldsymbol{\omega}))$ wrt $\mathcal{F}(z;\boldsymbol{\omega})$ [Frankot & Chellappa 1988]

$$\mathcal{F}(z;\boldsymbol{\omega}) = -\frac{j\omega_u}{|\boldsymbol{\omega}|^2}\, \mathcal{F}(p;\boldsymbol{\omega}) - \frac{j\omega_v}{|\boldsymbol{\omega}|^2}\, \mathcal{F}(q;\boldsymbol{\omega})$$

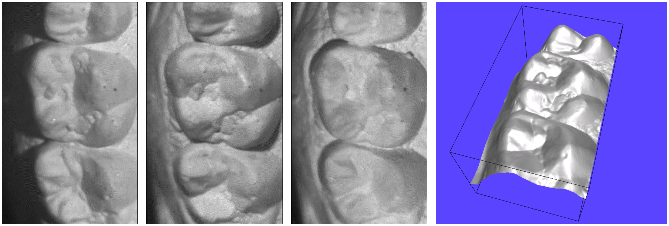
```
[m,n] = size(p);
Wu = fft2(fftshift([-1,0,1]/2),m,n); % discrete differential operator
Wv = fft2(fftshift([-1;0;1]/2),m,n);
Z = -(Wu.*fft2(p) + Wv.*fft2(q))./(abs(Wu).^2 + abs(Wv).^2 + eps);
z = real(ifft2(Z));
```

Photometric Stereo: Examples



3 input images

surface



3 input images

surface

- integrated by the FC algorithm from Slide 197
- bias due to interreflections can be removed

[Drew & Funt, JOSA-A 1992]

▶Integrability of a Vector Field

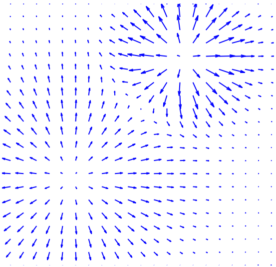
- not every vector field $p(u, v)$, $q(u, v)$ is integrable (born by a surface $z(u, v)$)
- integrability constraint

$$p_v(u, v) = q_u(u, v)$$

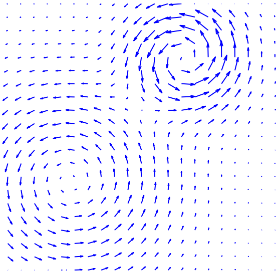
- this is because a regular surface has $\operatorname{rot} \nabla z(u, v) = 0$ irrotational gradient field

$$z_{uv}(u, v) = z_{vu}(u, v)$$

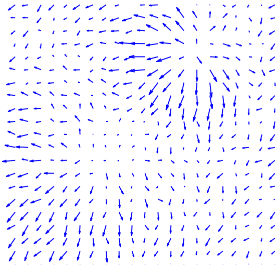
- noise causes non-integrability
- the FC algorithm finds the closest integrable surface



integrable



non-integrable



non-integrable (noisy)

Optimal Light Configurations

For n lights $\mathbf{S}$ the error $\Delta\mathbf{b} = \mathbf{S}^{-\top}\Delta\mathbf{J}$ in normal $\mathbf{b}$ due to error $\Delta\mathbf{J}$ in image is

$$\epsilon(\mathbf{S}) = E\left[\Delta\mathbf{b}^\top\Delta\mathbf{b}\right] = E\left[\Delta\mathbf{J}^\top(\mathbf{S}^\top\mathbf{S})^{-1}\Delta\mathbf{J}\right] = \sigma^2\,\mathrm{tr}\left[(\mathbf{S}\mathbf{S}^\top)^{-1}\right] \geq \frac{9\sigma^2}{n}.$$

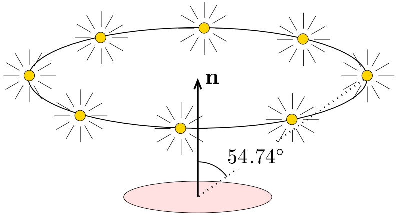
assuming pixel-independent normal camera noise $\Delta J_i \sim N(0,\sigma)$

The error ϵ is minimum if [Drbohlav & Chantler 2005]

$$\mathbf{S}\mathbf{S}^\top = \frac{n}{3}\,\mathbf{I}, \qquad \text{where} \quad \mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n]$$

- either $n \geq 3$ equidistant and equiradiant lights on a circle of uniform slant of $\arctan\sqrt{2} \approx 54.74°$
- $n-1$ lights in this configuration plus a light parallel to the sum $\sum_{i=1}^{n-1}\mathbf{s}_i$
- or light matrix $\mathbf{S}$ is a concatenation of optimal solutions (each of ≥ 3 lights)

eg. 3 optimally placed $(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$ + 3 lights $(\mathbf{s}_4, \mathbf{s}_5, \mathbf{s}_6) = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3) + \alpha$ rotated by angle α around $\mathbf{n}$



Uncalibrated Photometric Stereo

Factorization $\mathbf{J} = \mathbf{S}^\top \mathbf{B}$ [Hayakawa94]

LS solution by SVD decomposition of $\mathbf{J} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$

$\mathbf{S} = \mathbf{D}_{1:3}\mathbf{U}^\top$ scaled pseudo-lights

$\mathbf{B} = (\mathbf{V}_{1:3})^\top$ scaled pseudo-normals $\mathbf{V}_{1:3}$ are columns 1–3

Ambiguity $\mathbf{J} = \mathbf{S}^\top \mathbf{B} = \underbrace{\mathbf{S}^\top \mathbf{A}^{-1}}_{\bar{\mathbf{S}}^\top} \underbrace{\mathbf{A}\mathbf{B}}_{\bar{\mathbf{B}}}, \quad \mathbf{A} \in GL(3)$ [Koenderink94]

information	ambiguity		
3+ normals $\bar{\mathbf{B}}$ known	$\lambda\mathbf{I}$	(identity 3×3 mtx) $\bar{\mathbf{B}} = \mathbf{A}\mathbf{B} \;\Rightarrow\; \mathbf{A}$	$\mathbf{B}$ is measured
uniform albedo	$\lambda\mathbf{R}$	(orthogonal 3×3 mtx) 6 points: $\lVert\mathbf{A}\mathbf{b}_i\rVert = 1 \Rightarrow \mathbf{b}_i^\top \mathbf{A}^\top \mathbf{A}\mathbf{b}_i = 1 \Rightarrow \mathbf{A}^\top\mathbf{A} \Rightarrow \mathbf{A}$ up to rot. (Choleski)	[Drew92]
equal light intensity	$\lambda\mathbf{R}$	$\lVert\mathbf{s}_j \mathbf{A}^{-1}\rVert = 1 \Rightarrow \mathbf{A}$ up to rot.	[Hayakawa94]
integrable normals $p_v = q_u$ for $\mathbf{n} \sim (p, q, 1)$	$\begin{bmatrix} \lambda & 0 & \mu \\ 0 & \lambda & \nu \\ 0 & 0 & \tau \end{bmatrix}$	generalized bas-relief ambiguity	[Yuille99, Fan97, Belhumeur99]
uniform albedo and integrability	$\lambda\mathbf{I}$		
integrability and 2+ specular pts	$\lambda\mathbf{I}$		[Drbohlav & Chantler, ICCV 2005]

▶Generalized Bas Relief Ambiguity (GBR)

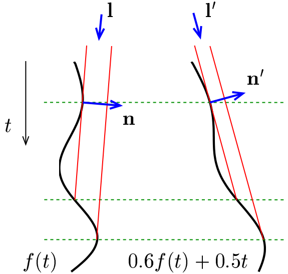
GBR maps surface $z'(u,v) = \lambda z(u,v) + \mu\, u + \nu\, v$, i.e. it maps normals to $\mathbf{n}' = \mathbf{G}\mathbf{n}$, where

$$\mathbf{G} = \begin{bmatrix} \lambda & 0 & -\mu \\ 0 & \lambda & -\nu \\ 0 & 0 & 1 \end{bmatrix}$$

Obs: If normals change $\mathbf{n}' = \mathbf{G}\mathbf{n}$ and lights change $\mathbf{l}' = \mathbf{G}^{-\top}\,\mathbf{l}$ then Lambertian shading does not change:

$$\mathbf{n}'^{\top}\mathbf{l}' = (\mathbf{n}^{\top}\mathbf{G}^{\top})(\mathbf{G}^{-\top}\mathbf{l}) = \mathbf{n}^{\top}\mathbf{l}$$

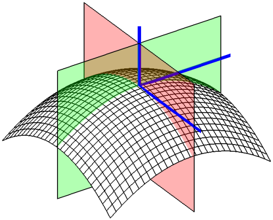

Reproduced from [Belhumeur et al. 1997]



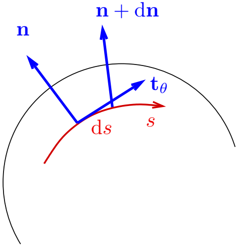
Obs: Shadow boundaries of surface $\mathcal{S}$ illuminated by light $\mathbf{l}$ are identical to those of surface $\mathcal{S}'$ transformed by GBR $\mathbf{G}$ and illuminated by light $\mathbf{l}' = \mathbf{G}^{-\top}\mathbf{l}$

weak assumptions [Belhumeur et al. 1997]

▶A Quick Glance at the Classical Differential Geometry of Surfaces



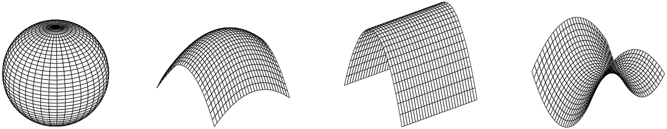
Darboux frame



$\kappa_\theta = \mathbf{t}_\theta^\top \dfrac{\mathrm{d}\mathbf{n}}{\mathrm{d}s}$	normal curvature, direction θ
κ_1, κ_2	principal curvatures
$K = \kappa_1 \cdot \kappa_2$	Gaussian curvature
$H = \kappa_1 + \kappa_2$	mean curvature

$$\kappa_\theta = \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta$$

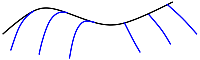
	umbilical	elliptical	parabolic	hyperbolic
convex	$\kappa_1 = \kappa_2 > 0$	$\kappa_1 > 0,\ \kappa_2 > 0$	$\kappa_1 > 0,\ \kappa_2 = 0$	$\kappa_1 > 0,\ \kappa_2 < 0$
concave		$\kappa_1 < 0,\ \kappa_2 < 0$		



the transition elliptic $\to$ parabolic $\to$ hyperbolic occurs at parabolic lines

non-umbilical surface like a torus

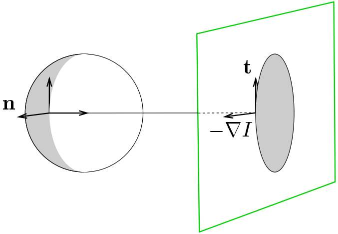
▶Occluding Contour Structure



smooth self-occlusion contour (back)
not smooth contour (mane)

- surface curves are tangent to smooth self-occlusion contour

- isophotes are surface curves ⇒ their density approaches infinity on smooth self-occlusion contour



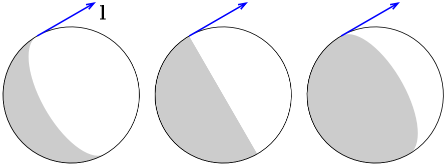
$$\mathbf{n} = \mathbf{Q}^\top \underline{\mathbf{t}} \qquad \text{optical plane normal}$$

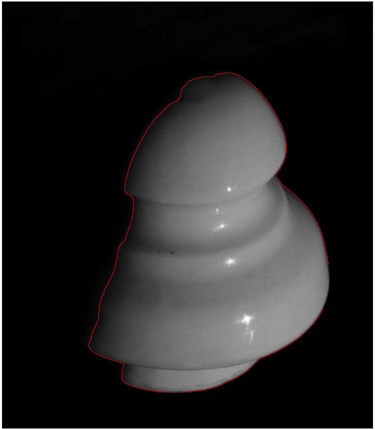
$$K = \kappa_s\, \kappa_t \quad \to \quad \operatorname{sign}(K) = \operatorname{sign}(\kappa_t)$$

