2D projective transformations (homographies)

Christiano Gava

christiano.gava@dfki.de

Gabriele Bleser

gabriele.bleser@dfki.de

Introduction

- Previous lectures:
  - From world to image
  - Homogeneous coordinates
  - Intrinsic and extrinsic camera parameters
  - Camera calibration, P-matrix estimation
- Today: (slides partly based on Marc Pollefeys)
  - Homographies
  - Homography estimation
  - Applications: panorama stitching, rotating camera, pose estimation from planar surfaces

Reminder: homogeneous coordinates

- Allow to manipulate n-dim vectors in a n+1-dim space
- For n=2: $R^2 \rightarrow P^2$

3 entries, but only 2 degrees of freedom (DOF)

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Converting *to* homogeneous image coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

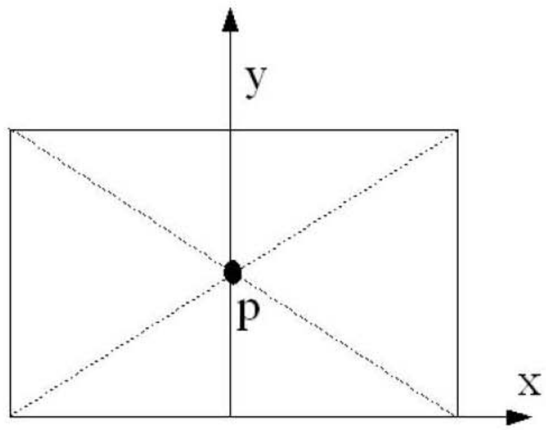
Converting *from* homogeneous image coordinates

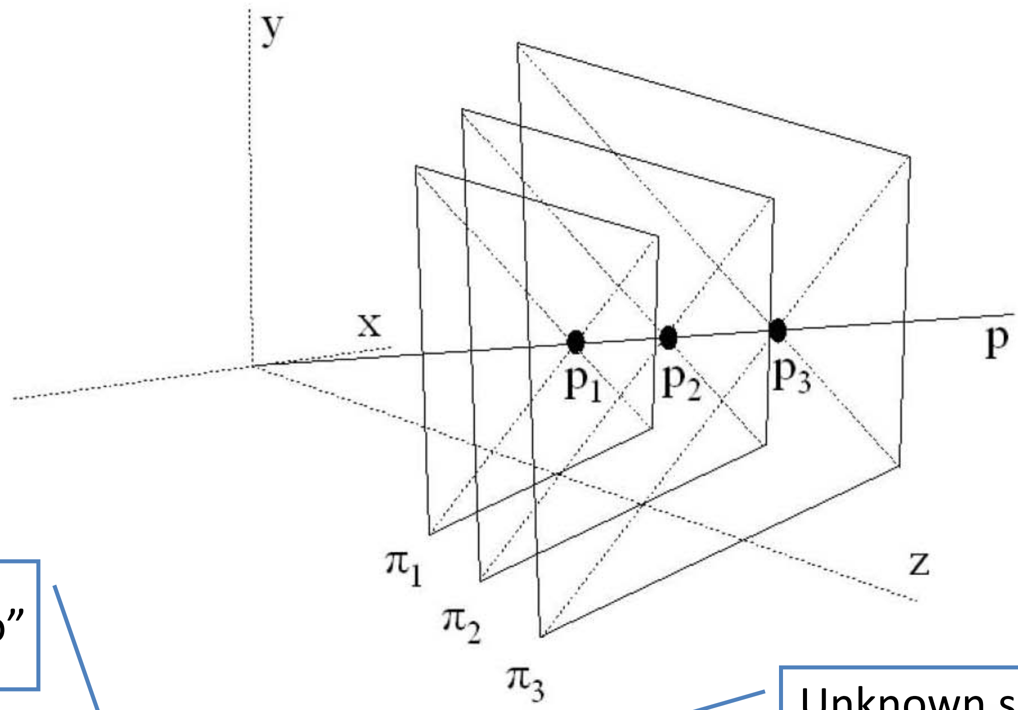
- Infinite points are represented with w=0

$$\left(\frac{x}{0}, \frac{y}{0}, 0 \right) \Rightarrow (\infty, \infty, 0)$$

Where is this useful in computer vision?

Reminder: homogeneous coordinates





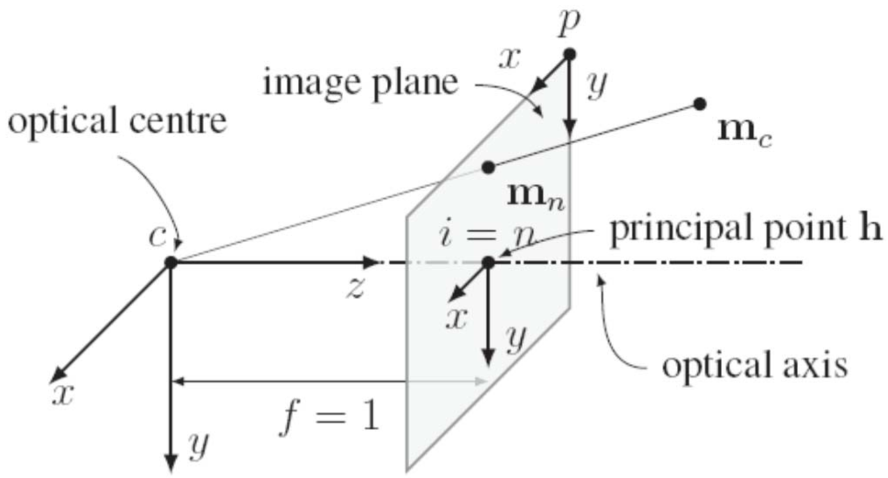
"proportional to"

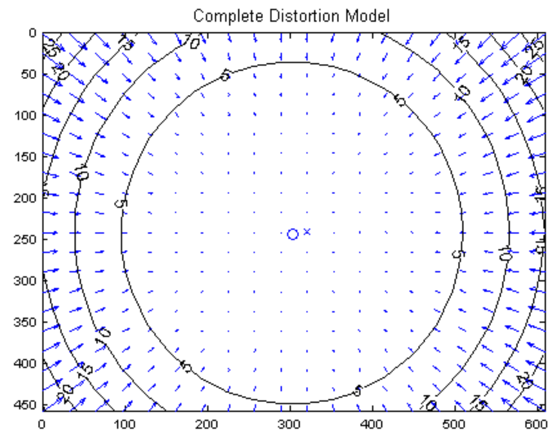
$$p = (x, y, 1)^T \propto w(x, y, 1)^T, \forall w \neq 0$$

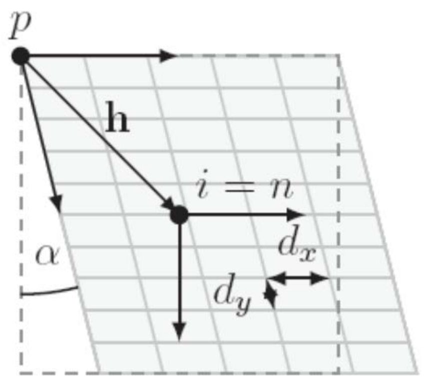
Unknown scale factor!

