Computer Vision I

From Single to Two-View Geometry – 03.07.2013

TECHNISCHE UNIVERSITÄT DARMSTADT

vi visual inference

Announcements

TECHNISCHE UNIVERSITÄT DARMSTADT

- Class next week
  - Will be held by Stephan Richter
- Class in two weeks
  - is cancelled (hence double lecture today)
- Q&A Session
  - Wednesday 24.7.2013, 9:50
  - Chance to ask detailed questions on the entire material
  - Participation optional (otherwise office hours, email, etc.)

Outline

TECHNISCHE UNIVERSITÄT DARMSTADT

- Part 1: Camera calibration
- Part 2: Single view geometry
- Homography
  - Perspective image of a plane
  - Views from same camera centre
- (Homography) estimation
  - Numerical conditioning of coordinates
  - Robust fitting with RANSAC
- Application: Panorama stitching

Projecting a planar object

TECHNISCHE UNIVERSITÄT DARMSTADT

- Why is this useful?
  - undistort perspective images of planar structures
  - map from one plane to another

Projecting a planar object

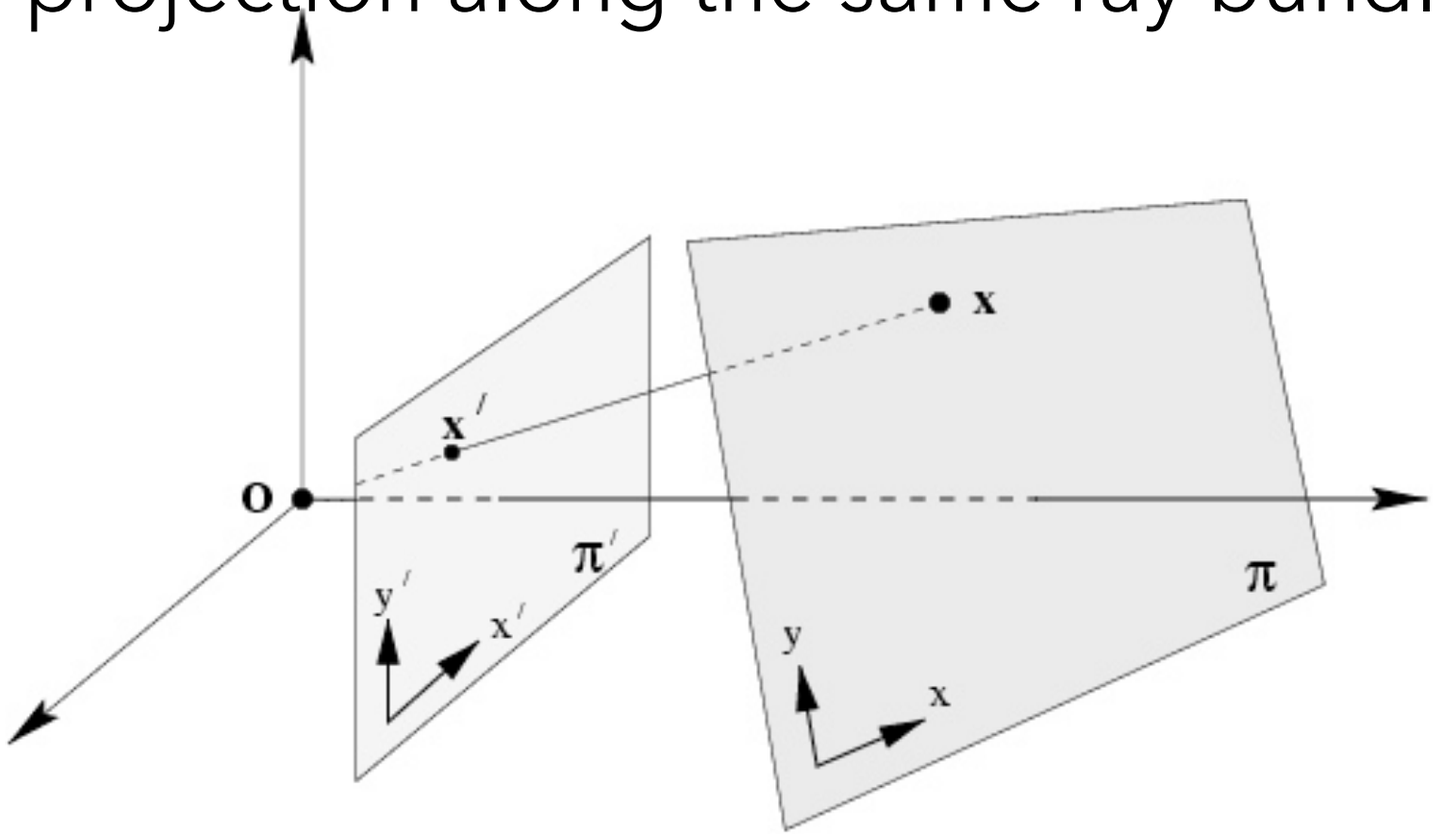
TECHNISCHE UNIVERSITÄT DARMSTADT

- Why is this useful?
  - undistort perspective images of planar structures
  - map from one plane to another
- geometrically: How does a pinhole camera map a plane?
- algebraically: How does a projection matrix act on co-planar points?

Geometric intuition

TECHNISCHE UNIVERSITÄT DARMSTADT

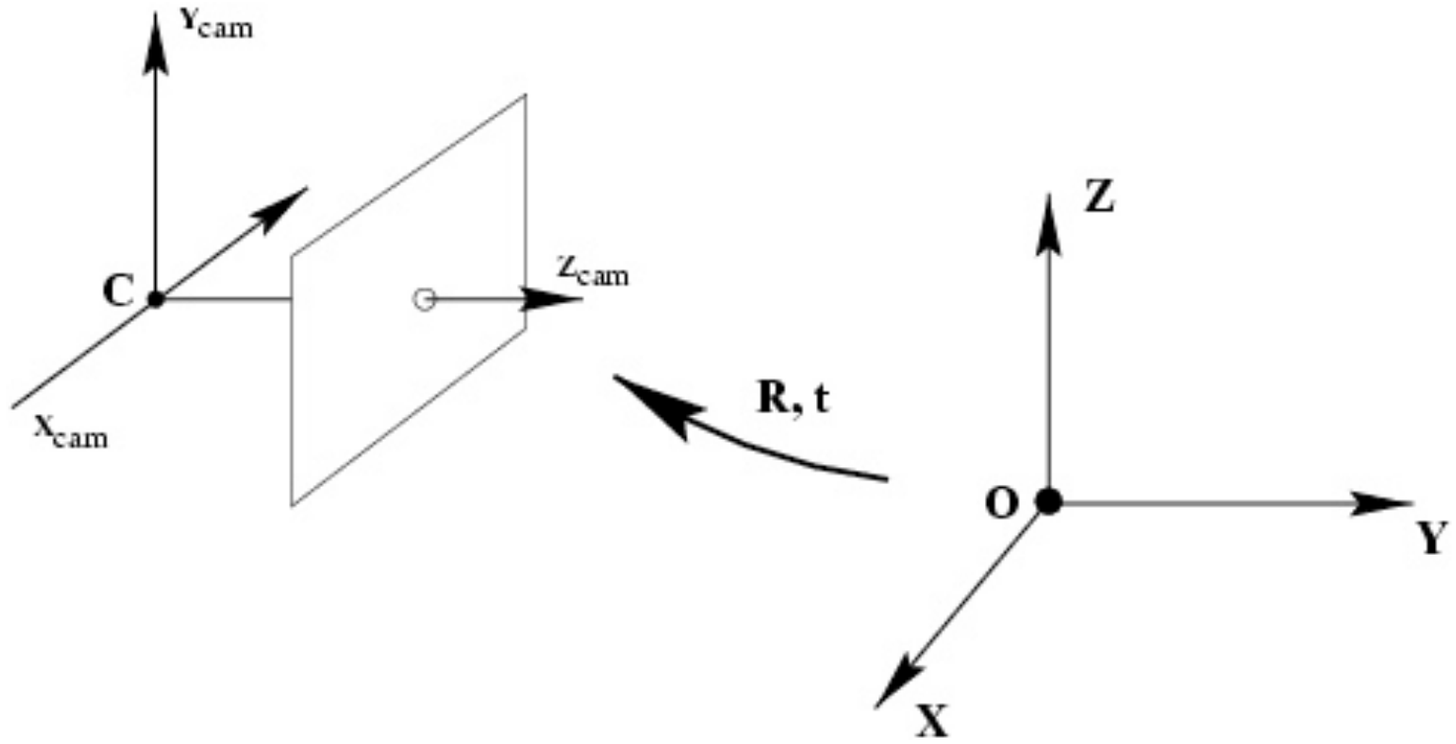
- Projection with a straight ray from a 2D plane to another 2D plane (the retina of the camera)
- No dimensionality reduction
- Ray intersects each plane exactly once
- Inverse projection along the same ray bundle



Pinhole camera model

TECHNISCHE UNIVERSITÄT DARMSTADT

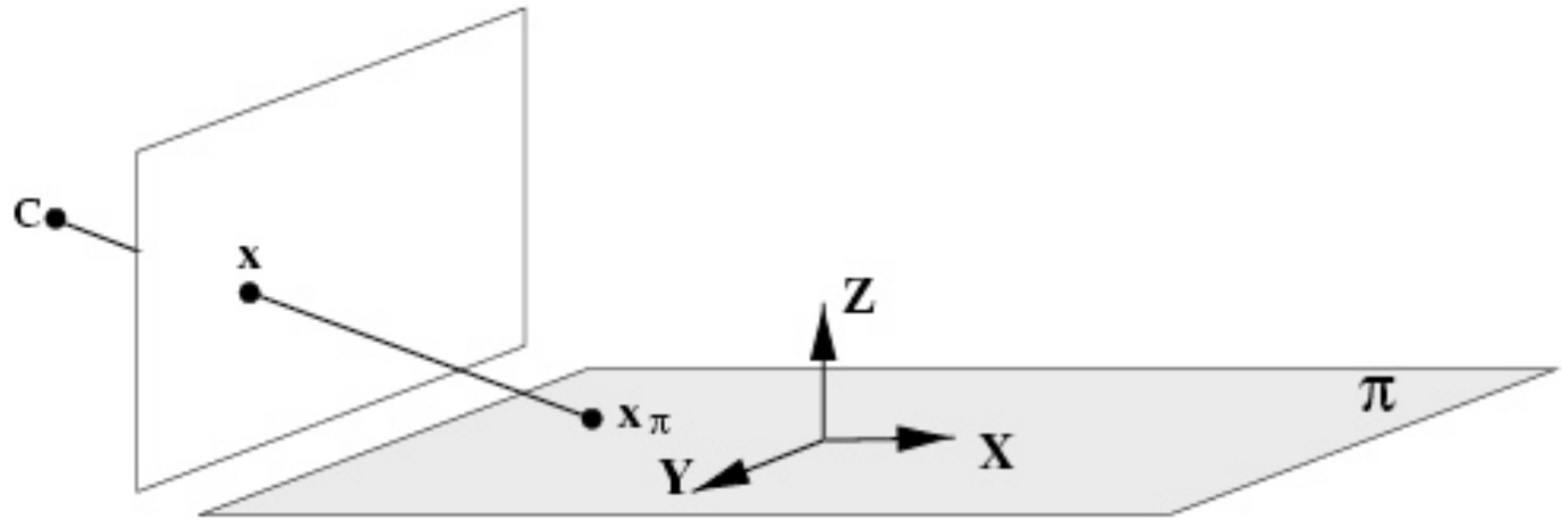
- Remember: a pinhole camera is represented by a (3x4) homogeneous projection matrix P



$$x = PX = K[R|-R\tilde{C}]X$$

Projecting a plane

- Coordinate system in the plane: Z=0

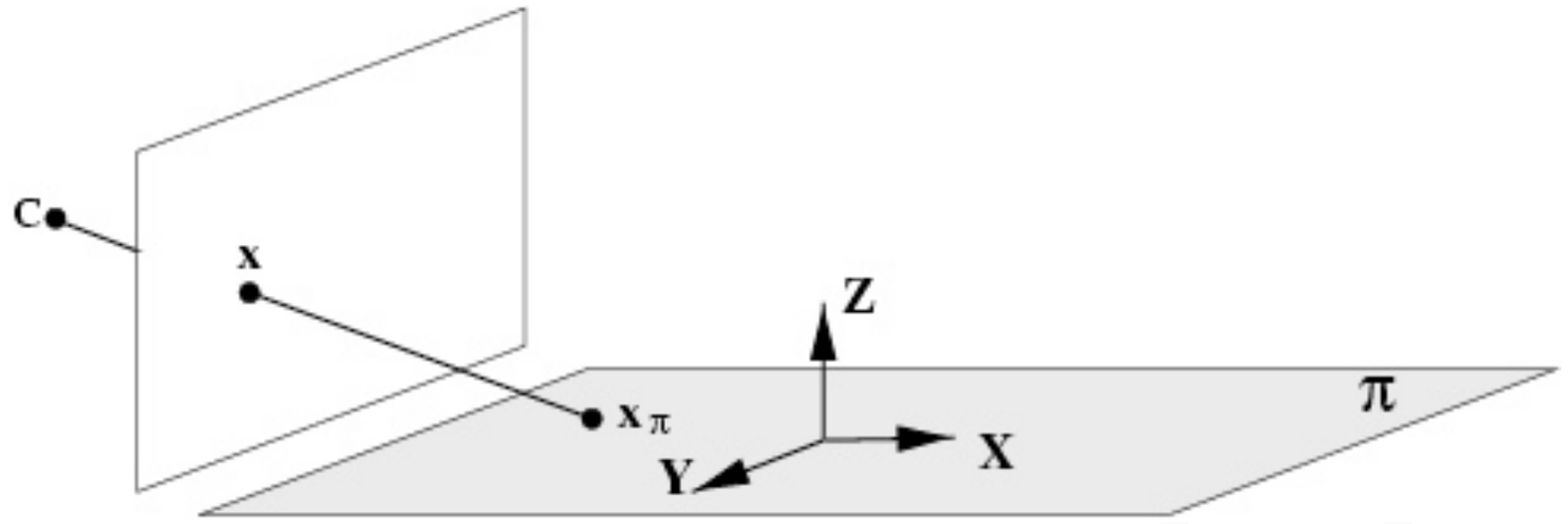


$$x = ?x_\pi$$

Projecting a plane

TECHNISCHE UNIVERSITÄT DARMSTADT

- Coordinate system in the plane: Z=0



$$x = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ 0 \\ 1 \end{bmatrix} =$$

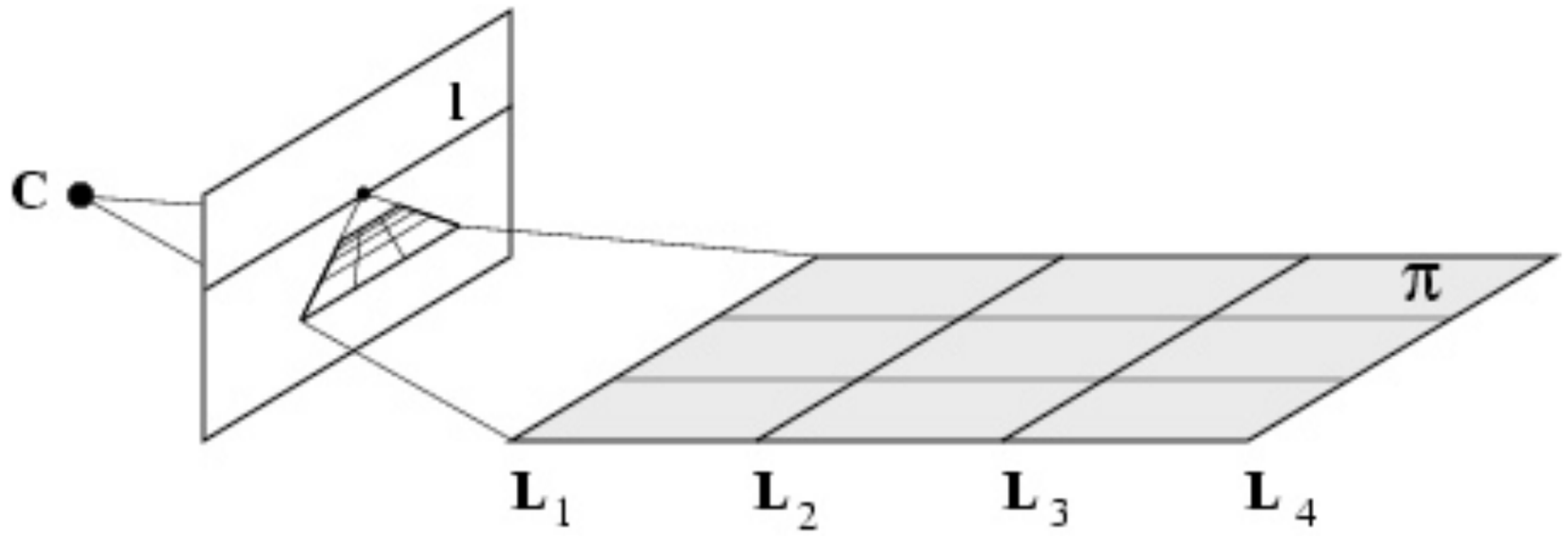
$$= \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ 1 \end{bmatrix} = Hx_\pi$$

Projecting a plane

TECHNISCHE UNIVERSITÄT DARMSTADT

- The mapping is given by a (3x3) matrix H
  - H is called a homography (or projectivity)
  - the most general projective 2D transformation
  - maps a square to a general quadrangle
  - has 8 degrees of freedom
  - H is a one-to-one mapping, hence invertible

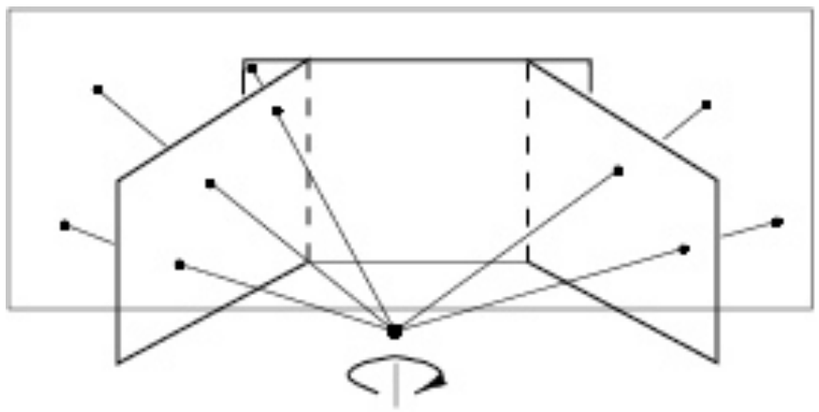
$$x = Hx_\pi \qquad\qquad x_\pi = H^{-1}x$$



Rotating camera

TECHNISCHE UNIVERSITÄT DARMSTADT

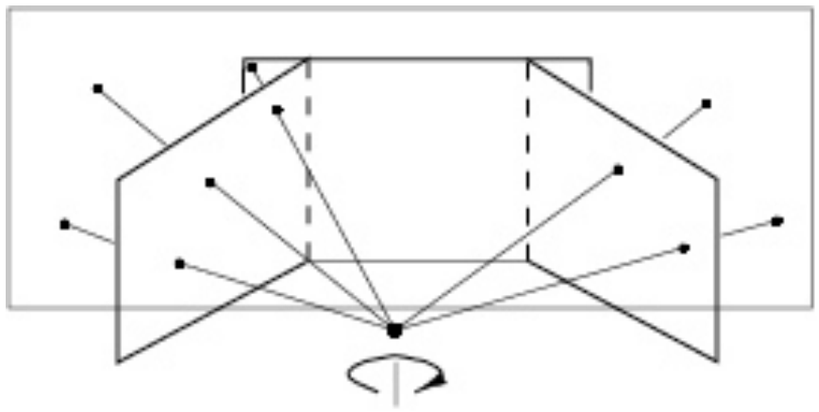
- Relation between two images, if the camera only rotates (baseline = 0)



Rotating camera

TECHNISCHE UNIVERSITÄT DARMSTADT

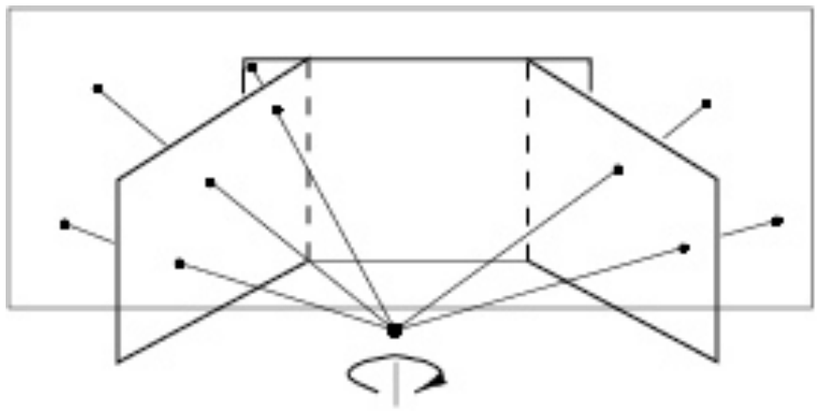
- Relation between two images, if the camera only rotates (baseline = 0)
  - Projection rays are identical
  - ray maps directly from one image plane to the other
  - the two images are related by a homography



Rotating camera

TECHNISCHE UNIVERSITÄT DARMSTADT

- Relation between two images, if the camera only rotates (baseline = 0)
  - Projection rays are identical
  - ray maps directly from one image plane to the other
  - the two images are related by a homography



$$x = K[I|0]X = K\tilde{X} \qquad \tilde{X} = K^{-1}x$$

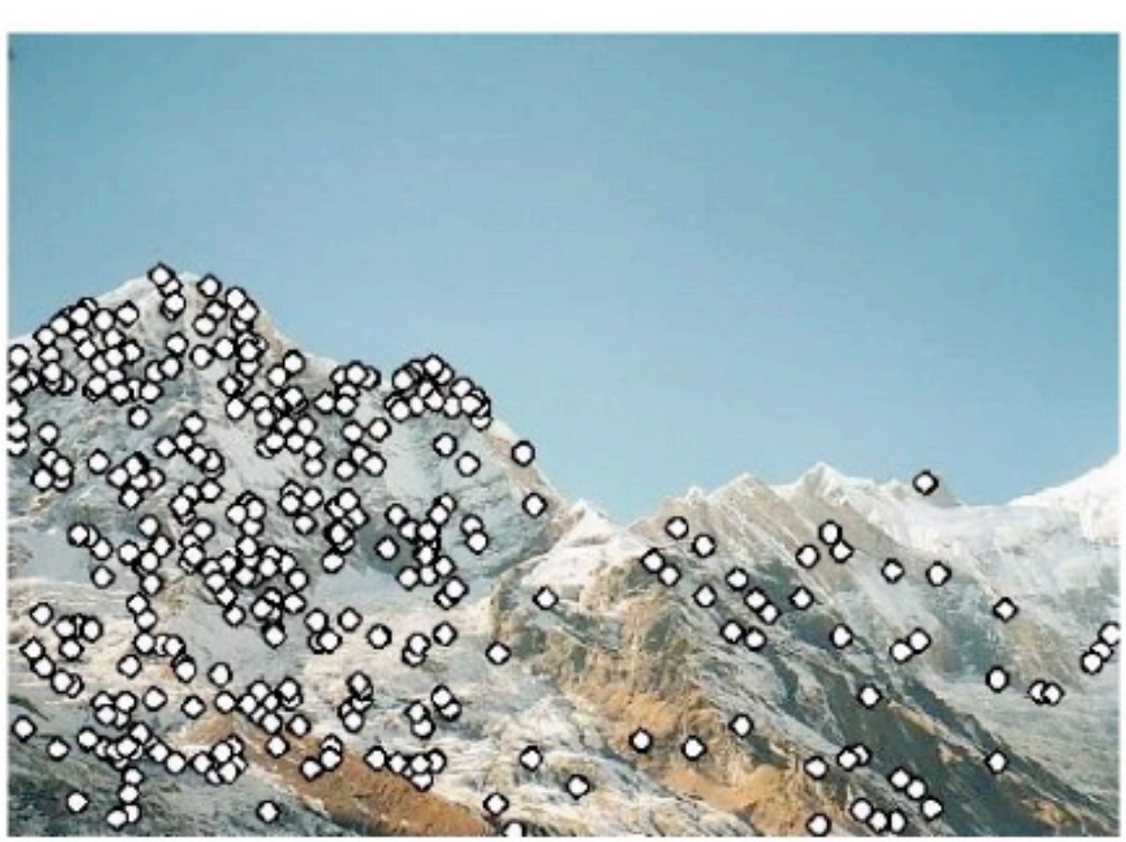
$$x' = K'[R|0]X = K'R\tilde{X} \qquad x' = \boxed{K'RK^{-1}}x$$

$$x' = Hx$$

Rotating camera

TECHNISCHE UNIVERSITÄT DARMSTADT

- Application
  - Panorama stitching
  - estimate the homography between two overlapping images, transform one into the image plane of the other



Rotating camera

TECHNISCHE UNIVERSITÄT DARMSTADT

- Application
  - Panorama stitching
  - estimate the homography between two overlapping images, transform one into the image plane of the other

Estimating the homography

TECHNISCHE UNIVERSITÄT DARMSTADT

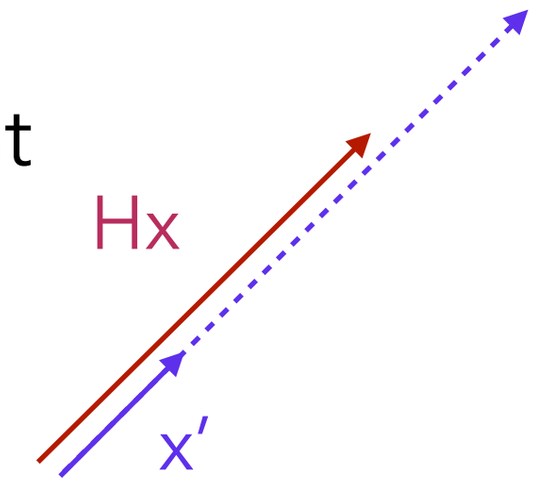
- Eight unknowns
- Two linearly independent equations per point
- Need 4 point correspondences

$$\lambda x' = Hx \qquad , \qquad x' \propto Hx$$

Estimating the homography

TECHNISCHE UNIVERSITÄT DARMSTADT

- Eight unknowns
- Two linearly independent equations per point
- Need 4 point correspondences



$$x' \propto Hx \qquad \rightarrow \qquad Hx \times x' = 0$$

$$\begin{bmatrix} 0 & 0 & 0 & x & y & 1 & -xy' & -yy' & -y' \\ -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

Estimating the homography

TECHNISCHE UNIVERSITÄT DARMSTADT

- Stack equations from at least 4 correspondences
- Remember
  - a homogeneous linear equation system

Solution ?

Estimating the homography

TECHNISCHE UNIVERSITÄT DARMSTADT

- Stack equations from at least 4 correspondences
- Remember
  - a homogeneous linear equation system
  - solve s.t. $||h||=1$, using SVD

$$Ah = 0$$

$$[U, D, V] = \mathrm{svd}(A)$$

$$h = \begin{bmatrix} v_{19} \\ v_{29} \\ \vdots \\ v_{99} \end{bmatrix}$$

Estimating the homography

TECHNISCHE UNIVERSITÄT DARMSTADT

- A note on numerical stability
  - Coefficients of an equation system should be in the same order of magnitude so as not to lose significant digits
  - in pixels: $xx' \sim 1\mathrm{E}6$
  - Conditioning: scale and shift points to be in [-1..1]
  - apply inverse transform to estimated homography
  - a general rule, not only for homography

$$\begin{bmatrix} 0 & 0 & 0 & x & y & 1 & -xy' & -yy' & -y' \\ -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

Estimating the homography

TECHNISCHE UNIVERSITÄT DARMSTADT

- A note on numerical stability
  - Conditioning: scale and shift points to be in [-1..1]

$$s = \frac{1}{2} \max_i(\|x_i\|)$$

$$t = \text{mean}(x_i)$$

$$s' = \frac{1}{2} \max_i(\|x'_i\|)$$

$$t' = \text{mean}(x'_i)$$

$$T = \begin{bmatrix} \frac{1}{s} & 0 & -\frac{t_x}{s} \\ 0 & \frac{1}{s} & -\frac{t_y}{s} \\ 0 & 0 & 1 \end{bmatrix}$$

$$T' = \begin{bmatrix} \frac{1}{s'} & 0 & -\frac{t'_x}{s'} \\ 0 & \frac{1}{s'} & -\frac{t'_y}{s'} \\ 0 & 0 & 1 \end{bmatrix}$$

$$u = Tx$$

$$u' = T'x'$$

SVD

$$u' = \bar{H}u$$

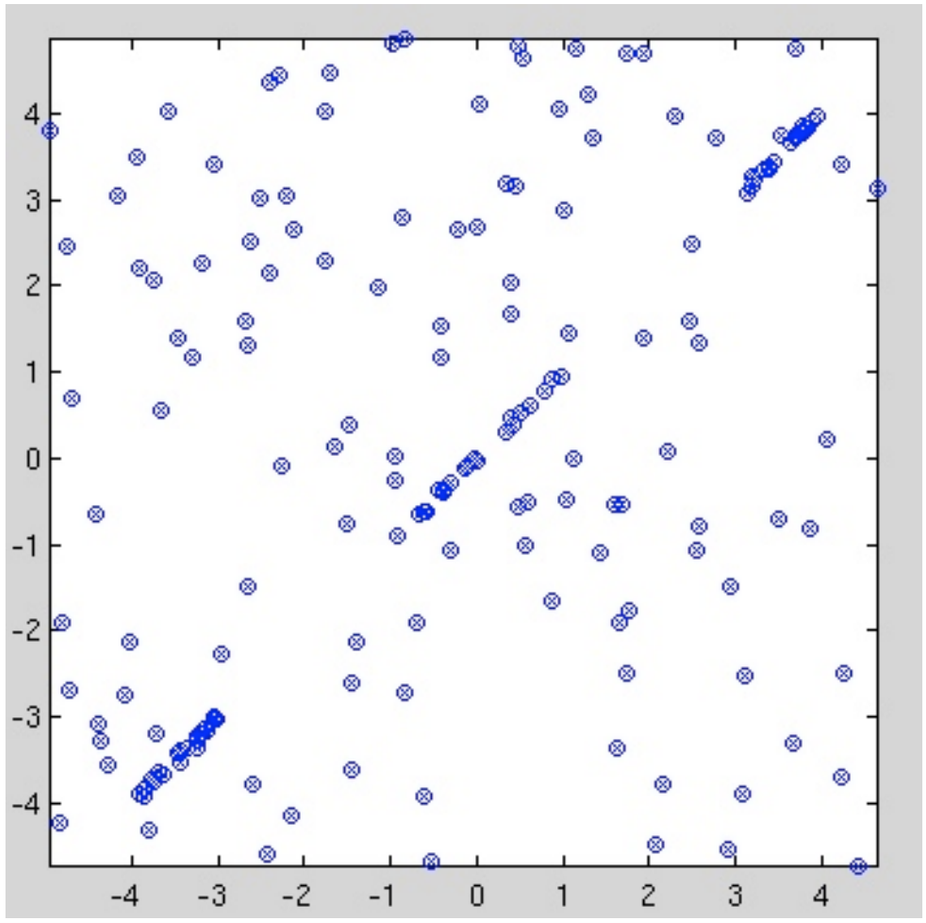
$$T'x' = \bar{H}Tx$$

$$H = (T')^{-1}\bar{H}T$$

Robust Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

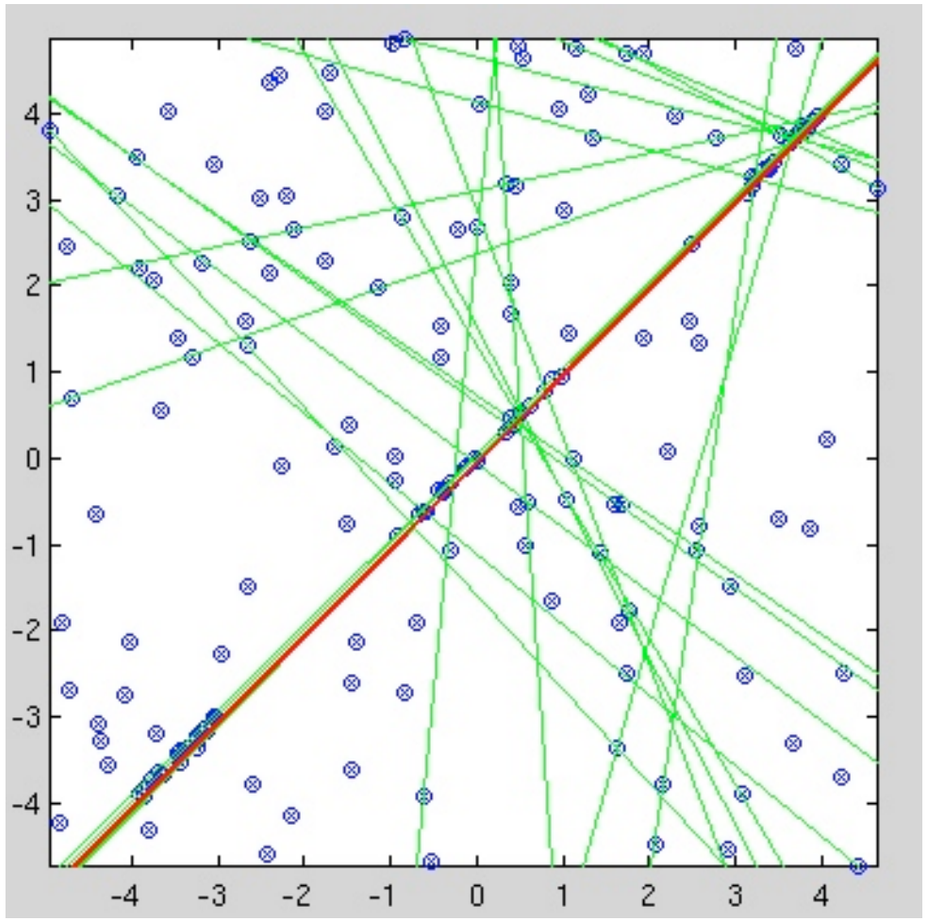
- Problem: some correspondences may be wrong
- Using false correspondences will corrupt the estimate
- Need a method to get rid of matching errors
- Again general, not only for homography



Robust Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- RANSAC (random sample consensus)
  - iterate:
    - randomly pick a minimal set of points
    - construct the model (a line, a homography, ...) from the points
    - measure the support (count number of points with small error)
  - keep the best hypothesis



Robust Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- RANSAC (random sample consensus)
  - we cannot try all possible samples - how many is “enough”?