$\kappa_s > 0$ – curvature in the direction of sight
κ_t – occluding contour curvature
$\mathbf{x}_{st} = 0$ since $\mathbf{x}_s \simeq \mathbf{v}$ [Koenderink 84]

- this is a basis for shape from occluding contour

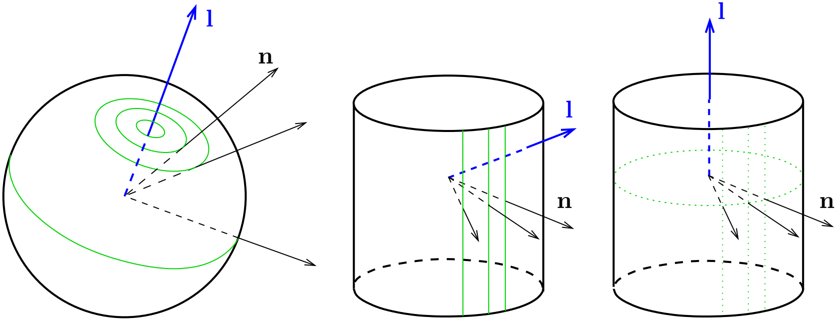
Self-Shadow Contour Structure





- loci where occluding and self-shadow meet: the projection of light direction vector to image plane is tangent to the contour there

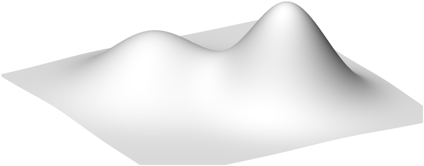
Isophotes on Simple Lambertian Surfaces



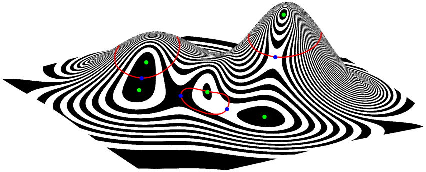
Surface is parameterized by: σ – slant, τ – tilt, where $\mathbf{n}^\top \mathbf{l} = \cos\sigma$

- isophotes – green
- apex – where $\mathbf{n} \simeq \mathbf{l}$
- isophotes parallel to rulings on developable surfaces
- illuminant on cylinder axis: constant reflectance cylindrical part illumination w/o shading
- in general: isophotes are parallel to zero-curvature principal direction

Isophotes on a Complex Surface



shaded Lambertian surface



isophotes w/ approximate parabolic curves

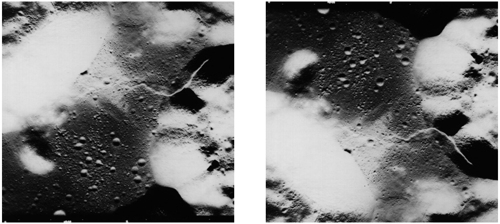
singular image points

- Lambertian apex: move with light, $\mathbf{n} = \mathbf{l}$ (T1)
- extrema and saddles on parabolic lines: move along parabolic lines (T2)
- planar points: do not move (not shown)
- specular points: move with light and/or viewer but slower (not shown)

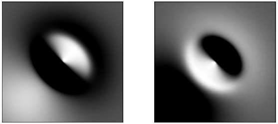
[Koenderink & van Doorn 1980]

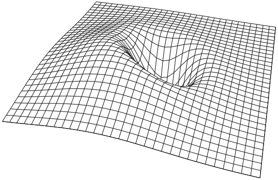
The Crater Illusion

Ambiguity in Local Shading and The Human Vision Preference



Apollo 17 landing site (Taurus-Littrow); courtesy of NASA





bottom: crater-like surface
top: surface illuminated from lower-left and top-right

Shading at Lambertian apex:

$$K^2 = \det\left(\mathbf{H}\mathbf{G}^{-1}\right)$$

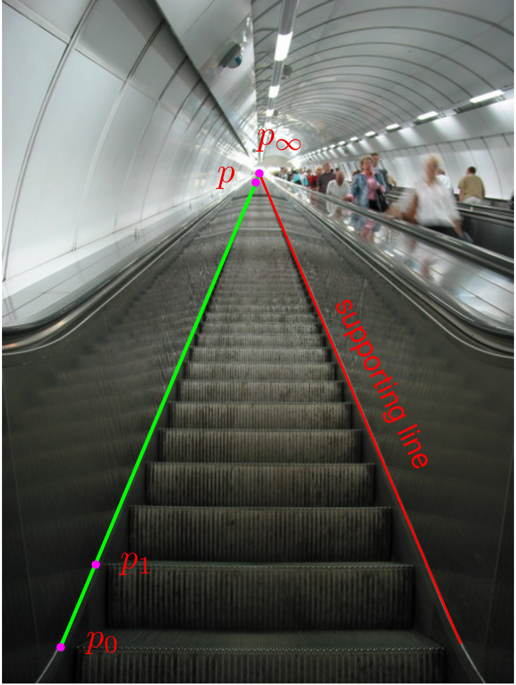
$$2H^2 - K = -\frac{1}{2}\,\mathrm{tr}\left(\mathbf{H}\mathbf{G}^{-1}\right)$$

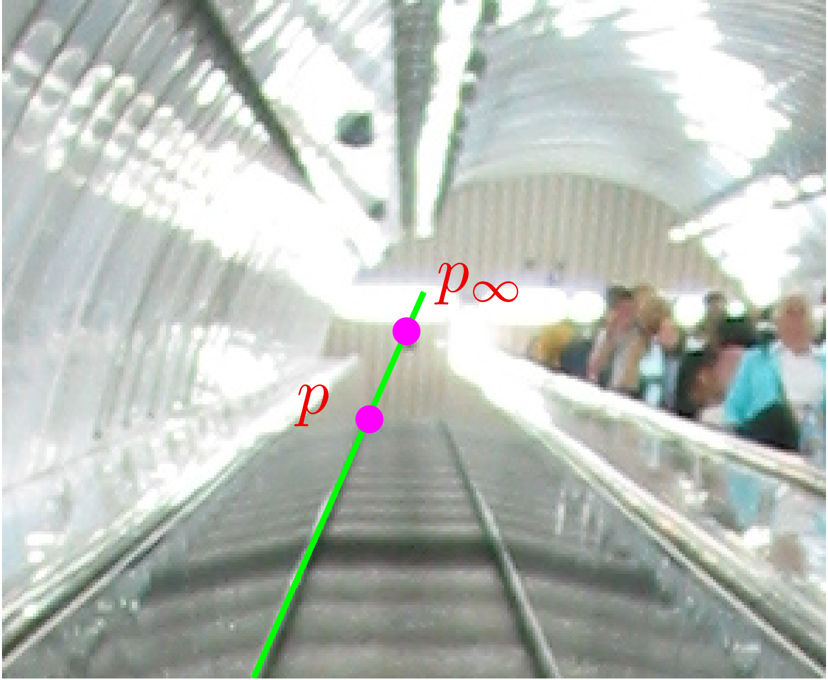
$$\mathbf{H} = \begin{bmatrix} I_{uu} & I_{uv} \\ I_{uv} & I_{vv} \end{bmatrix}$$ image Hessian

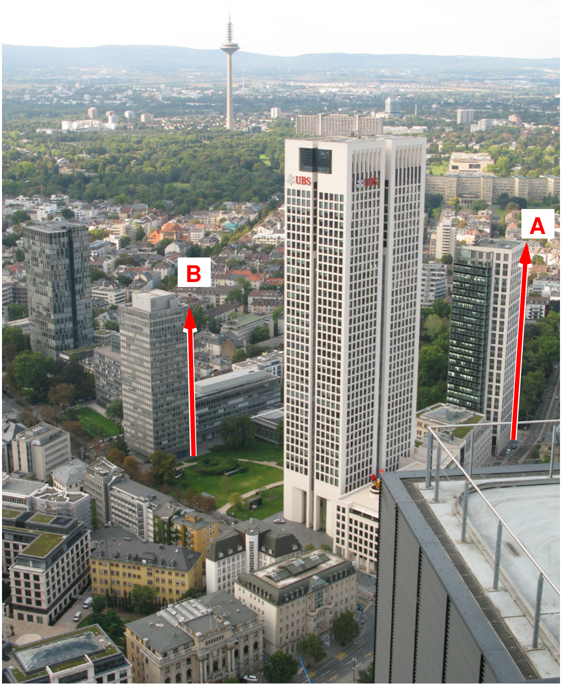
$$\mathbf{G} = \begin{bmatrix} 1 + l_1^2 & l_1 l_2 \\ l_1 l_2 & 1 + l_2^2 \end{bmatrix}$$ from light dir. $\mathbf{l} = (l_1, l_2, l_3)$

Apex: Up to 4 solutions for surface principal curvatures: convex/concave × elliptic/hyperbolic