- A vector in P is just a representative of an equivalence class of vectors
- Everything is up-to-scale!

Follow-up: intrinsic parameters





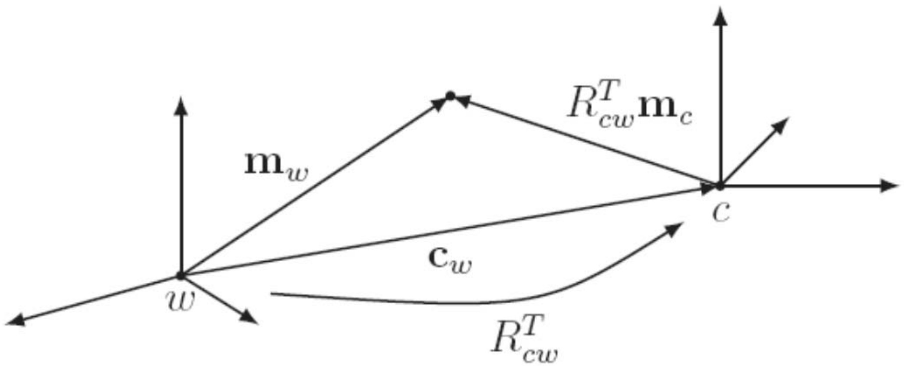


$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \underbrace{\left(1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6\right) \begin{bmatrix} x_n \\ y_n \end{bmatrix}}_{\text{radial distortion}}$$

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s_\alpha & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix}}_{K} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$$

Distorted coordinats

Follow-up: extrinsic parameters and P-matrix



$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{cw} & \mathbf{w}_c \\ \mathbf{0}_3^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \propto z_c \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s_\alpha & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{cw} & \mathbf{w}_c \\ \mathbf{0}_3^T & 1 \end{bmatrix}}_{P} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

"proportional to"

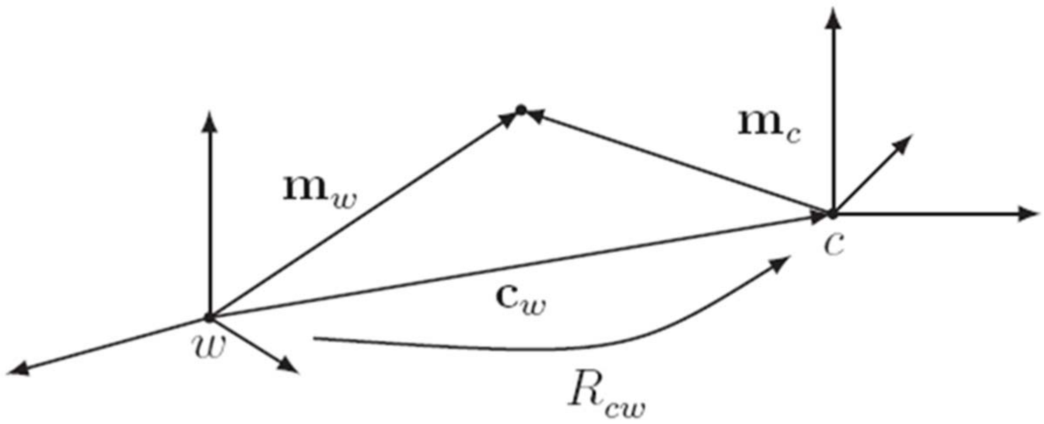
P does not include lens distortion!

This makes it a nonlinear function.

$$\mathbf{m}_p = \mathcal{P}(\mathbf{m}_w) = (\mathcal{K} \circ \mathcal{D} \circ \mathcal{P}_n \circ \mathcal{T})(\mathbf{m}_w)$$

How do c_w and w_c relate? $\mathbf{w}_c = -R_{cw}\mathbf{c}_w$

Follow-up: extrinsic parameters and P-matrix



$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \propto z_c \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s_\alpha & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{cw} & \mathbf{w}_c \\ \mathbf{0}_3^T & 1 \end{bmatrix}}_{P} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

P does not include lens distortion!

This makes it a nonlinear function.

How do m_w and m_c relate?

$$m_w = R_{cw}^T m_c + c_w$$

or

$$m_c = R_{cw} m_w + w_c$$

w_c???

$$\mathbf{w}_c = -R_{cw}\mathbf{c}_w$$

In matrix form:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{cw} & \mathbf{w}_c \\ \mathbf{0}_3^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

2D homography (projective transformation)

Definition:

Line preserving

A 2D *homography* is an invertible mapping h from P^2 to itself such that three points x_1,x_2,x_3 lie on the same line if and only if $h(x_1)$,$h(x_2)$,$h(x_3)$ do.

Theorem:

A mapping h: $P^2 \rightarrow P^2$ is a homography if and only if there exist a non-singular 3x3 matrix **H** such that for any point in P^2 represented by a vector x it is true that $h(\mathrm{x})=\mathbf{H}\mathrm{x}$

Definition: Homography

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \text{or} \quad \mathrm{x}' = \mathbf{H}\,\mathrm{x}$$

