3D Computer Vision

Radim Šára Martin Matoušek

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

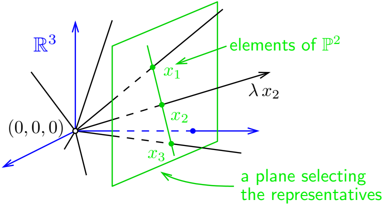
https://cw.fel.cvut.cz/wiki/courses/tdv/start

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

rev. October 6, 2020

Open Informatics Master's Course

▶Homography in $\mathbb{P}^2$



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

including 'points at infinity'

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

Homography in $\mathbb{P}^2$: Non-singular linear mapping in $\mathbb{P}^2$

an analogic definition for $\mathbb{P}^3$

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

Defining properties

- collinear points are mapped to collinear points

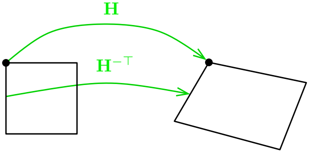
  lines of points are mapped to lines of points
- concurrent lines are mapped to concurrent lines

  concurrent = intersecting at a point
- and point-line incidence is preserved

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

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

▶Mapping 2D Points and Lines by Homography



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

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

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

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

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

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

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

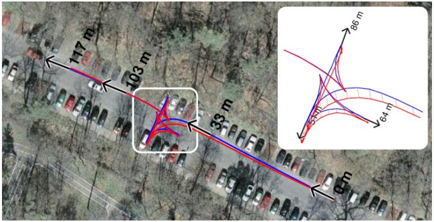
Some Homographic Tasters

Rectification of camera rotation: →59 (geometry), →127 (homography estimation)

$$\mathbf{H} \simeq \mathbf{K}\mathbf{R}^\top\mathbf{K}^{-1}$$

maps from image plane to facade plane

Homographic Mouse for Visual Odometry: [Mallis 2007]



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

$$\mathbf{H} \simeq \mathbf{K}\left(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^\top}{d}\right)\mathbf{K}^{-1}$$

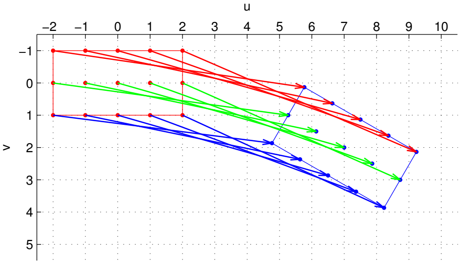
[H&Z, p. 327]

▶Homography Subgroups: Euclidean Mapping (aka Rigid Motion)

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

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

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



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

EM = The most general homography preserving

1. **areas**: $\det \mathbf{H} = 1 \;\Rightarrow\;$ unit Jacobian
2. **lengths**: Let $\underline{\mathbf{x}}'_i = \mathbf{H}\underline{\mathbf{x}}_i$ (check we can use $=$ instead of $\simeq$). Let $(x_i)_3 = 1$, Then

$$\|\underline{\mathbf{x}}'_2 - \underline{\mathbf{x}}'_1\| = \|\mathbf{H}\underline{\mathbf{x}}_2 - \mathbf{H}\underline{\mathbf{x}}_1\| = \|\mathbf{H}(\underline{\mathbf{x}}_2 - \underline{\mathbf{x}}_1)\| = \cdots = \|\underline{\mathbf{x}}_2 - \underline{\mathbf{x}}_1\|$$

3. **angles** check the dot-product of normalized differences from a point $(\mathbf{x}-\mathbf{z})^\top(\mathbf{y}-\mathbf{z})$ (Cartesian(!))

