CSED441:Introduction to Computer Vision (2016F)

Lecture7: Image Matching & Geometric Model Fitting (1)

Minsu Cho

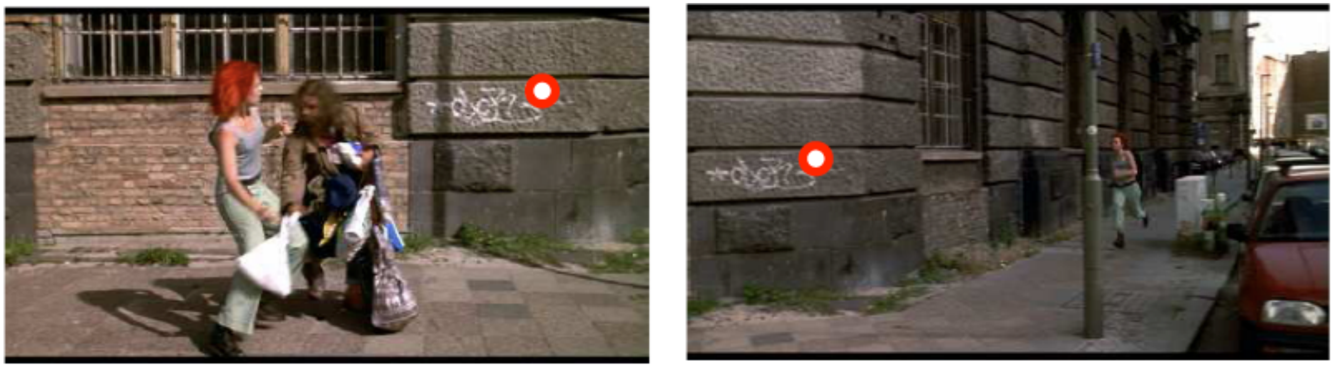
CSE, POSTECH

mscho@postech.ac.kr

Some materials for this lecture are the courtesy of Josef Sivic, Andrew Zisserman, Bohyung Han