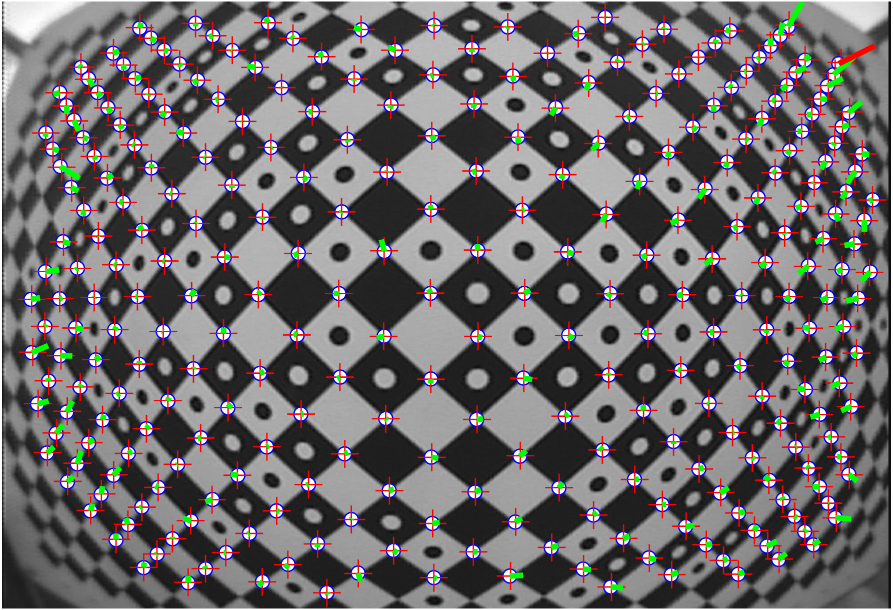
Thank You

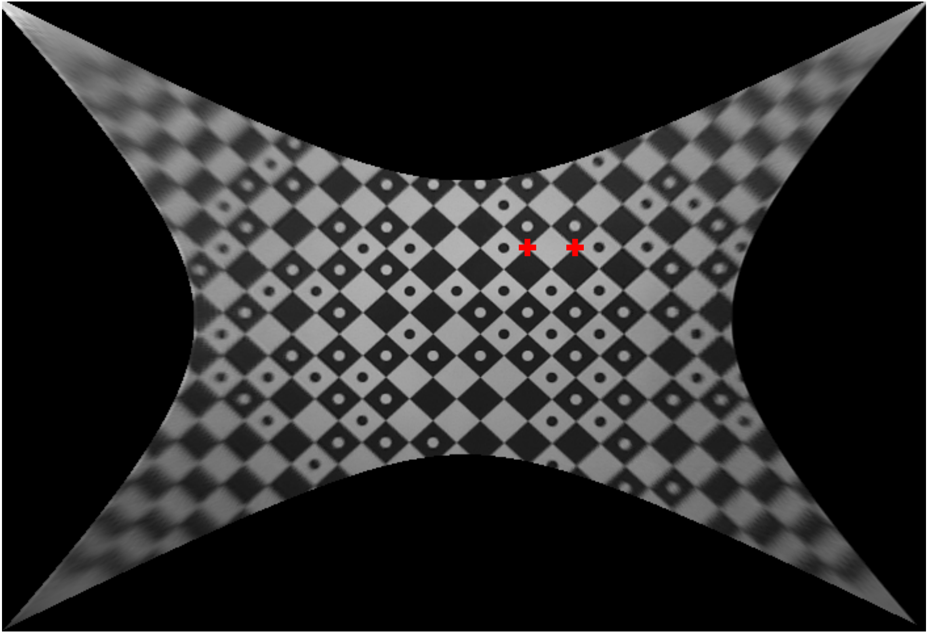

p_∞
p
supporting line
p_1
p_0

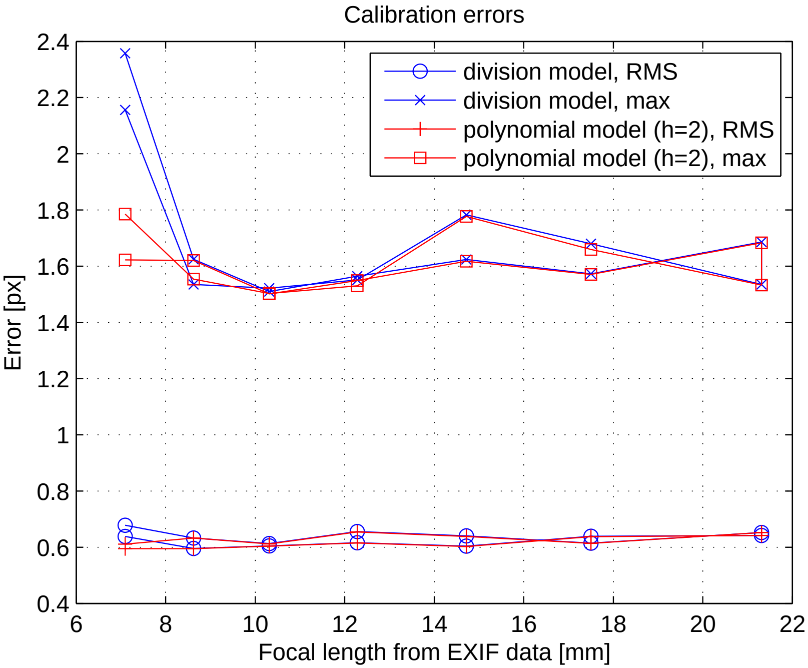

p_∞
p


UBS
A
B

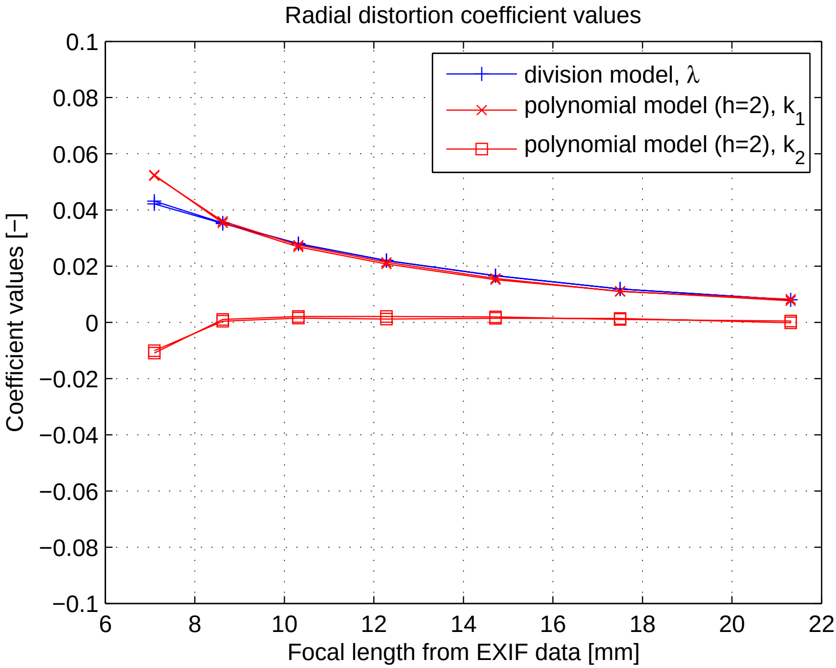
Camera 0, im. 6: Reprojection errors (16x)

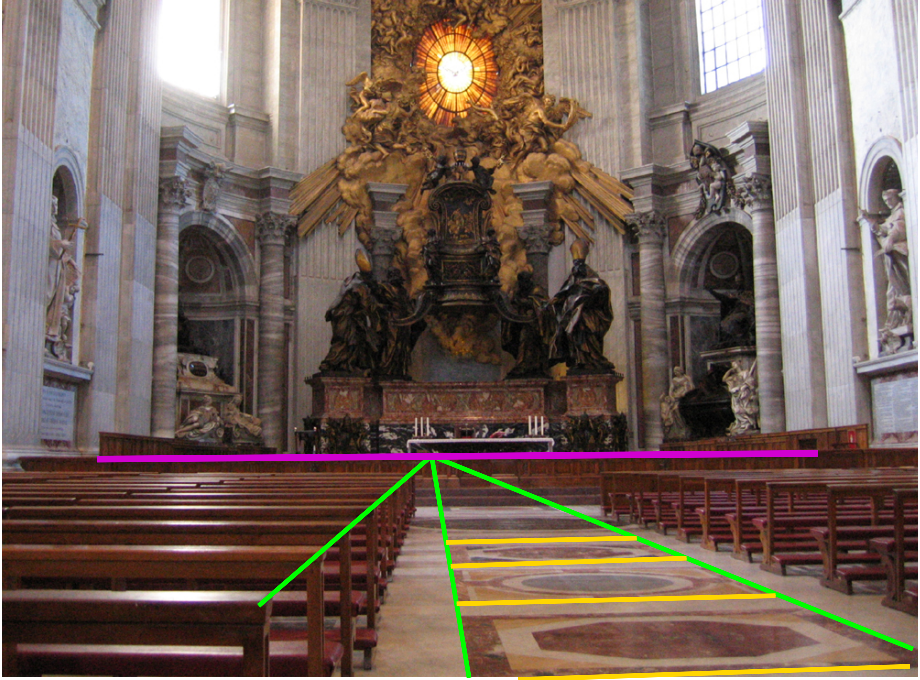






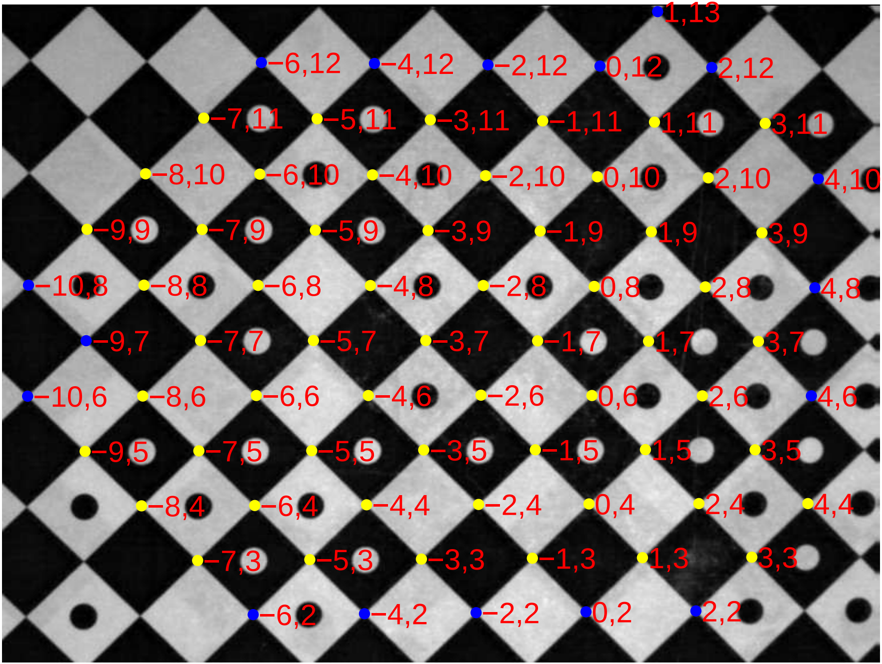
Calibration errors
division model, RMS
division model, max
polynomial model (h=2), RMS
polynomial model (h=2), max
Error [px]
Focal length from EXIF data [mm]
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
6
8
10
12
14
16
18
20
22

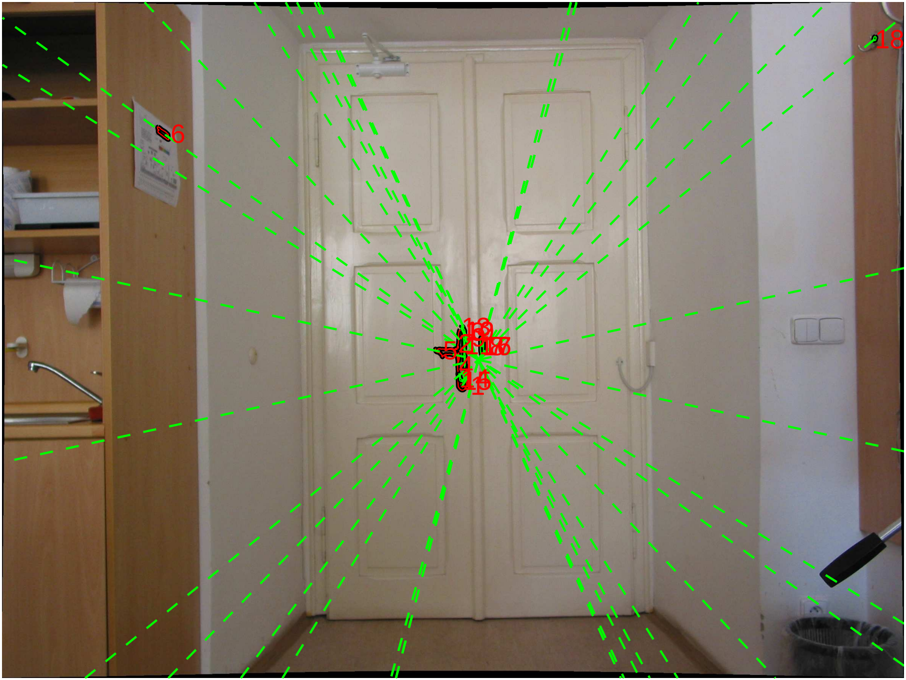

Radial distortion coefficient values
division model, λ
polynomial model (h=2), k1
polynomial model (h=2), k2
Coefficient values [−]
Focal length from EXIF data [mm]
0.1
0.08
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
−0.08
−0.1
6
8
10
12
14
16
18
20
22