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

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

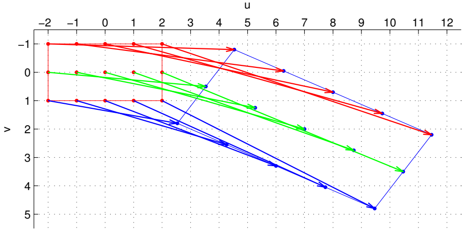
$\mathbf{e}_2$, $\mathbf{e}_3$ – circular points, i – imaginary unit

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

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

▶Homography Subgroups: Affine Mapping

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



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

AM = The most general homography preserving

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

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

does not preserve

- lengths
- angles
- areas
- circular points

Euclidean mappings preserve all properties affine mappings preserve, of course

▶Homography Subgroups: General Homography

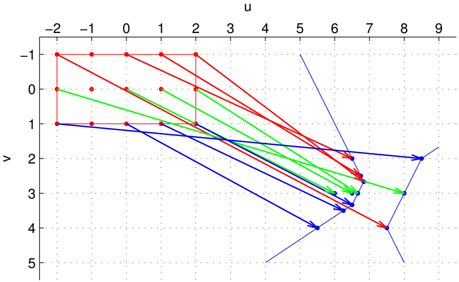
$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

preserves only

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

does not preserve

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

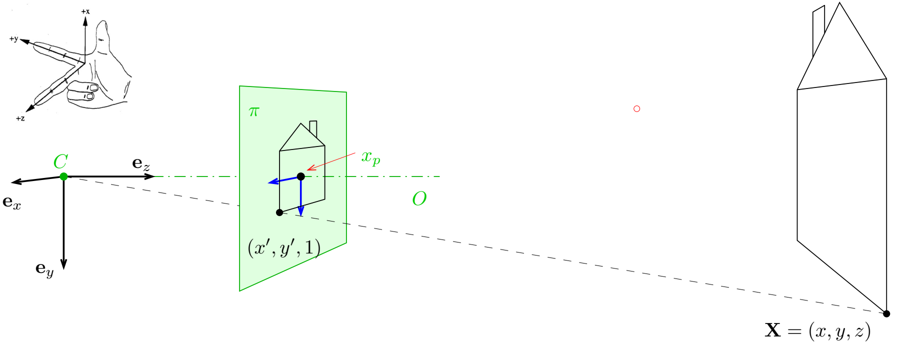


$$\mathbf{H} = \begin{bmatrix} 7 & -0.5 & 6 \\ 3 & 1 & 3 \\ 1 & 0 & 1 \end{bmatrix}$$

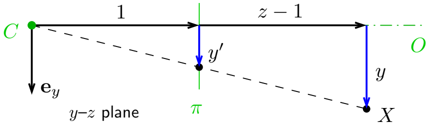
line $\underline{\mathbf{n}} = (1, 0, 1)$ is mapped to $\underline{\mathbf{n}}_\infty$: $\mathbf{H}^{-\top}\underline{\mathbf{n}} \simeq \underline{\mathbf{n}}_\infty$

(where in the picture is the line n?)

►Canonical Perspective Camera (Pinhole Camera, Camera Obscura)



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



projected point in the natural image coordinate system:

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

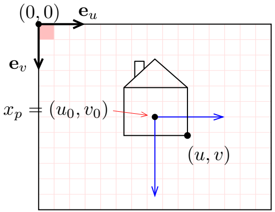
▶Natural and Canonical Image Coordinate Systems

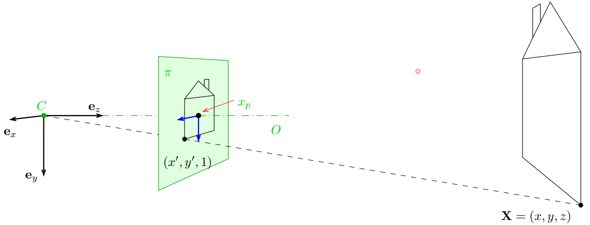
projected point **in canonical camera** $(z \neq 0)$

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

projected point **in scanned image**

scale by f and translate by $(-u_0, -v_0)$





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

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

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

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

►Computing with Perspective Camera Projection Matrix

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

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

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

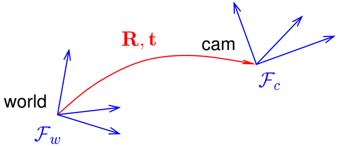
(u_0, v_0) – principal point in pixels

Perspective Camera:

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

▶Changing The Outer (World) Reference Frame

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



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

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

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

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

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

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

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

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

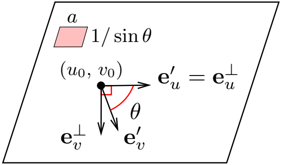
$\mathbf{r}_3^\top$ – optical axis in the world reference frame $\mathcal{F}_w$ — third row of $\mathbf{R}$: $\mathbf{r}_3 = \mathbf{R}^{-1}[0, 0, 1]^\top$