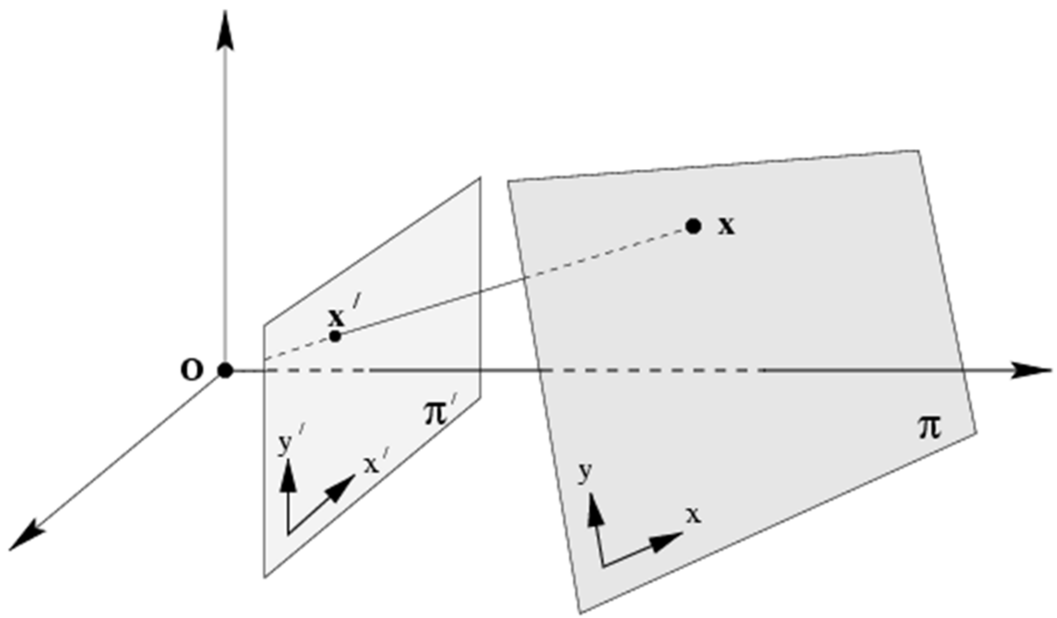
8DOF

Homography=projective transformation=projectivity=collineation

General homography

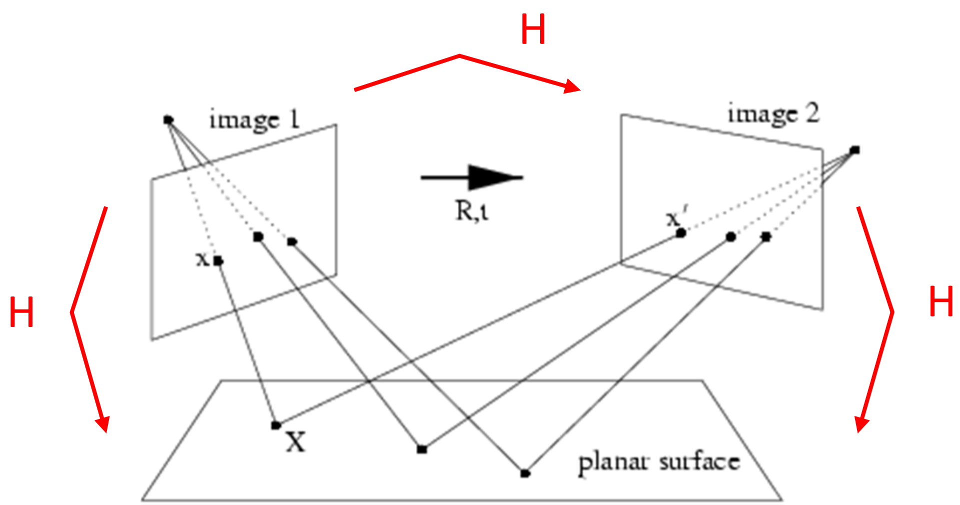
- Note: homographies are not restricted to P^2
- General definition:
  A homography is a non-singular, line preserving, projective mapping h: $P^n \rightarrow P^n$.
  It is represented by a square $(n + 1)$-dim matrix with $(n + 1)^2$-1 DOF

- Now back to the 2D case…
- Mapping between planes



Homographies in computer vision

Rotating/translating camera, planar world

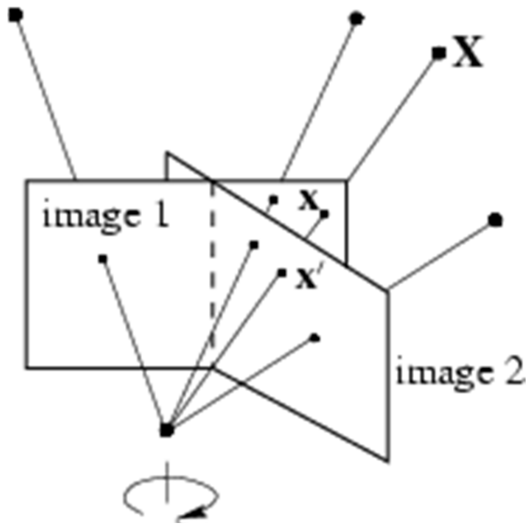


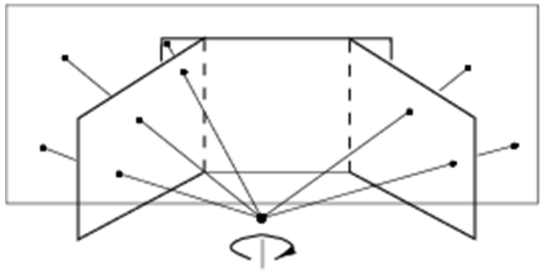
$$(x, y, 1)^T = x \propto PX = K[r_1 r_2 \cancel{r_3} t]\begin{pmatrix} X \\ Y \\ \cancel{0} \\ 1 \end{pmatrix} = H\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

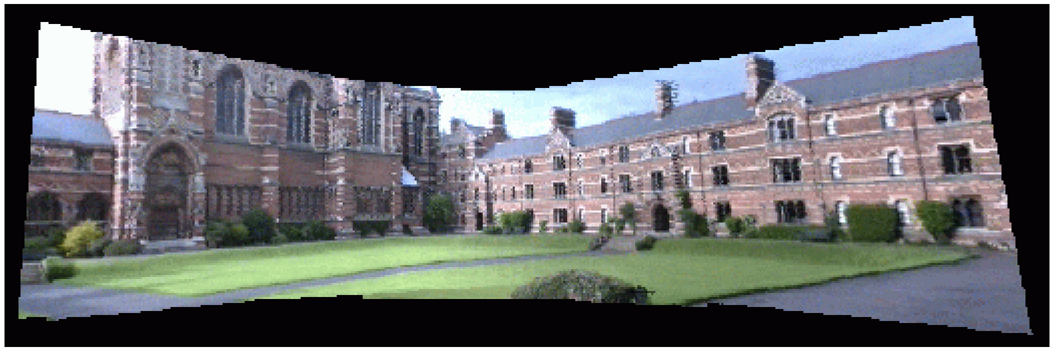
What happens to the P-matrix, if Z is assumed zero?