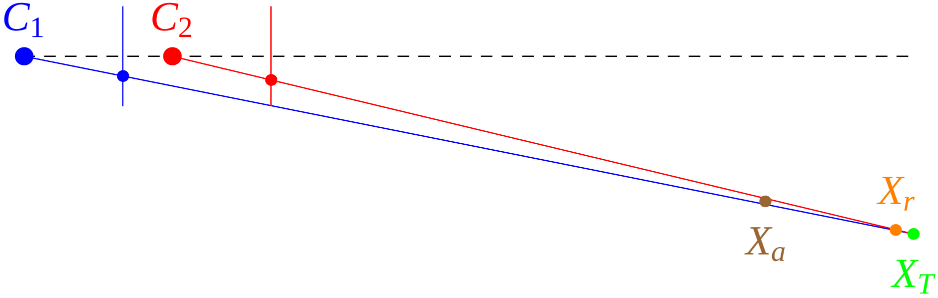


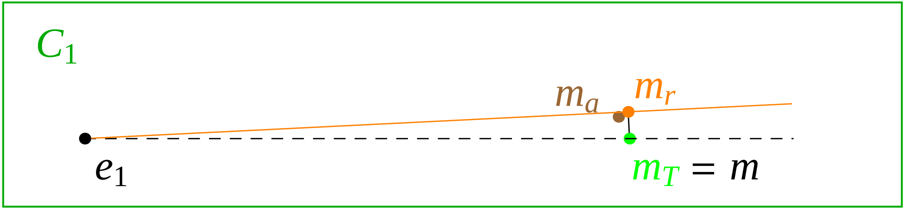

VIII · CLEMENS · XII · PONT · MAX · ANN · SAC · MDCCXXXVII · PONT ·

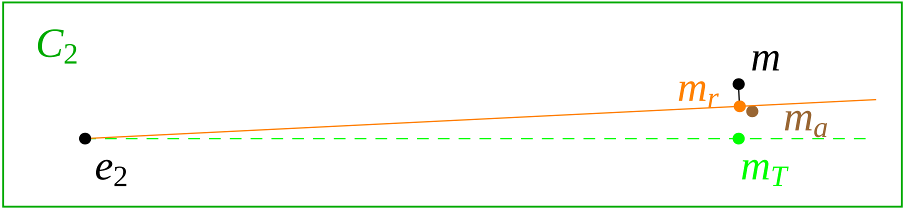

BOOKS
FILMS
TICKETS
BOOKS


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

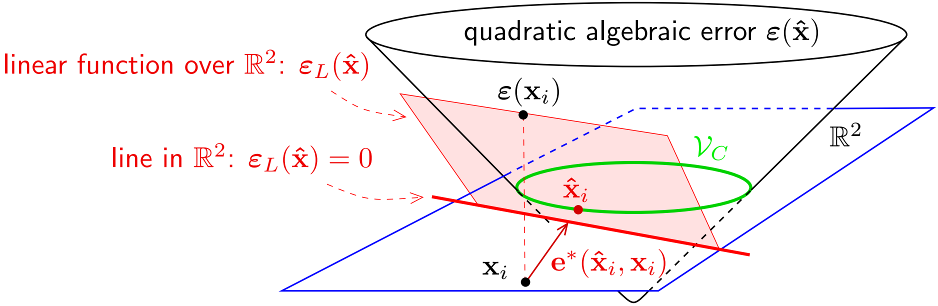

6
18

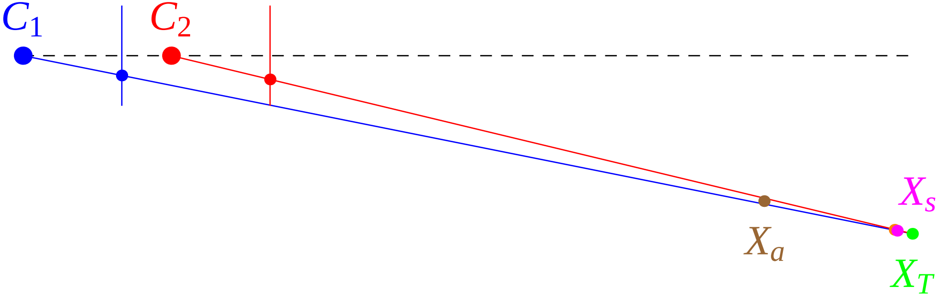

C_1
C_2
X_r
X_a
X_T

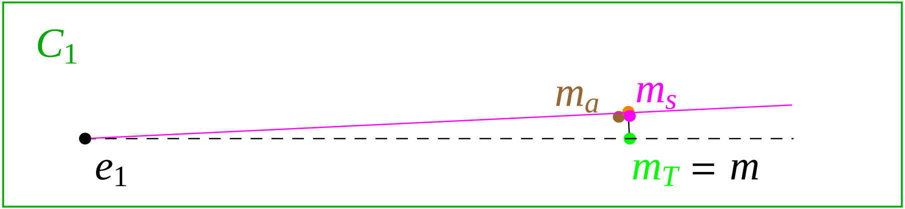

C_1
m_a
m_r
e_1
$m_T = m$

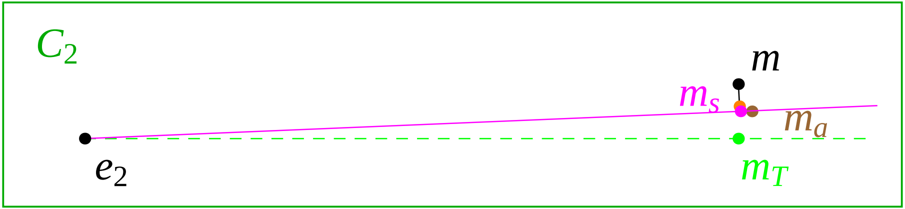


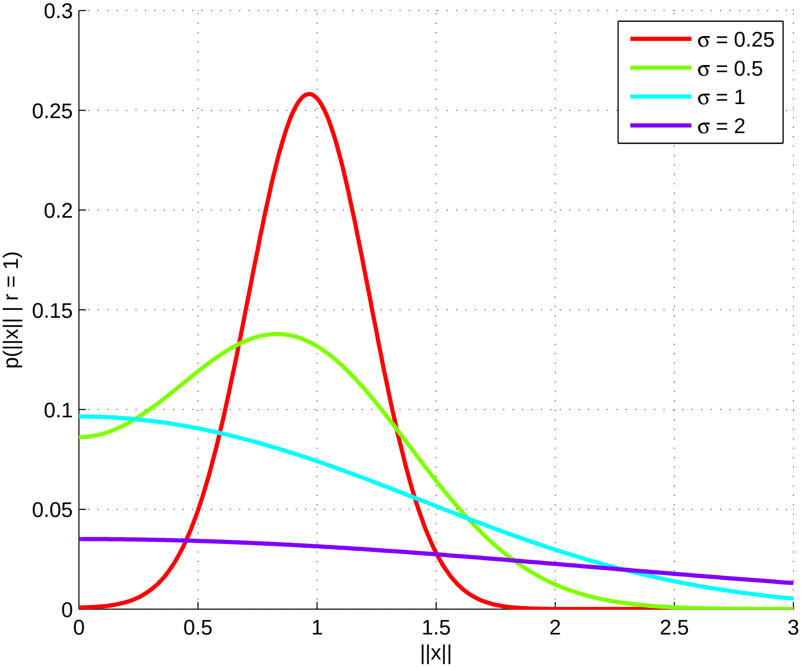
C_2
m
m_r
m_a
e_2
m_T


quadratic algebraic error $\varepsilon(\hat{\mathbf{x}})$
linear function over $\mathbb{R}^2$: $\varepsilon_L(\hat{\mathbf{x}})$
$\varepsilon(\mathbf{x}_i)$
$\mathbb{R}^2$
$\mathcal{V}_C$
line in $\mathbb{R}^2$: $\varepsilon_L(\hat{\mathbf{x}}) = 0$
$\hat{\mathbf{x}}_i$
$\mathbf{x}_i$
$\mathbf{e}^*(\hat{\mathbf{x}}_i, \mathbf{x}_i)$

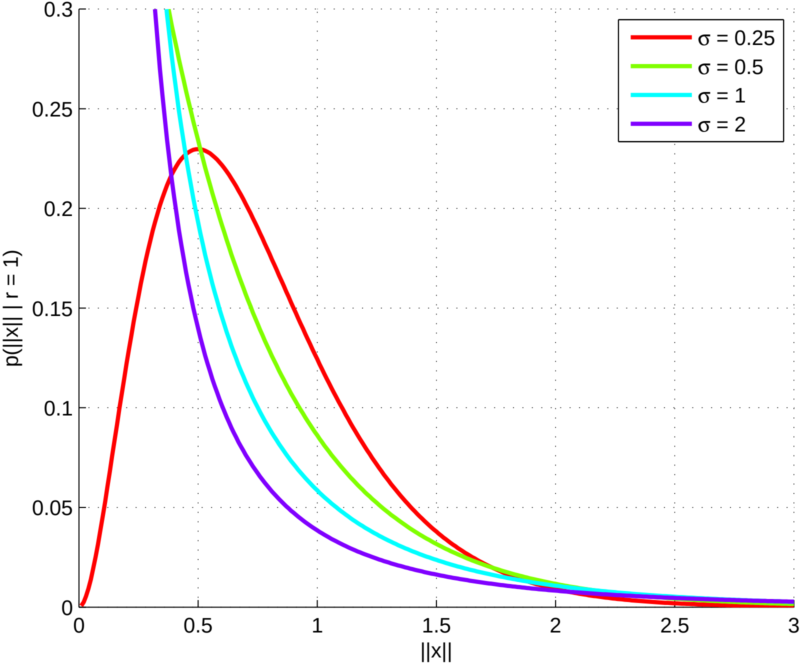

C_1
C_2
X_s
X_a
X_T


C_1
m_a
m_s
e_1
$m_T = m$

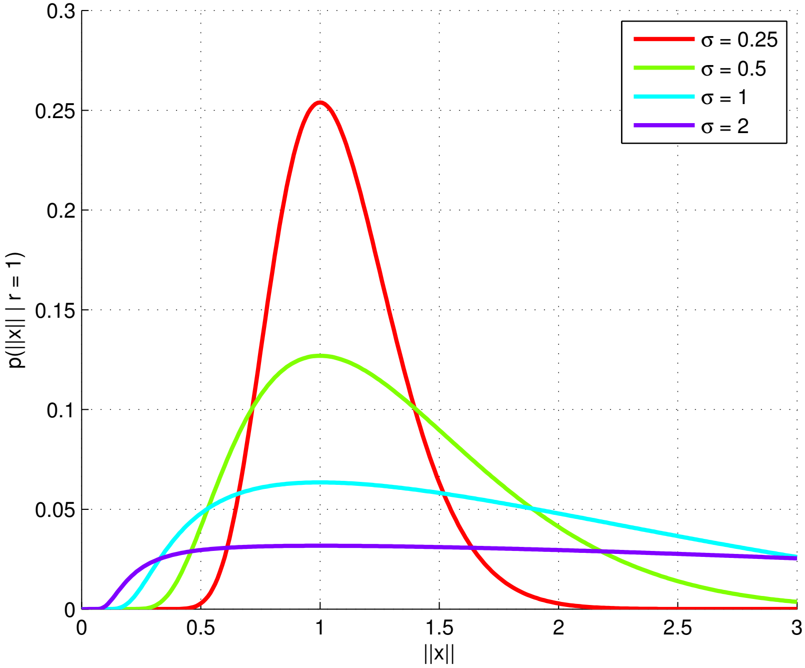

C_2
m
m_s
m_a
e_2
m_T



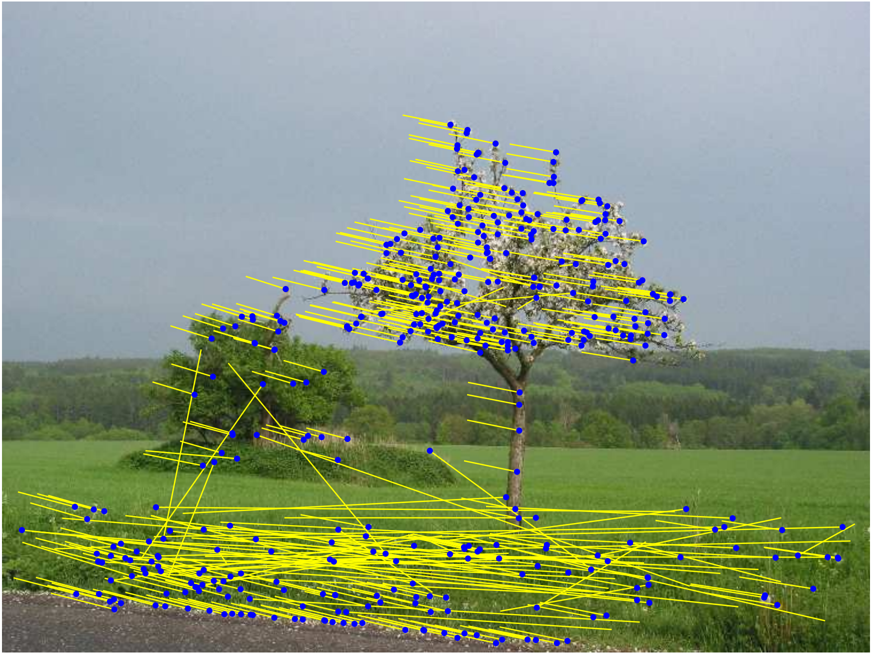
σ = 0.25
σ = 0.5
σ = 1
σ = 2
p(||x|| | r = 1)
||x||
0
0.05
0.1
0.15
0.2
0.25
0.3
0
0.5
1
1.5
2
2.5
3