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

▶Changing the Inner (Image) Reference Frame

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

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



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

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

⊛ H1; 2pt: Verify this $\mathbf{K}$; deadline LD+2 wk

Hints:

1. image projects to orthogonal system $F^{\perp}$, then it maps by skew to F', then by scale $a\,f$, f to F'', then by translation by u_0, v_0 to F'''
2. Skew: Express point $\mathbf{x}$ as

$$\mathbf{x} = u'\mathbf{e}_{u'} + v'\mathbf{e}_{v'} = u^{\perp}\mathbf{e}_u^{\perp} + v^{\perp}\mathbf{e}_v^{\perp}$$

   $\mathbf{e}_{:}$ are unit basis vectors

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

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

4. figure drawn 'after the transformation' but in $F^{\perp}$

►Summary: Projection Matrix of a General Finite Perspective Camera

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

a recipe for filling $\mathbf{P}$

general finite perspective camera has 11 parameters:

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

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

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

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

▶Projection Matrix Decomposition

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

$\mathbf{Q} \in \mathbb{R}^{3,3}$ full rank (if finite perspective camera; see [H&Z, Sec. 6.3] for cameras at infinity)
$\mathbf{K} \in \mathbb{R}^{3,3}$ upper triangular with positive diagonal elements
$\mathbf{R} \in \mathbb{R}^{3,3}$ rotation: $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$ and $\det \mathbf{R} = +1$

1. $\begin{bmatrix}\mathbf{Q} & \mathbf{q}\end{bmatrix} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix} = \begin{bmatrix}\mathbf{KR} & \mathbf{Kt}\end{bmatrix}$ also →35
2. RQ decomposition of $\mathbf{Q} = \mathbf{KR}$ using three Givens rotations [H&Z, p. 579]

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

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

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

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

- RQ decomposition nonuniqueness: $\mathbf{KR} = \mathbf{KT}^{-1}\mathbf{TR}$, where $\mathbf{T} = \mathrm{diag}(-1,-1,1)$ is also a rotation, we must correct the result so that the diagonal elements of $\mathbf{K}$ are all positive
  'thin' RQ decomposition
- care must be taken to avoid overflow, see [Golub & van Loan 2013, sec. 5.2]

RQ Decomposition Step

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

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

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

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

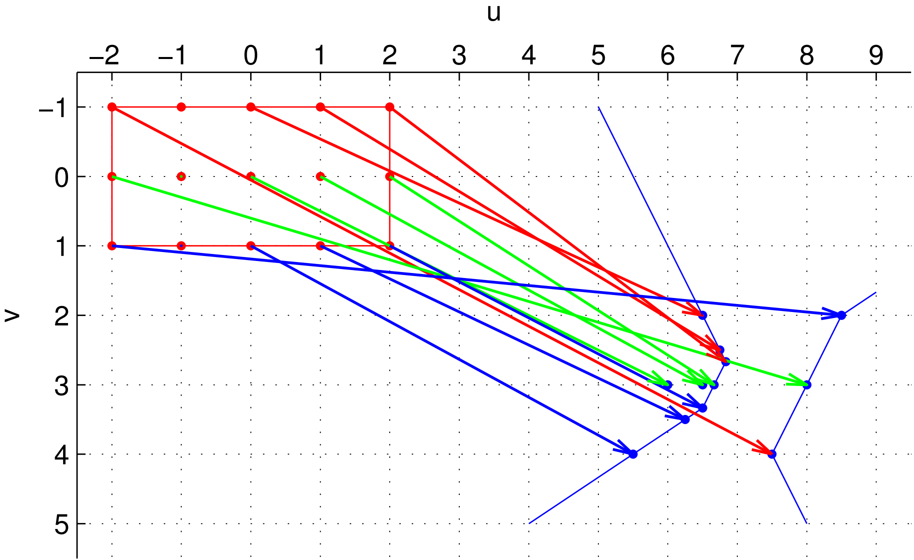
```
s1 = Solve [{Q1[[3]][[2]] == 0, c^2 + s^2 == 1}, {c, s}][[2]]
```

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

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

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

Thank You


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