Robust Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- RANSAC (random sample consensus)
  - we cannot try all possible samples - how many is “enough”?
  - assume we have a conservative guess of the fraction of “inliers” (correct points)

- say 20% of the correspondences are wrong
- how many random samples do we have to pick, in order to “almost certainly” find the right homography?

Robust Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- RANSAC (random sample consensus)
  - we cannot try all possible samples - how many is “enough”?
  - assume we have a conservative guess of the fraction of “inliers” (correct points)

▪ probability of picking an uncontaminated sample	w^p
▪ probability of seeing no uncontaminated sample in *k* trials	$z = (1 - w^p)^k$
▪ required *k* to be *z*% (say, 99%) sure there is an uncontaminated sample	$k = \frac{\log(z)}{\log(1 - w^p)}$

Robust Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- RANSAC (random sample consensus)
  - an immensely successful brute-force method
  - even the best hypothesis may be a rather bad fit
  - should always refit to all inliers

p	proportion of outliers						
	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Estimating the homography

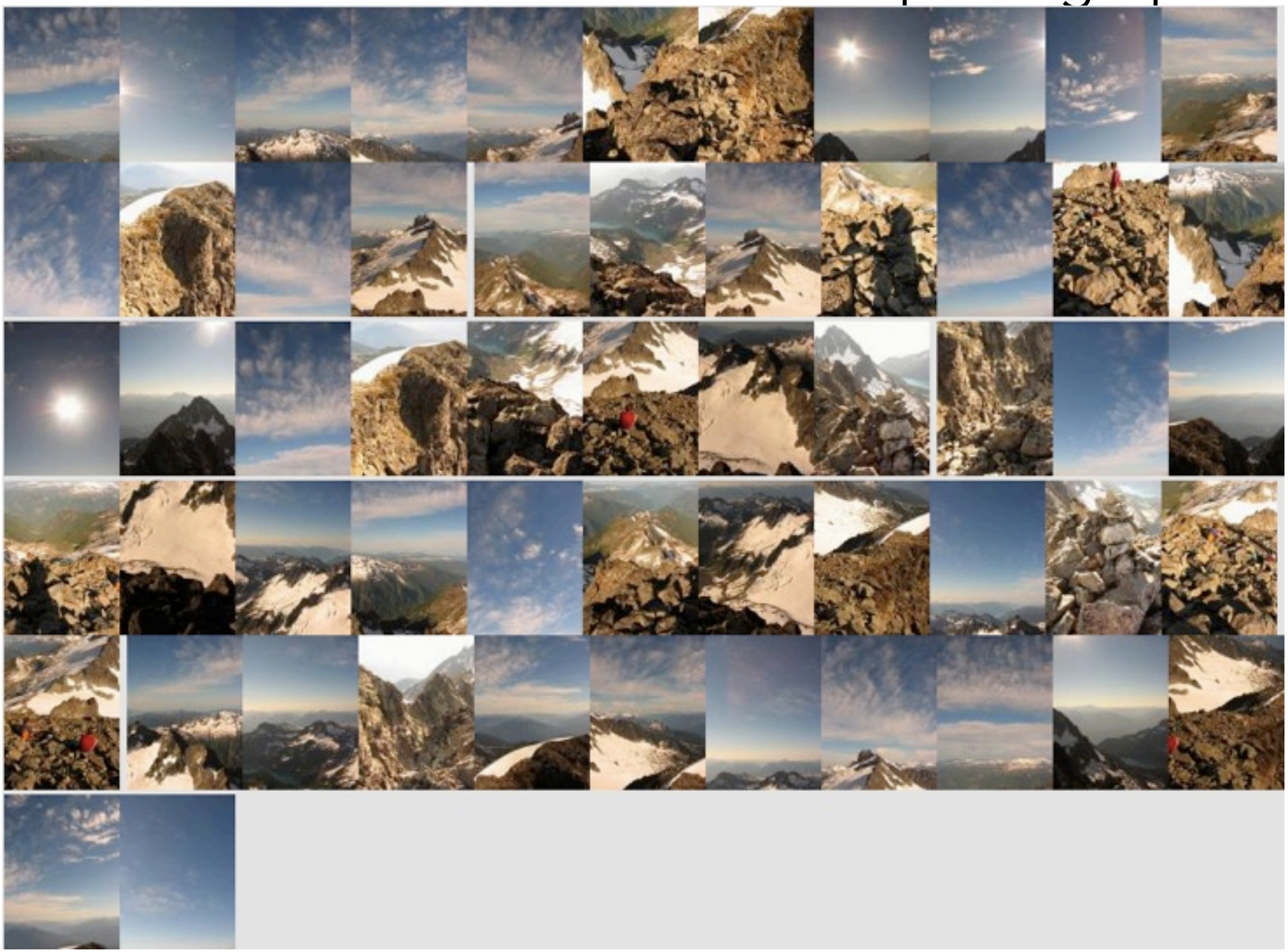
TECHNISCHE UNIVERSITÄT DARMSTADT

- The cookbook recipe:
  - condition all image coordinates
  - RANSAC loop: iterate
    - pick 4 correspondences
    - build equations, estimate homography
    - transform all points x
    - measure distance to all points x' (in non-homogeneous coordinates!)
    - count inliers (scale down threshold too!)
    - re-estimate final homography
  - undo conditioning

Application: AutoStitch

TECHNISCHE UNIVERSITÄT DARMSTADT

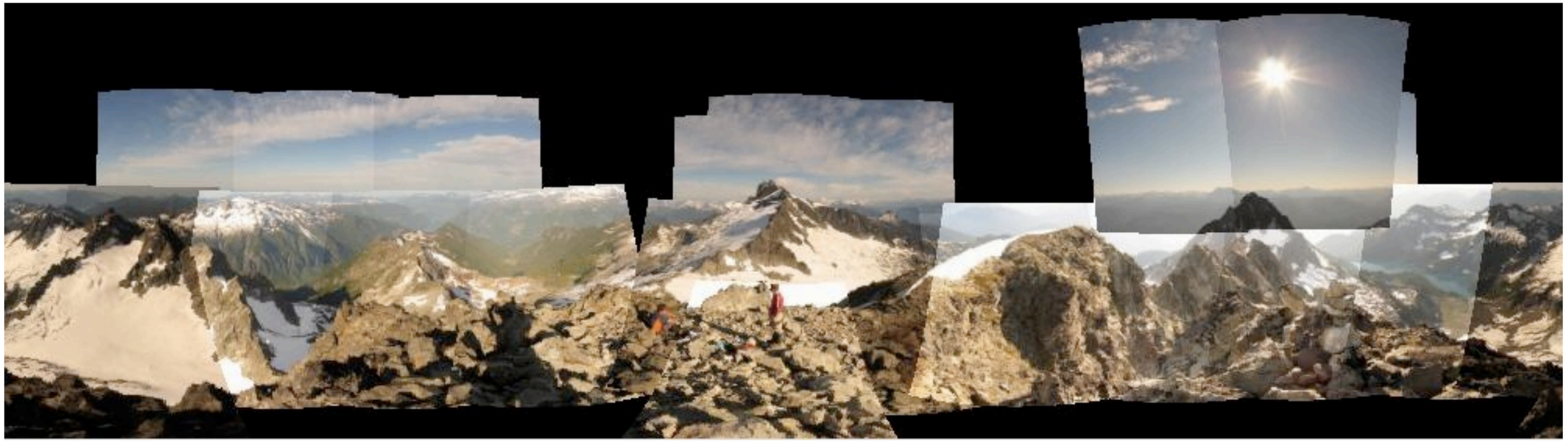
- Given a collection of unordered photographs...



Application: AutoStitch

TECHNISCHE UNIVERSITÄT DARMSTADT

- ... automatically create a panorama
  - find images related by a homography
  - extract relative rotation
  - remap to a sphere
  - blend colors along seams



Application: AutoStitch

TECHNISCHE UNIVERSITÄT DARMSTADT

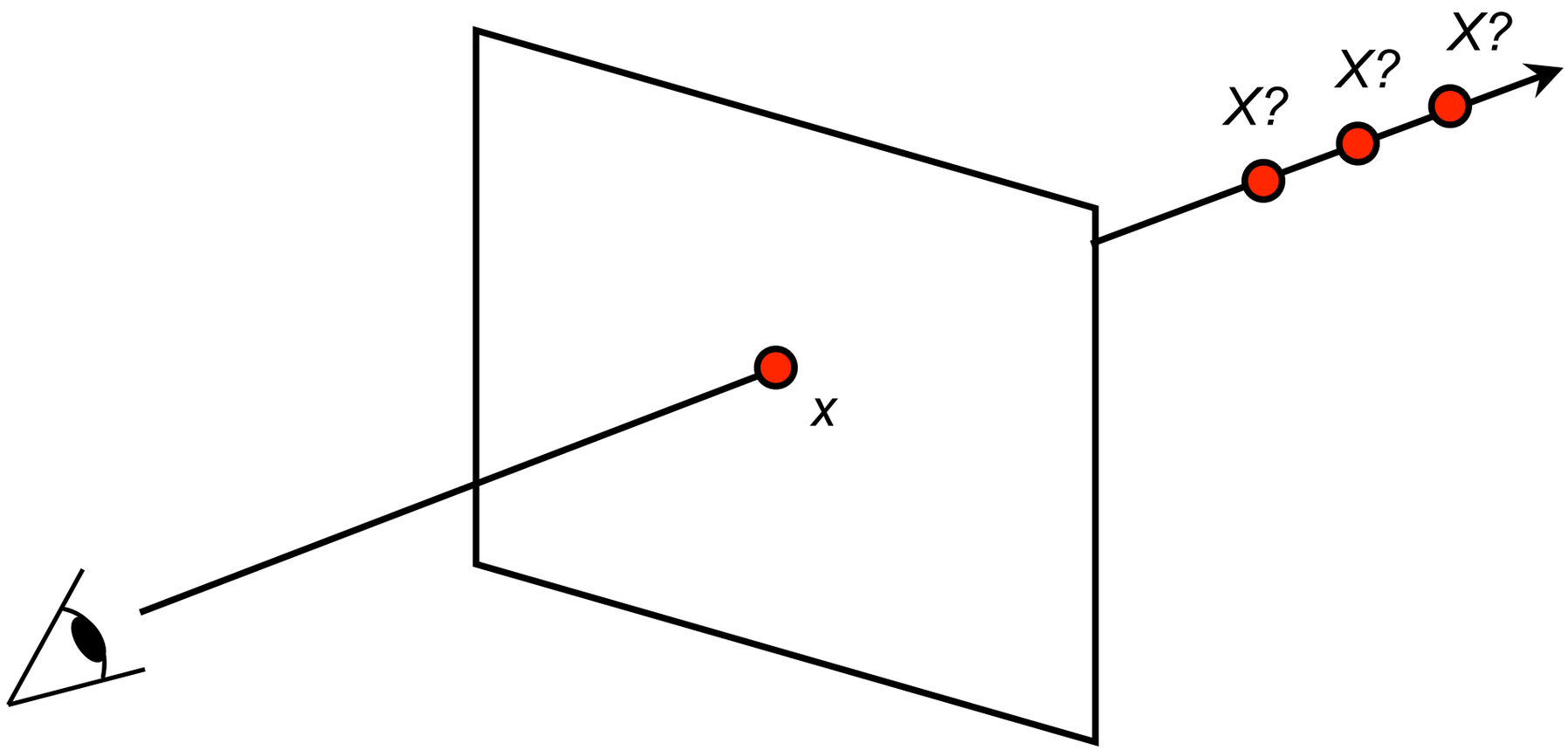
- ... automatically create a panorama
  - find images related by a homography
  - extract relative rotation
  - remap to a sphere
  - blend colors along seams

Application: AutoStitch

TECHNISCHE UNIVERSITÄT DARMSTADT

- ... automatically create a panorama
  - find images related by a homography
  - extract relative rotation
  - remap to a sphere
  - blend colors along seams

Our goal: Recovery of 3D structure



Our goal: Recovery of 3D structure

TECHNISCHE UNIVERSITÄT DARMSTADT

- Recovery of depth from one image is inherently ambiguous

Our goal: Recovery of 3D structure

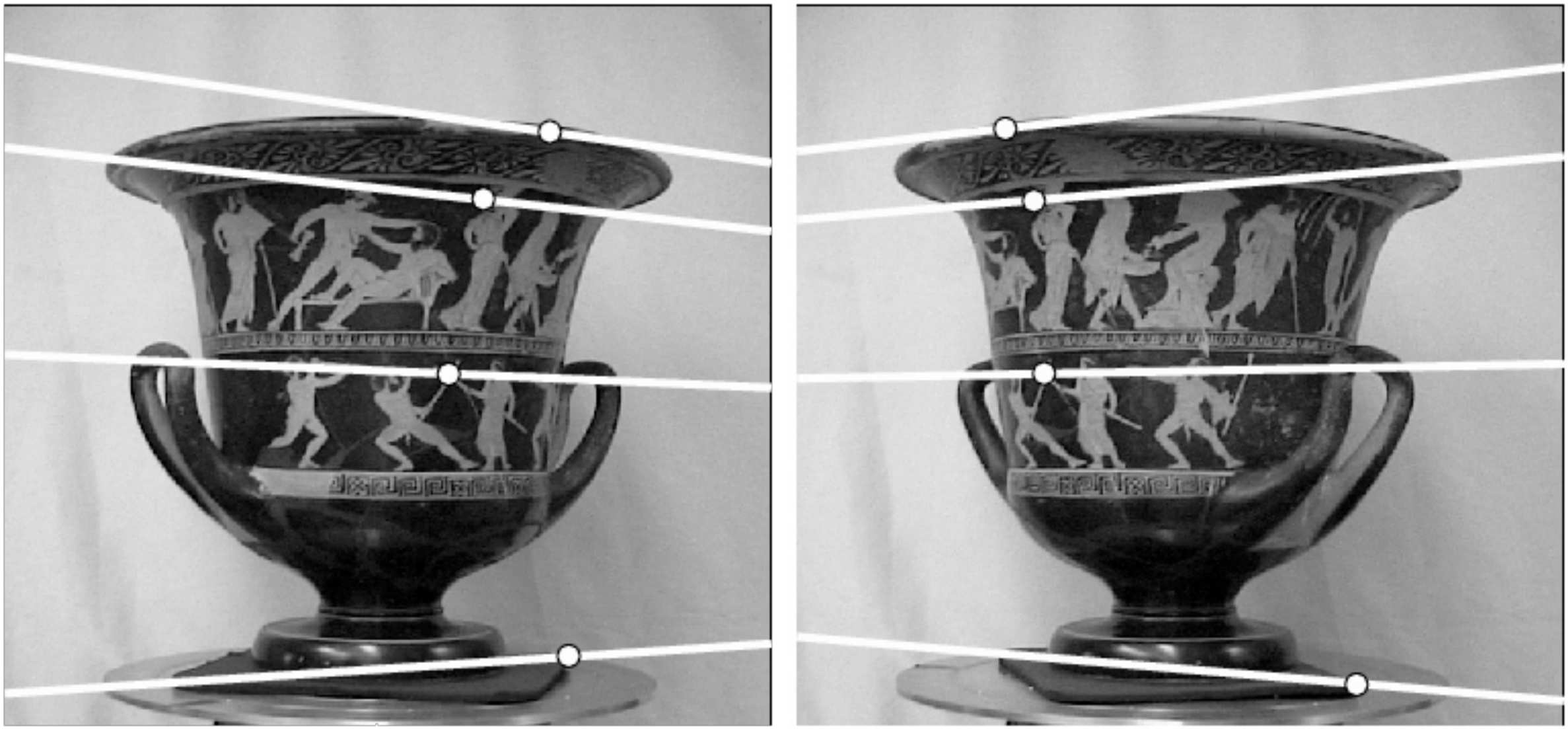
TECHNISCHE UNIVERSITÄT DARMSTADT

- Recovery of depth from one image is inherently ambiguous

Our goal: Recovery of 3D structure

TECHNISCHE UNIVERSITÄT DARMSTADT

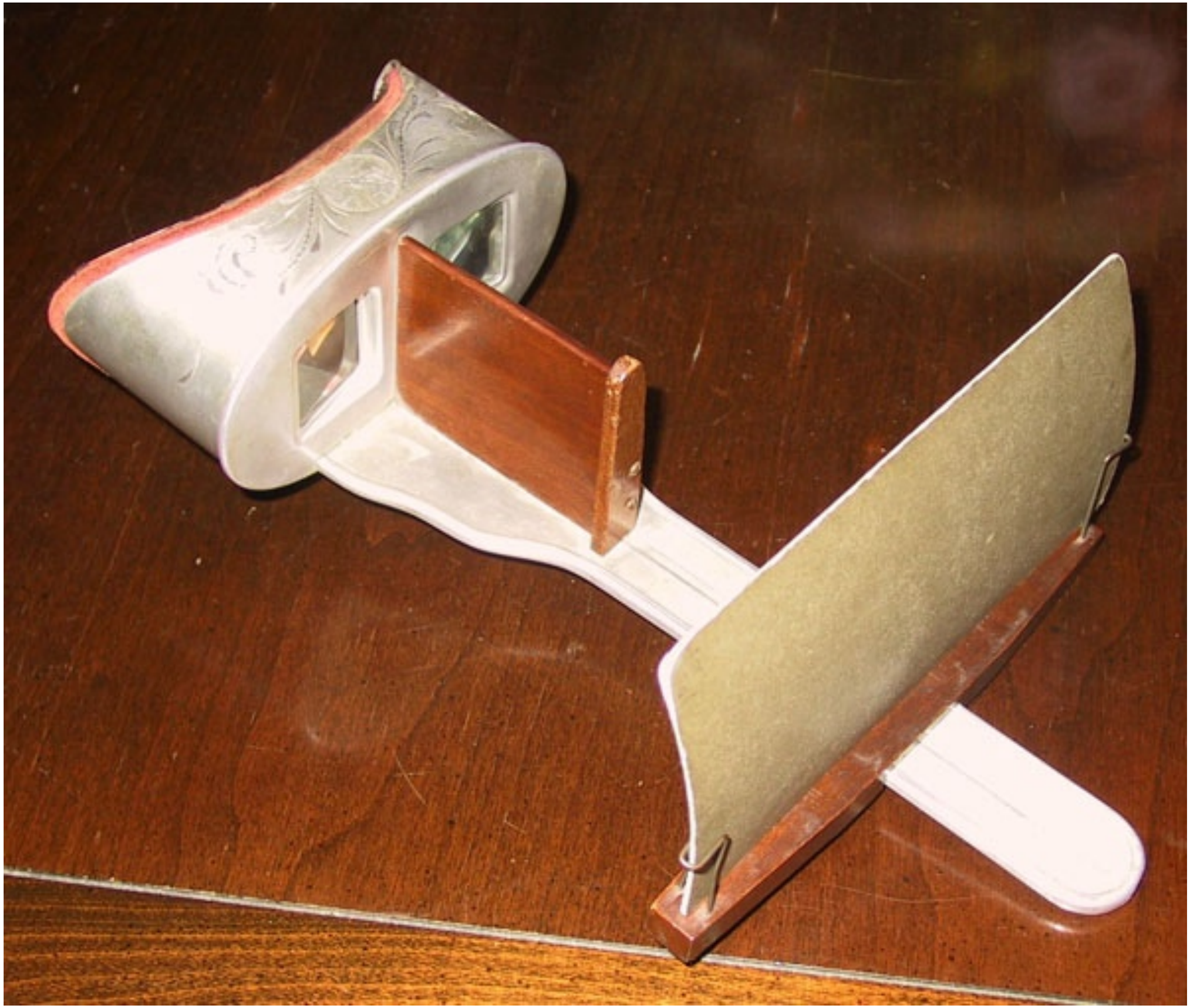
- Next: geometric methods, that rely on two views



What is Stereo (Vision)?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Stereo vision, stereopsis, or short stereo is the perception or measurement of depth from two projections.
  - Was first described in technical detail by Charles Wheatstone in the 1830s with the invention of the stereoscope.
  - The human visual system heavily relies on stereo vision:
    - Our eyes give two slightly shifted projections of the scene.
    - But only relative depth can be judged accurately by humans.

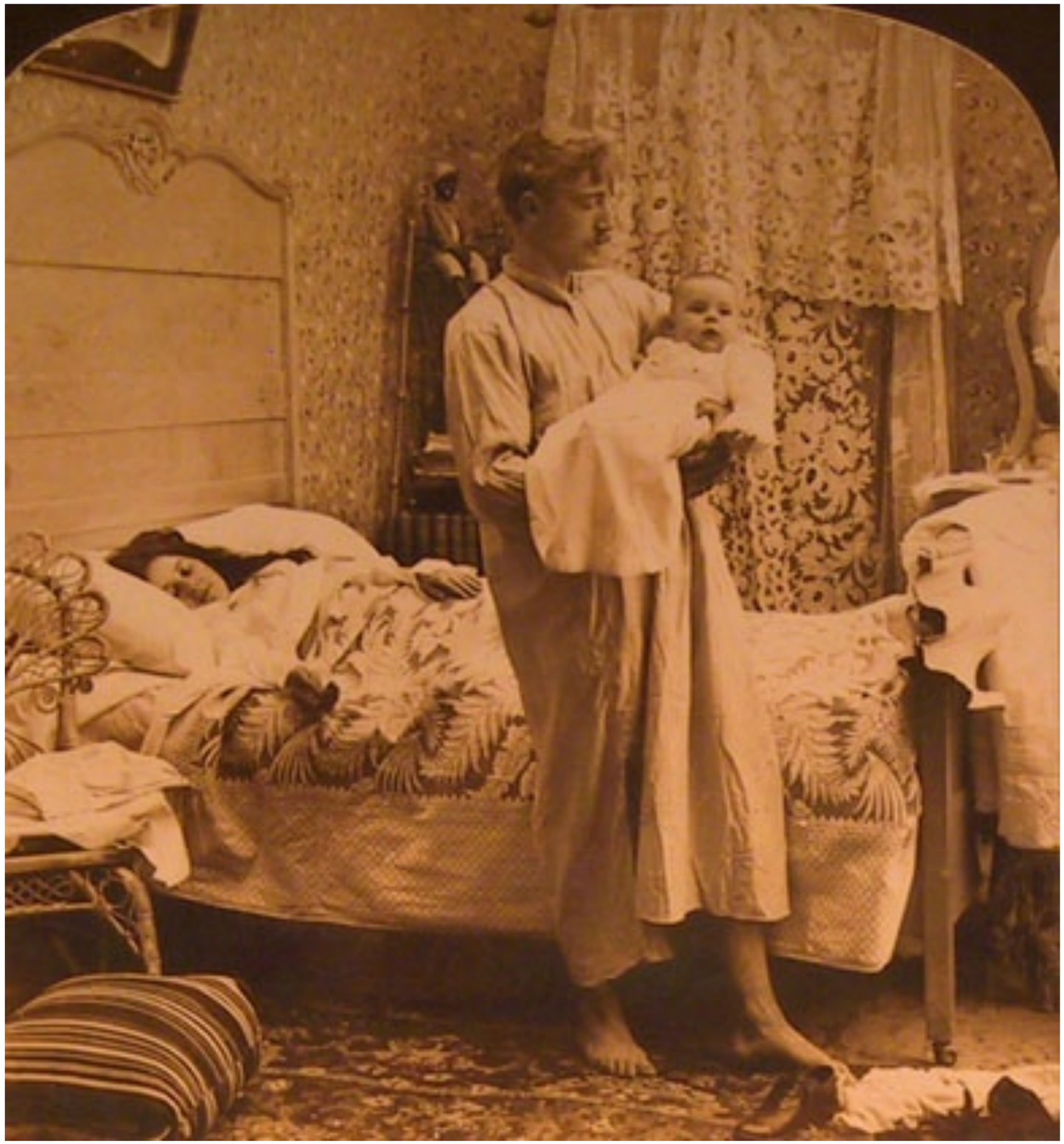


[Black]

[Black]

[Black]

Left image



[Black]

Right image

[Black]

Outline

TECHNISCHE UNIVERSITÄT DARMSTADT

- Part 2: two-view geometry

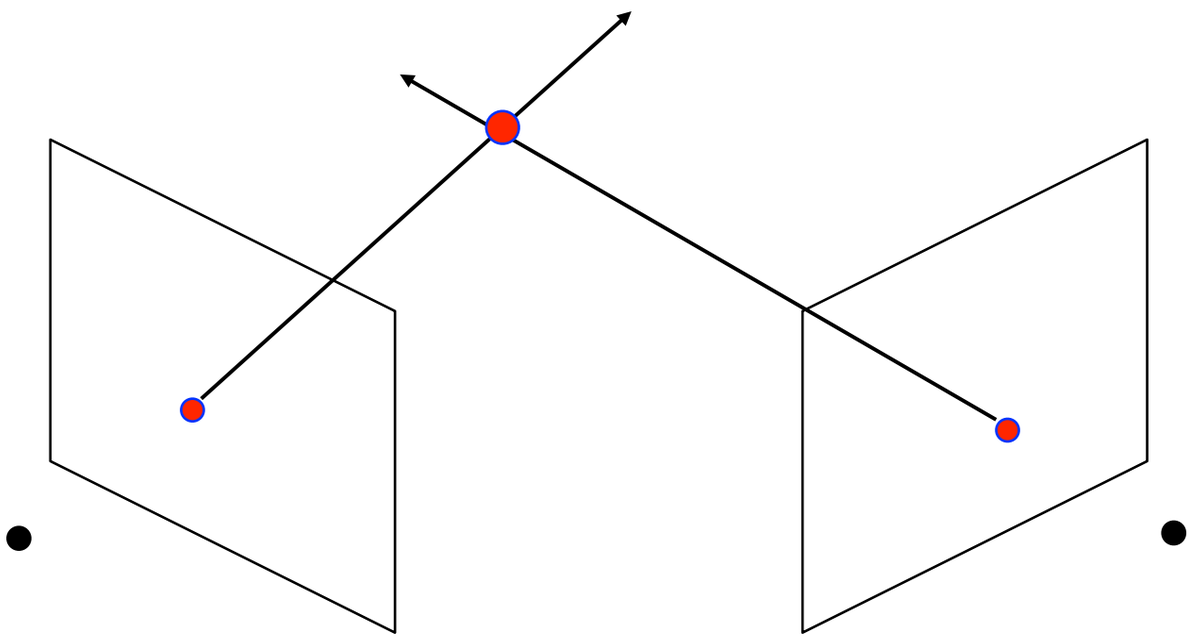
- Triangulation
  - 3D point reconstruction by ray intersection
    - simple geometric method
    - linear method
    - non-linear geometric method

- Application: robot SLAM

Depth from Stereo

TECHNISCHE UNIVERSITÄT DARMSTADT

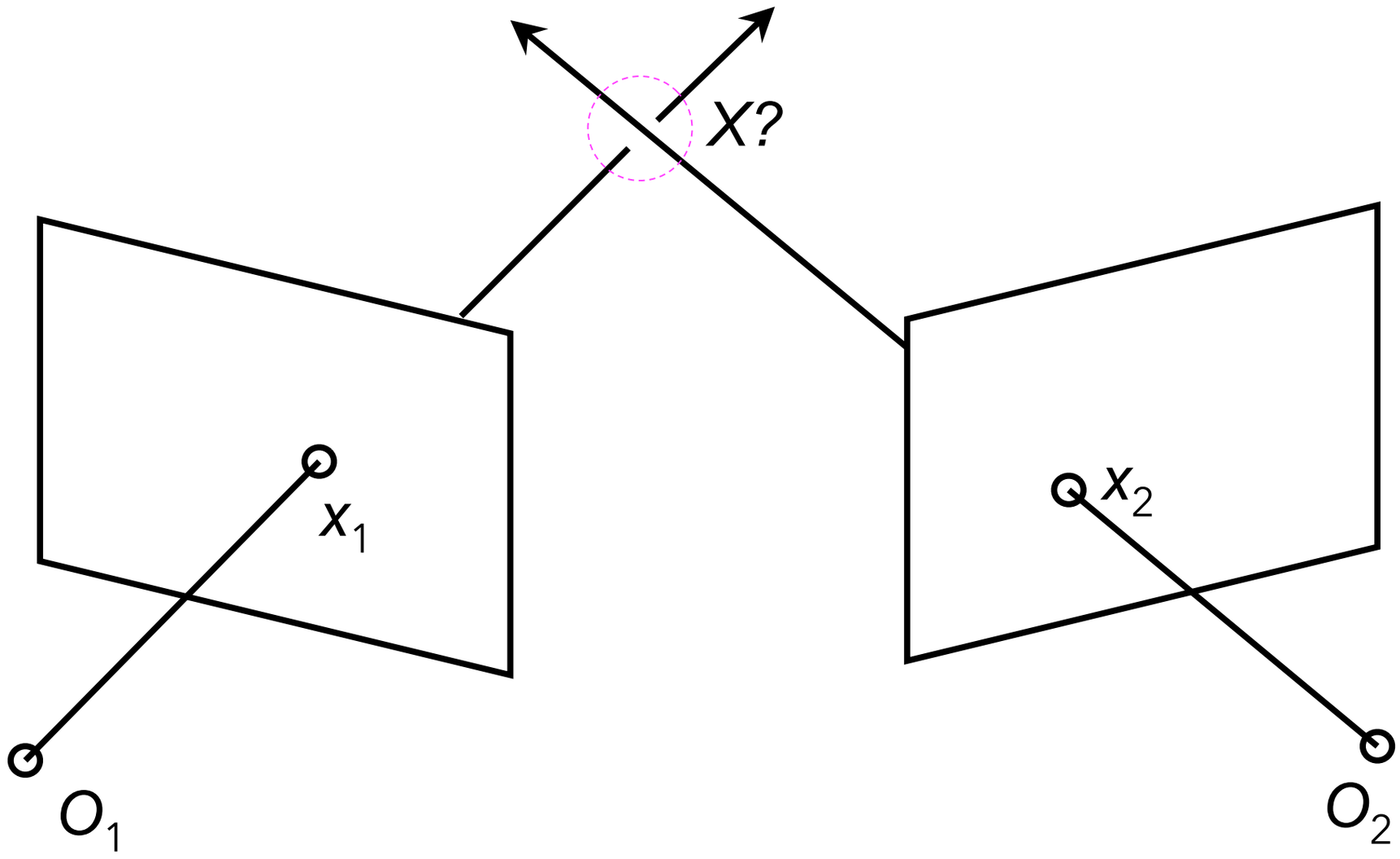
- Triangulation
  - intersect two rays originating from the same point in the scene
  - Requires correspondences (knowledge which pixels are images of “the same point”)
  - requires camera pose (in order to construct the 3D rays)



Triangulation

TECHNISCHE UNIVERSITÄT DARMSTADT

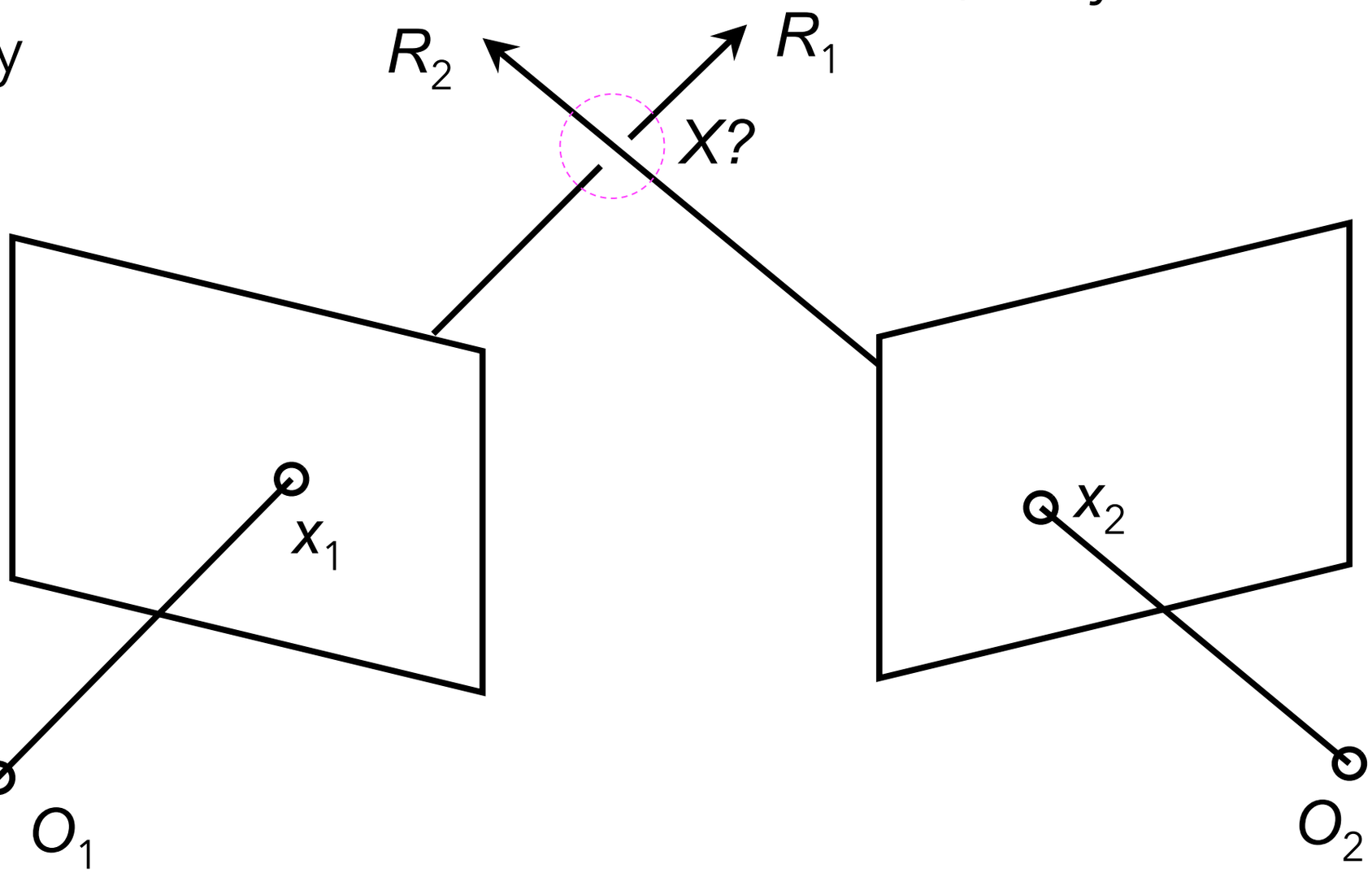
- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