Homographies in computer vision

Rotating camera, arbitrary world







$$(x, y, 1)^T = x \propto PX = K\left(r_1 r_2 r_3 \cancel{t}\right)\begin{pmatrix} X \\ Y \\ Z \\ \cancel{1} \end{pmatrix} \propto KRK^{-1}x' = Hx'$$

What happens to the P-matrix, if t is assumed zero?

Transformation hierarchy: isometries

(*iso*=same, *metric*=measure)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \varepsilon\cos\theta & -\sin\theta & t_x \\ \varepsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \varepsilon = \pm 1$$

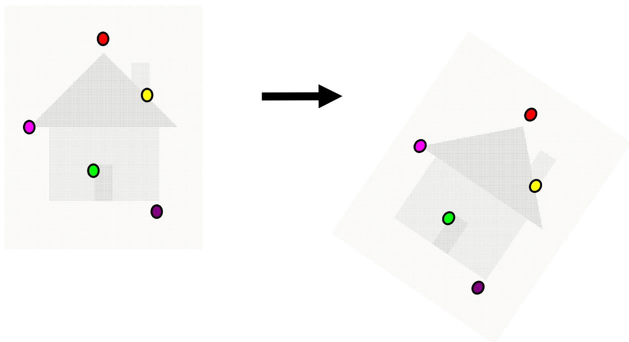
orientation preserving: $\varepsilon = 1$

orientation reversing: $\varepsilon = -1$

$$\mathrm{x}' = \mathbf{H}_E\,\mathrm{x} = \begin{bmatrix} \mathbf{R} & \mathrm{t} \\ 0^\mathsf{T} & 1 \end{bmatrix} \mathrm{x} \qquad \mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{I}$$

3DOF (1 rotation, 2 translation)

special cases: pure rotation, pure translation



Invariants: length, angle, area

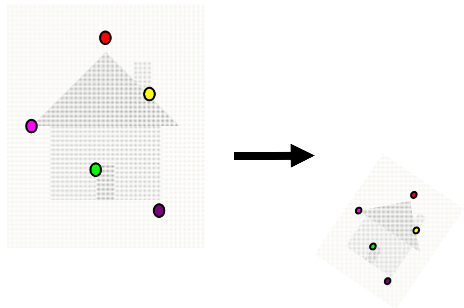
Transformation hierarchy: similarities

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(*isometry + scale*)

$$\mathrm{x}' = \mathbf{H}_S\,\mathrm{x} = \begin{bmatrix} s\mathbf{R} & \mathrm{t} \\ 0^\mathsf{T} & 1 \end{bmatrix} \mathrm{x}$$

$$\mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{I}$$



4DOF (1 scale, 1 rotation, 2 translation)

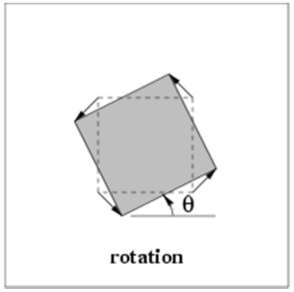
also know as *equi-form* (shape preserving)

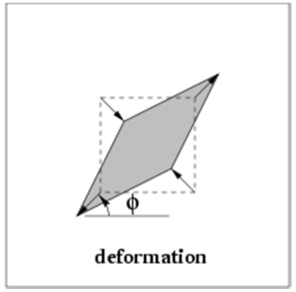
metric structure = structure up to similarity (in literature)

Invariants: ratios of length, angle, ratios of areas, parallel lines

Transformation hierarchy: affine transformations

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$





$$\mathrm{x}' = \mathbf{H}_A\, \mathrm{x} = \begin{bmatrix} \mathbf{A} & \mathrm{t} \\ 0^{\mathsf{T}} & 1 \end{bmatrix} \mathrm{x}$$

$$\mathbf{A} = \mathbf{R}(\theta)\mathbf{R}(-\phi)\mathbf{D}\mathbf{R}(\phi) \qquad \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

6DOF (2 scale, 2 rotation, 2 translation)

non-isotropic scaling! (2DOF: scale ratio and orientation)

Invariants: parallel lines, ratios of parallel lengths,
ratios of areas

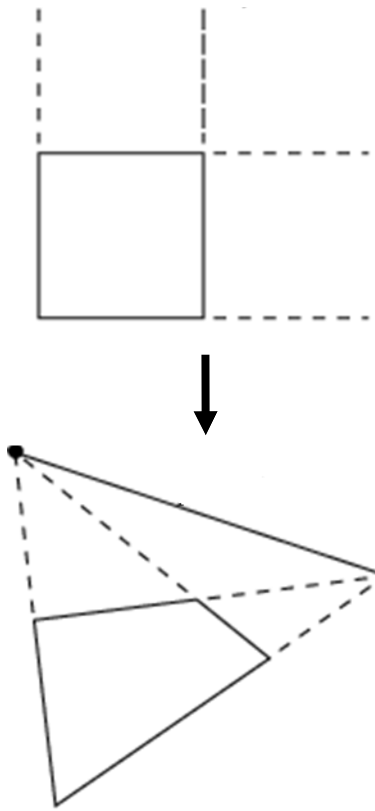
Transformation hierarchy: homographies

$$\mathbf{H}_P = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \vec{t} \\ \vec{v}^T & v \end{pmatrix}$$

Changes homogeneous coordinate!

$$\mathrm{x}' = \mathbf{H}_P\, \mathrm{x} = \begin{bmatrix} \mathbf{A} & \mathrm{t} \\ \mathrm{v}^{\mathrm{T}} & v \end{bmatrix} \mathrm{x}$$

$$\mathrm{v} = (v_1, v_2)^{\mathrm{T}}$$



8DOF (2 scale, 2 rotation, 2 translation, 2 line at infinity)

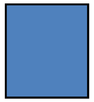
Acts non-homogeneous over the plane

Invariants: cross-ratio of four points on a line (ratio of ratio)

Allows to observe vanishing points, horizon

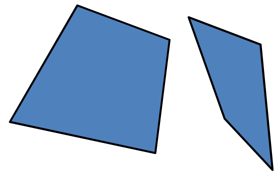
2D transformation hierarchy

A square transforms to:



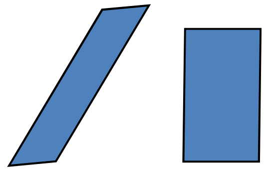
Projective
8dof

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



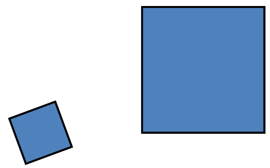
Affine
6dof

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



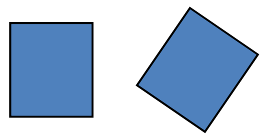
Similarity
4dof

$$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



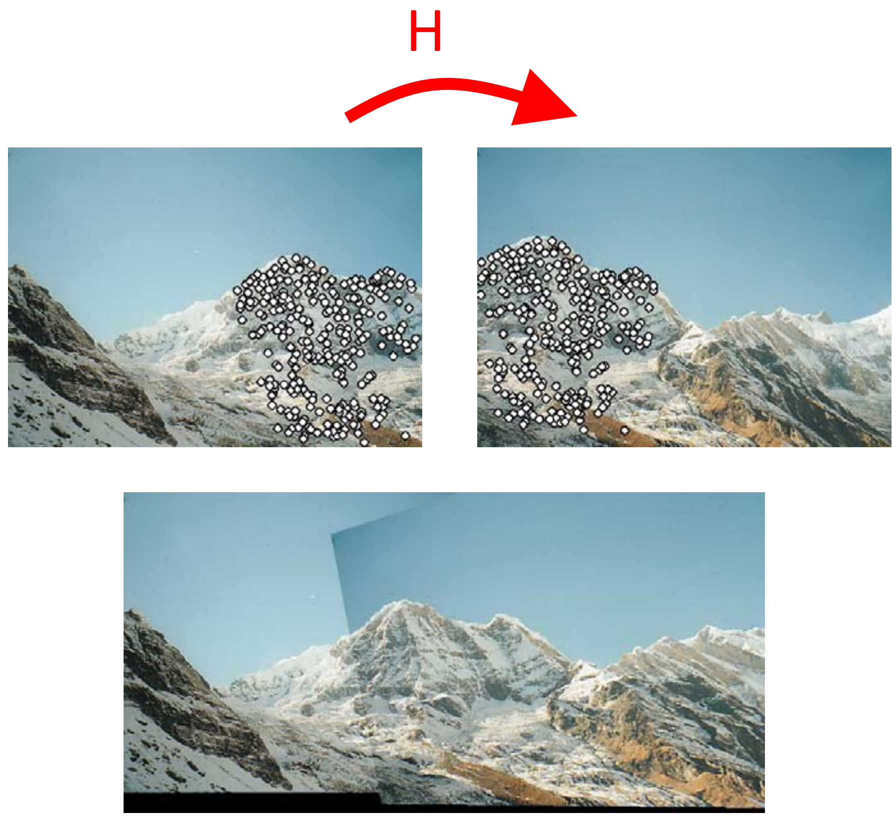
Euclidean
3dof

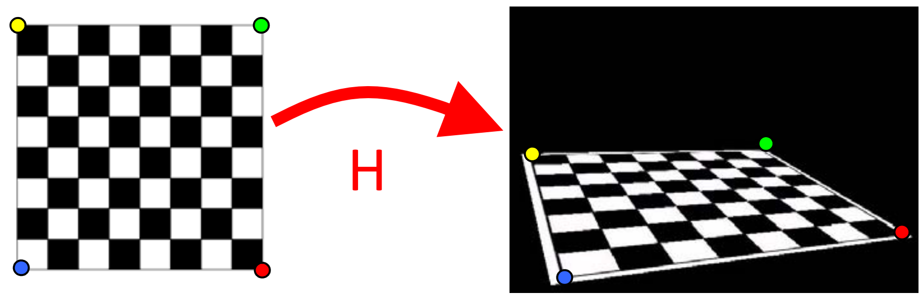
$$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

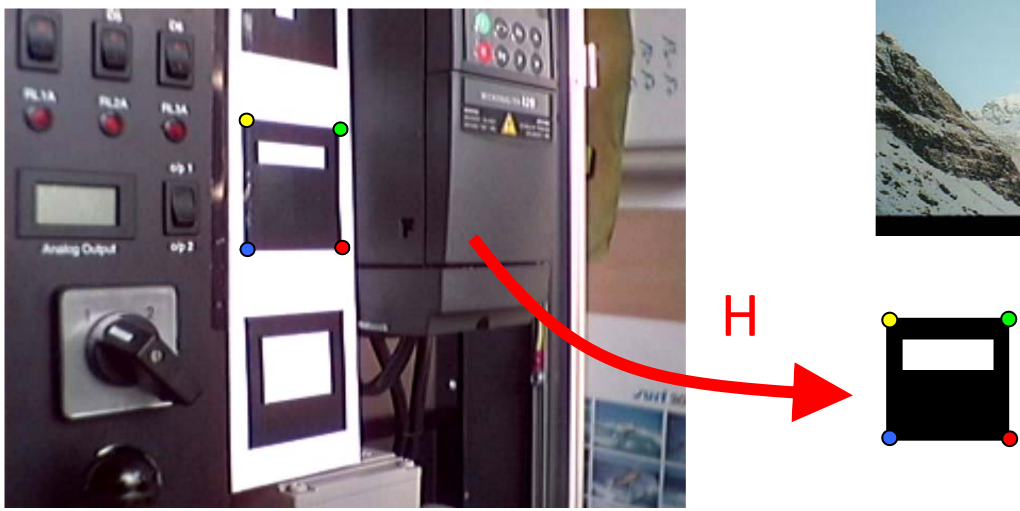


- Before:
  - What is a homography and how does it act on vectors/points
- Now:
  - How to estimate a homography from point correspondences
  - Same procedure as for P-matrix

- Estimate homography from point correspondences between:
  - two images
  - model plane and image
- Assumption: planar motion!