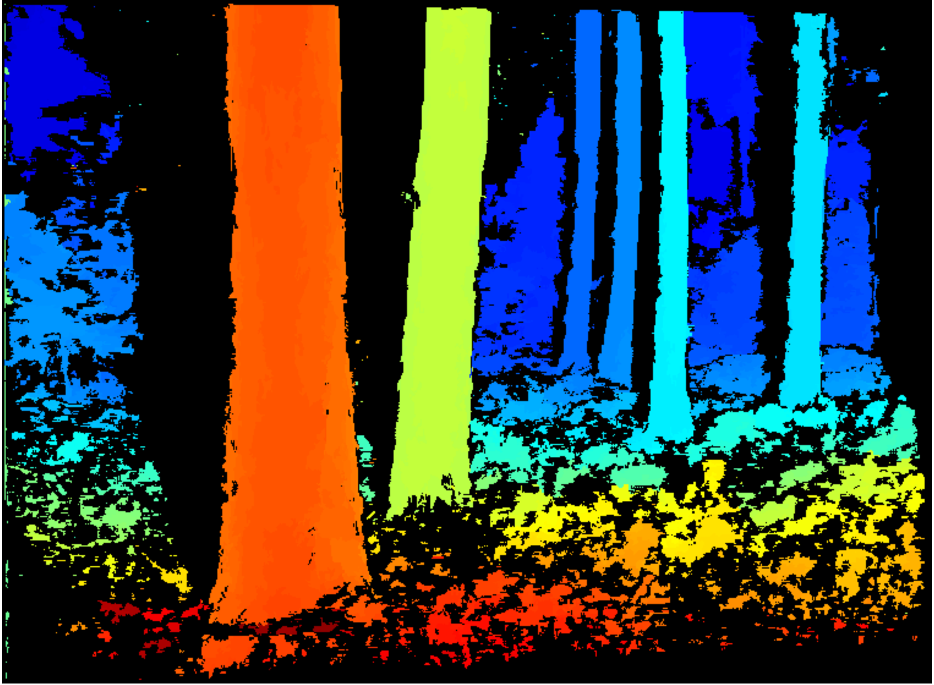


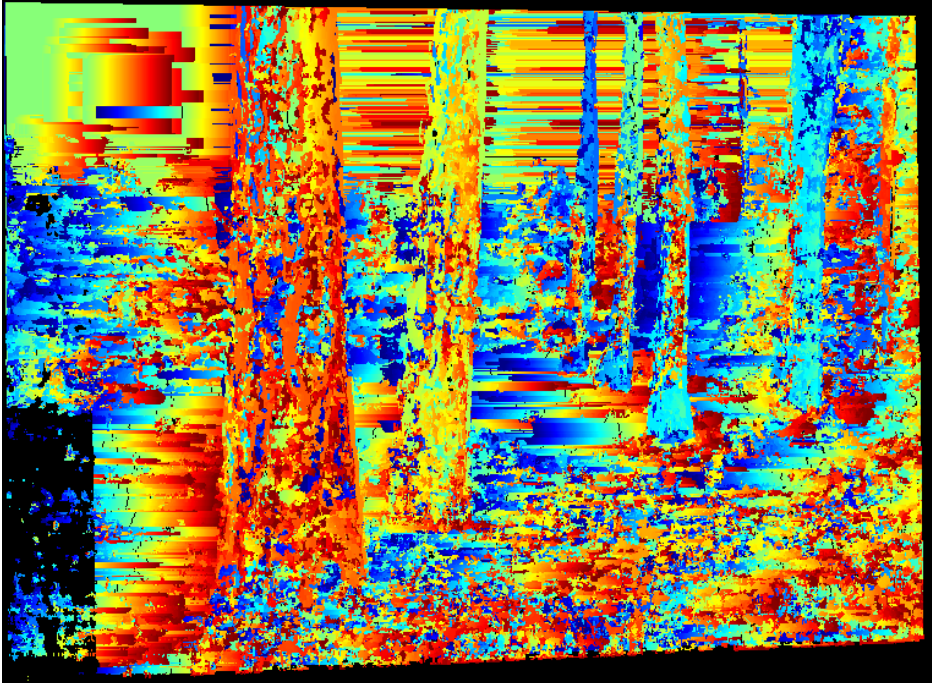
σ = 0.25
σ = 0.5
σ = 1
σ = 2
p(||x|| | r = 1)
||x||
0
0.05
0.1
0.15
0.2
0.25
0.3
0
0.5
1
1.5
2
2.5
3

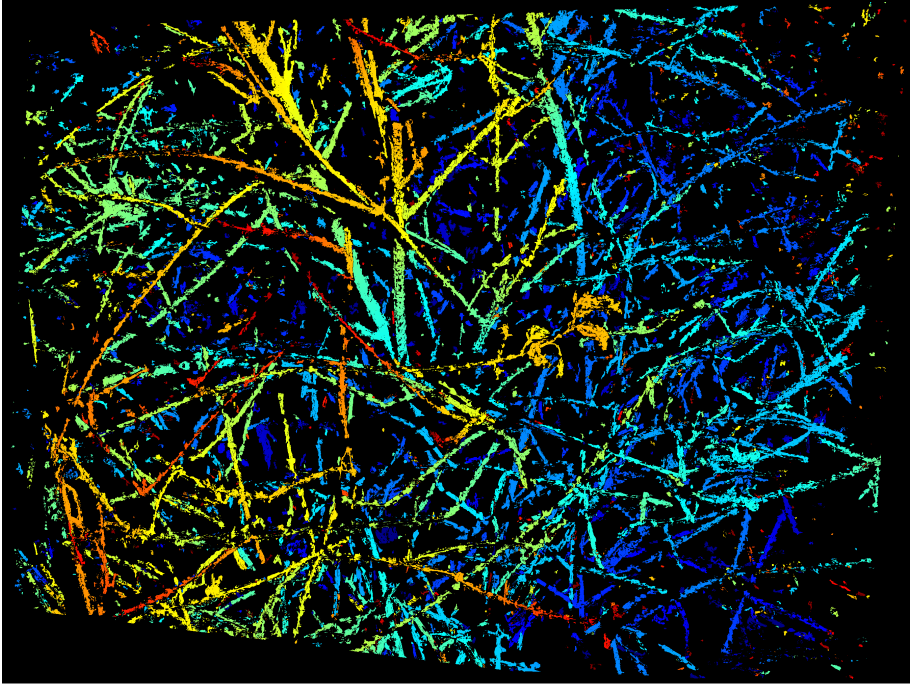


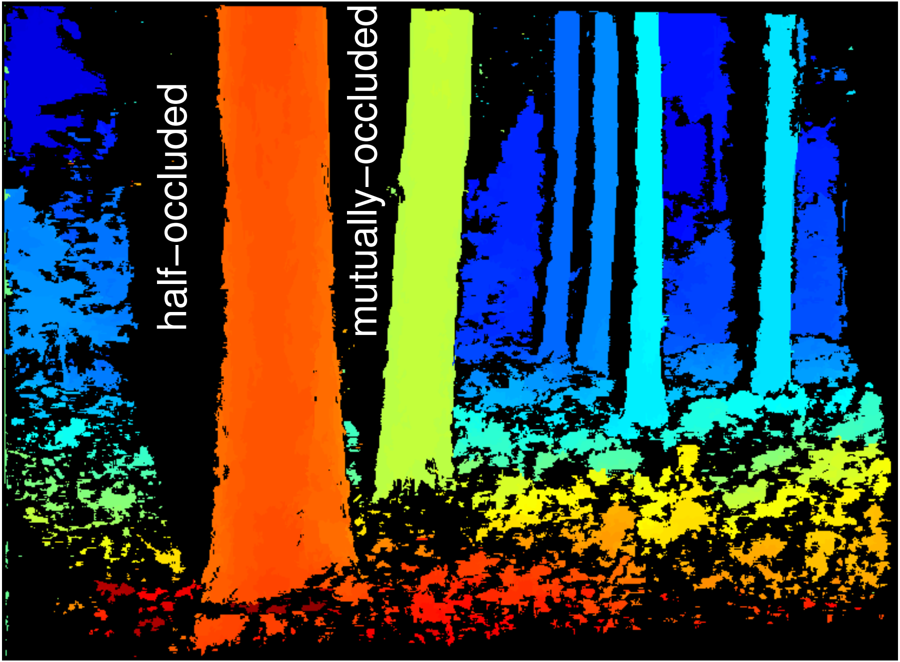
σ = 0.25
σ = 0.5
σ = 1
σ = 2
p(||x|| | r = 1)
||x||
0
0.05
0.1
0.15
0.2
0.25
0.3
0
0.5
1
1.5
2
2.5
3