Triangulation

TECHNISCHE UNIVERSITÄT DARMSTADT

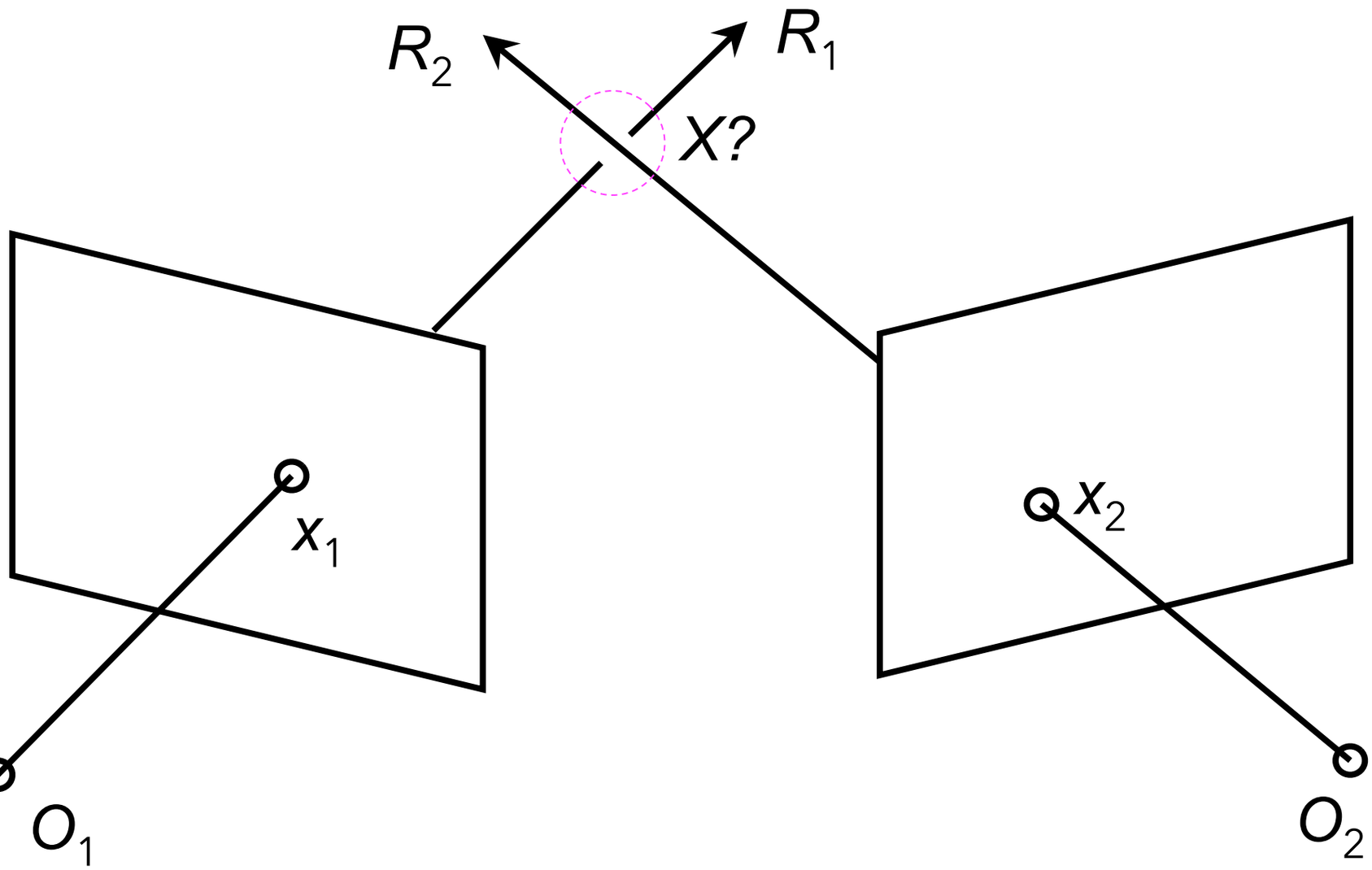
- We want to intersect the two viewing rays corresponding to x_1 and x_2
- Because of noise and numerical errors, they do not meet exactly



Triangulation

TECHNISCHE UNIVERSITÄT DARMSTADT

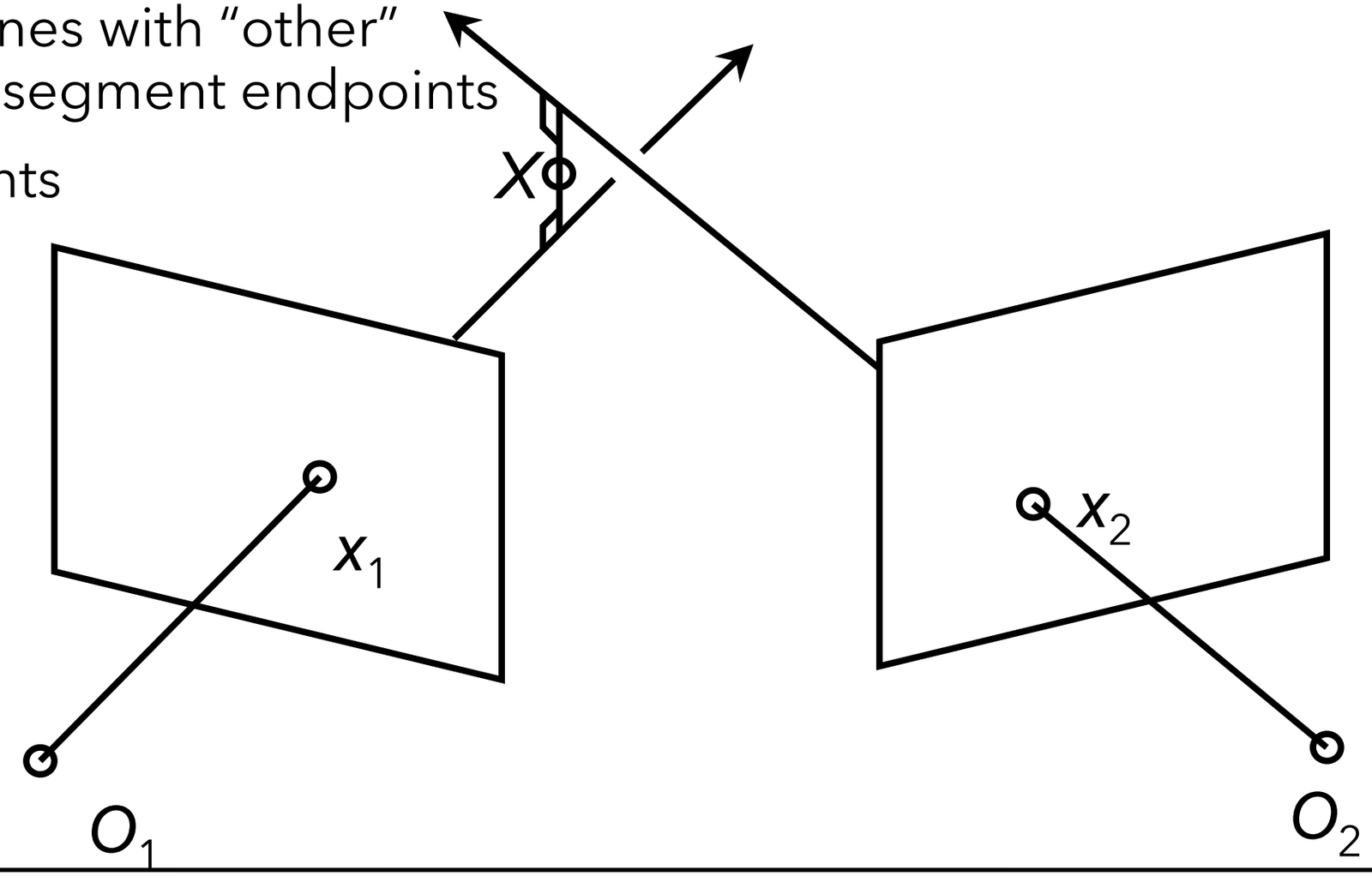
- Ideas?



Triangulation: geometric midpoint

TECHNISCHE UNIVERSITÄT DARMSTADT

- Find shortest segment connecting the two viewing rays and let X be the midpoint of that segment
  - find the segment direction (normal to both rays)
  - construct 2 planes each containing the segment and one ray
  - intersect planes with “other” rays to yield segment endpoints
  - average points

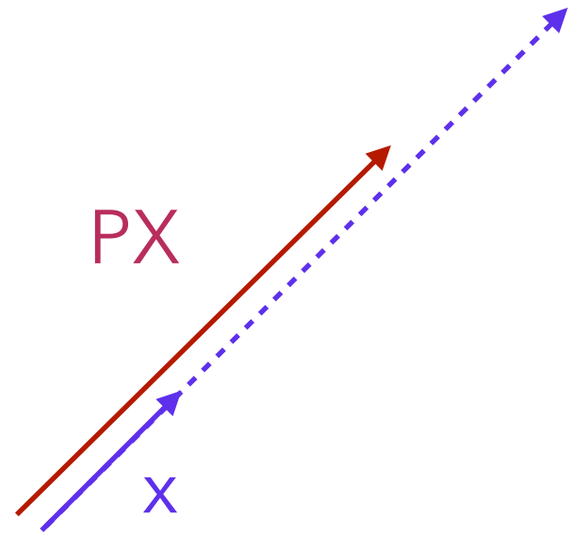


Triangulation: Linear approach

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\lambda_1 x_1 = P_1 X \qquad x_1 \times P_1 X = 0$$

$$\lambda_2 x_2 = P_2 X \qquad x_2 \times P_2 X = 0$$



- Two independent equations per image point for three unknown entries of X
- ... again, a homogeneous linear equation system
- ... again, solve with SVD
- note: directly generalizes to >2 views, just stack more equations

Side note

TECHNISCHE UNIVERSITÄT DARMSTADT

- An algebraic trick (useful here and later):
- how to get there without expanding the cross-product and re-ordering: cross-product can be written as a matrix-vector multiplication

$$x \times PX = 0$$

$$a \times b = 0$$

$$[a_\times]b = 0$$

$$[a_\times] = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

$$\boxed{[x_\times]P}X = 0$$

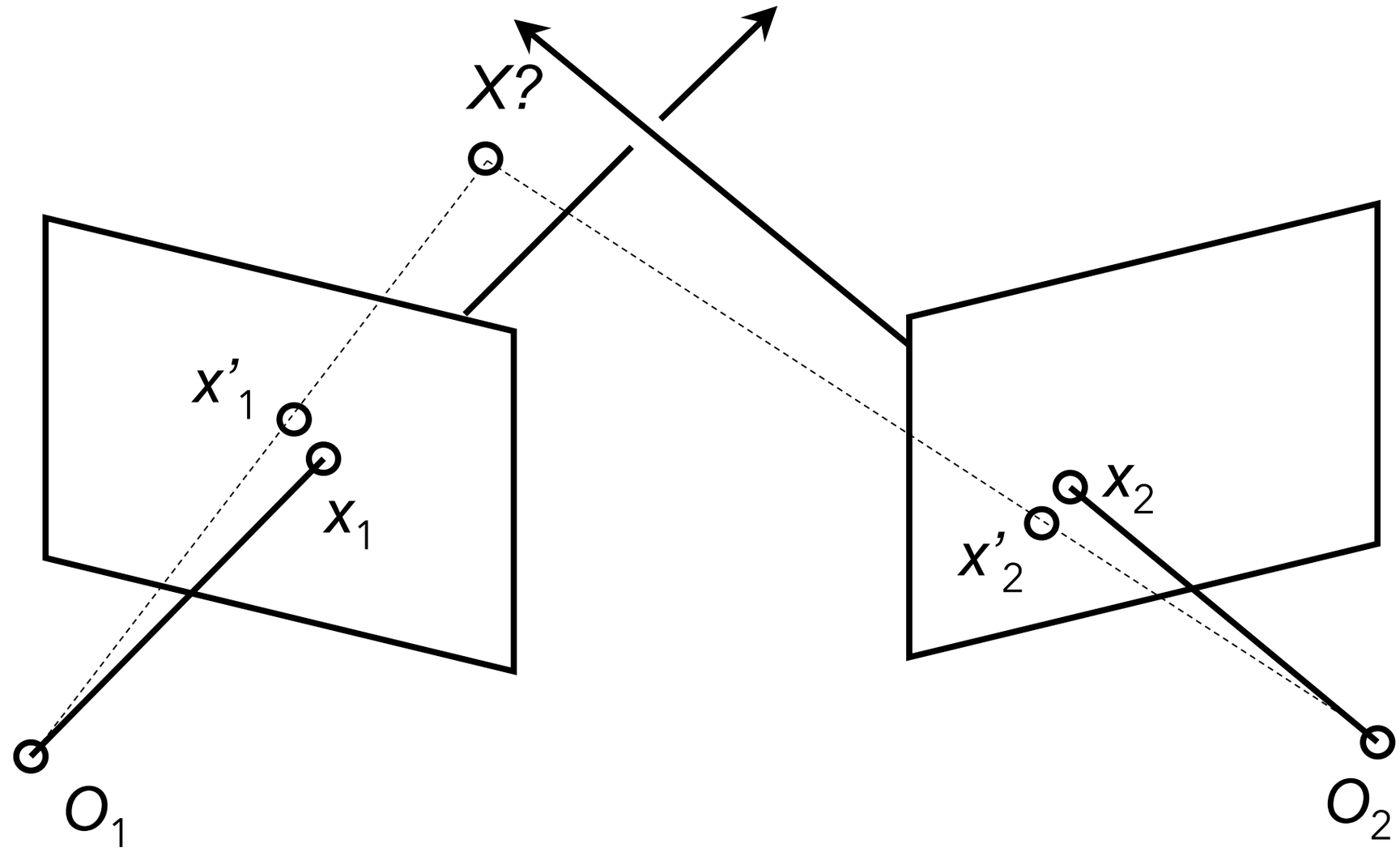
a matrix

Triangulation: non-linear approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- Find X that minimizes the (squared) reprojection error:

$$d^2(x_1, P_1X) + d^2(x_2, P_2X)$$



Triangulation: non-linear approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- Find X that minimizes the (squared) reprojection error:

$$d^2(x_1, P_1X) + d^2(x_2, P_2X)$$

  - The most accurate method, but is more complex than the other two
  - With 2 cameras
    - find roots of a 6th degree polynomial (Hartley & Zisserman, chapter 12)
  - With >2 cameras
    - initialize with linear estimate
    - optimize with iterative methods (Gauss-Newton, Levenberg-Marquardt - HZ, appendix 6)

Outline

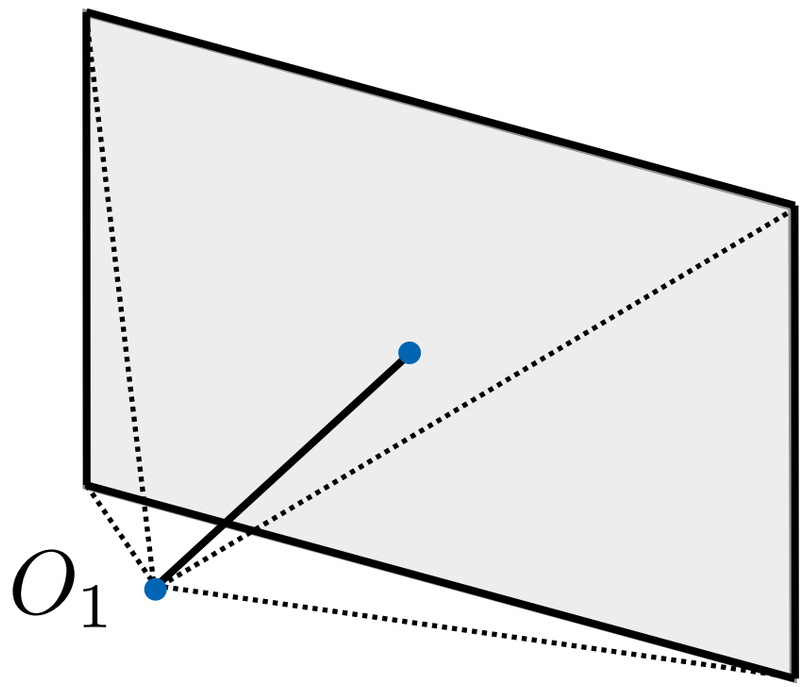
TECHNISCHE UNIVERSITÄT DARMSTADT

- Goal: find the relation between two different views of the same scene
  - recover camera placement
  - matching and triangulation for multiple views
  - scene modeling only from images
- Geometrical relations
  - epipolar constraint
  - epipolar geometry of 2 views
- Algebraic relations
  - calibrated cameras: essential matrix
  - uncalibrated cameras: fundamental matrix
  - the normalized 8-point algorithm
  - relation to camera matrices

Epipolar Geometry

TECHNISCHE UNIVERSITÄT DARMSTADT

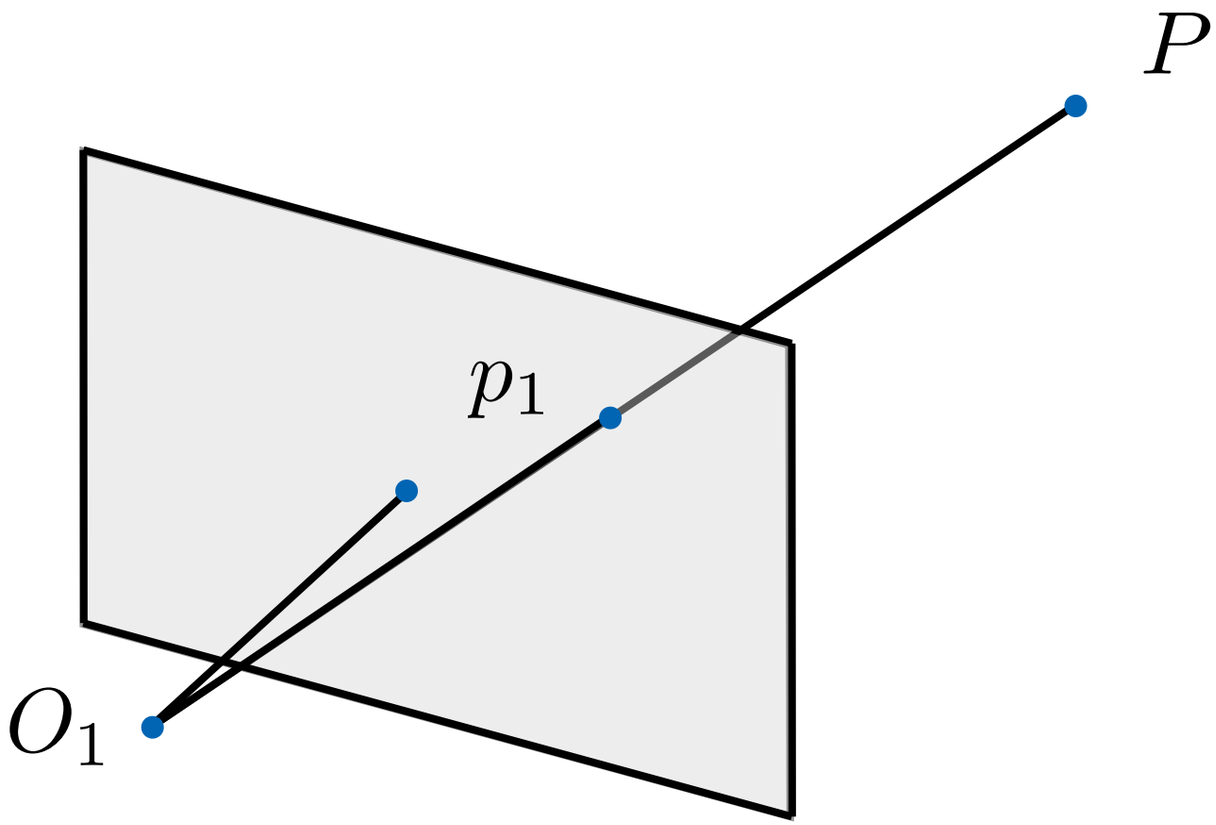
P point in the world



Camera 1

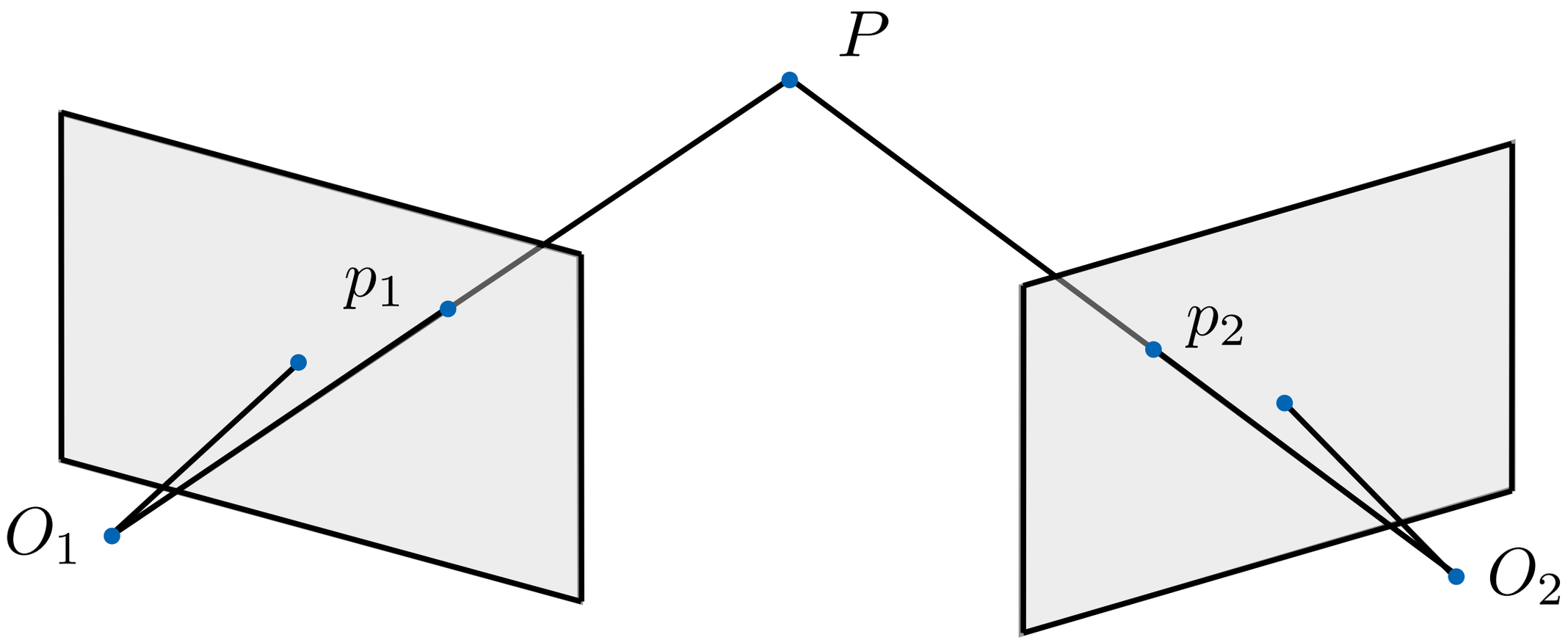
Epipolar Geometry

TECHNISCHE UNIVERSITÄT DARMSTADT



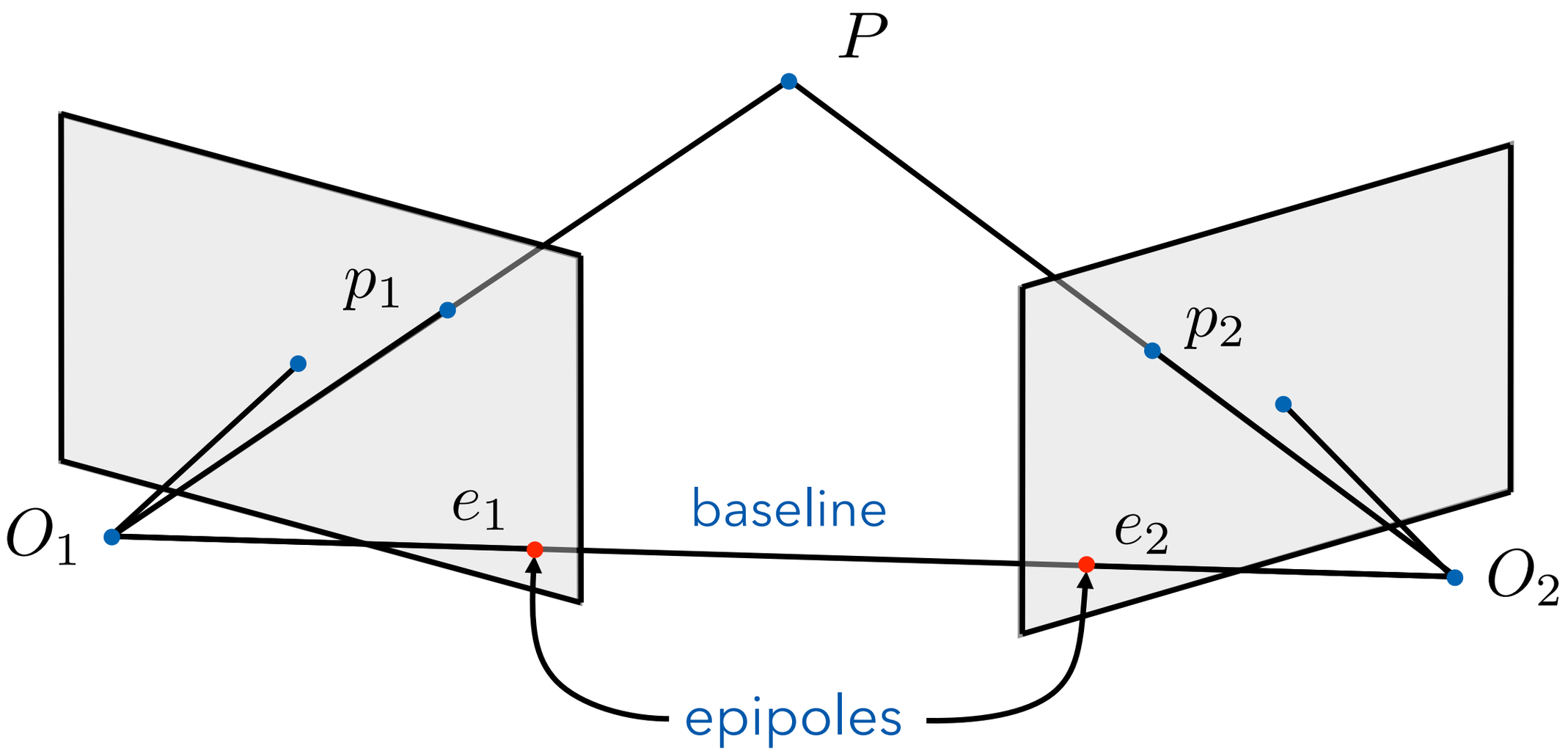
Epipolar Geometry

TECHNISCHE UNIVERSITÄT DARMSTADT



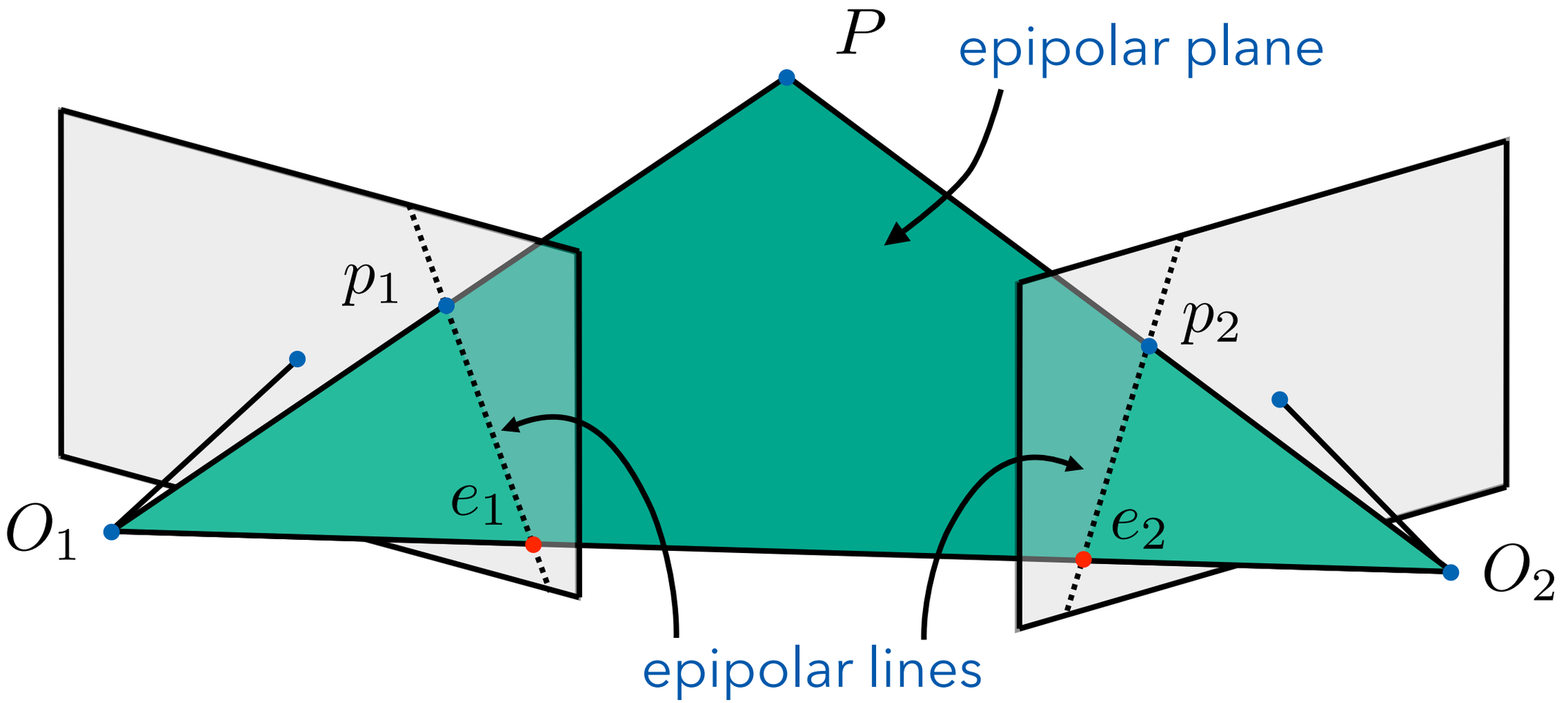
Epipolar Geometry

TECHNISCHE UNIVERSITÄT DARMSTADT



- Epipole: Image location of the optical center of the other camera.
  - Can be outside of the visible area.