Homography estimation

- Homography mapping from image to image
  (holds under planar camera motion as mentioned before):

$$x' \propto Hx$$

$$\Leftrightarrow$$

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homography estimation

- Equation for one point correspondence i:

$$\lambda \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

i^{th} row of H

$$\lambda \mathbf{x}_i' = \mathbf{H}\,\mathbf{x}_i = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix} \mathbf{x}_i$$

Homogeneous coordinate, might be 1

9 entries, 8 degrees of freedom
(scale is arbitrary)

$$\mathbf{x}_i' \times \mathbf{H}\,\mathbf{x}_i = 0$$

$$\mathrm{x}_i' \times \mathrm{H}\mathrm{x}_i = \begin{pmatrix} y_i' \mathrm{h}^{3^\mathsf{T}} \mathrm{x}_i - w_i' \mathrm{h}^{2^\mathsf{T}} \mathrm{x}_i \\ w_i' \mathrm{h}^{1^\mathsf{T}} \mathrm{x}_i - x_i' \mathrm{h}^{3^\mathsf{T}} \mathrm{x}_i \\ x_i' \mathrm{h}^{2^\mathsf{T}} \mathrm{x}_i - y_i' \mathrm{h}^{1^\mathsf{T}} \mathrm{x}_i \end{pmatrix}$$

$$\begin{bmatrix} 0^\mathsf{T} & -w_i' \mathrm{x}_i^\mathsf{T} & y_i' \mathrm{x}_i^\mathsf{T} \\ w_i' \mathrm{x}_i^\mathsf{T} & 0^\mathsf{T} & -x_i' \mathrm{x}_i^\mathsf{T} \\ \cancel{-y_i' \mathrm{x}_i^\mathsf{T}} & \cancel{x_i' \mathrm{x}_i^\mathsf{T}} & \cancel{0^\mathsf{T}} \end{bmatrix} \begin{pmatrix} \mathrm{h}^1 \\ \mathrm{h}^2 \\ \cancel{\mathrm{h}^3} \end{pmatrix} = 0$$

3 equations, only 2 linearly independent, drop third row

Direct linear transform

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} h = 0 \Rightarrow Ah = 0$$

- H has 8 DOF (9 parameters, but scale is arbitrary)
- One correspondence gives two linearly independent equations
- Four matches needed for a minimal solution (null space of 8x9 matrix)
- More points: search for “best” according to some cost function

Direct linear transform

- No exact solution because of inexact measurements due to noise
- With n correspondences: size A is 2nx9, rank most likely not 8

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} h = 0 \qquad Ah = 0$$

- Find approximate solution
  - Additional constraint needed to avoid 0, e.g. $\|h\| = 1$
  - $Ah = 0$ not possible, so minimize $\|Ah\|$

DLT algorithm

<u>Objective</u>

Given n≥4 2D to 2D point correspondences $\{x_i \leftrightarrow x_i'\}$, determine the 2D homography matrix H such that $x_i'=Hx_i$

<u>Algorithm</u>

(i) For each correspondence $x_i \leftrightarrow x_i'$ compute A_i. Usually only two first rows needed.

(ii) Assemble n 2x9 matrices A_i into a single 2nx9 matrix A

(iii) Obtain SVD of A. Solution for h is last column of V,
=singular value of A
=eigen vector to the smallest eigen value of A^TA

(iv) Determine H from h

Inhomogeneous solution

Since h can only be computed up to scale, impose constraint pick h_j=1, e.g. h_9=1, and solve for 8-vector

$$\begin{bmatrix} 0 & 0 & 0 & -x_i w_i' & -y_i w_i' & -w_i w_i' & x_i y_i' & y_i y_i' \\ x_i w_i' & y_i w_i' & w_i w_i' & 0 & 0 & 0 & x_i x_i' & y_i x_i' \end{bmatrix} \tilde{\mathrm{h}} = \begin{pmatrix} -w_i y_i' \\ w_i x_i' \end{pmatrix}$$

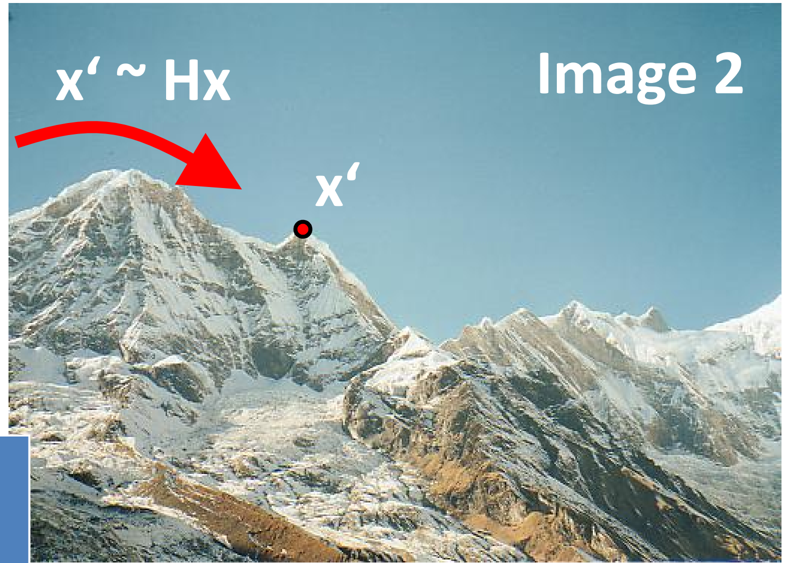
Can be solved using linear least-squares

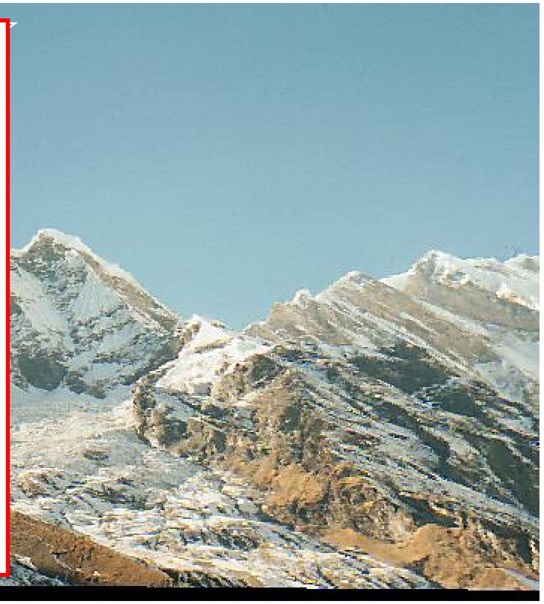
However, if h_9=0 this approach fails
Also poor results if h_9 close to zero
Therefore, not recommended

And what now?