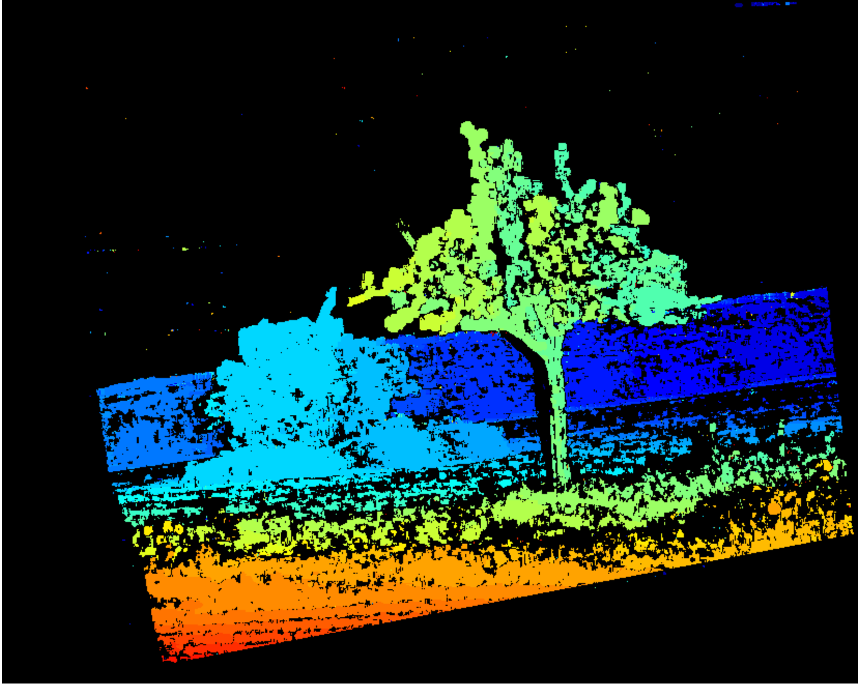


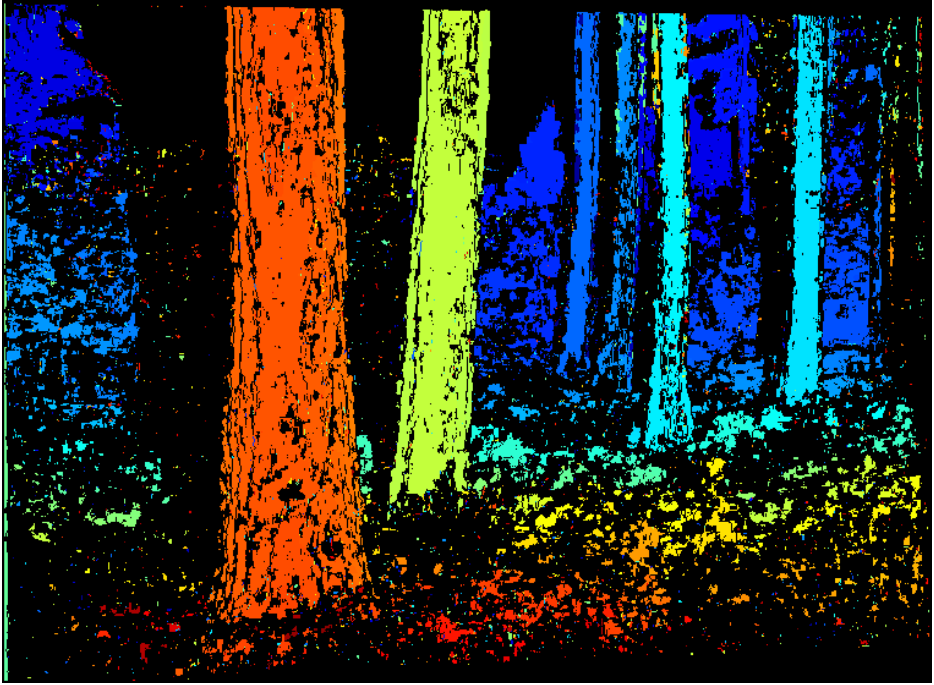


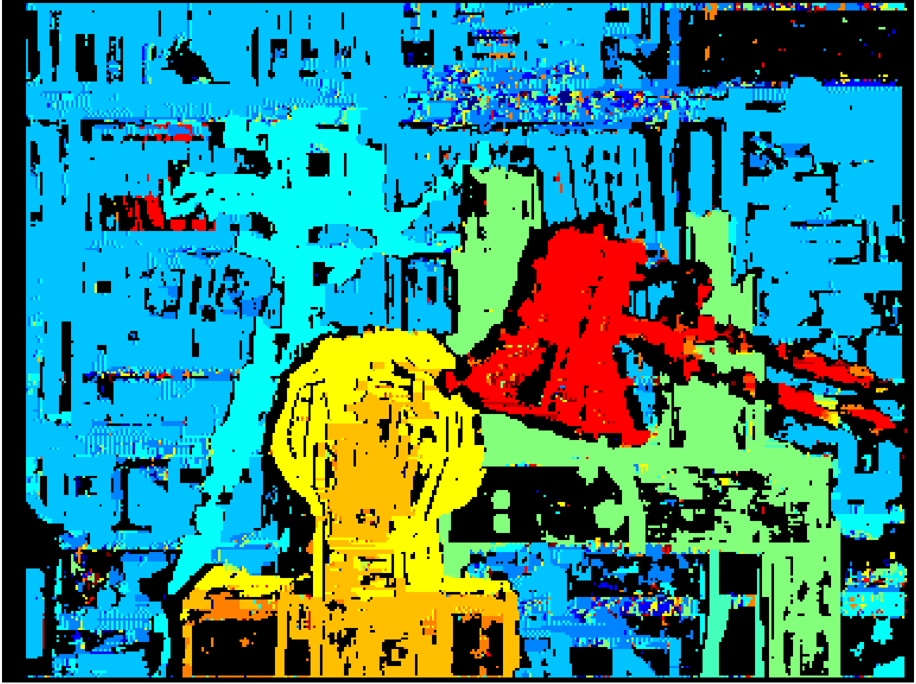


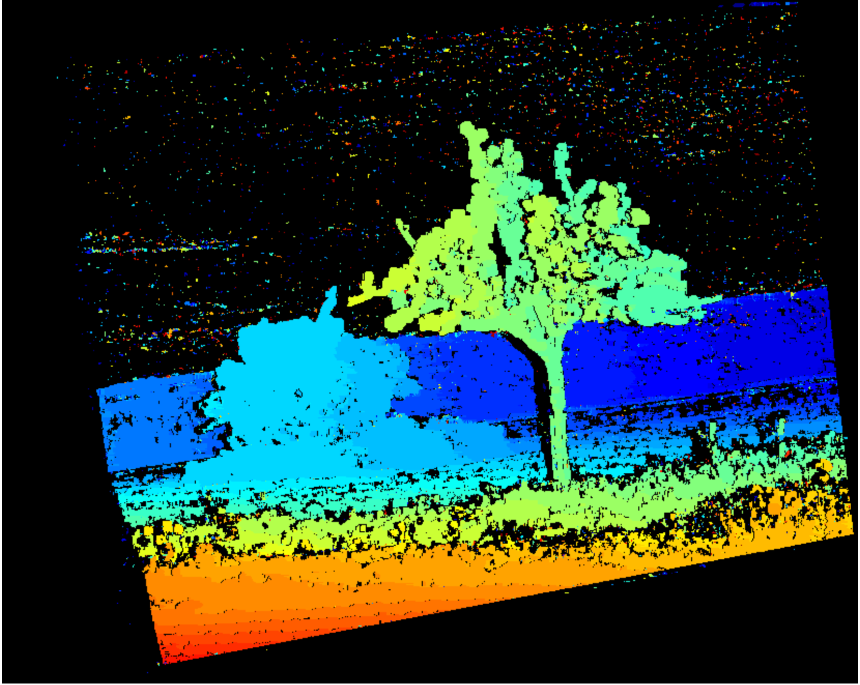


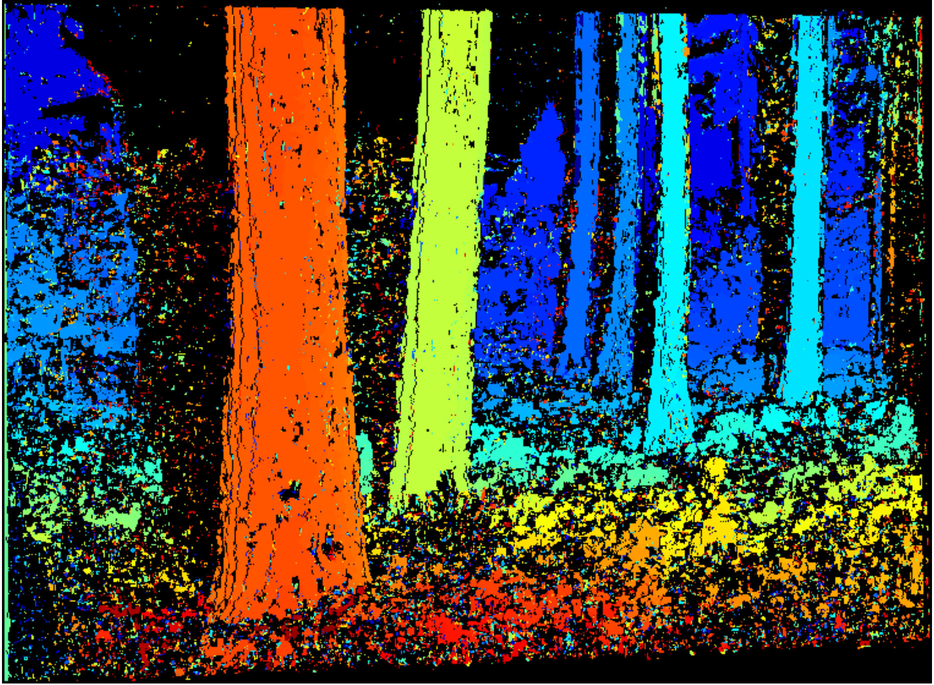

half–occluded
mutually–occluded

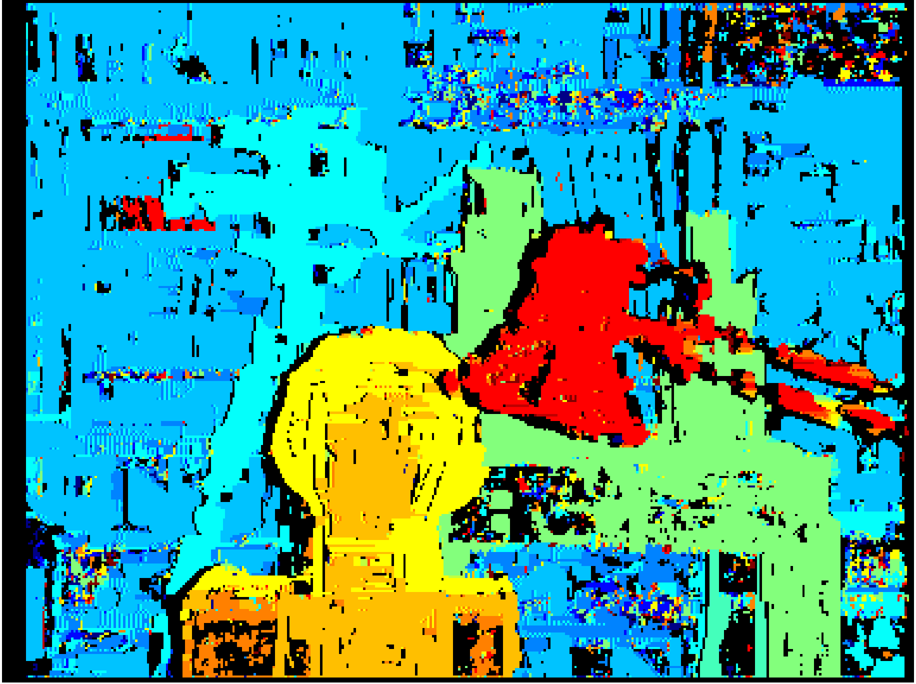


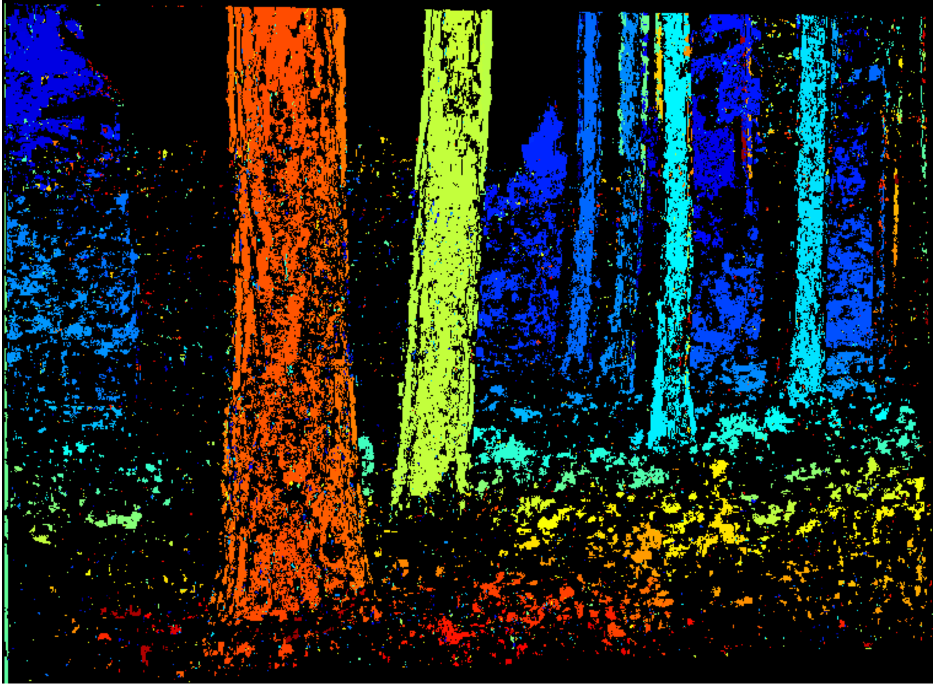


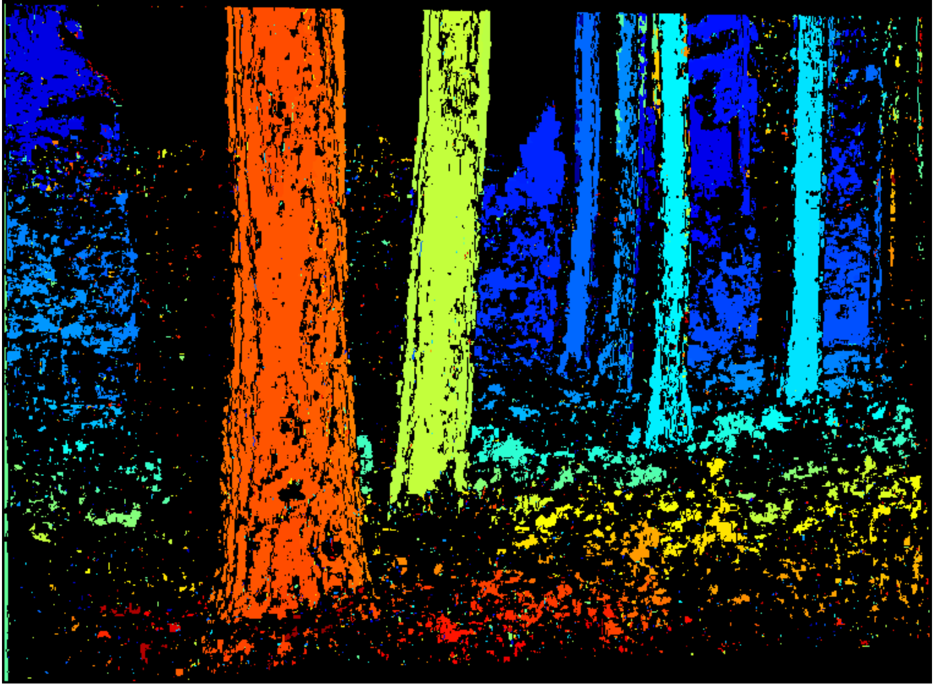