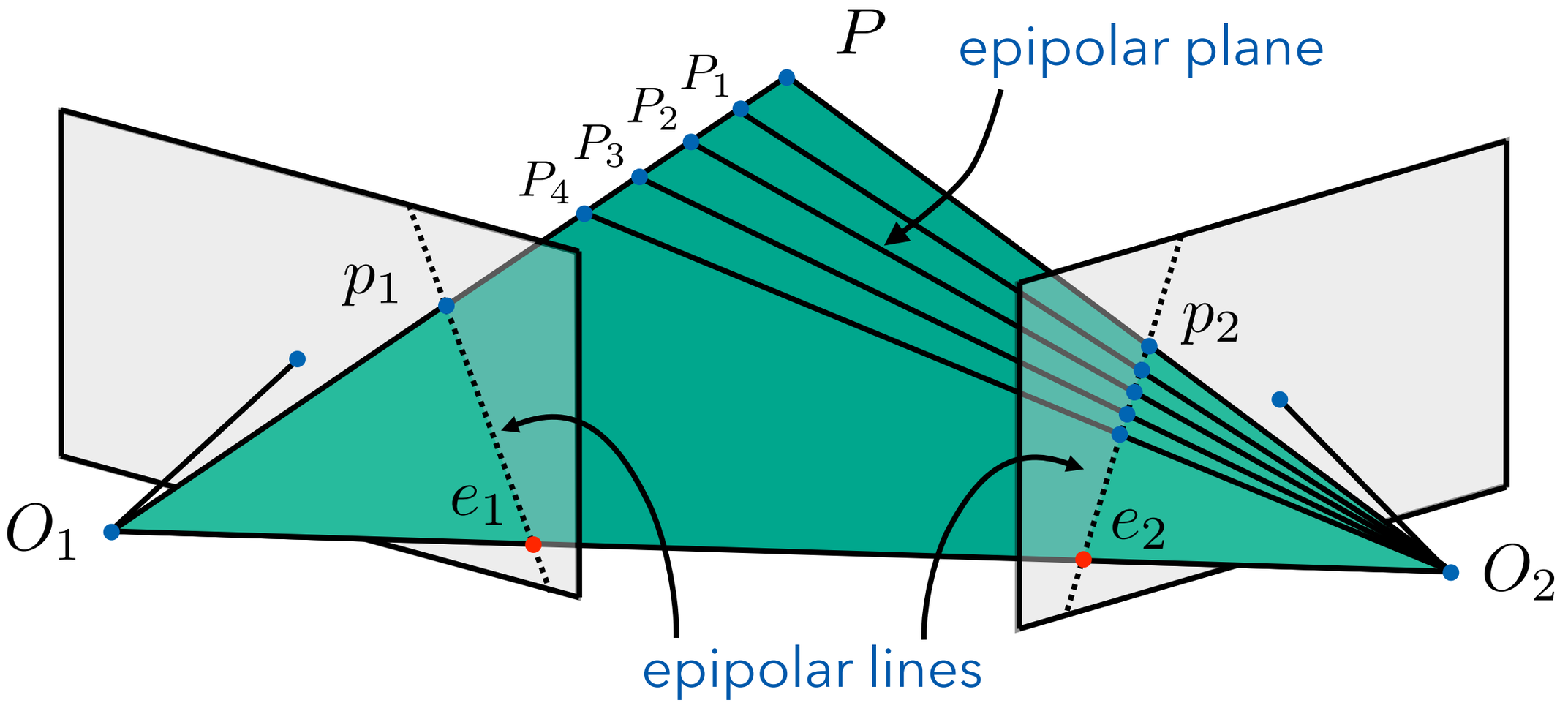
Epipolar Geometry



- Epipolar plane: Plane through both camera centers and world point.

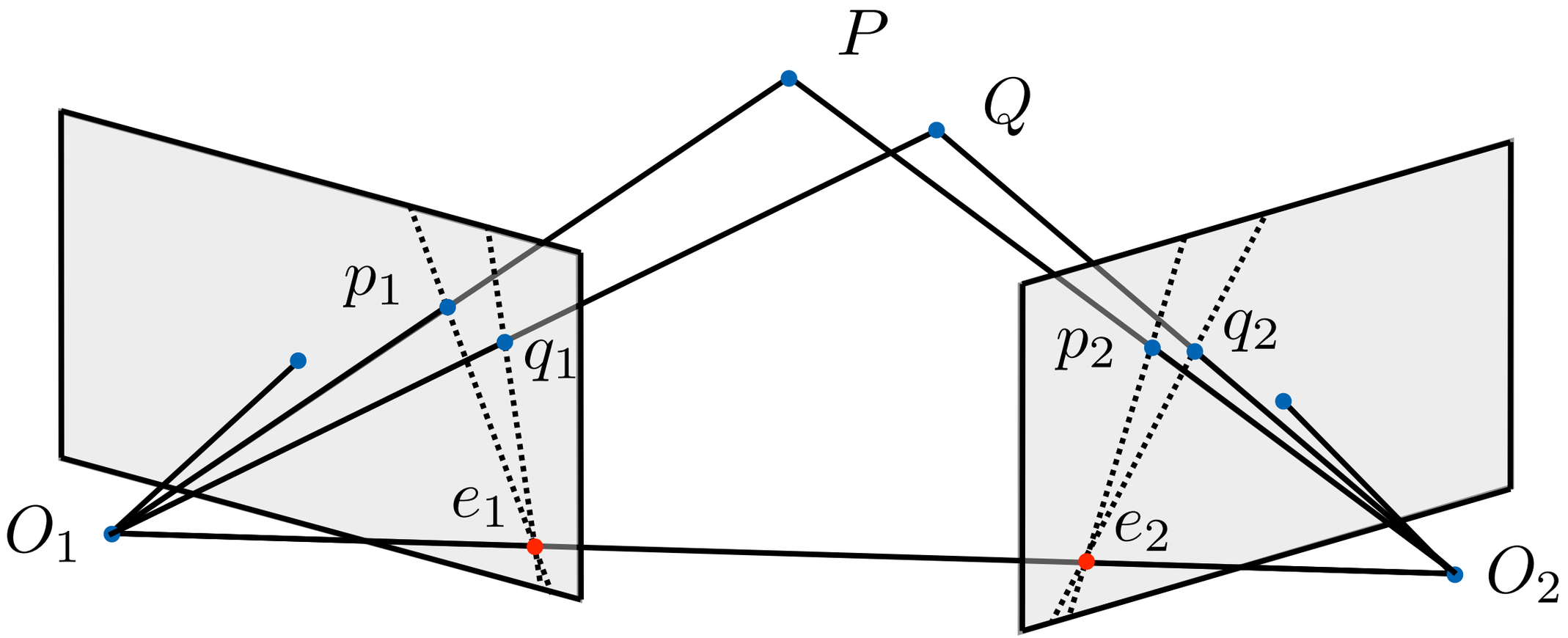
Epipolar Geometry

TECHNISCHE UNIVERSITÄT DARMSTADT



- Epipolar line: Constrains the location where a particular feature from one view can be found in the other.
  - Here: Possible matches for p_1

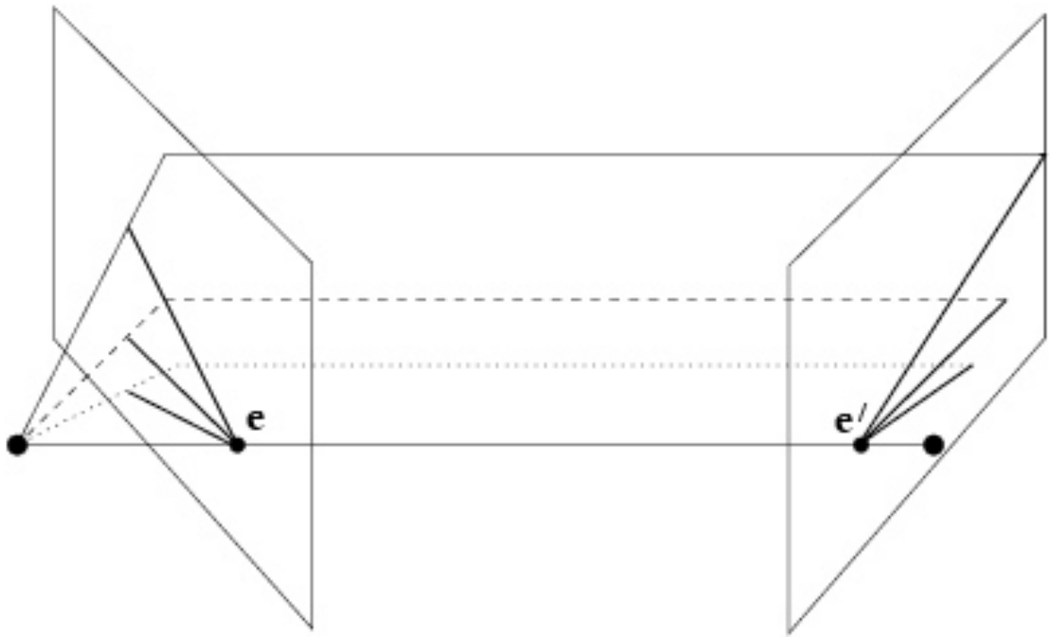
Epipolar Geometry

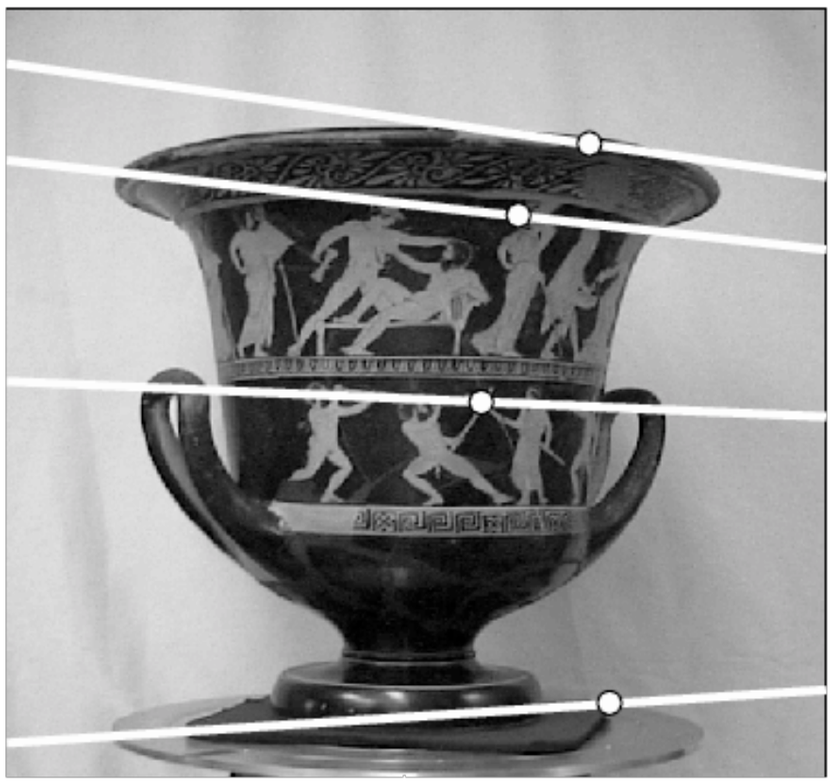


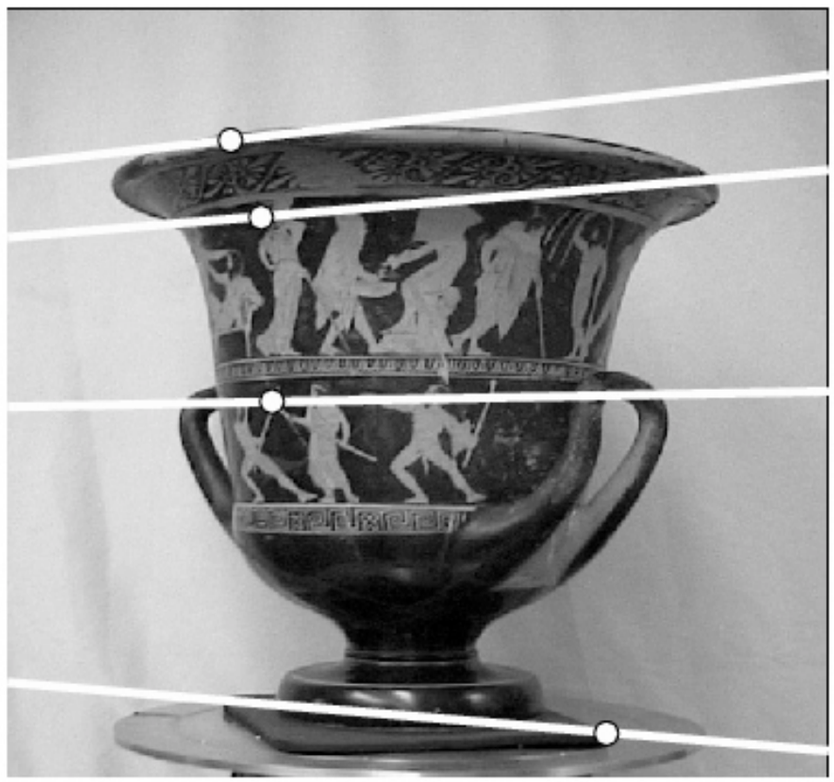
- Epipolar lines:
  - Intersect at the epipoles.
  - In general not parallel.

Example: Converging Cameras

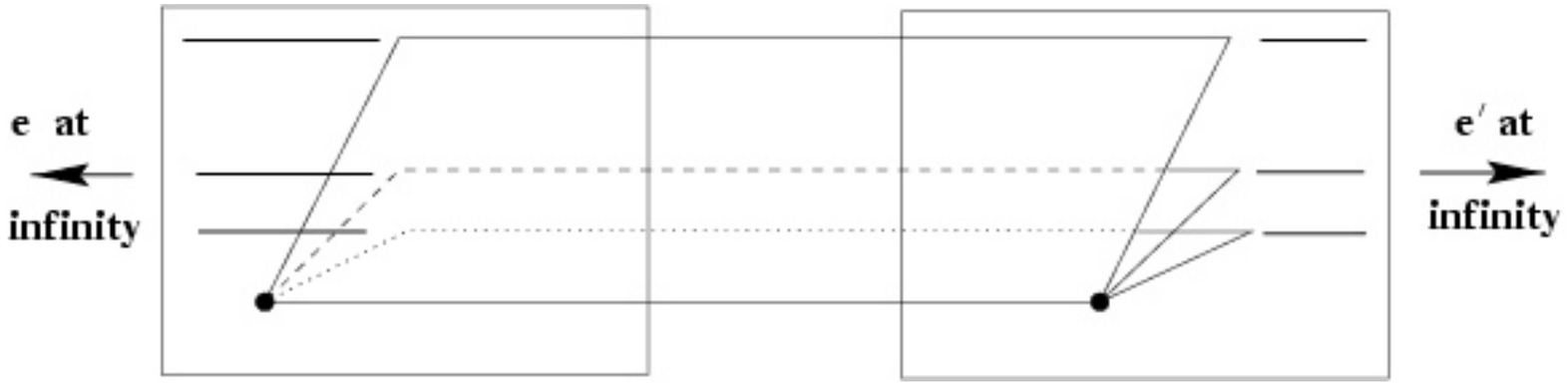
TECHNISCHE UNIVERSITÄT DARMSTADT





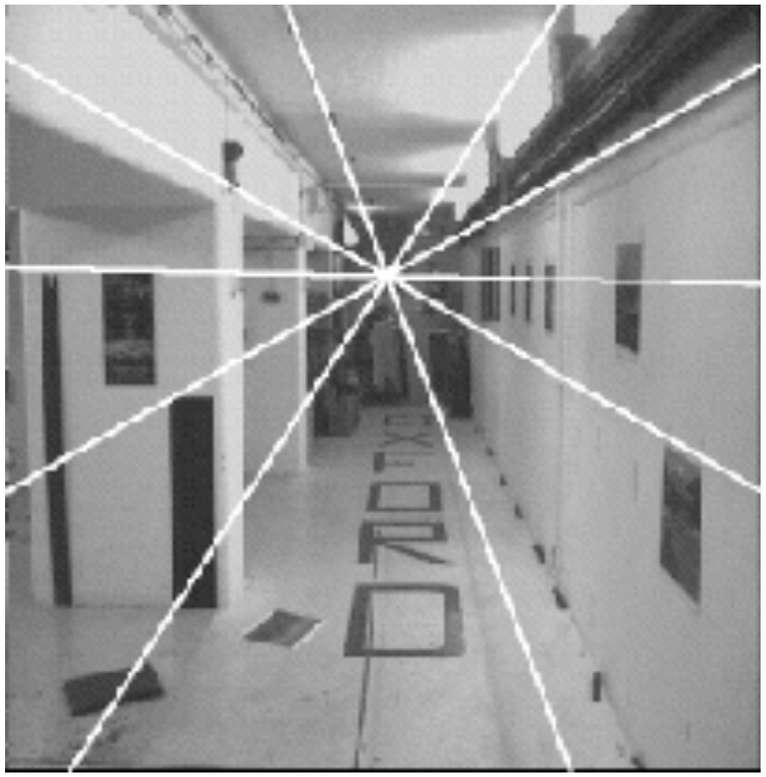


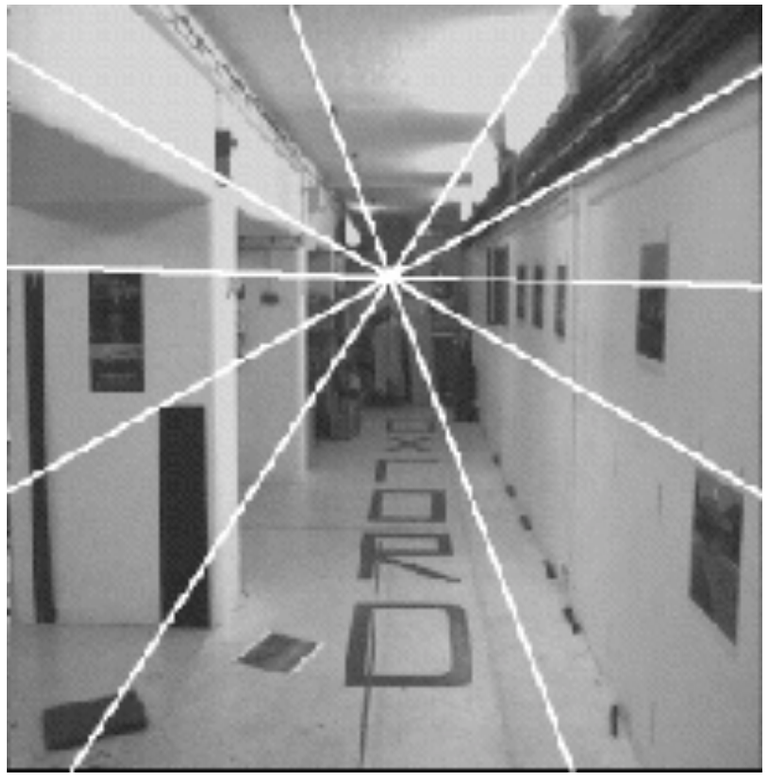
Example: Motion Parallel to Image Plane

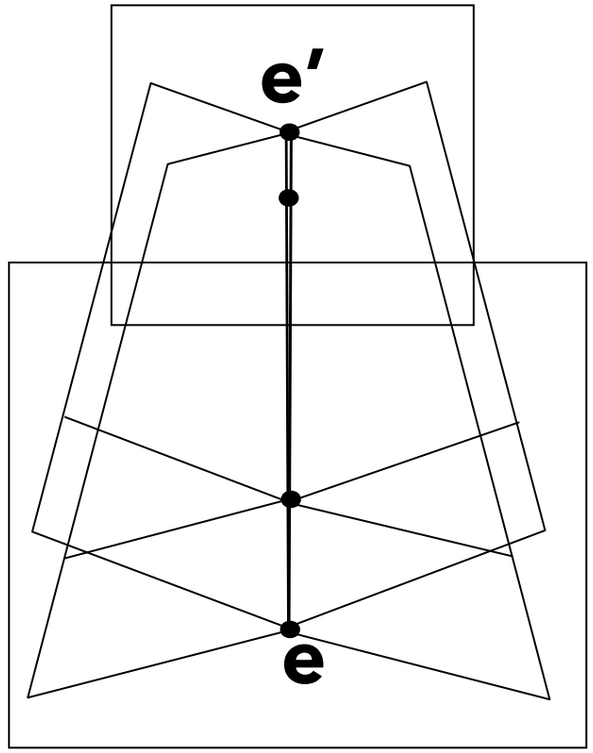


Example: Forward Motion

TECHNISCHE UNIVERSITÄT DARMSTADT



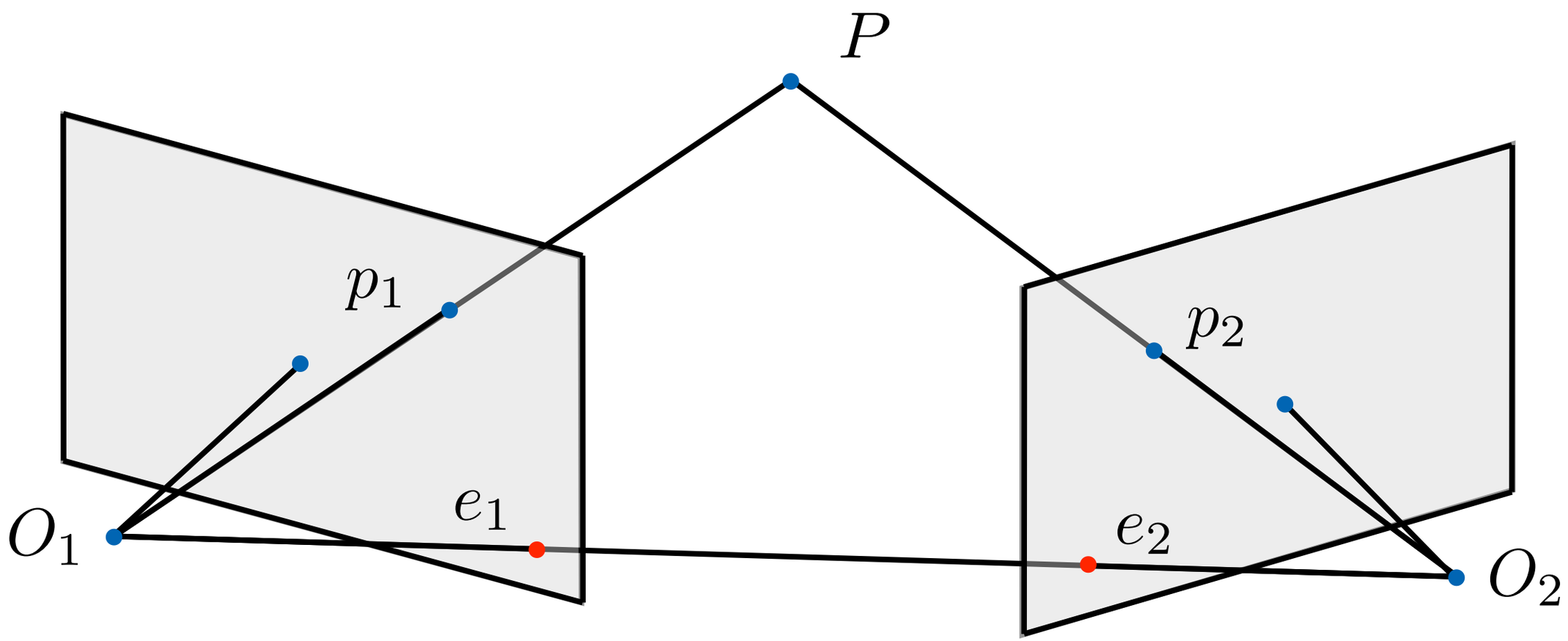




- Epipole has same coordinate in both images
- Point move along lines radiating from epipole
  - “focus of expansion”

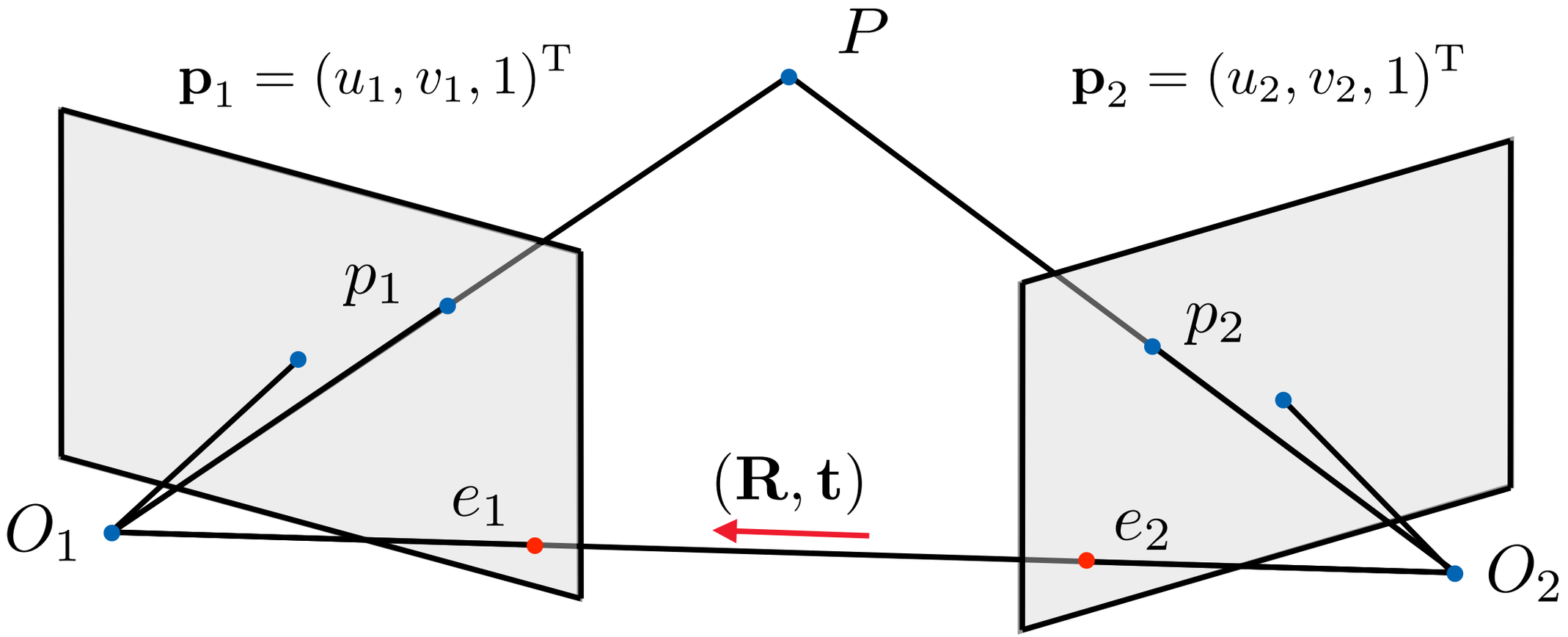
Epipolar Geometry

TECHNISCHE UNIVERSITÄT DARMSTADT



- Mathematically: $\overrightarrow{O_1P},\ \overrightarrow{O_2P},\ \overrightarrow{O_1O_2}$ are coplanar.
- Equivalently: $\overrightarrow{O_1p_1},\ \overrightarrow{O_2p_2},\ \overrightarrow{O_1O_2}$ are coplanar.
- In other terms: $\overrightarrow{O_1p_1} \cdot \left[\overrightarrow{O_1O_2} \times \overrightarrow{O_2p_2}\right] = 0$

Epipolar Geometry



- Rewrite in camera coordinate system of camera 1:

$$\overrightarrow{O_1 p_1} \cdot \left[\overrightarrow{O_1 O_2} \times \overrightarrow{O_2 p_2} \right] = 0$$

$$\Leftrightarrow \quad \mathbf{p}_1^{\mathrm{T}} \left[\mathbf{t} \times (\mathbf{R}\mathbf{p}_2) \right] = 0$$

Epipolar Geometry

TECHNISCHE UNIVERSITÄT DARMSTADT

- Epipolar constraint: $\mathbf{p}_1^{\mathrm{T}} \left[\mathbf{t} \times (\mathbf{R}\mathbf{p}_2) \right] = 0$
- Express cross-product as matrix: $\mathbf{t} \times \mathbf{v} = [\mathbf{t}]_\times \mathbf{v}$
- **Essential matrix:** $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$
  - captures the relation of two views [Longuet-Higgins, 1981]
- Epipolar constraint with essential matrix:

$$\begin{aligned} 0 &= \mathbf{p}_1^{\mathrm{T}} \left[\mathbf{t} \times (\mathbf{R}\mathbf{p}_2) \right] \\ &= \mathbf{p}_1^{\mathrm{T}} \left[([\mathbf{t}]_\times \mathbf{R}) \mathbf{p}_2 \right] \\ &= \mathbf{p}_1^{\mathrm{T}} \mathbf{E} \mathbf{p}_2 \end{aligned}$$

Epipolar Geometry

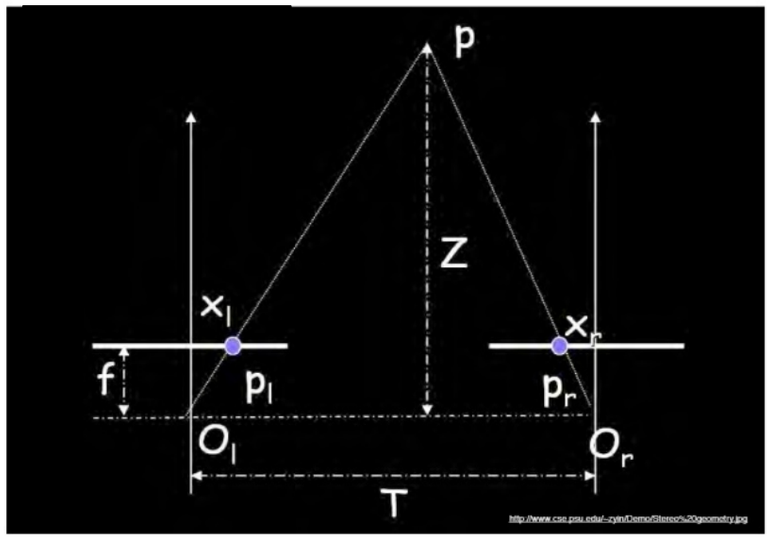
TECHNISCHE UNIVERSITÄT DARMSTADT

- Two views of the same 3D point must satisfy $\mathbf{p}_1^{\mathrm{T}}\mathbf{E}\mathbf{p}_2 = 0$
- Other important properties...
- Epipolar lines: $\mathbf{l}_1 = \mathbf{E}\mathbf{p}_2$ and $\mathbf{l}_2 = \mathbf{E}^{\mathrm{T}}\mathbf{p}_1$
- Epipoles are the (left/right) null-space of the essential matrix:

$$\mathbf{e}_1^{\mathrm{T}}\mathbf{E} = \mathbf{E}^{\mathrm{T}}\mathbf{e_1} = 0 \qquad \mathbf{E}\mathbf{e}_2 = 0$$

- Essential matrix is singular; has rank 2.
- The two remaining eigenvalues are equal.
- 5 degrees of freedom (translation + rotation have 6, but scale is arbitrary)

Binocular Stereo: Parallel cameras



$$\mathbf{R} = \mathbf{I}$$

$$\mathbf{t} = [-b, 0, 0]$$

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & b \\ 0 & -b & 0 \end{bmatrix}$$

$$0 = \mathbf{p}_1^{\mathrm{T}} \mathbf{E} \mathbf{p}_2$$

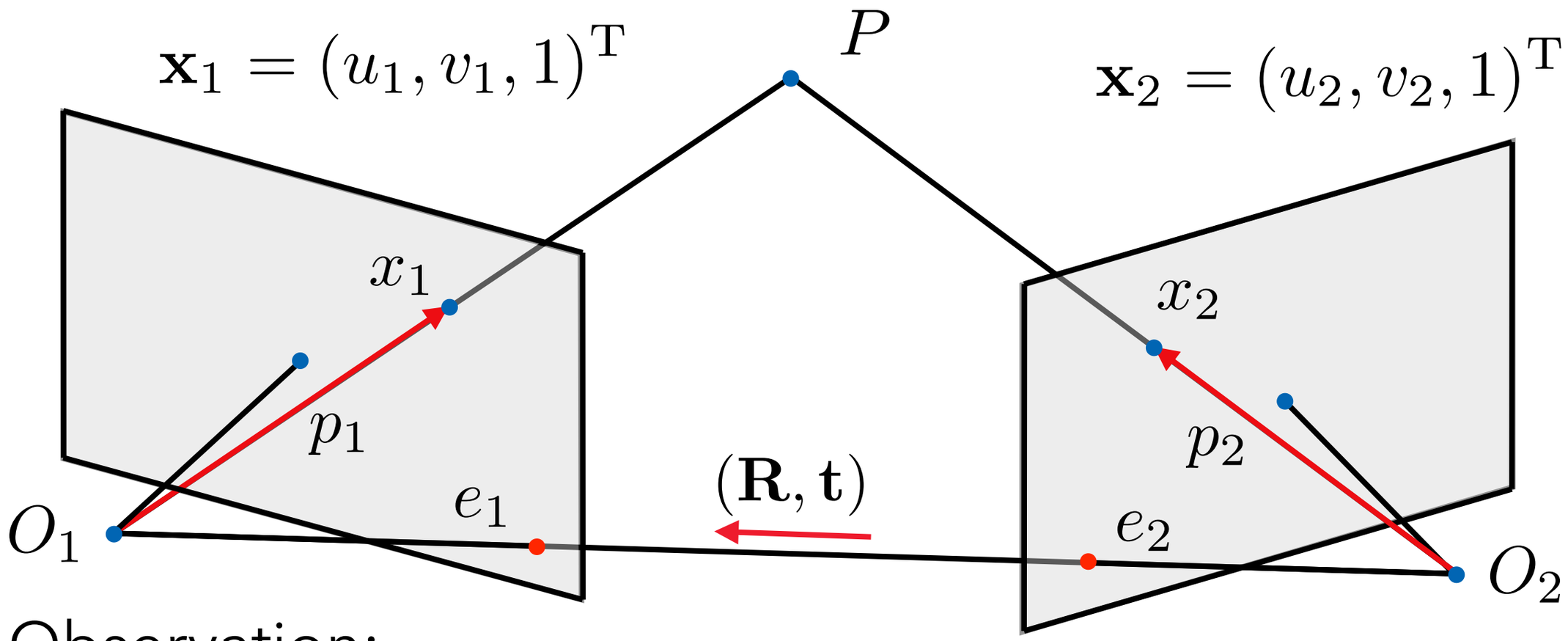
$$= [x_1, y_1, 1] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & b \\ 0 & -b & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

$$= [x_1, y_1, 1] \begin{bmatrix} 0 \\ b \\ -by_2 \end{bmatrix} = by_1 - by_2$$

$$y_1 = y_2$$

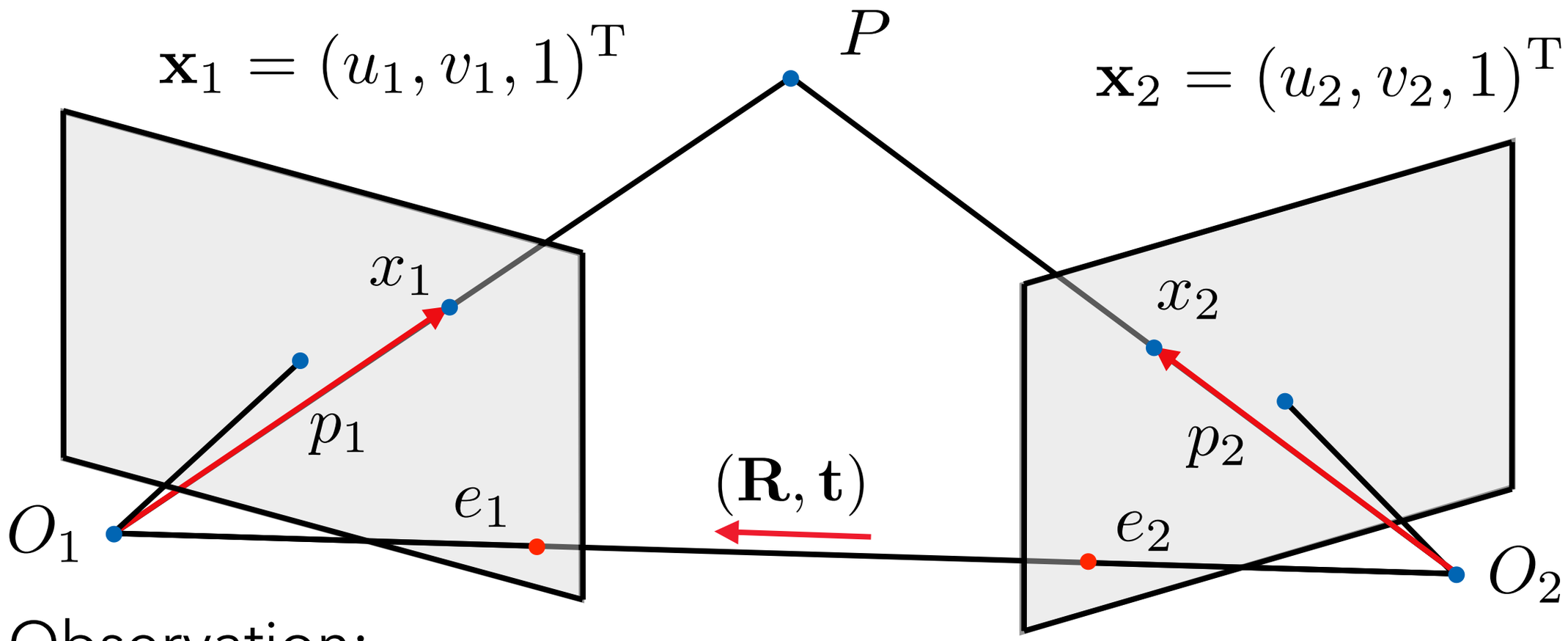
x_2 arbitrary

Uncalibrated Cameras



- Observation:
  - the epipolar constraint is derived by intersecting lines and planes
  - these lines and planes exist for any two pinhole cameras
  - if we do not know the calibration, the constraint must still hold