What can we do when knowing the homography between two images

Application (1): panorama stitching





Panorama stitching:

1. Undistort images
2. Find point correspondences between images
3. Compute homography H
4. Resample:
   1. Loop over image 1
   2. Project into image 2 using H
   3. Bilinear interpolation in image 2

Application (2): camera pose estimation

- Assuming that K (intrinsic calibration matrix) is known, derive the 3D camera pose from H
- Enables augmentation of 3D virtual objects (augmented reality)
  - Set virtual camera to real camera
  - Render virtual scene
  - Compose with real image
- Enables localization/navigation
- Recall the two cases of planar motion:
  - purely rotating camera, arbitrary scene
  - Rotating and translating camera, planar scene

Camera pose estimation

- Purely rotating camera:

$$\boldsymbol{x} \sim \boldsymbol{K}\begin{bmatrix} \boldsymbol{I} & | & \boldsymbol{0} \end{bmatrix}\boldsymbol{X}$$

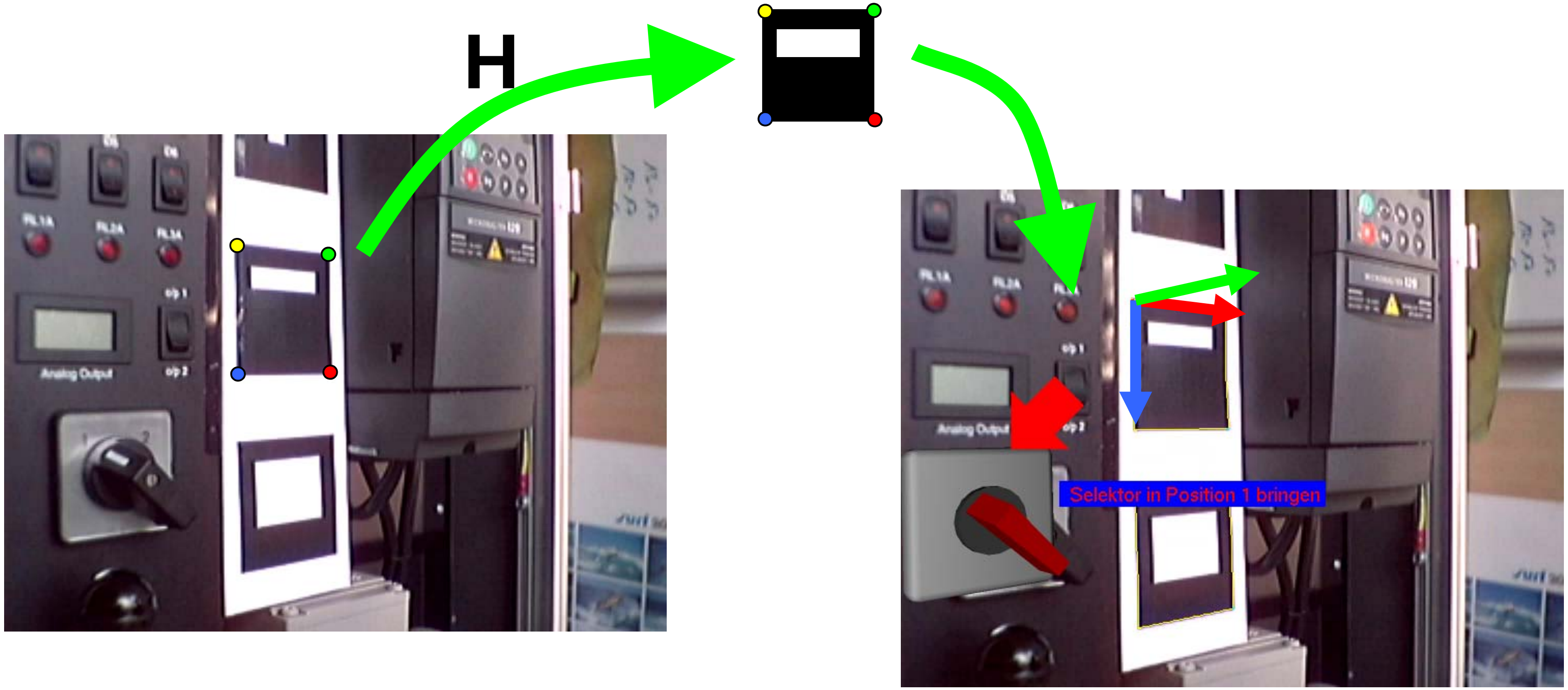
$$\begin{aligned}
\boldsymbol{x}' &\sim \boldsymbol{K}\begin{bmatrix} \boldsymbol{R} & | & \boldsymbol{0} \end{bmatrix}\boldsymbol{X} \\
&\sim \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{-1}\boldsymbol{K}\begin{bmatrix} \boldsymbol{I} & | & \boldsymbol{0} \end{bmatrix}\boldsymbol{X} \\
&\sim \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{-1}\boldsymbol{x} \\
&\sim \boldsymbol{H}\boldsymbol{x}
\end{aligned}$$

$$\boldsymbol{H}_i \sim \boldsymbol{K}\boldsymbol{R}_i\boldsymbol{K}^{-1}$$

⇨ Movie

Camera pose estimation

- Planar scene (example marker tracker, applies to any planar scene):



Camera pose estimation

Assume all points lie in one plane with Z=0:

$$\boldsymbol{X} = (X, Y, 0, 1)$$

$$\begin{aligned}
\boldsymbol{x} &= \boldsymbol{P}\boldsymbol{X} \\
&= \boldsymbol{K}\left[\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{r}_3\boldsymbol{t}\right]\begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} \\
&= \boldsymbol{K}\left[\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{t}\right]\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \\
&= \boldsymbol{H}\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}
\end{aligned}$$

$$\boldsymbol{H} = \lambda\boldsymbol{K}\left[\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{t}\right]$$

$$\boldsymbol{K}^{-1}\boldsymbol{H} = \lambda\left[\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{t}\right]$$

– r_1 and r_2 are unit vectors ⇨ find lambda
– Use this to compute t
– Rotation matrices are orthogonal ⇨ find r3

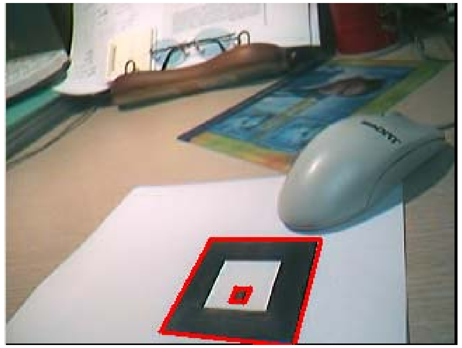
$$\boldsymbol{P} = \boldsymbol{K}\begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & (\boldsymbol{r}_1 \times \boldsymbol{r}_2) & \boldsymbol{t} \end{bmatrix}$$