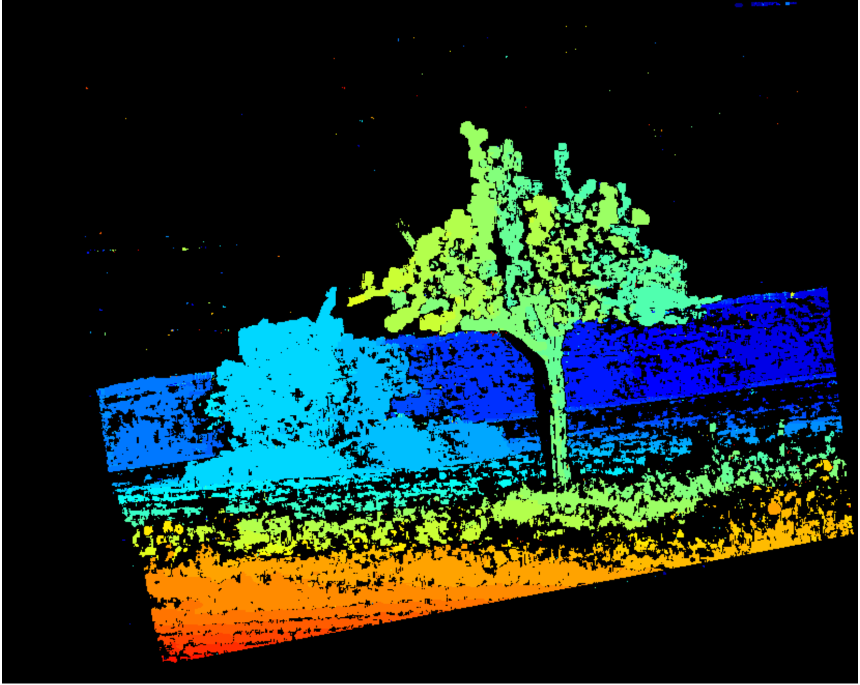


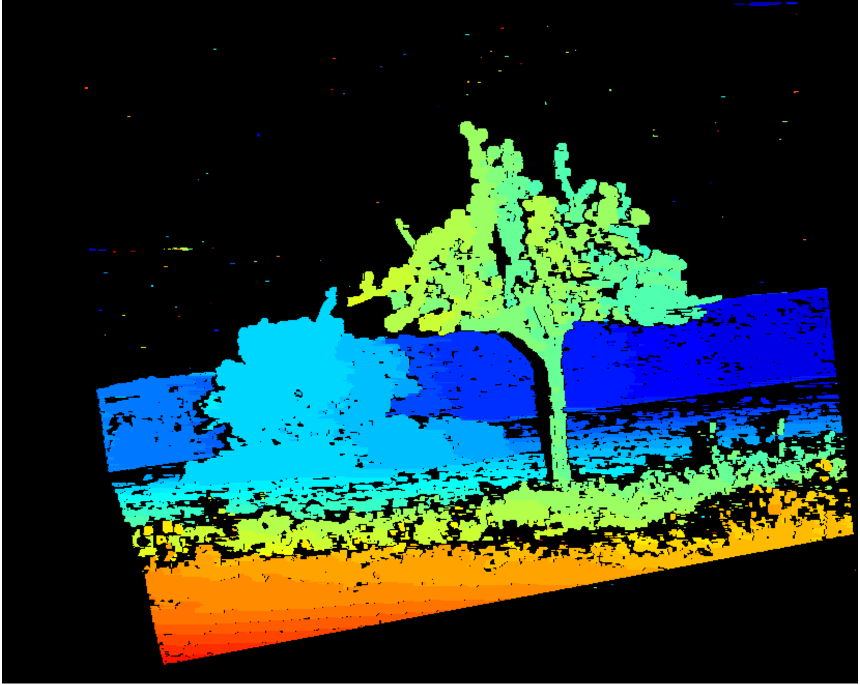


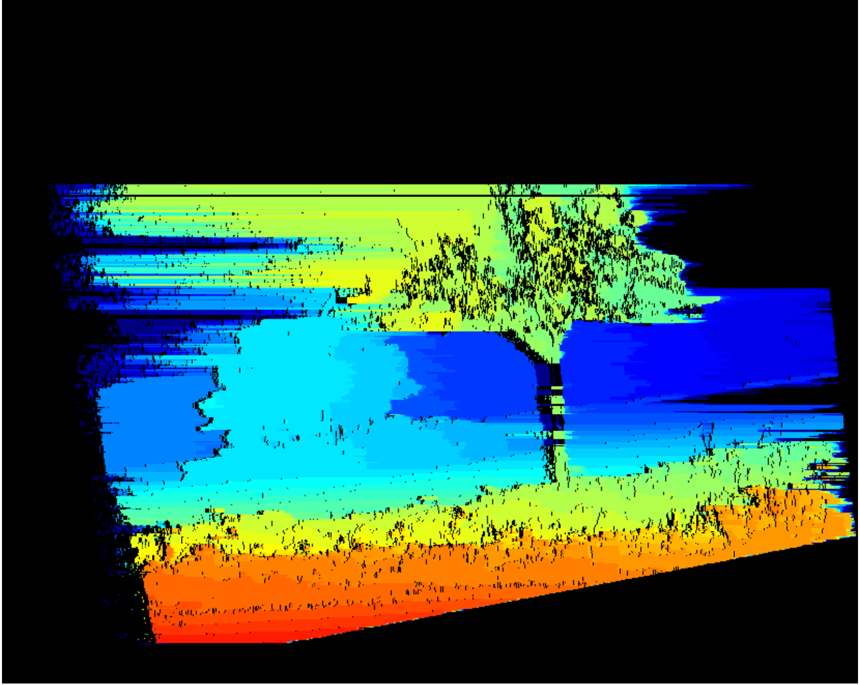


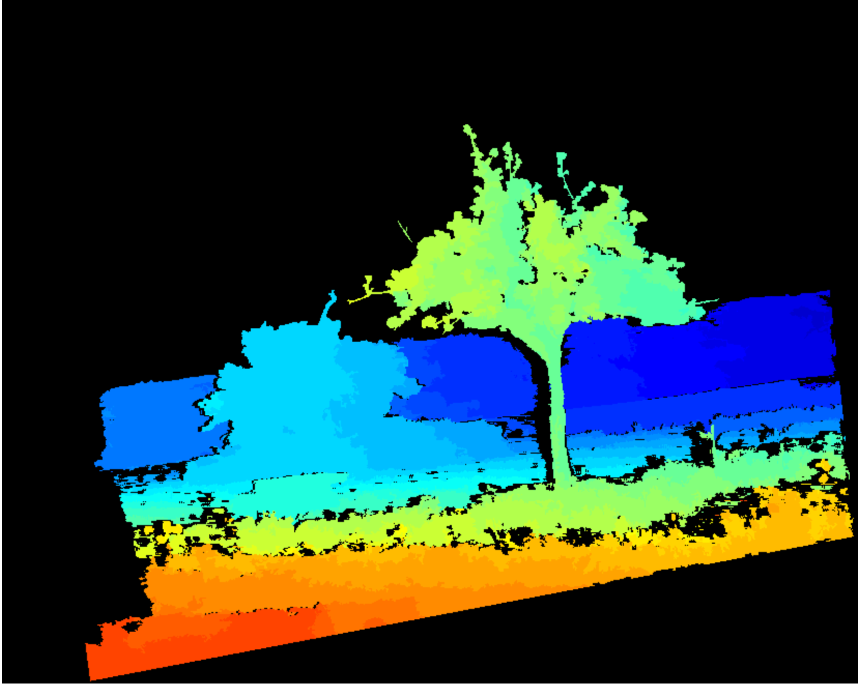


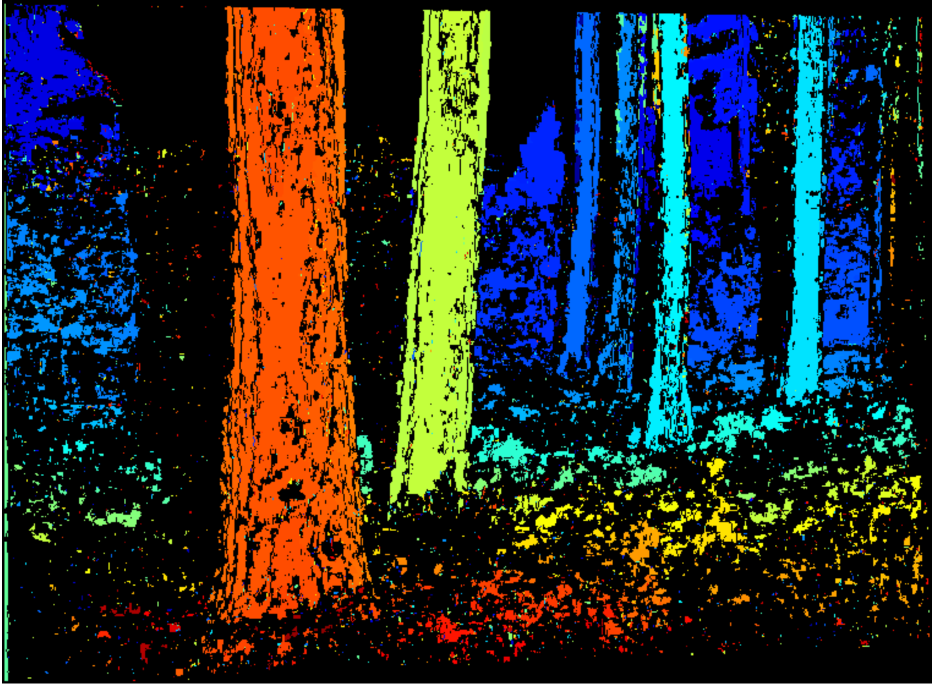


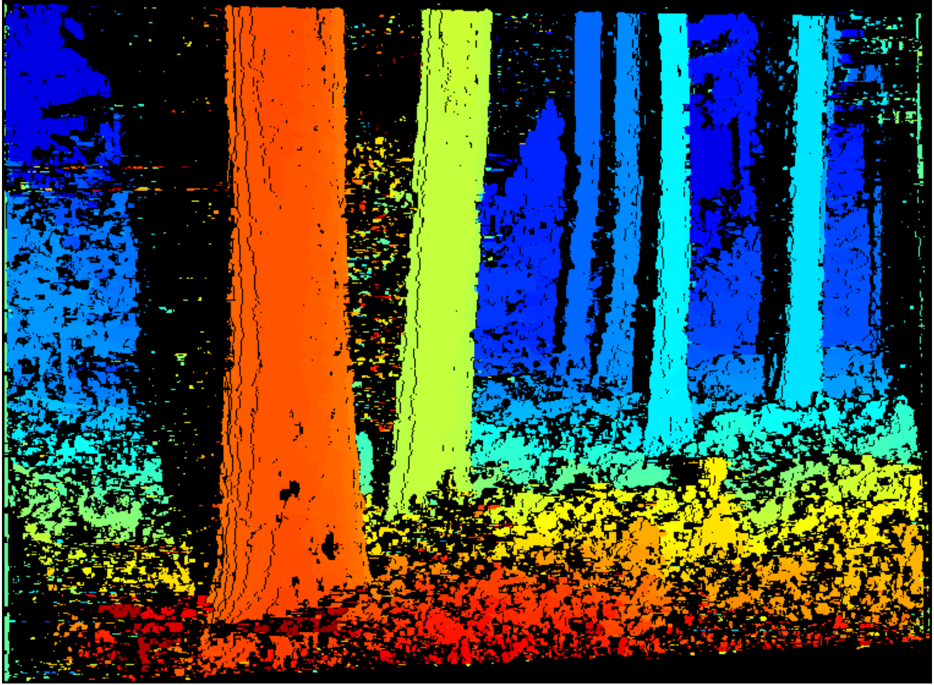


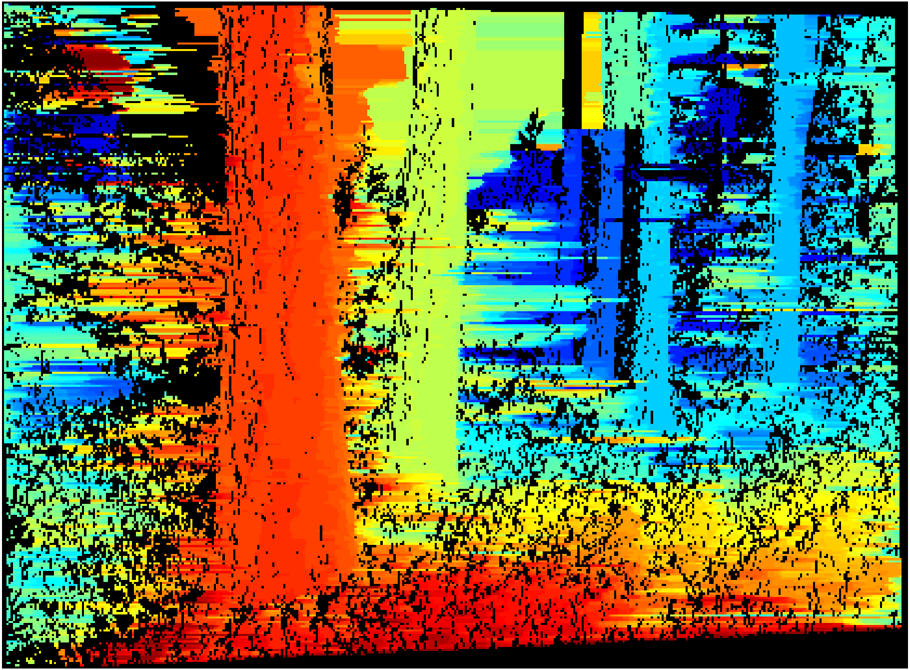


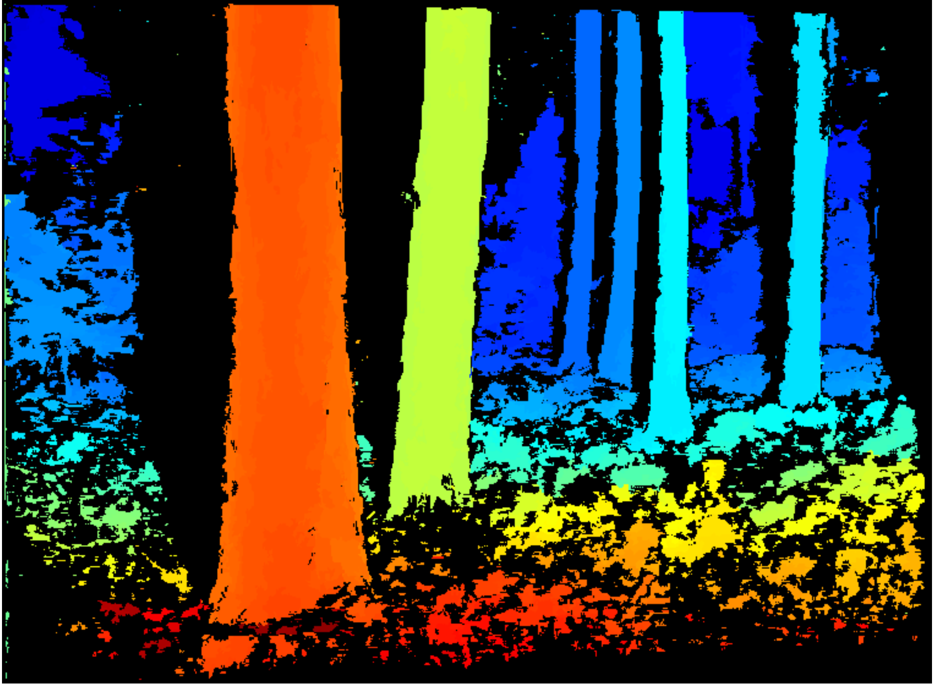