Uncalibrated Cameras

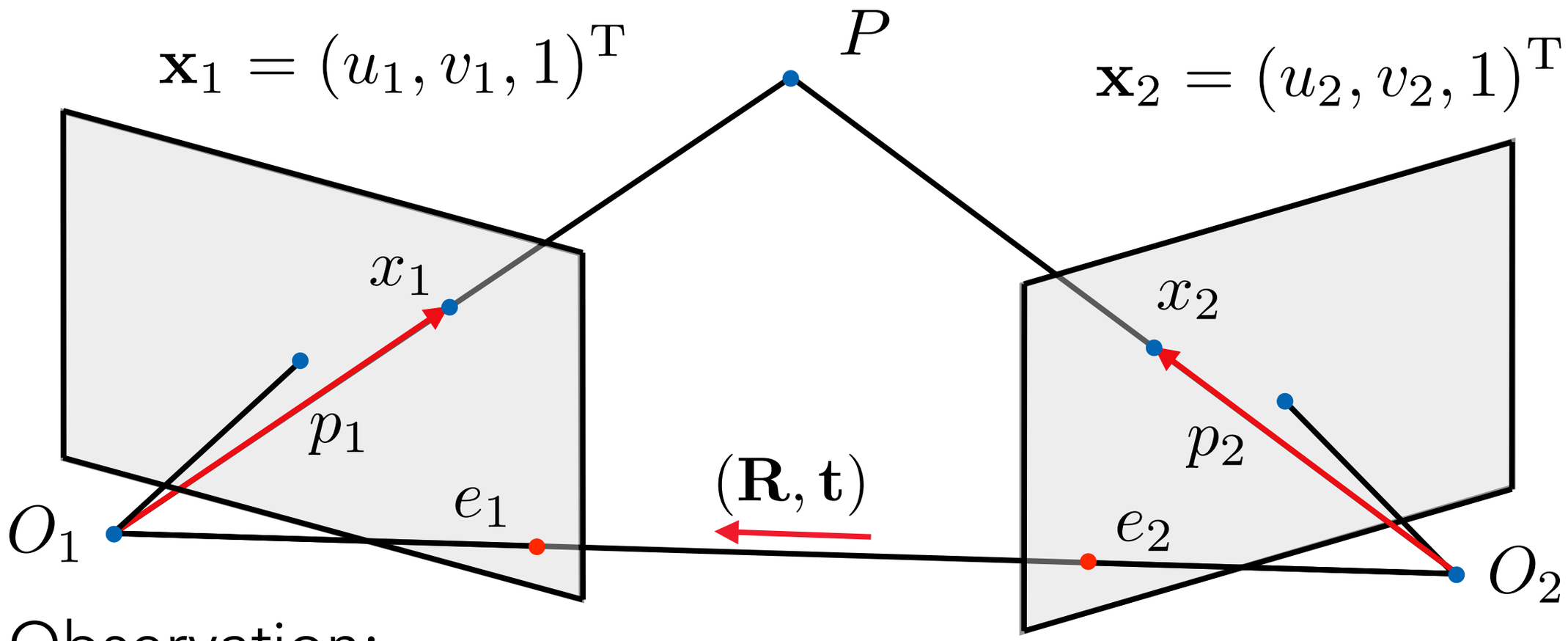


- Observation:
  - The transformation from rays $\mathbf{p}$ to image points $\mathbf{x}$ is the calibration $\mathbf{K}$
  - The transformation exists and is invertible, even if it is unknown

$$\mathbf{x}_1 = \mathbf{K}_1 \mathbf{p}_1 \qquad \mathbf{x}_2 = \mathbf{K}_2 \mathbf{p}_2$$

$$\mathbf{p}_1 = \mathbf{K}_1^{-1} \mathbf{x}_1 \qquad \mathbf{p}_2 = \mathbf{K}_2^{-1} \mathbf{x}_2$$

Uncalibrated Cameras



- Observation:
  - We silently assumed that everything was done in world coordinates
  - We can simply plug in the pixel coordinates:

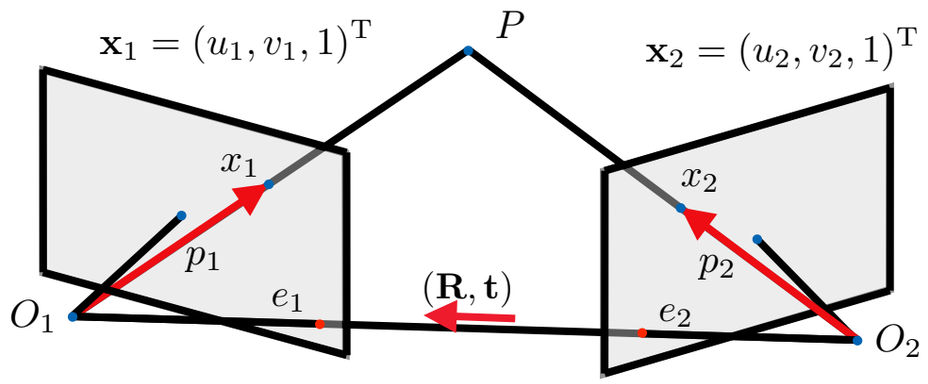
$$0 = \mathbf{p}_1^{\mathrm{T}} \mathbf{E} \mathbf{p}_2 = \mathbf{x}_1^{\mathrm{T}} \mathbf{K}_1^{-\mathrm{T}} \mathbf{E} \mathbf{K}_2^{-1} \mathbf{x}_2$$

$$0 = \mathbf{x}_1^{\mathrm{T}} \mathbf{F} \mathbf{x}_2$$

Fundamental matrix

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\mathbf{x}_1^{\mathrm{T}} \mathbf{F} \mathbf{x}_2 = 0$$



- holds for calibrated and uncalibrated cameras
- geometrically: two rays must be coplanar, independent of any specific $\mathbf{K}$, $\mathbf{R}$, $\mathbf{t}$
- $\mathbf{F}$ is called the fundamental matrix, and encapsulates the relative geometry of the two uncalibrated views [Faugeras and Luong 1992, Hartley 1992]

Fundamental matrix

TECHNISCHE UNIVERSITÄT DARMSTADT

- Two image coordinates of the same 3D point must satisfy

$$\mathbf{x}_1^{\mathrm{T}}\mathbf{F}\mathbf{x}_2 = 0$$

- Other important properties...
- Epipolar lines: $\mathbf{l}_1 = \mathbf{F}\mathbf{x}_2$ and $\mathbf{l}_2 = \mathbf{F}^{\mathrm{T}}\mathbf{x}_1$
- Epipoles are the (left/right) null-space of the fundamental matrix:

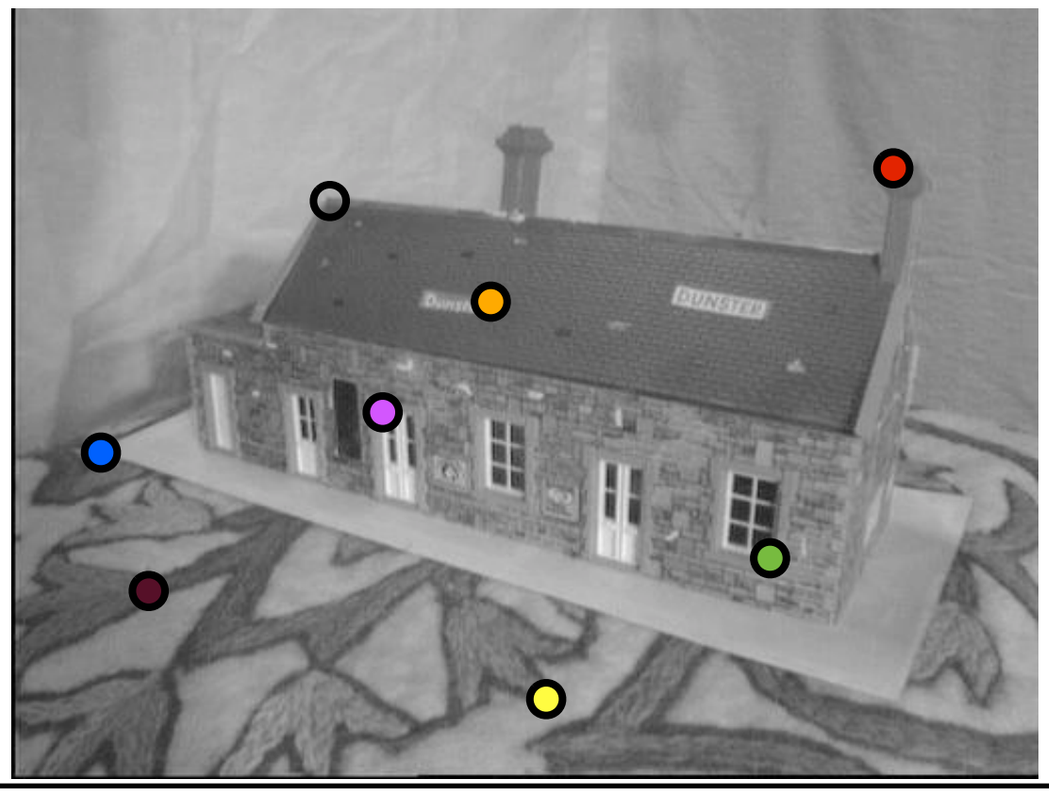
$$\mathbf{e}_1^{\mathrm{T}}\mathbf{F} = \mathbf{F}^{\mathrm{T}}\mathbf{e_1} = 0 \qquad \mathbf{F}\mathbf{e}_2 = 0$$

  - Fundamental matrix is singular; has rank 2.
  - 7 degrees of freedom

Estimating the fundamental matrix

TECHNISCHE UNIVERSITÄT DARMSTADT

- Seven degrees of freedom
- One equation per correspondence
- Need at least 7 pairs of corresponding points, but the solution is non-linear
- Linear solution with at least 8 point pairs: the normalized eight-point algorithm



Eight-point algorithm

TECHNISCHE UNIVERSITÄT DARMSTADT

- Stack equations from at least 8 correspondences
- Remember
  - a homogeneous linear equation system
  - solve s.t. $|| f ||=1$, using SVD

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} xx' & yx' & x' & xy' & yy' & y' & x & y & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0 \quad \rightarrow \quad Af = 0$$

Eight-point algorithm

TECHNISCHE UNIVERSITÄT DARMSTADT

- Remember: numerical stability
  - Coefficients of an equation system should be in the same order of magnitude
  - in pixels: xx' ~ $1\mathrm{E}6$
  - Conditioning: scale and shift points to be in [-1..1]
  - apply inverse transform to estimated matrix

$$s = \frac{1}{2} \max_i(\|x_i\|)$$
$$t = \mathrm{mean}(x_i)$$

$$T = \begin{bmatrix} \frac{1}{s} & 0 & -\frac{t_x}{s} \\ 0 & \frac{1}{s} & -\frac{t_y}{s} \\ 0 & 0 & 1 \end{bmatrix}$$

$$u = Tx$$
$$u' = T'x'$$

$$u'^{\top} \bar{F} u = 0$$

Eight-point algorithm

TECHNISCHE UNIVERSITÄT DARMSTADT

- Problem: the result should be of rank(2)
  - Find the most similar rank(2)-matrix, i.e. force the smallest eigenvalue to be 0

$$A\bar{f} = 0$$

SVD ↓ to ensure unit scale

$$U_A D_A V_A^\top = A$$

$$\widetilde{F} = \begin{bmatrix} v_{19} & v_{29} & v_{39} \\ v_{49} & v_{59} & v_{69} \\ v_{79} & v_{89} & v_{99} \end{bmatrix}$$

rank(3)

SVD

$$U_F \widetilde{D}_F V_F^\top = \widetilde{F}$$

$$D_F = \widetilde{D}_F \quad , \quad D_{F,33} = 0$$

rank(2)

$$\bar{F} = U_F D_F V_F^\top$$

un-conditioning

$$F = T'^\top \bar{F} T$$

Estimating the fundamental matrix

TECHNISCHE UNIVERSITÄT DARMSTADT

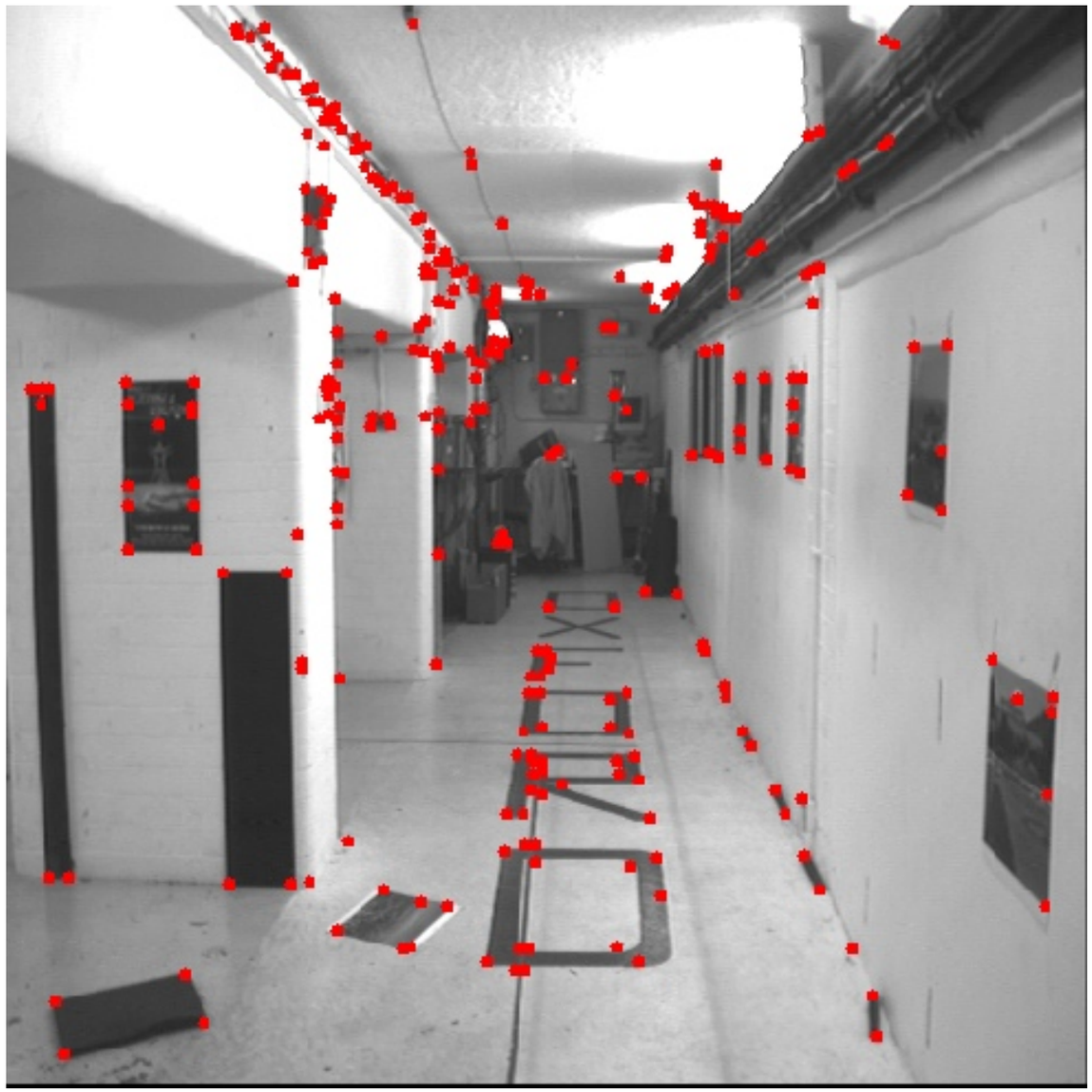
- Calibration not required, hence can deal with
  - archival images
  - photos not taken for reconstruction purposes
  - varying intrinsics
- So, where to start?

- The cookbook recipe
  - find interest points in both images
  - match them without epipolar constraint
  - compute epipolar geometry
  - ... refine

Example

TECHNISCHE UNIVERSITÄT DARMSTADT

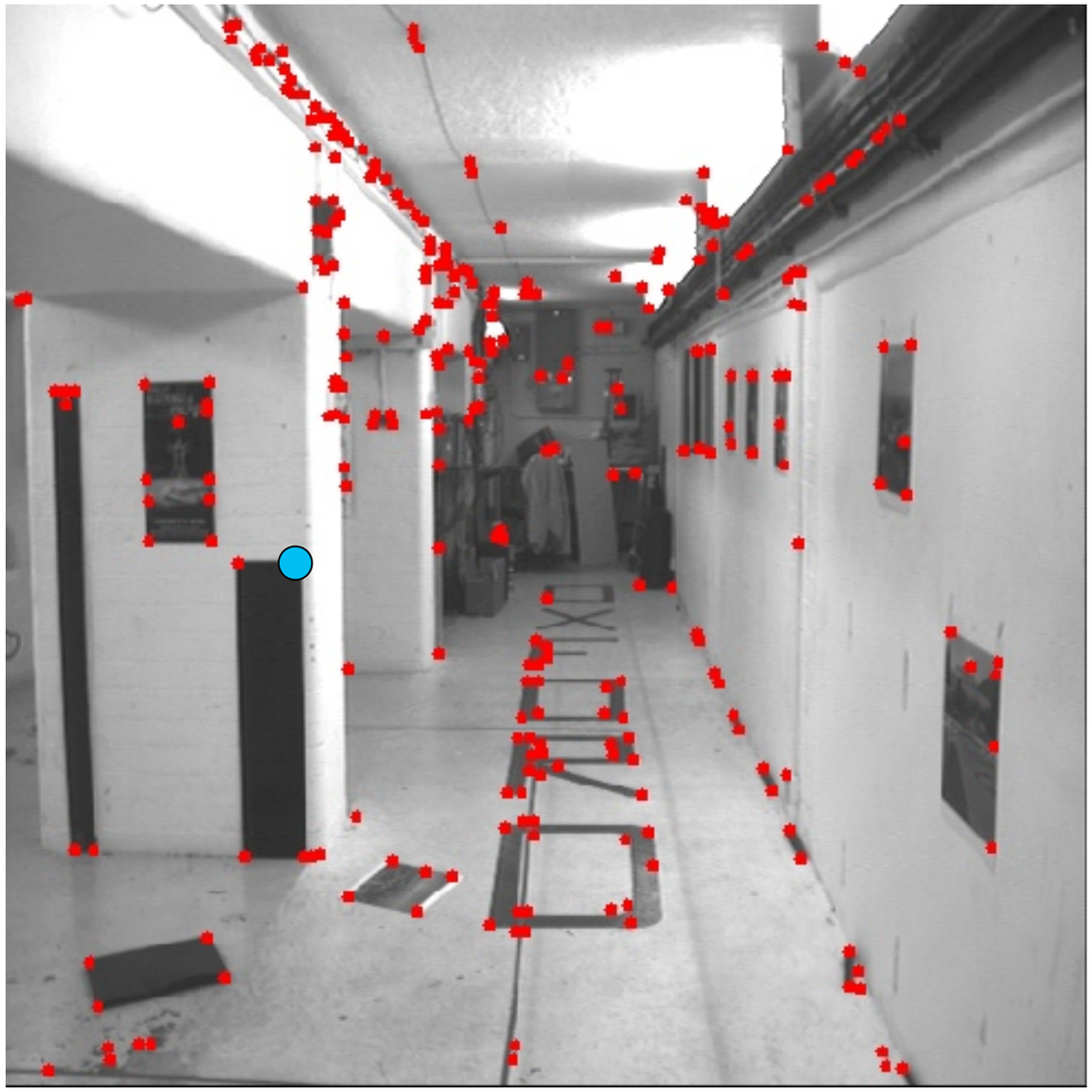
- Interest point (here: Harris corner detector)

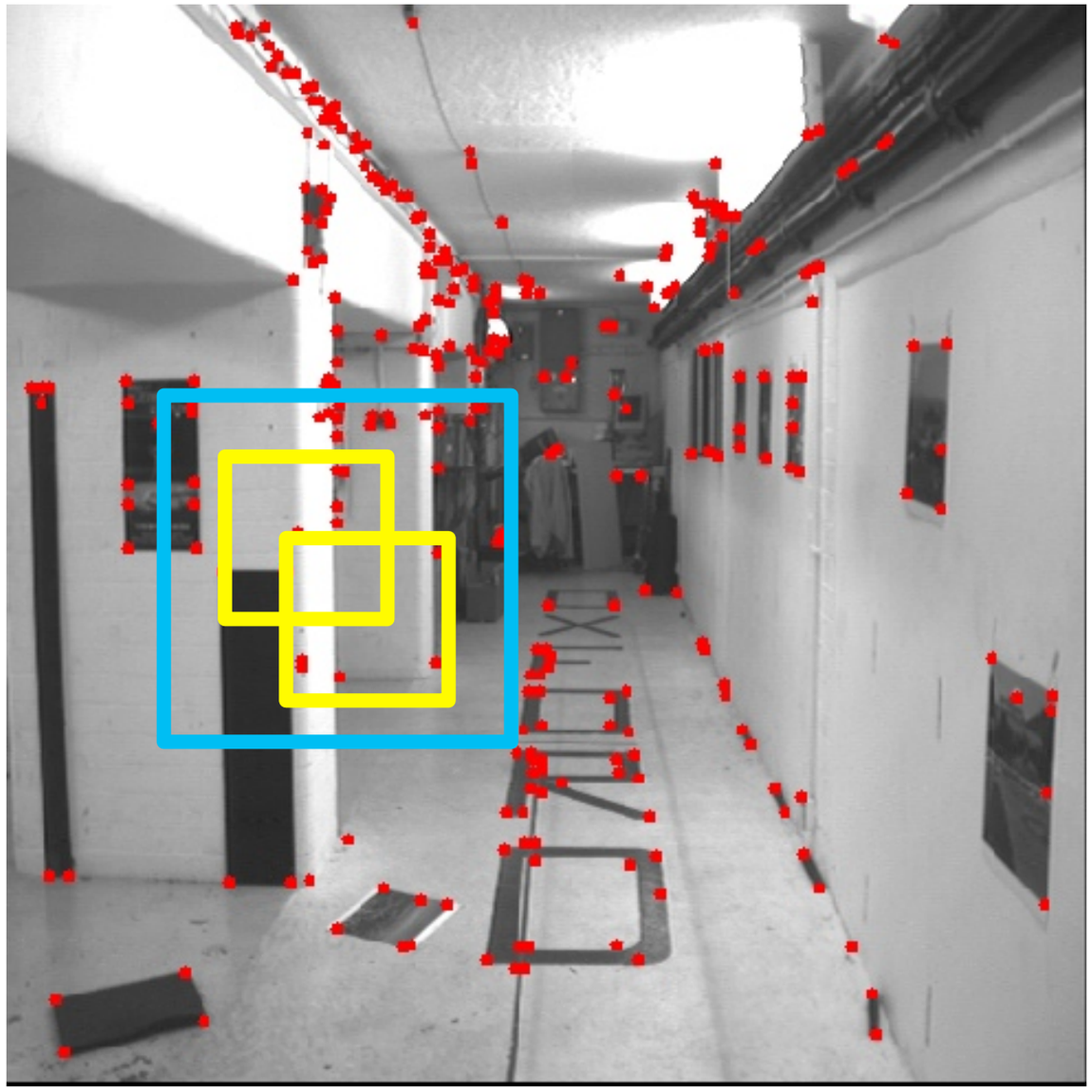


Example

TECHNISCHE UNIVERSITÄT DARMSTADT

- Match using only weak constraints (here: proximity)

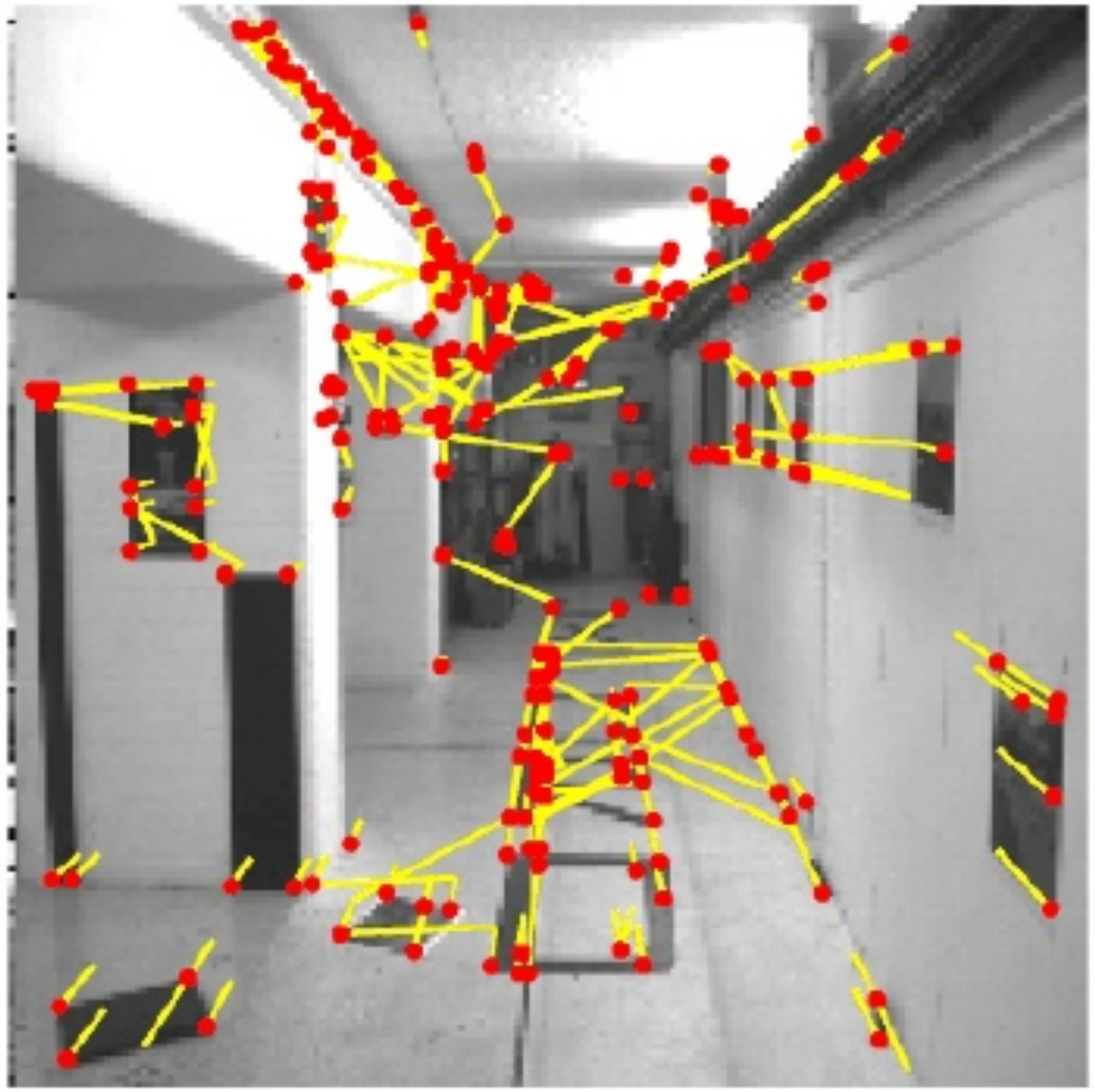




Example

TECHNISCHE UNIVERSITÄT DARMSTADT

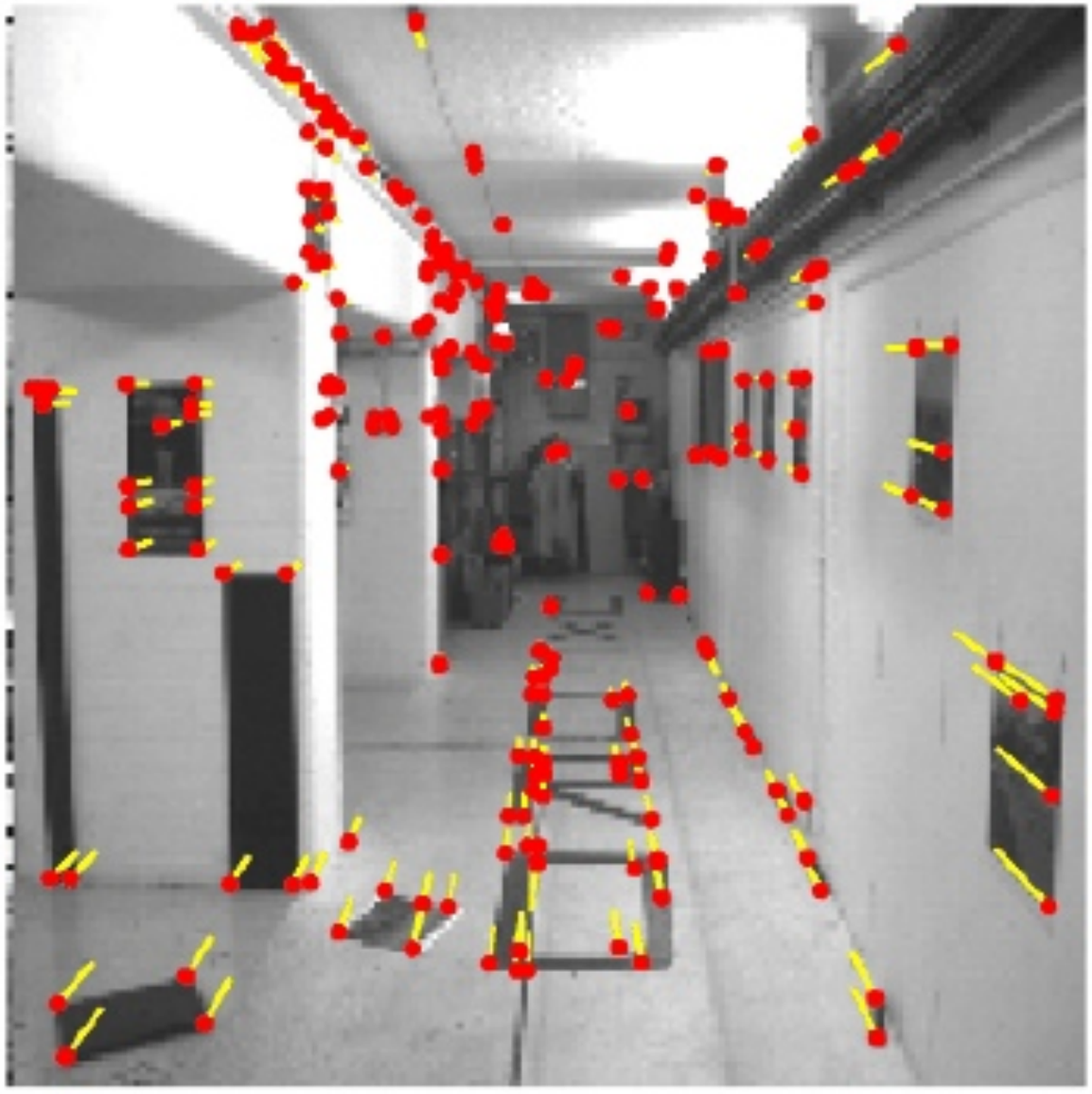
- many wrong matches, but enough to estimate $\mathbf{F}$ (RANSAC)