Problems

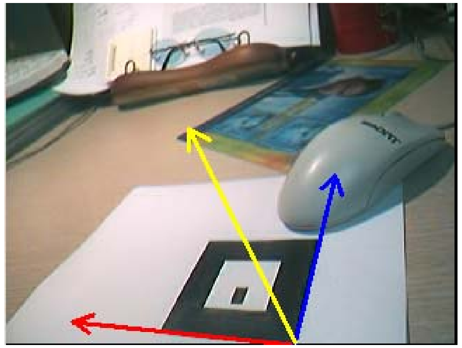
- Problem:
  - The vectors r_1 and r_2 might not yield the same lambda
- Solution:
  - Use the average value
- Problem:
  - The estimated rotation matrix might not be orthogonal
- Solution: orthogonalize R'
  - Obtain SVD ⇨ $R'=UWV^T$
  - Set singular values to 1 ⇨ $R=UV^T$

Example: marker tracker

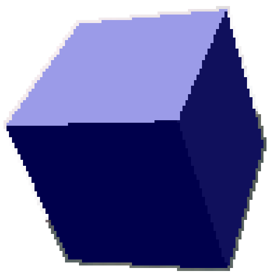
Video-input



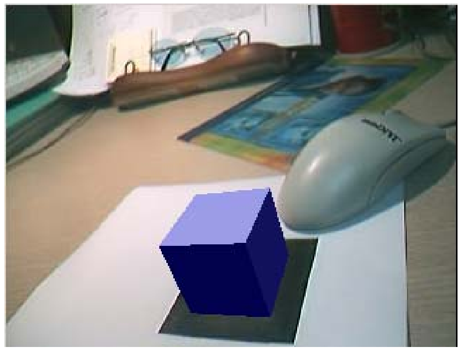
Pattern recognition
(point correspondences
from 4 corners)



Homography ⇨ 3D
pose



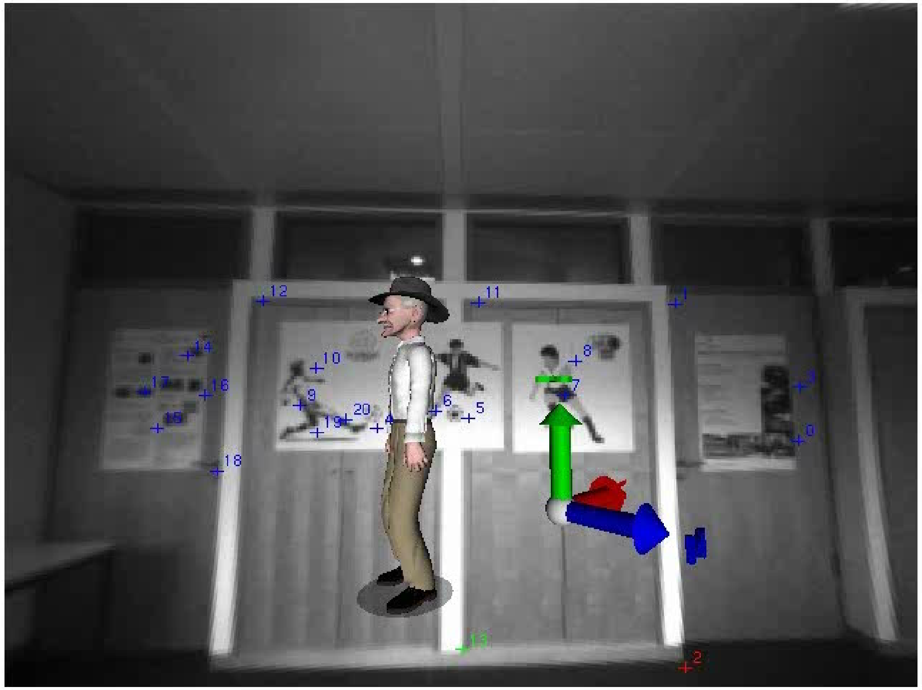
Rendering of the
virtual object



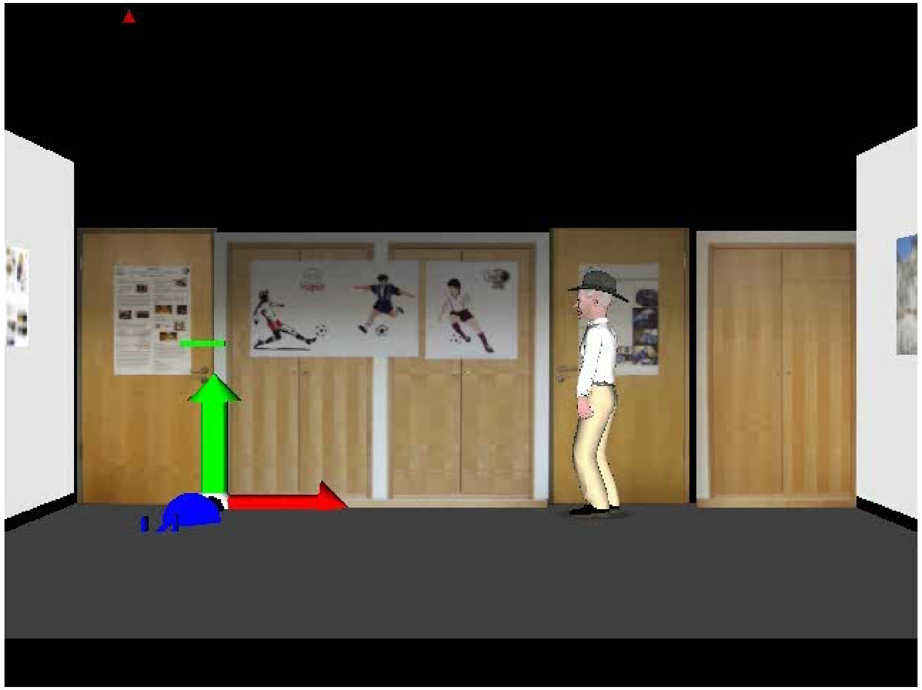
Synthesis and
overlay

Example: natural feature tracking

Life image with augmentations



Virtual scene



References

- Homography estimation from planes:
  - Zhang: Flexible camera calibration by viewing a plane from unknown orientations, ICCV, 1999.
- Homography estimation from purely rotating camera
  - Hartley: Self-Calibration from Multiple Views with a Rotating Camera, ECCV, 1994
  - Brown and Lowe: Recognizing Panoramas, ICCV, 2003.

Thank you!

Next lecture:

Linear/nonlinear/robust estimation techniques