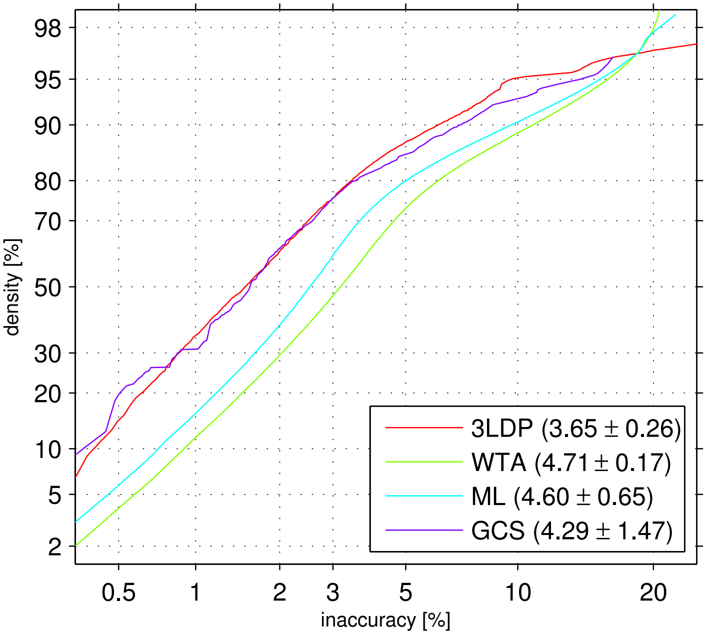


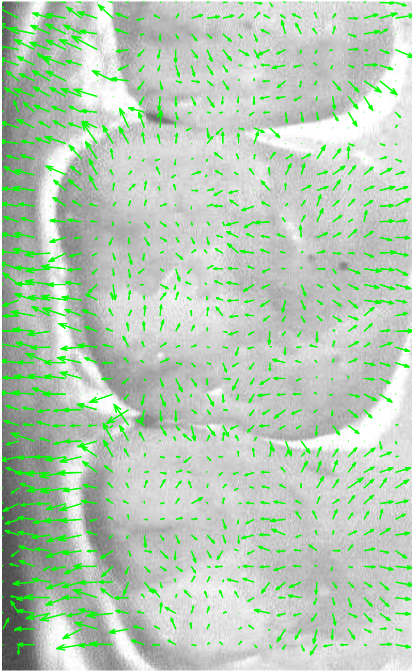


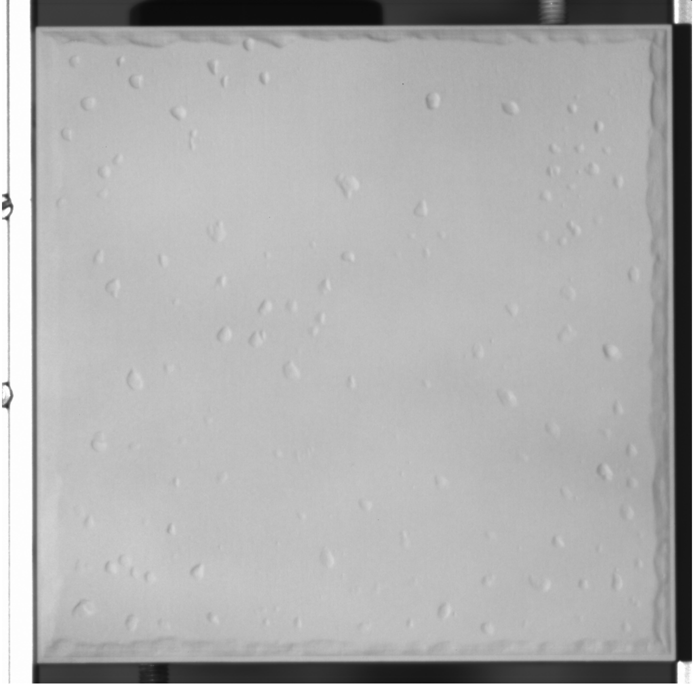


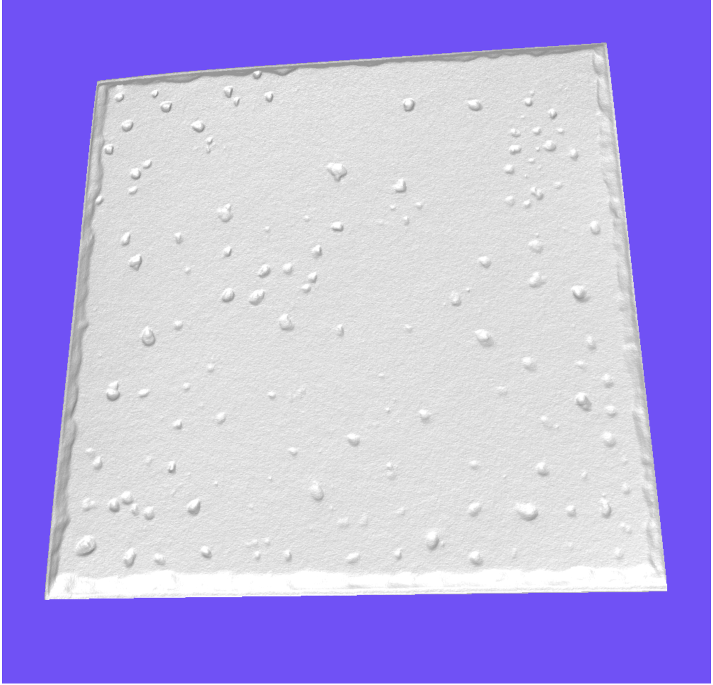


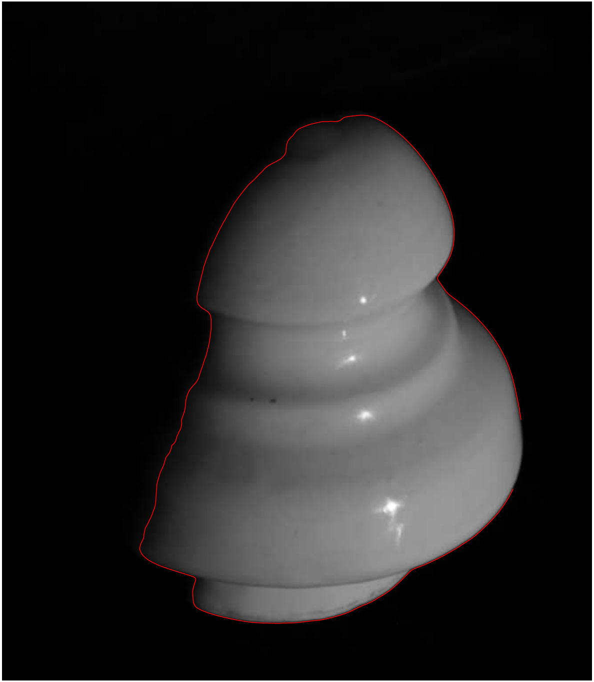
ROC curves and their average error rate bounds

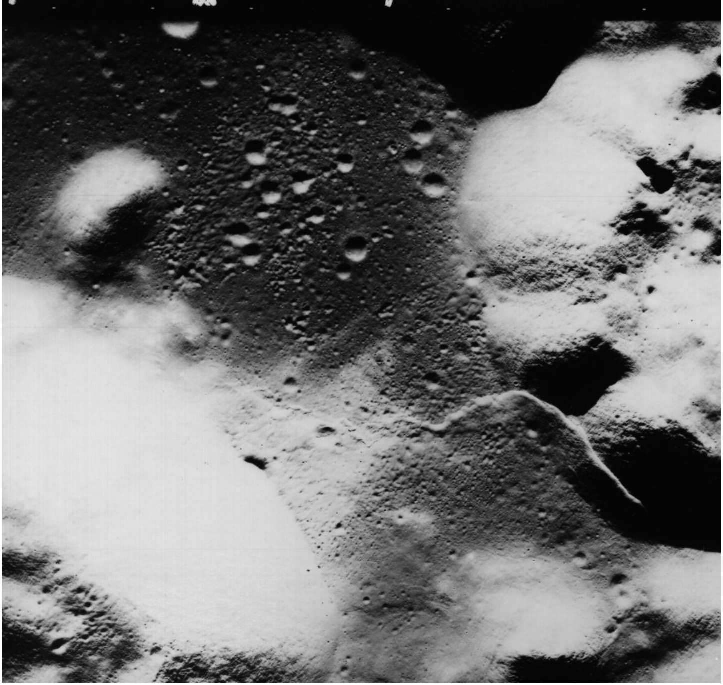












OPPA European Social Fund
Prague & EU: We invest in your future.