Example

TECHNISCHE UNIVERSITÄT DARMSTADT

- correspondences consistent with epipolar geometry



Example

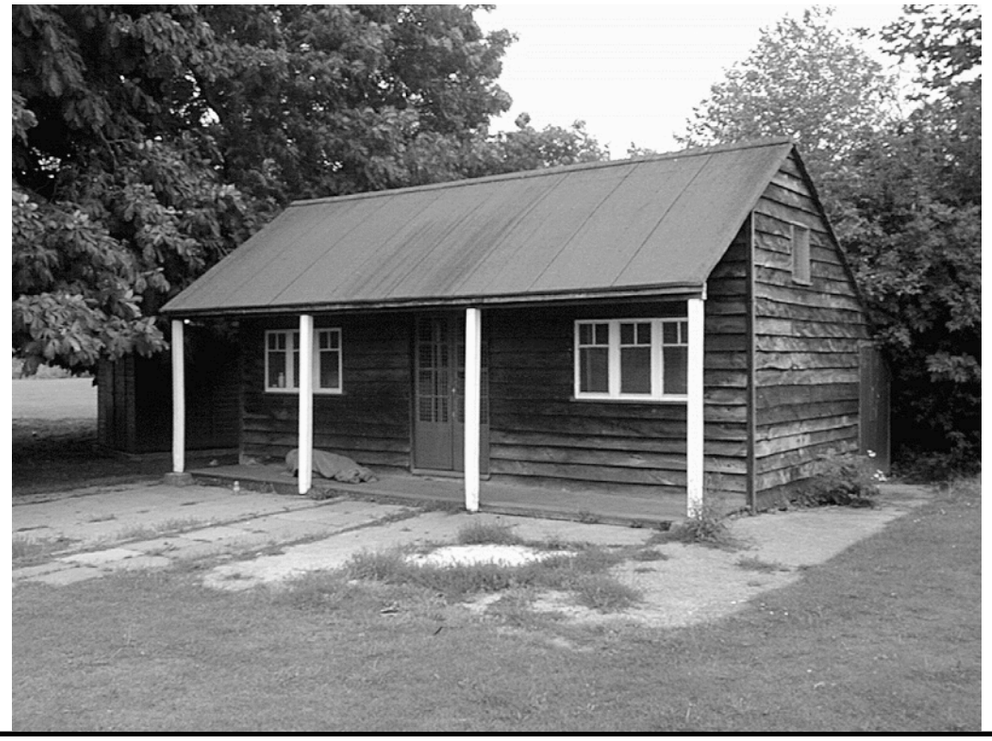
TECHNISCHE UNIVERSITÄT DARMSTADT

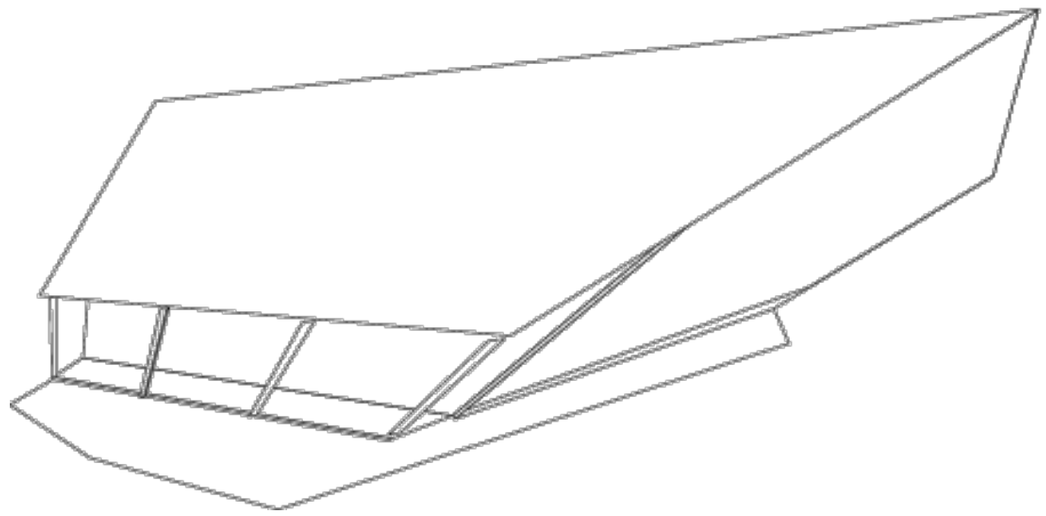
- Resulting epipolar geometry

What can we do with 2 views?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Given two views of a static scene, and a small number of correspondences
- If the views were recorded with any pinhole camera, we can
  - reconstruct the scene up to a projective ambiguity
  - determine line intersections, coplanarity,...
  - need at least a 3rd view to find the calibration

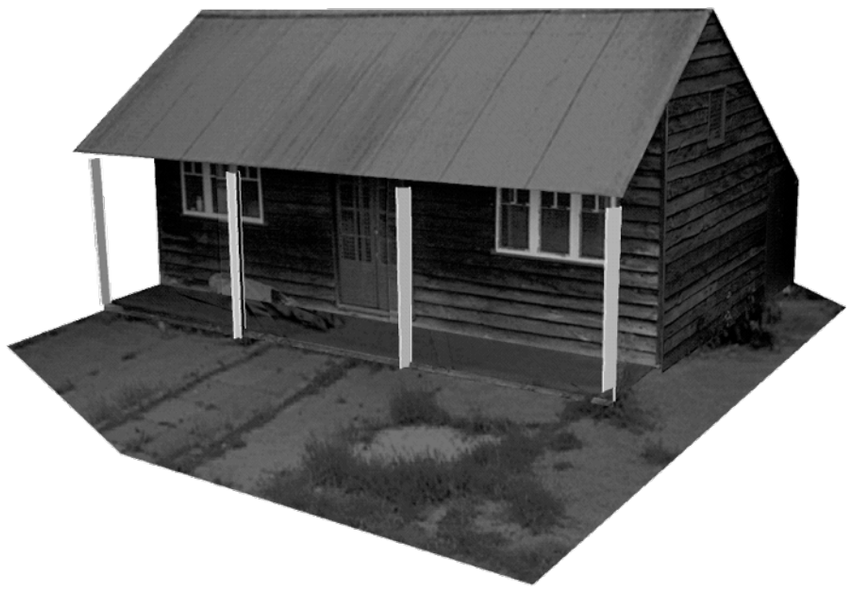




What can we do with 2 views?

TECHNISCHE UNIVERSITÄT DARMSTADT

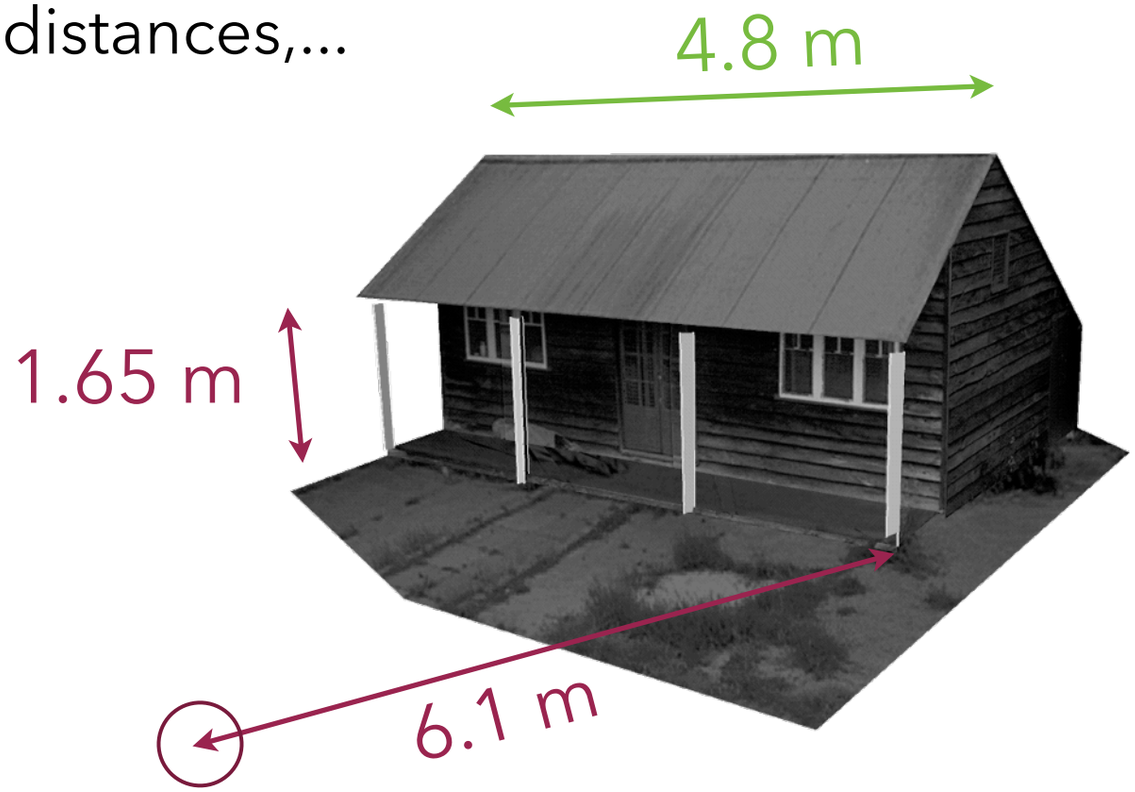
- Given two views of a static scene, and a small number of correspondences
- If the internal camera parameters have been calibrated, we can
  - reconstruct the scene up to scale
  - determine angles, relative distances,...
  - allows 3D modeling



What can we do with 2 views?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Given two views of a static scene, and a small number of correspondences
- If the baseline length (or any other distance in the scene) is known, we can
  - determine the global scale
  - measure absolute distances,...
  - allows navigation



Computer Vision I

Dense Stereo

TECHNISCHE UNIVERSITÄT DARMSTADT

visual inference

Stefan Roth, FB Informatik, TU Darmstadt – 03.07.2013

Stereo Vision

TECHNISCHE UNIVERSITÄT DARMSTADT

- Given two views of the same scene, estimate its 3D shape
  - Densely, not just at interest points...

Stereo Vision - Easier problem

TECHNISCHE UNIVERSITÄT DARMSTADT

- Narrower formulation: given two views with similar (but not identical!) viewpoint, fuse them to obtain a depth image

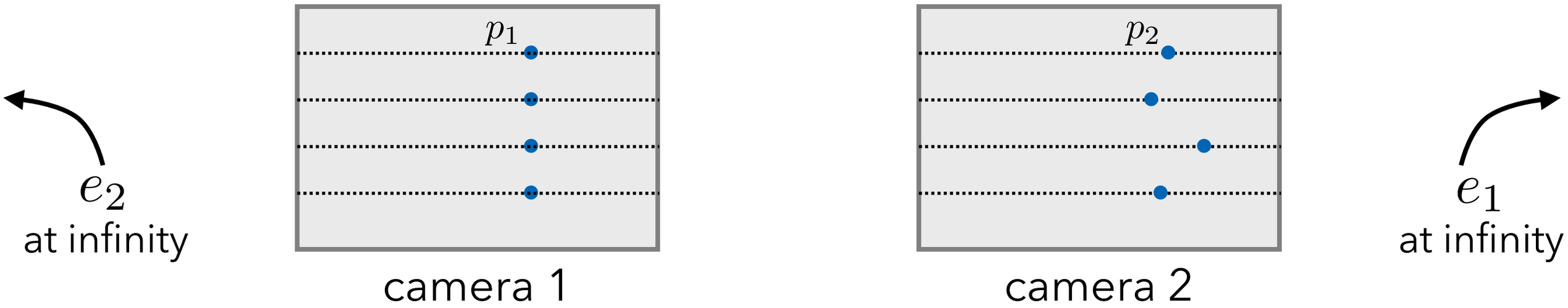
Image 1

Image 2

Dense depth map

Binocular Stereo

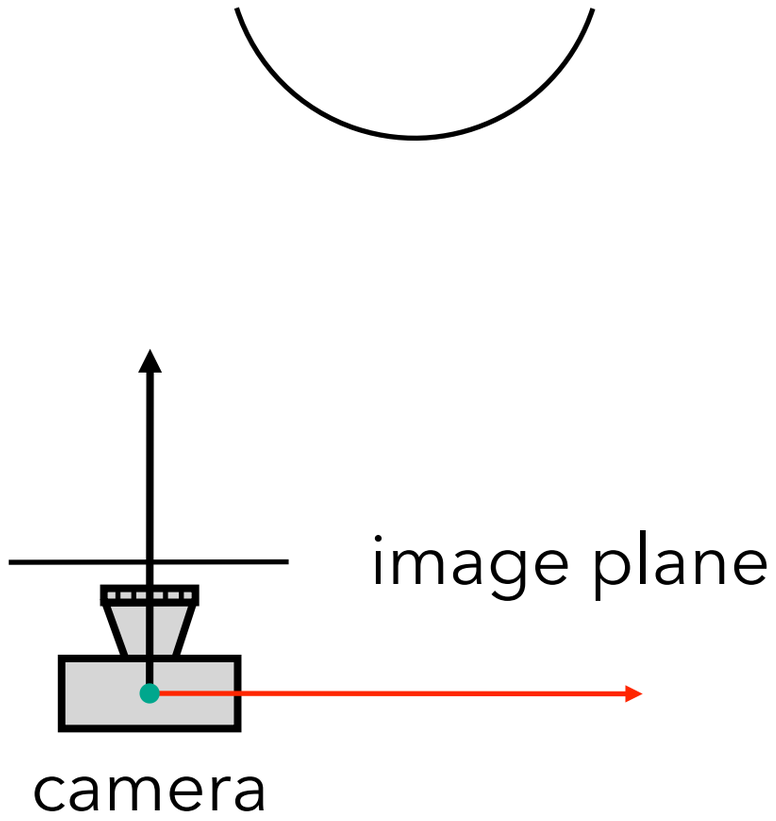
TECHNISCHE UNIVERSITÄT DARMSTADT



- Special case of epipolar geometry for stereo cameras with a standard binocular setup:
  - Epipoles are at infinity.
  - Epipolar lines are parallel.
  - Points correspond along the scanlines of the image.

Binocular Stereo

Left



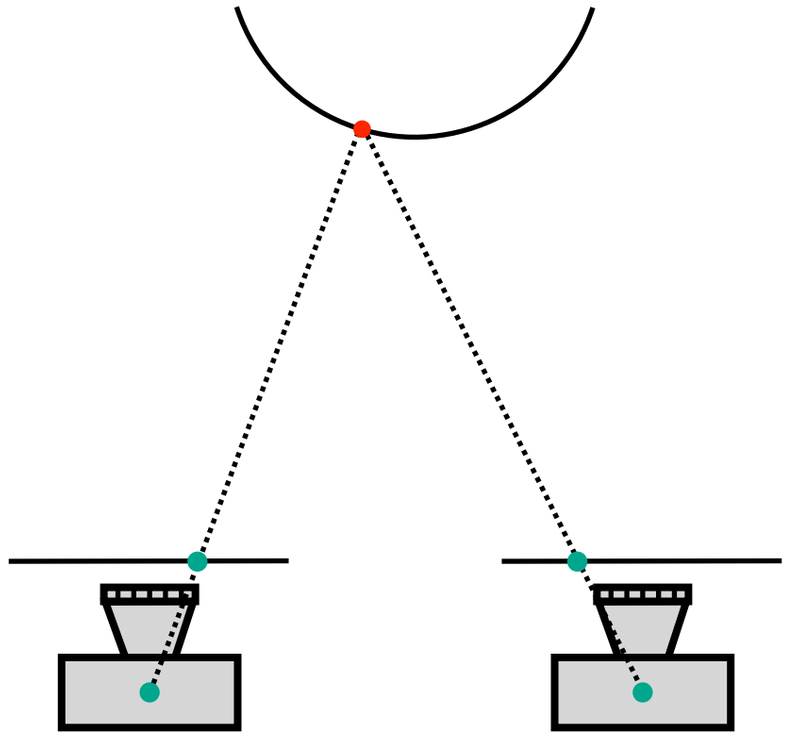
Binocular Stereo

TECHNISCHE UNIVERSITÄT DARMSTADT

Left

Right

Binocular Stereo

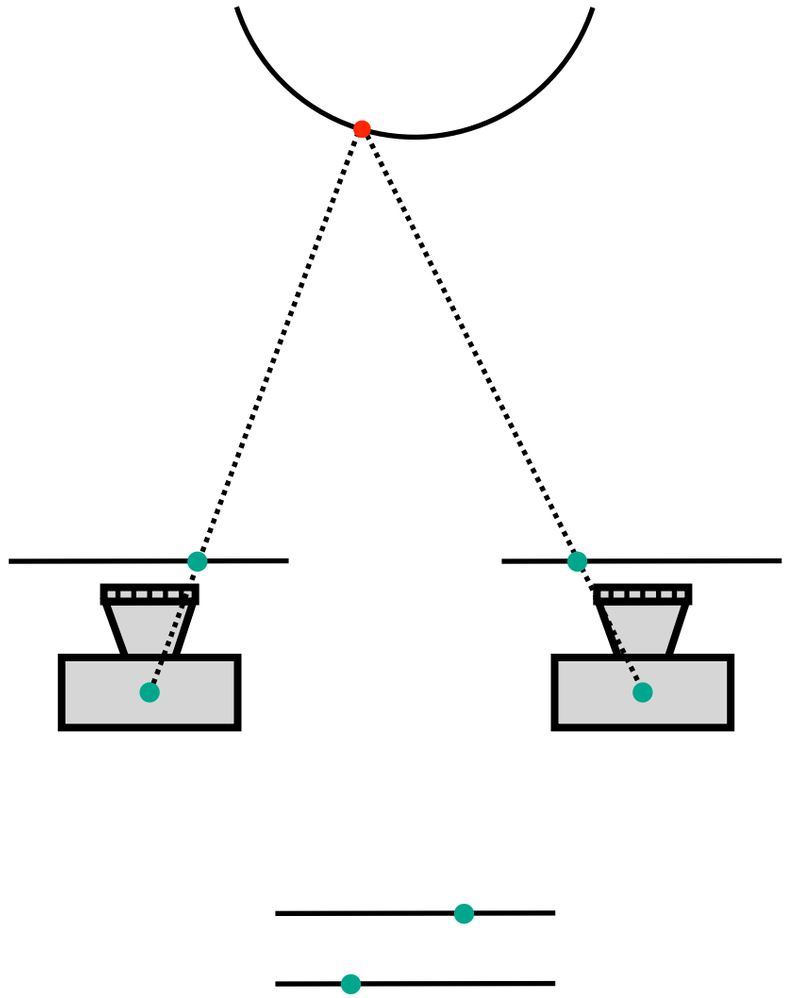


Left

Right

Binocular Stereo

TECHNISCHE UNIVERSITÄT DARMSTADT

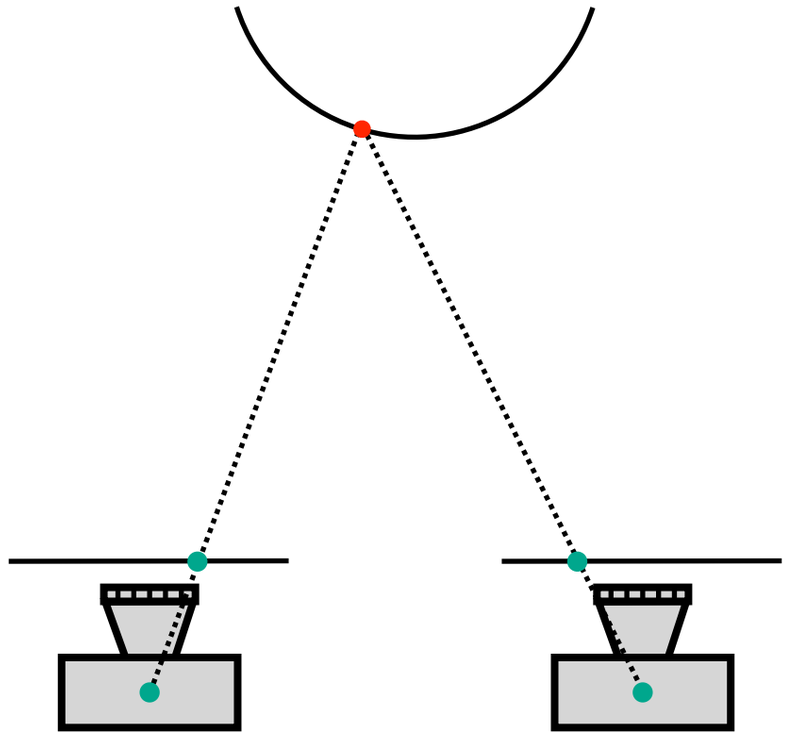


Left

Right

Binocular Stereo

TECHNISCHE UNIVERSITÄT DARMSTADT

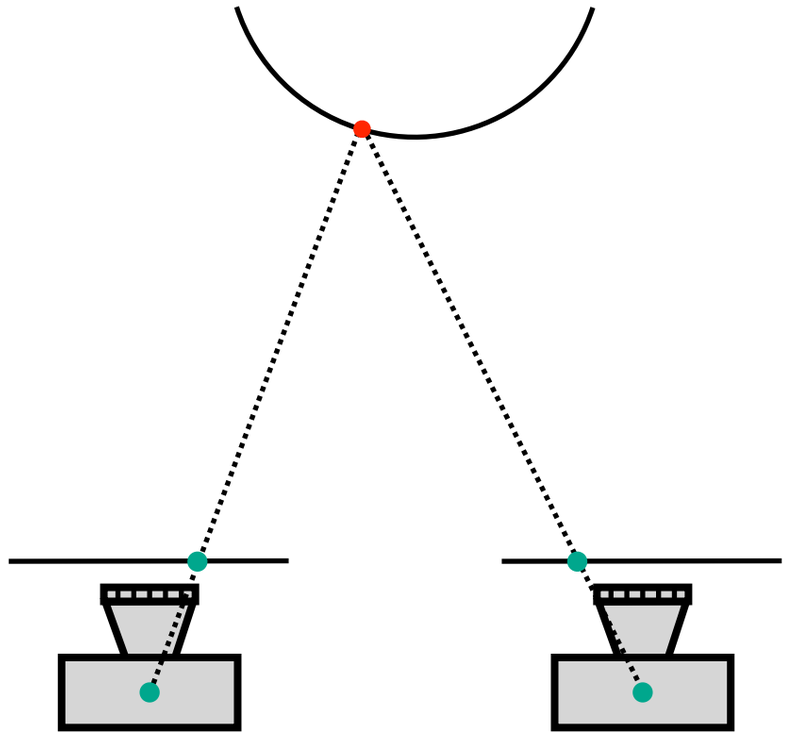


Left

Right

binocular disparity

Binocular Stereo



binocular disparity

From known geometry of the cameras and estimated disparity, recover depth in the scene

Left

Right

Triangulation

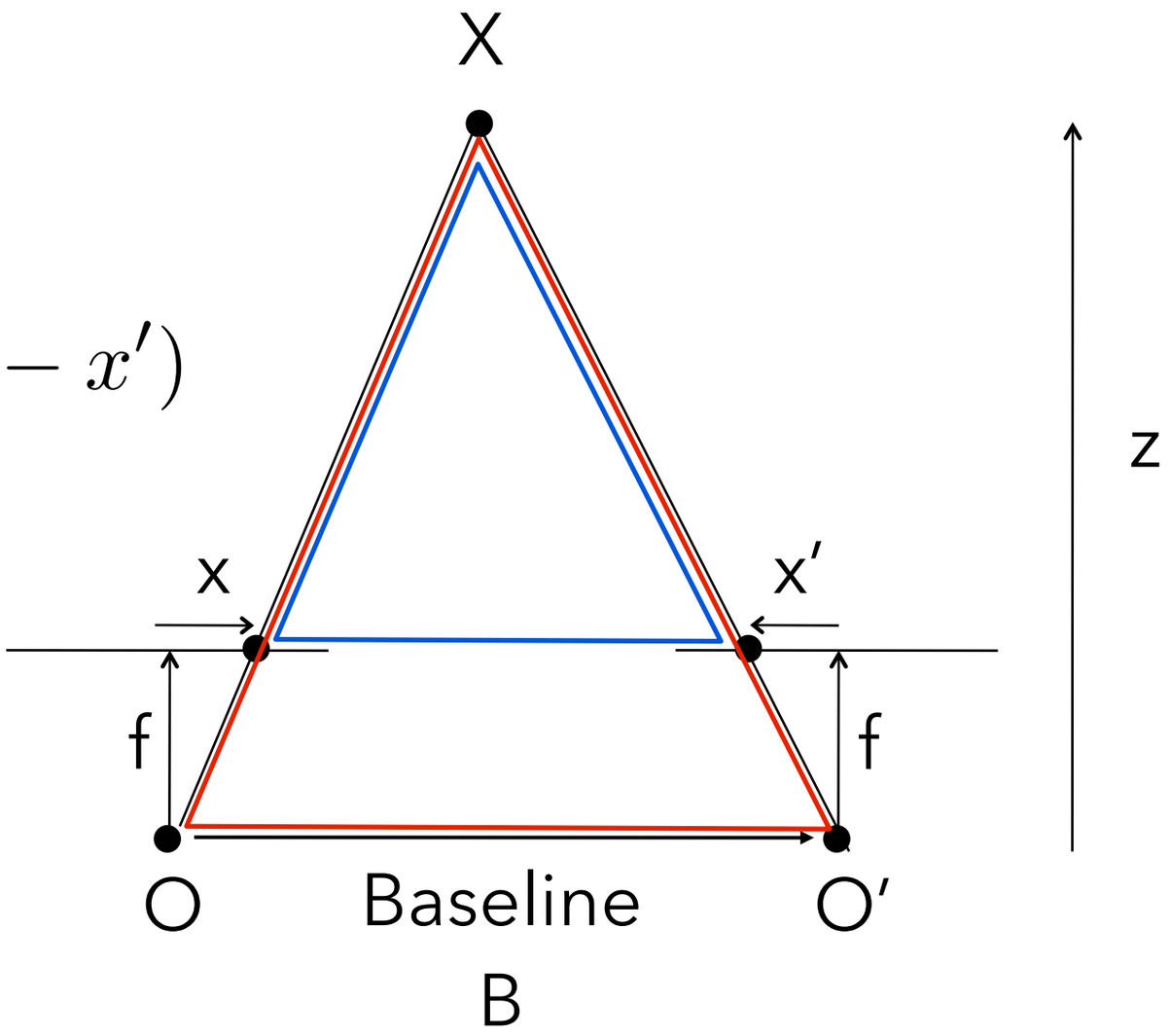
$$\frac{Z-f}{B-(x-x')} = \frac{Z}{B}$$

$$B \cdot Z - B \cdot f = B \cdot Z - Z \cdot (x-x')$$

$$Z = \frac{B \cdot f}{x-x'}$$

$$d = x - x' = \frac{B \cdot f}{Z}$$

disparity



- Parallel image planes: disparity from equal triangles

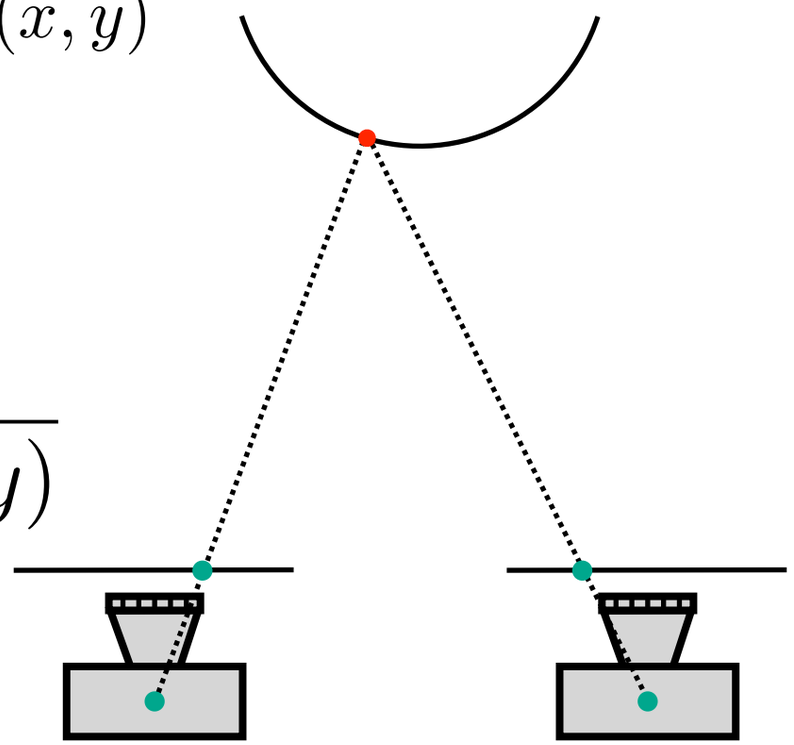
Binocular Disparity

$Z(x,y)$ is depth at pixel (x,y)

$d(x,y)$ is disparity

Estimate:

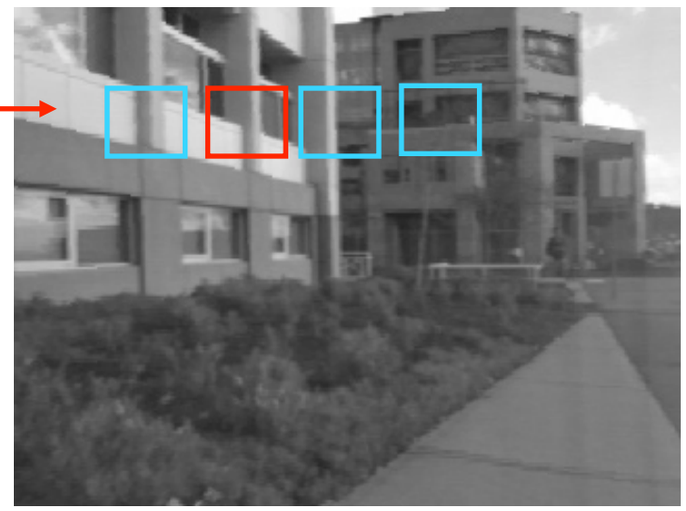
$$Z(x,y) = \frac{fB}{d(x,y)}$$



Left

Search for best match

Right



[Black]

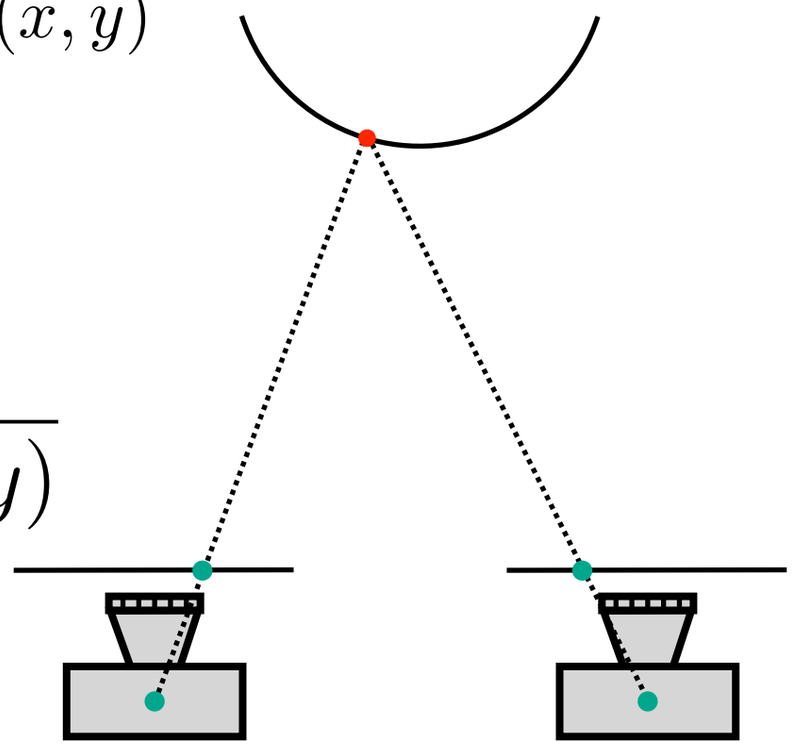
Binocular Disparity

$Z(x,y)$ is depth at pixel (x,y)

$d(x,y)$ is disparity

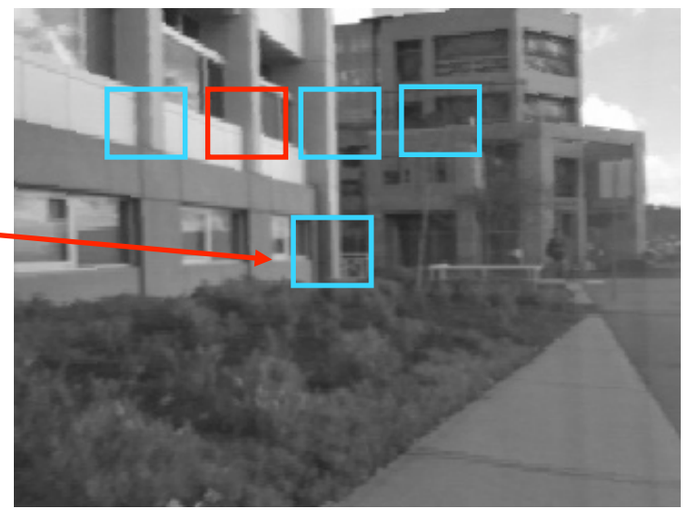
Estimate:

$$Z(x,y) = \frac{fB}{d(x,y)}$$



Left

Right



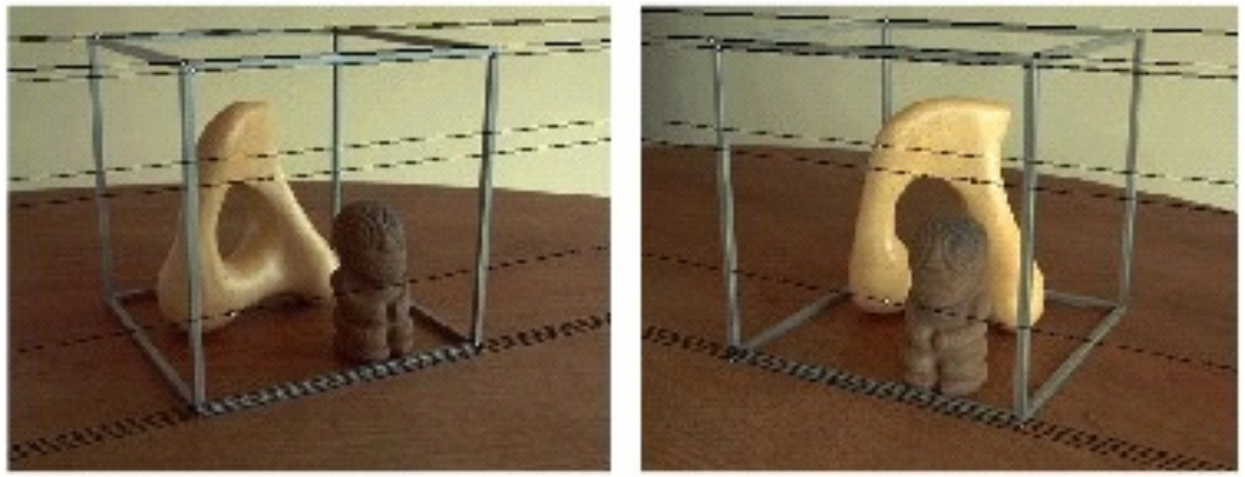
Do I need to consider this region?

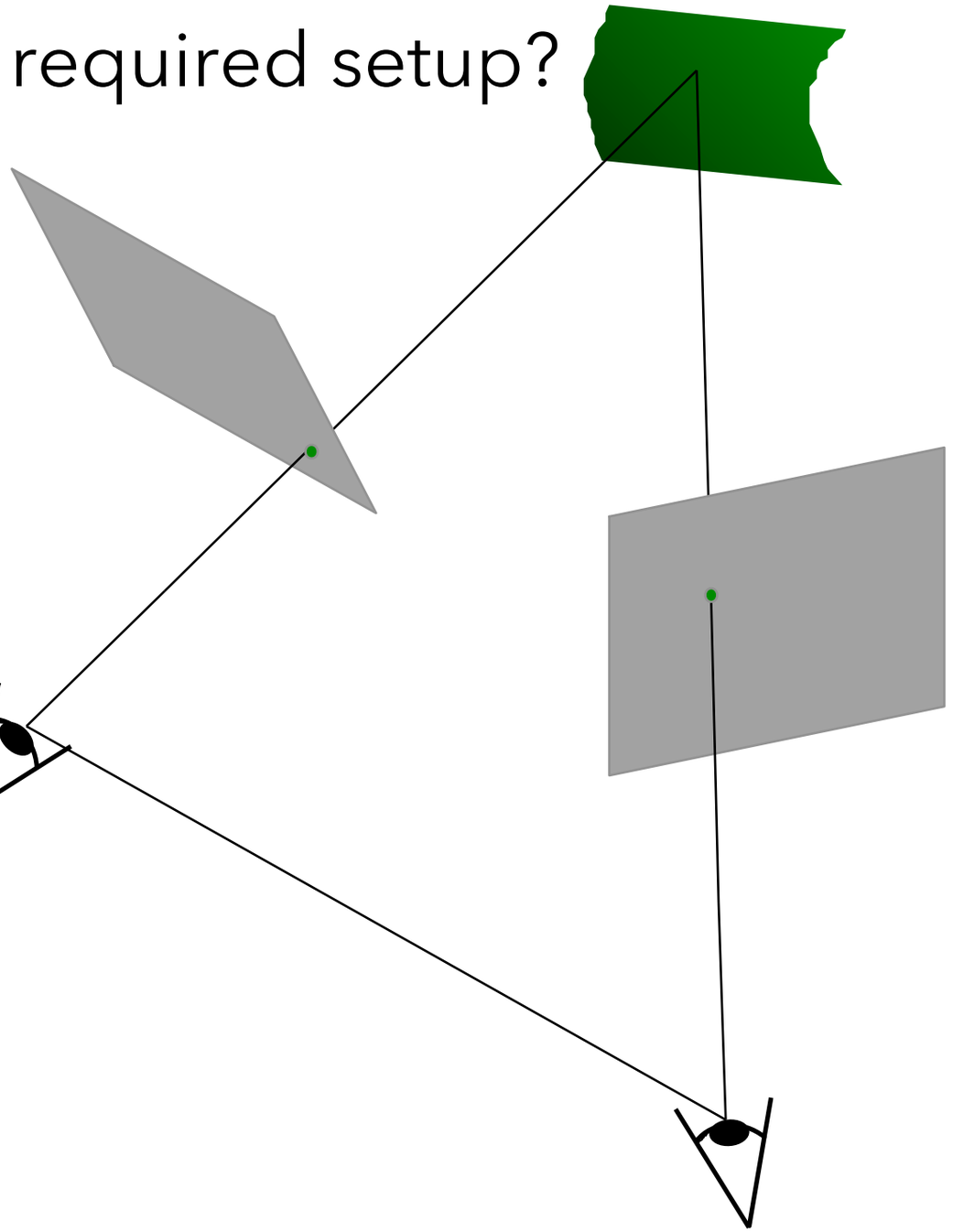
[Black]

Stereo rectification

TECHNISCHE UNIVERSITÄT DARMSTADT

- What if the images are not in the required setup?

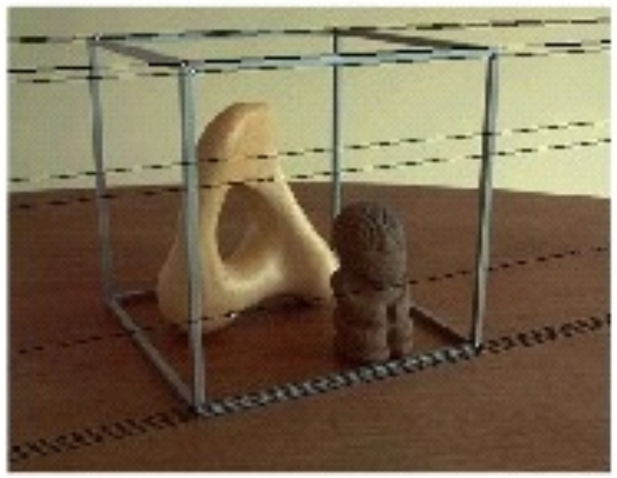


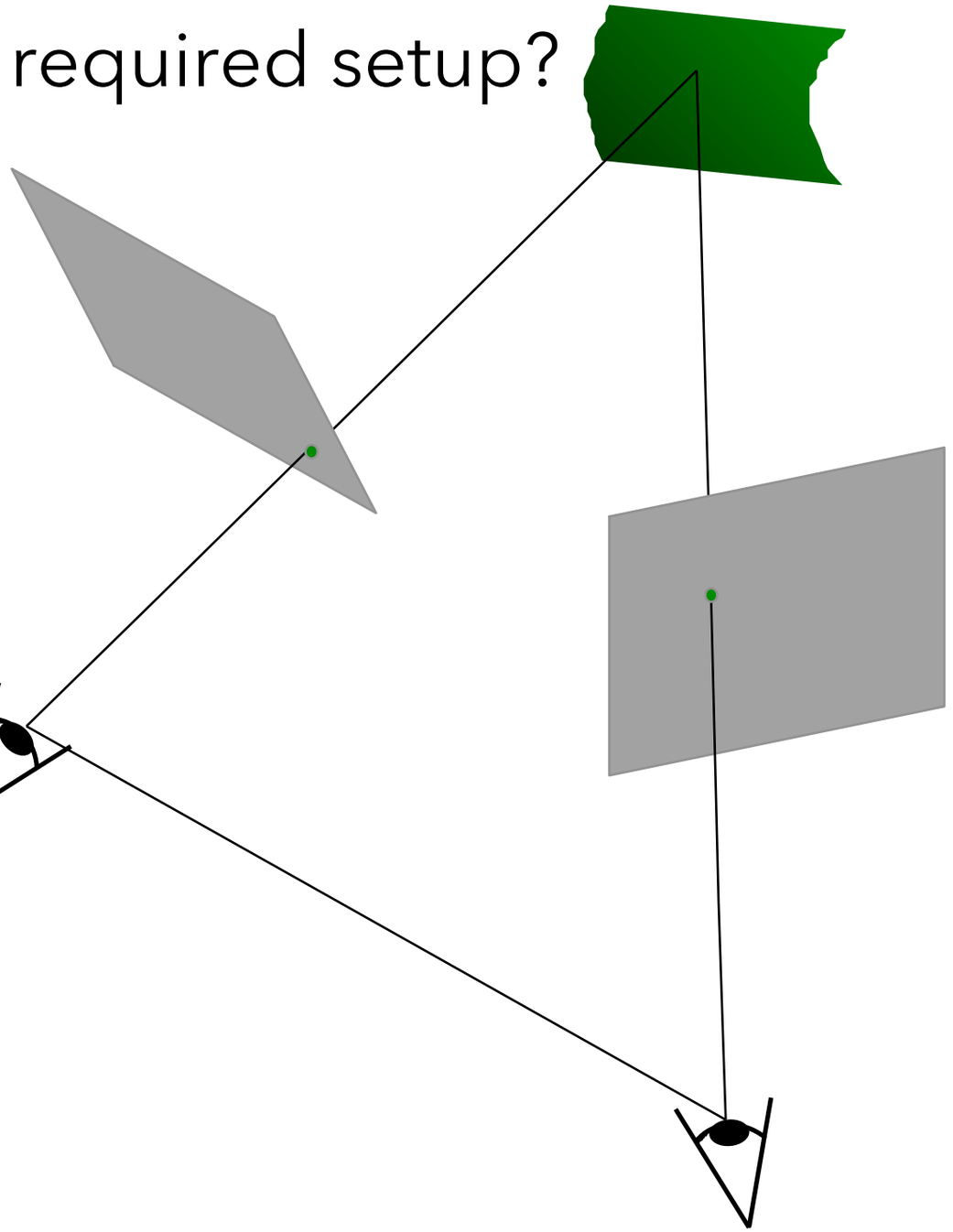


Stereo rectification

TECHNISCHE UNIVERSITÄT DARMSTADT

- What if the images are not in the required setup?
- Rewarp them such that they are!

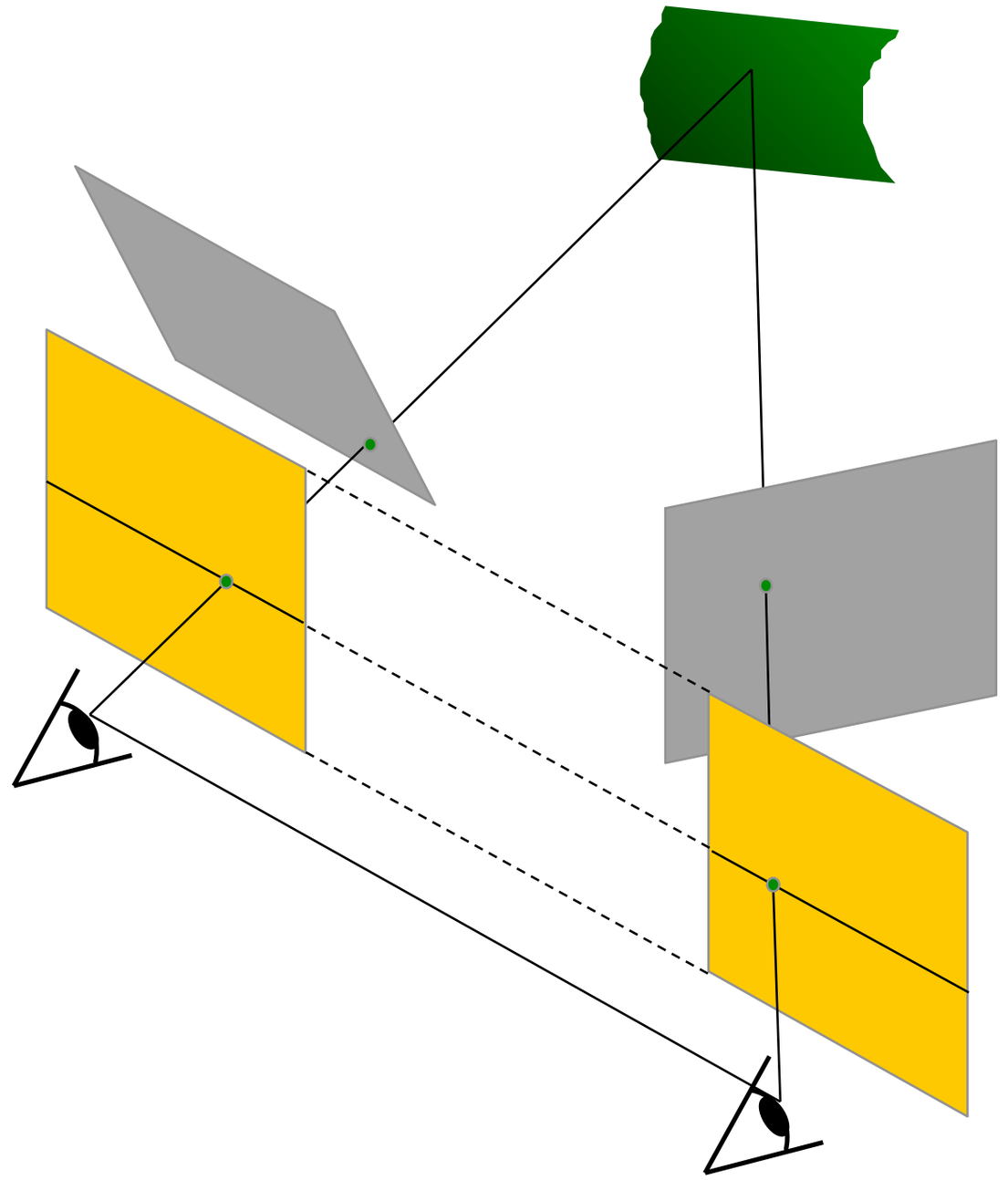




Stereo rectification

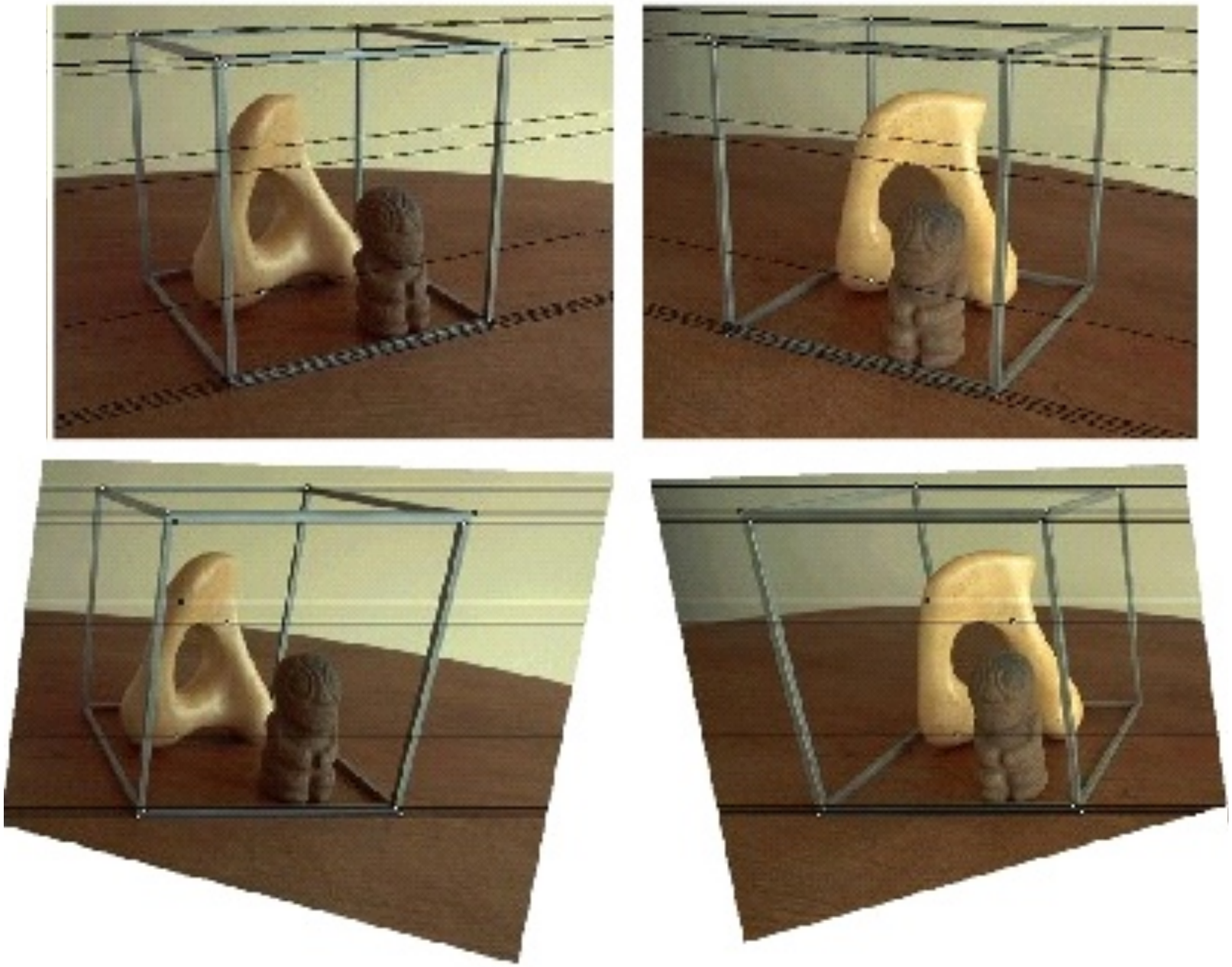
TECHNISCHE UNIVERSITÄT DARMSTADT

- Solution: map both image planes to a common plane parallel to the baseline
- requires a homography for each image
- after the transform, pixel motion is horizontal again



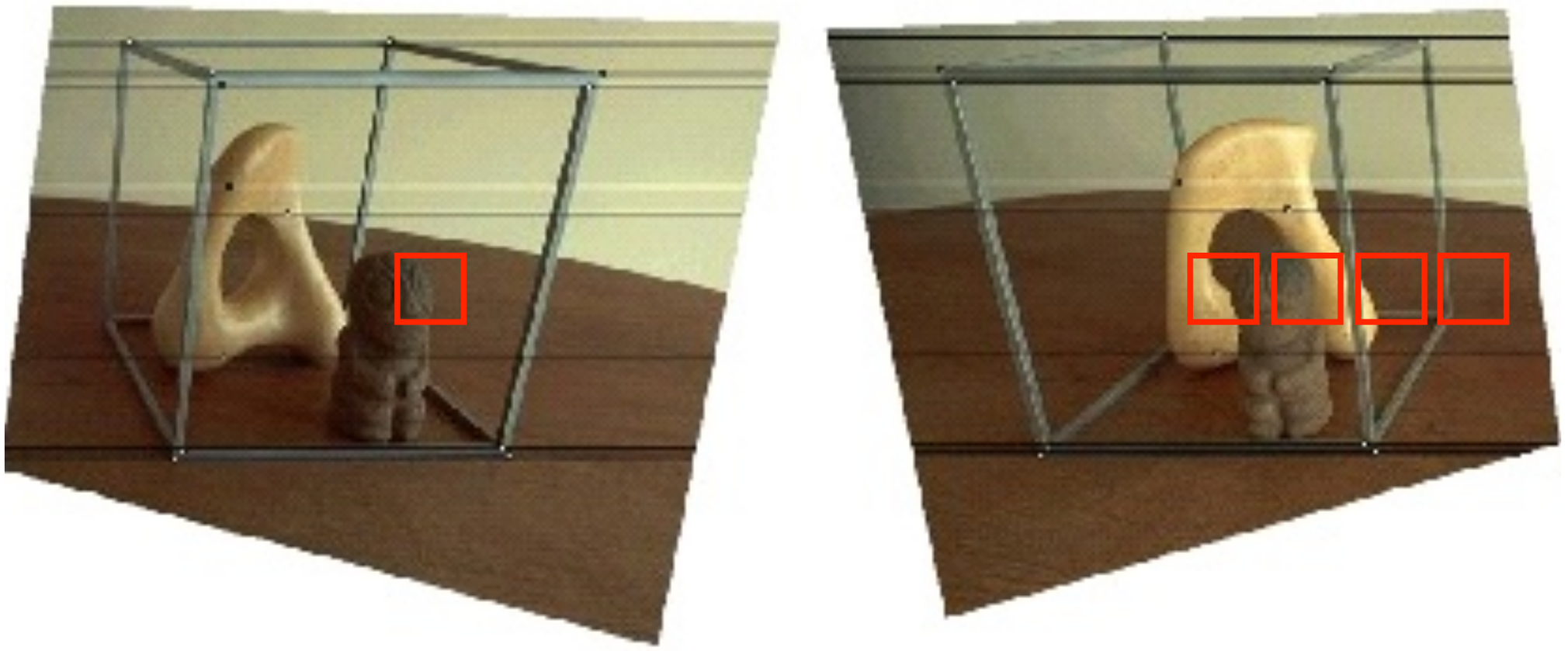
Stereo rectification

TECHNISCHE UNIVERSITÄT DARMSTADT



[Szeliski and Fleet]

Matching



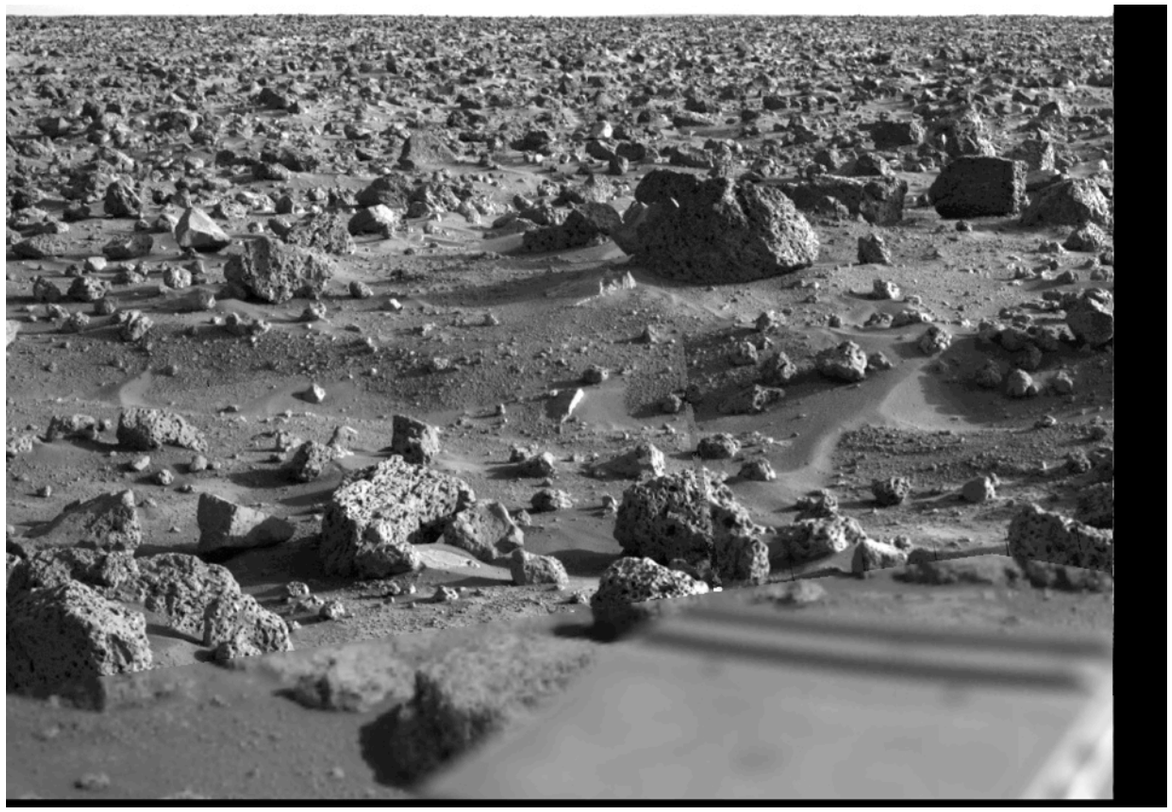
- Matching only has to occur along epipolar lines.
- Now in the simpler binocular case where the cameras are pointing forward.

[Szeliski and Fleet]

Stereo Correspondence

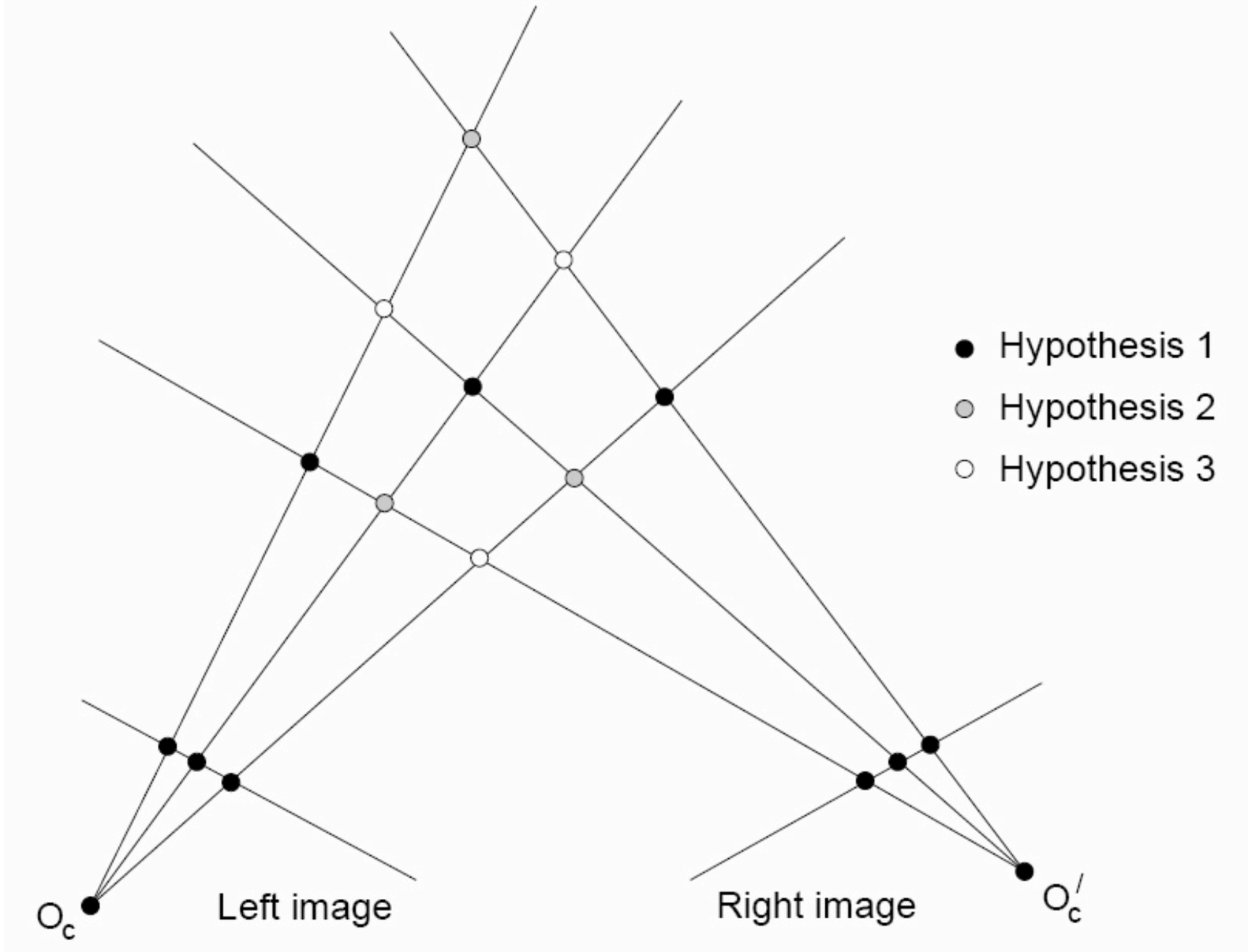
TECHNISCHE UNIVERSITÄT DARMSTADT

- Search over disparity to find correspondences
- Range of disparities to search over can change dramatically within a single image pair.



[Black]

Correspondence problem

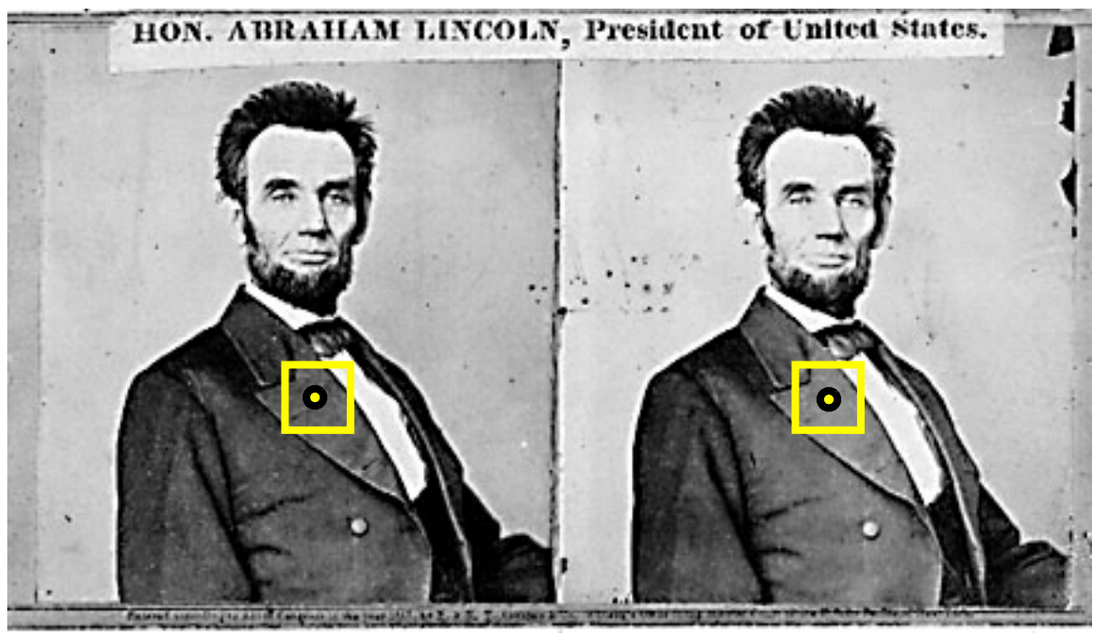


Even when constrained to 1D, there are multiple matching hypotheses - which one is correct?

Correspondence problem

TECHNISCHE UNIVERSITÄT DARMSTADT

- Let's make some assumptions to simplify the matching problem
  - The baseline is relatively small (compared to the depth of scene points) - "narrow-baseline stereo"
  - Matching regions are similar in appearance
  - Most scene points are visible in both views

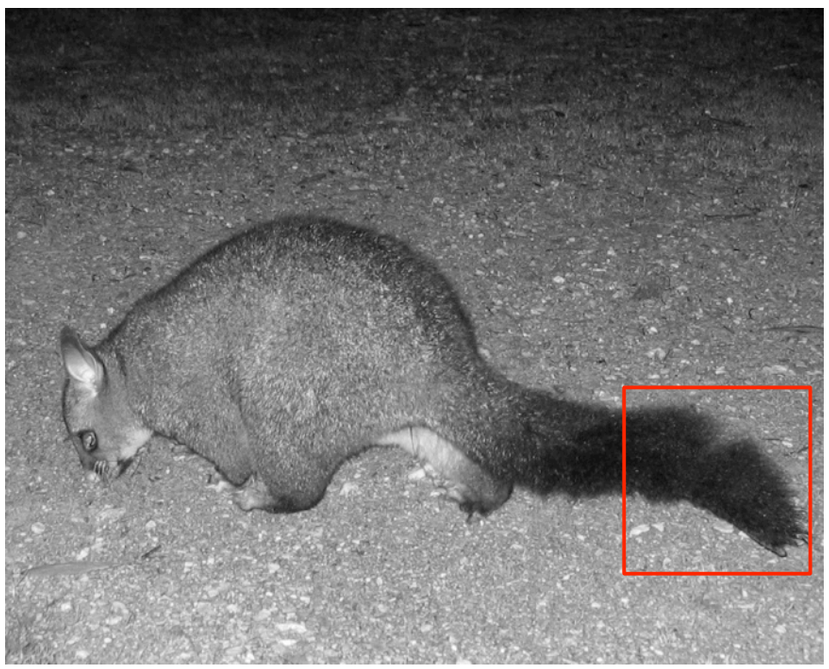


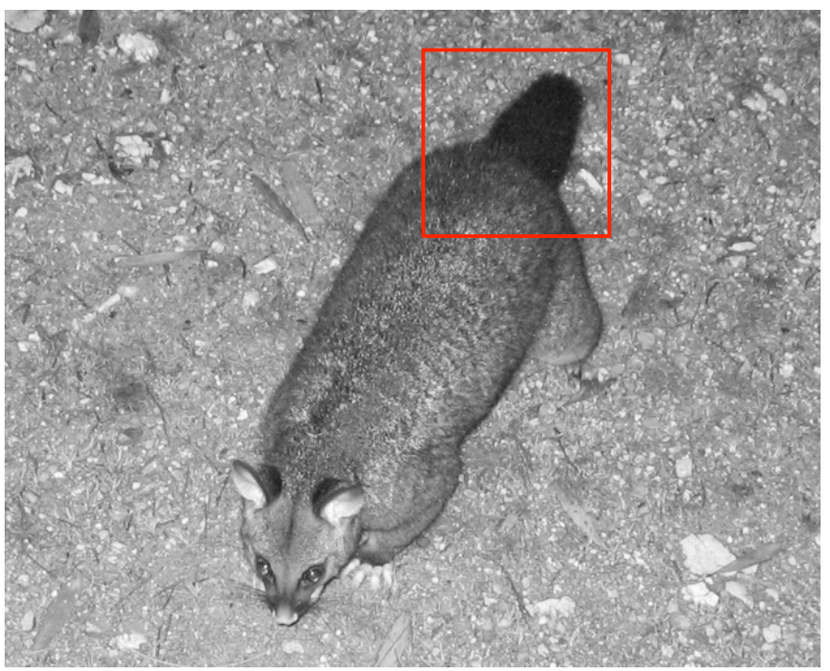
note: the 1st assumption makes the others a lot more likely

Correspondence problem

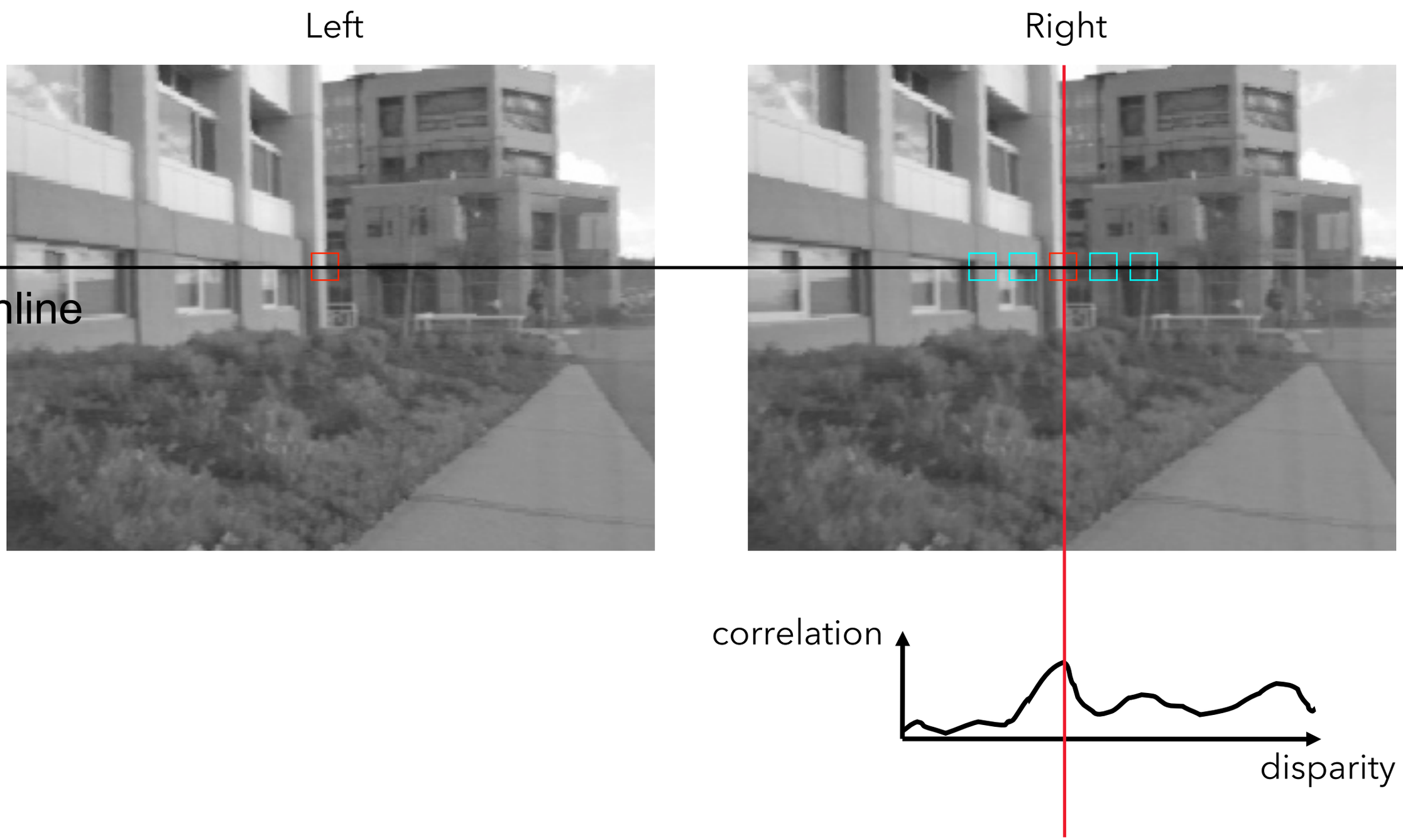
TECHNISCHE UNIVERSITÄT DARMSTADT

- Matching with narrow baseline
  - What are two “similar pixels”?
  - An ill-posed question, a single intensity sample does not reveal the local image structure
  - Measure similarity of the surrounding region
  - Area-based similarity measures



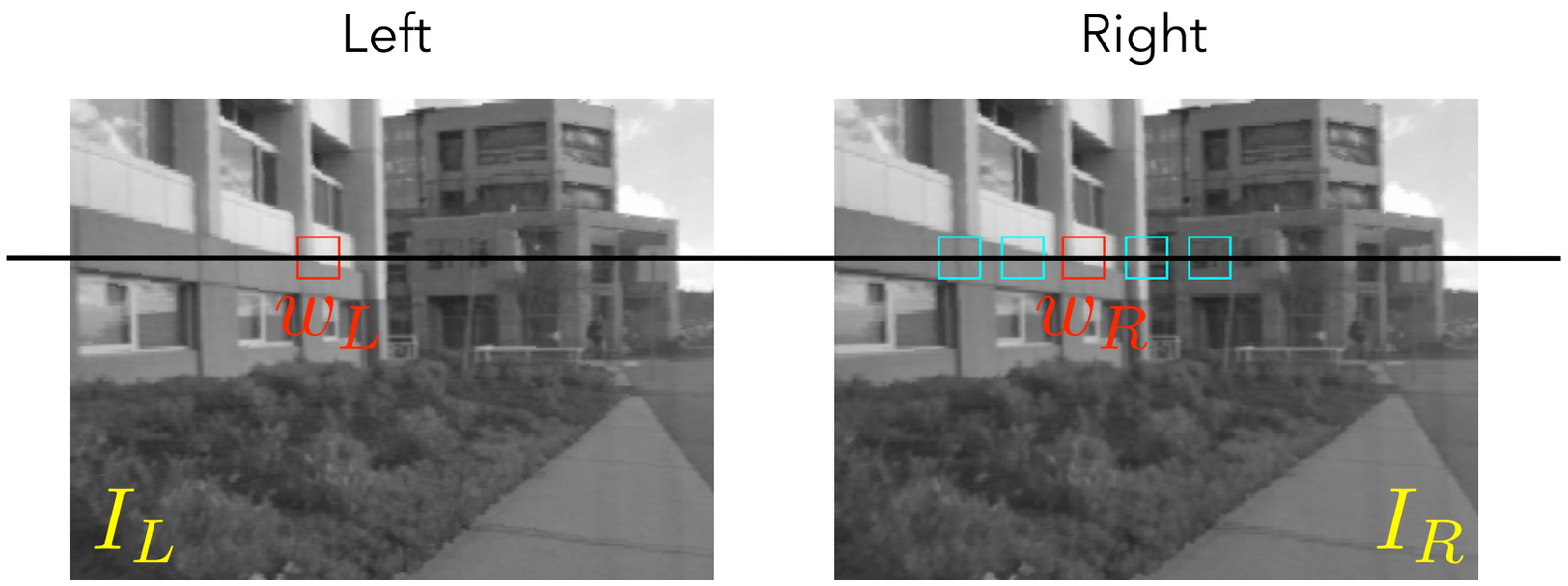


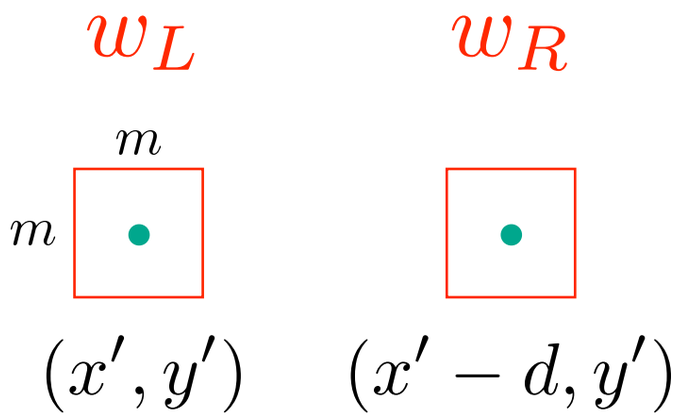
Correspondence Using Correlation



[Black]

Normalized Correlation



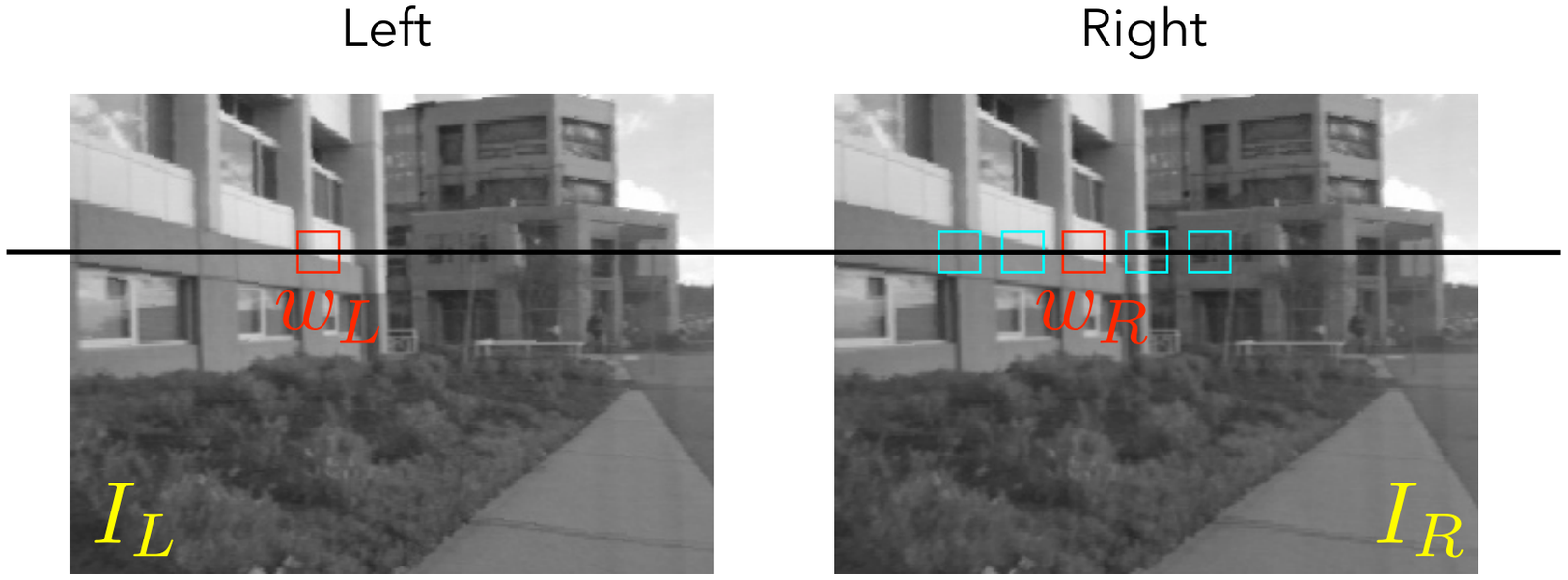


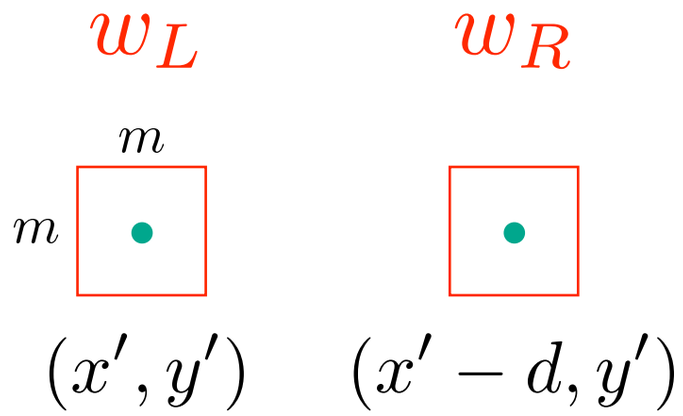
- w_L and w_R are corresponding $m \times m$ windows of pixels. We also write them as vectors $\mathbf{w}_L$ and $\mathbf{w}_R$.
- The normalized correlation computes the cosine of the angle between the patches:

$$\mathrm{NC}(x, y, d) = \frac{(\mathbf{w}_L(x, y) - \bar{\mathbf{w}}_L(x, y))^{\mathrm{T}} (\mathbf{w}_R(x - d, y) - \bar{\mathbf{w}}_R(x - d, y))}{|\mathbf{w}_L(x, y) - \bar{\mathbf{w}}_L(x, y)| \, |\mathbf{w}_R(x - d, y) - \bar{\mathbf{w}}_R(x - d, y)|}$$

patch mean

Even simpler: Sum of Squared (Pixel) Differences





- w_L and w_R are corresponding $m \times m$ windows of pixels.
- The SSD cost measures the intensity difference as a function of disparity:

$$\mathrm{SSD}_R(x, y, d) = \sum_{(x',y') \in w_L(x,y)} (I_L(x', y') - I_R(x' - d, y'))^2$$

[Black]

Window-based matching

TECHNISCHE UNIVERSITÄT DARMSTADT

- In practice:
  - Choose some disparity range $[0, d_{\max}]$
  - For all pixels (x, y) try all disparities and choose the one that maximizes the normalized correlation or minimizes the SSD.

Left

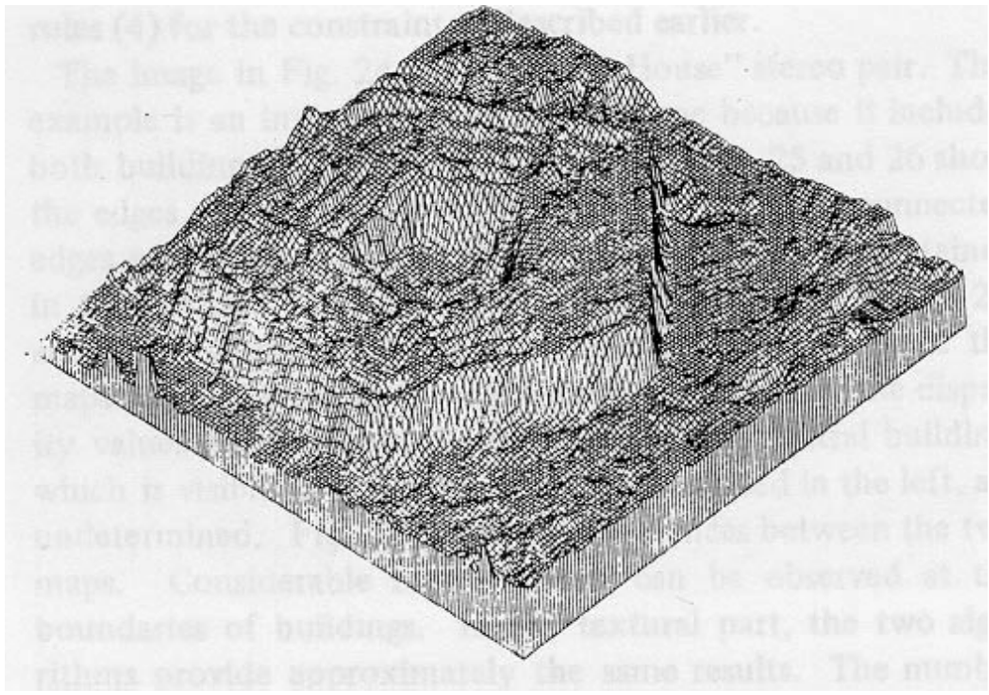
Disparity Map

[Point Grey Research]

Convert disparity to depth

TECHNISCHE UNIVERSITÄT DARMSTADT

- Of course, we can convert the disparity back to depth:



From [Ohta & Kanade, 1985]

Window-based matching

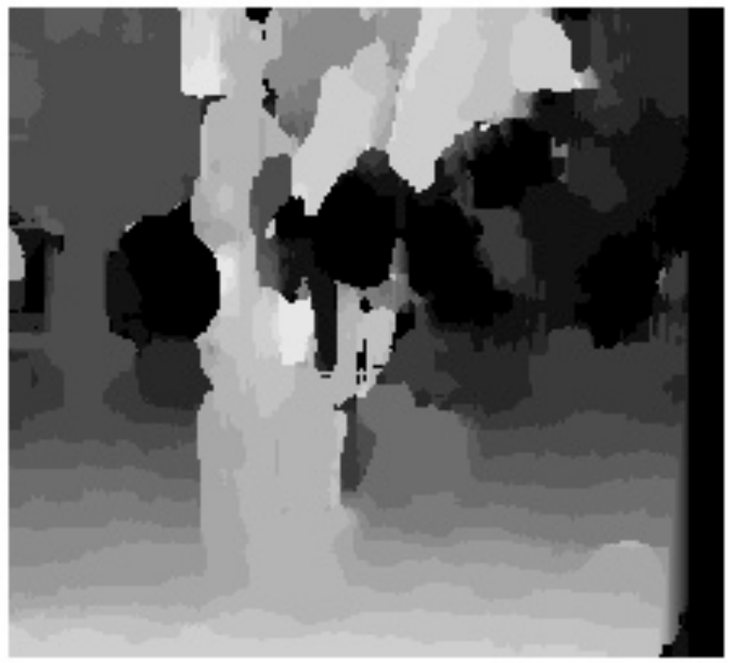
TECHNISCHE UNIVERSITÄT DARMSTADT

- In practice:
  - Choose some disparity range $[0, d_{\max}]$
  - For all pixels (x, y) try all disparities and choose the one that maximizes the normalized correlation or minimizes the SSD.

- Challenges and Problems:
  - How do we choose the right window size m?
  - Mismatches often lead to relatively poor results quality.

Influence of window size

$m = 3$



$m = 20$

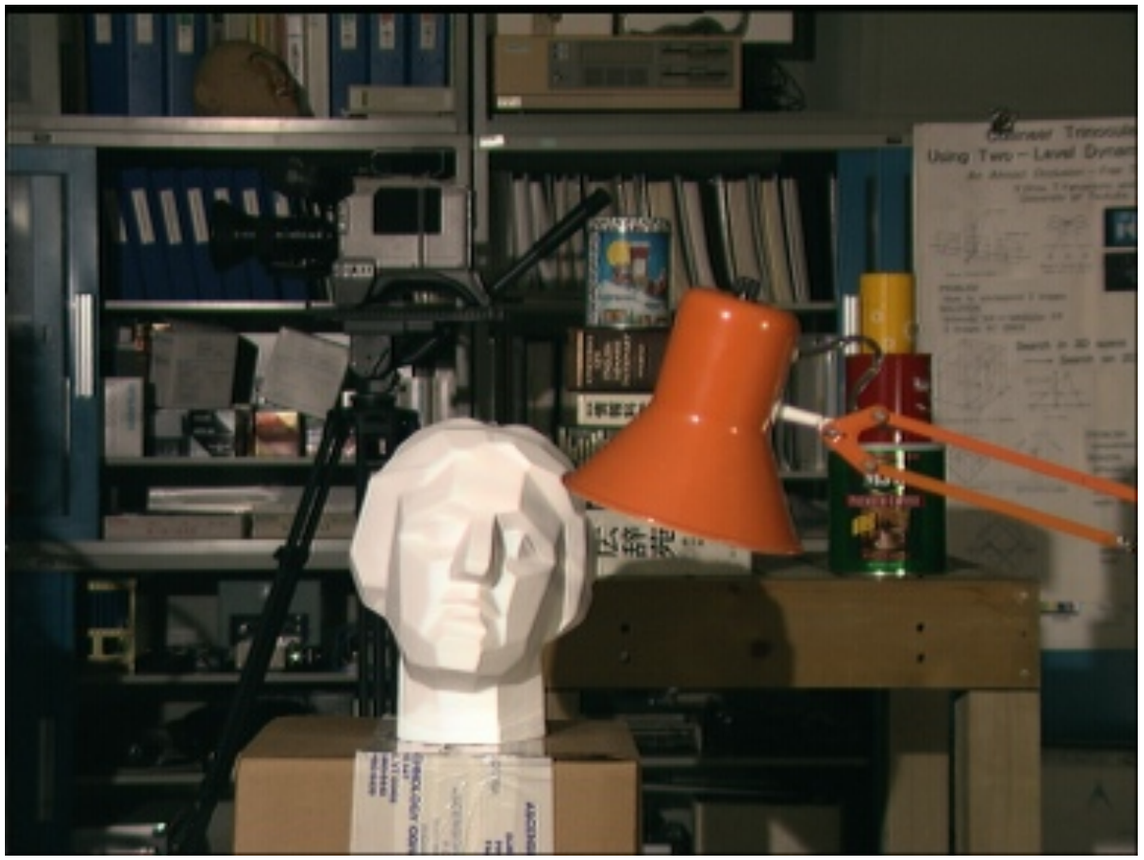
- Better results with adaptive window:
  - T. Kanade and M. Okutomi: A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment, Proc. International Conference on Robotics and Automation, 1991.
  - D. Scharstein and R. Szeliski: Stereo matching with nonlinear diffusion. International Journal of Computer Vision, 28(2):155-174, July 1998 .

[Seitz]

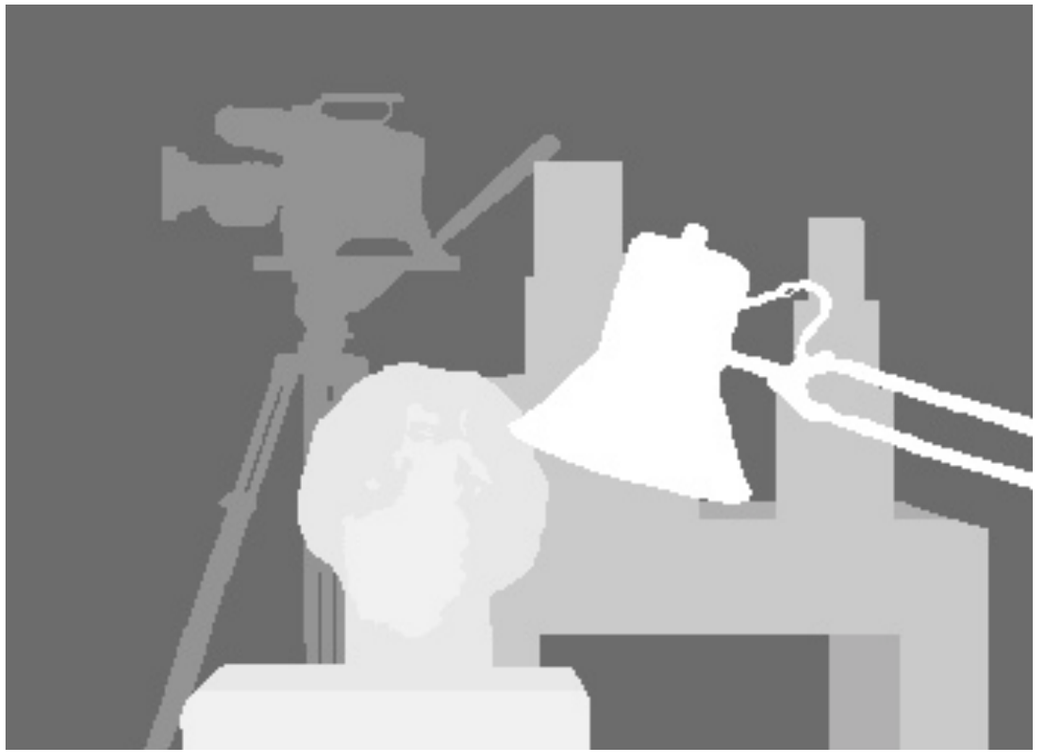
Preview: Stereo results

TECHNISCHE UNIVERSITÄT DARMSTADT

- Data from University of Tsukuba:



Scene

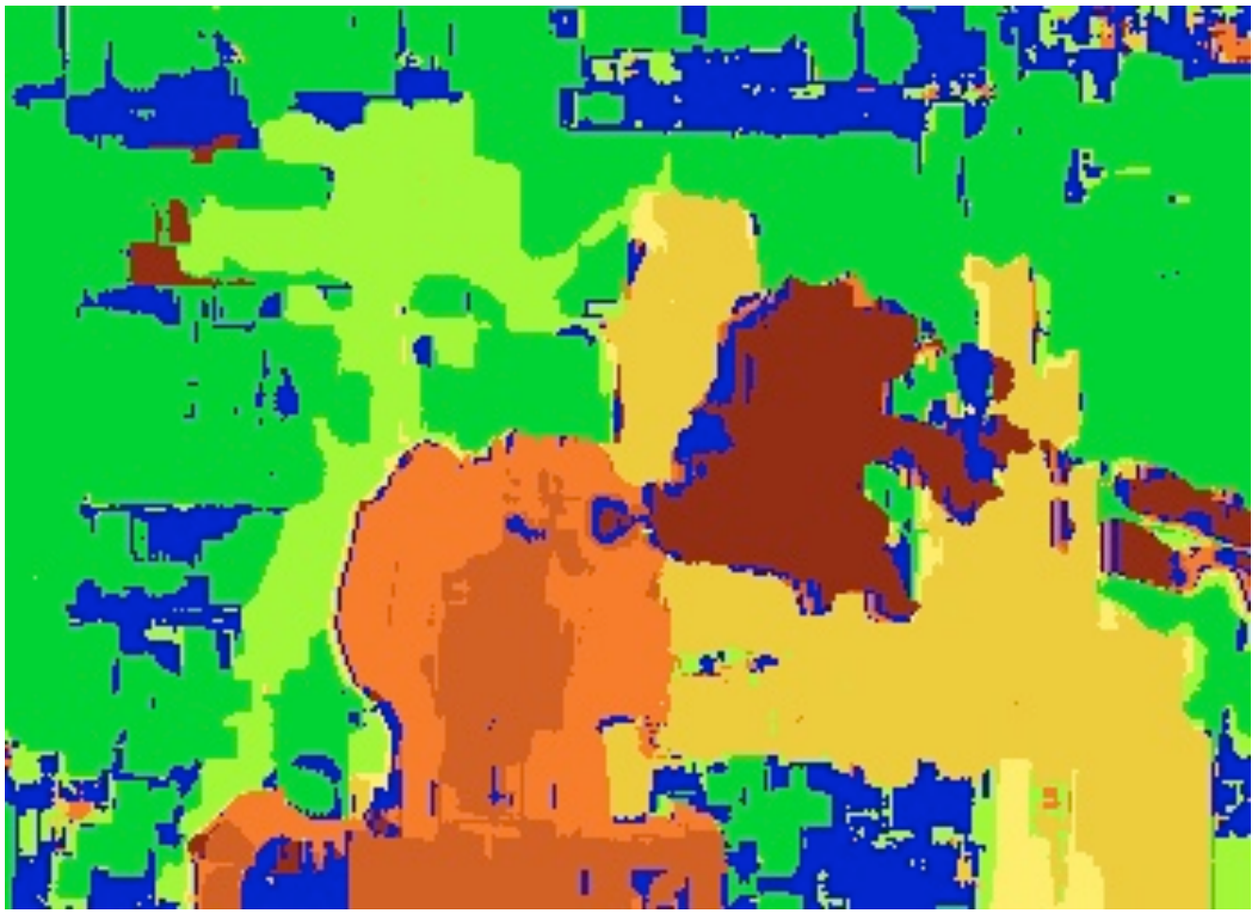


Ground truth

[Seitz]

Results with window correlation

TECHNISCHE UNIVERSITÄT DARMSTADT



Window-based matching
(best window size)

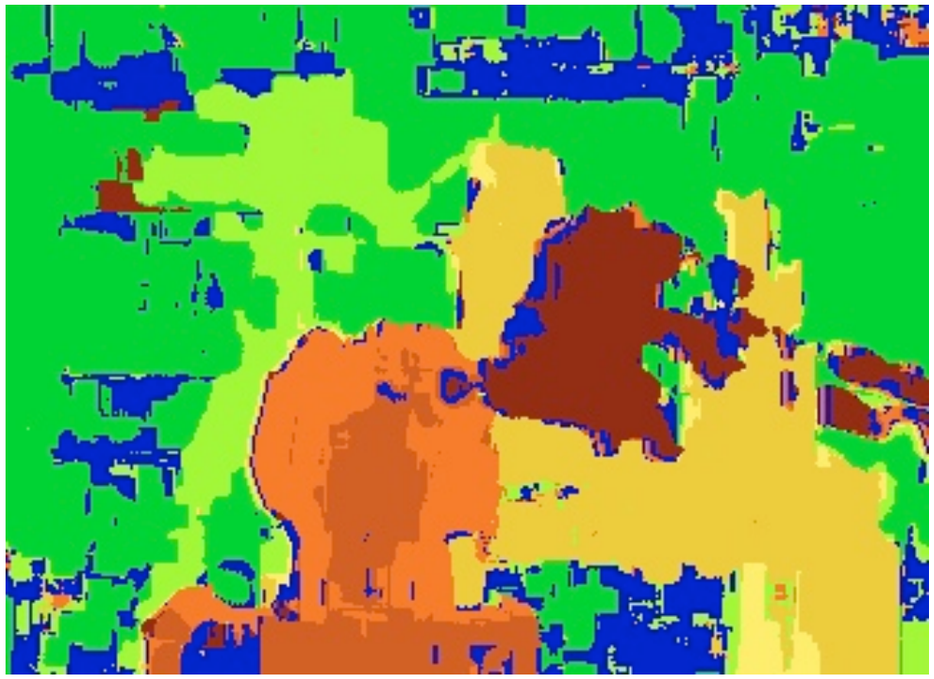
Ground truth

[Seitz]

How can we improve window-based matching?

TECHNISCHE UNIVERSITÄT DARMSTADT

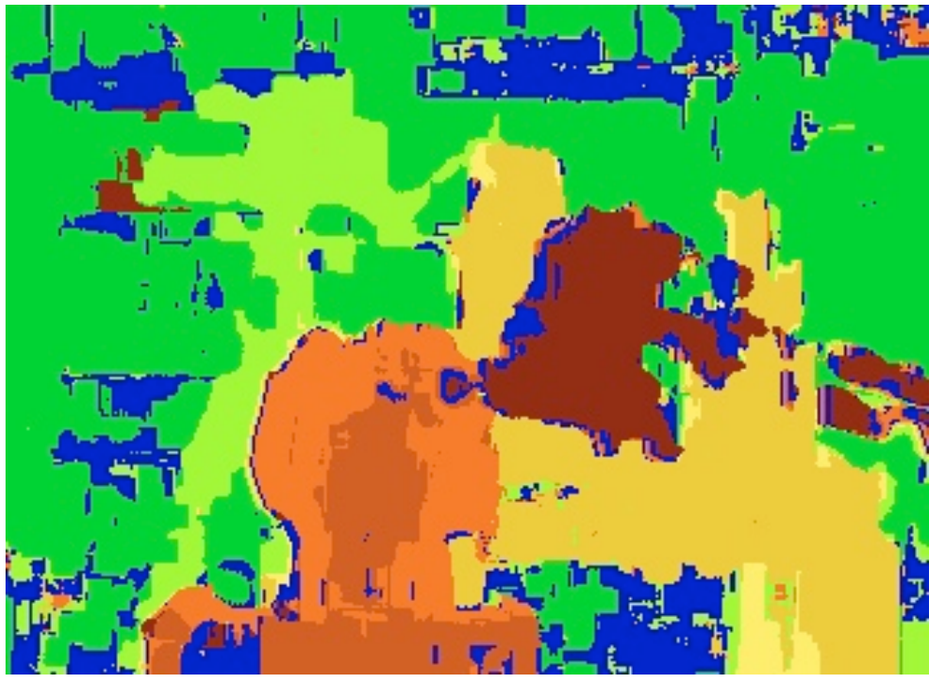
- The similarity constraint is local
  - each reference window is matched independently
  - other points do not influence the result



How can we improve window-based matching?

TECHNISCHE UNIVERSITÄT DARMSTADT

- The similarity constraint is local
  - each reference window is matched independently
  - other points do not influence the result
- Need to enforce non-local correspondence constraints



Slide by Lana Lazebnik

Where do we go from here?

TECHNISCHE UNIVERSITÄT DARMSTADT

Graph cuts

Ground truth

- How can be improve the results?
  - We need to impose spatial regularity to improve results!
  - Stay tuned for Computer Vision 2