Image matching with local features

- The goal: establish correspondence between images



$$\mathbf{x} = \mathrm{P}\mathbf{X} \qquad \mathbf{x}' = \mathrm{P}'\mathbf{X}$$

P : 3×4 matrix

$\mathbf{X}$: 4-vector

$\mathbf{x}$: 3-vector

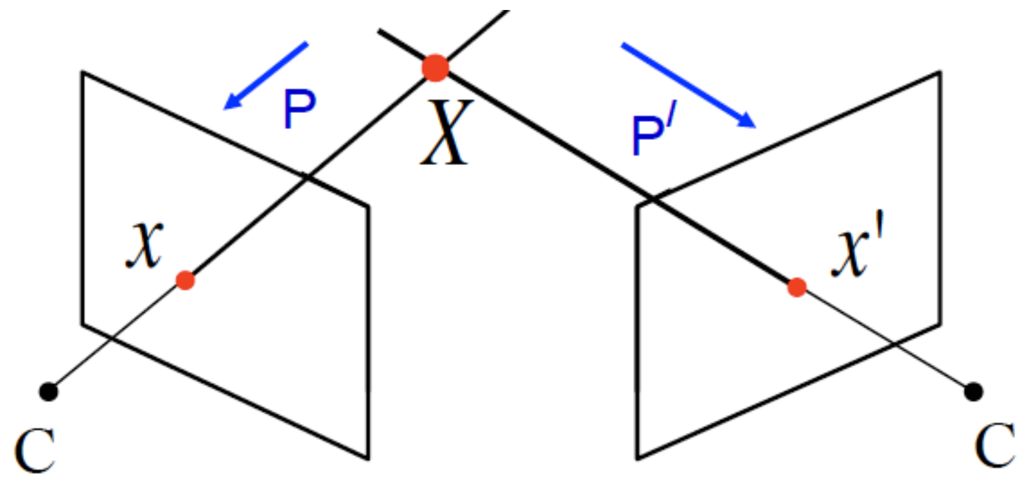


Image matching with local features

- Example I: Wide baseline matching and 3D reconstruction

Image matching with local features

- Example I: Wide baseline matching and 3D reconstruction

Image matching with local features

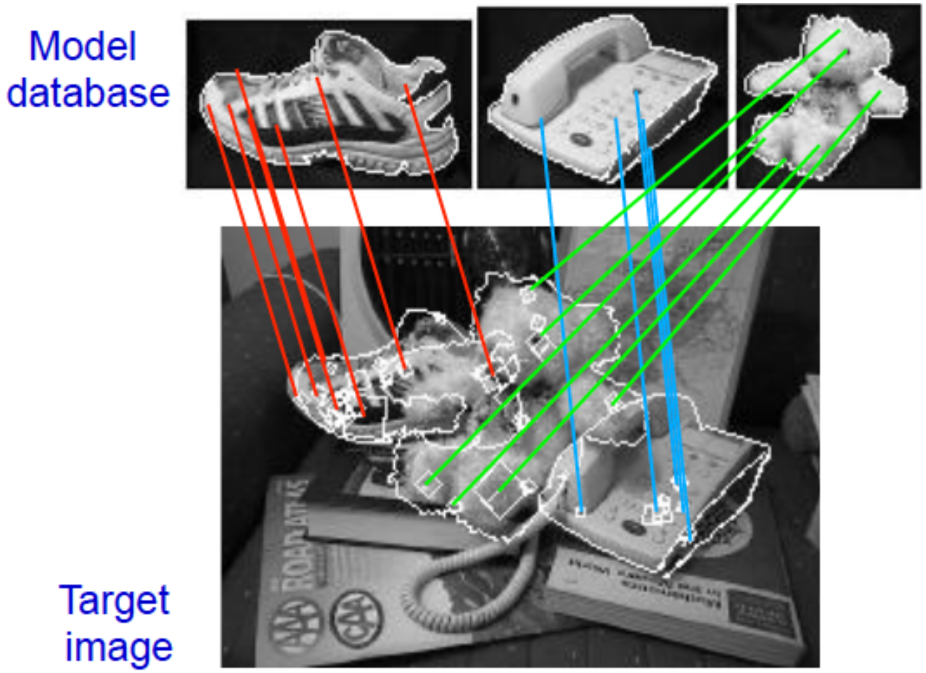
- [Agarwal, Snavely, Simon, Seitz, Szeliski, ICCV'09] – Building Rome in a Day

57,845 downloaded images, 11,868 registered images.

Image matching with local features

- Example II: Object recognition

Establish correspondence between the target image and (multipl e) images in the model database.

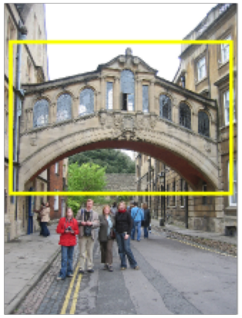


[D. Lowe, 1999]

Image matching with local features

- Example III: Visual search

Given a query image, find images depicting the same place / obj ect in a large unordered image collection.

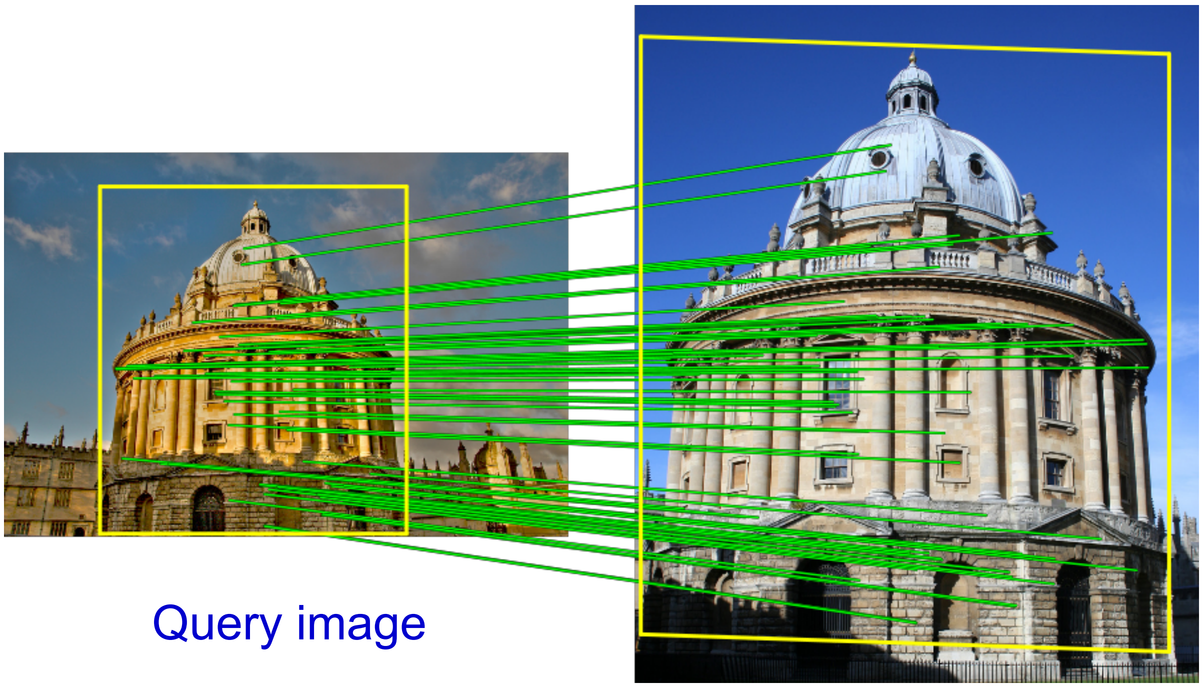


Find these landmarks ...in these images and 1M more

Image matching with local features

- Example III: Visual search

Establish correspondence between the query image and all imag es from the database depicting the same object / scene.



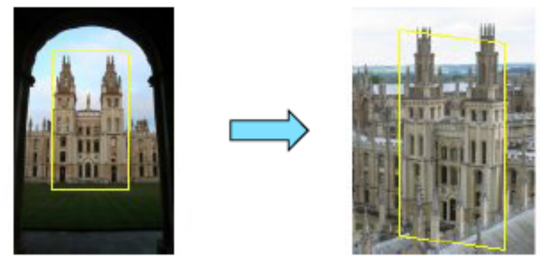
Query image

Database image(s)

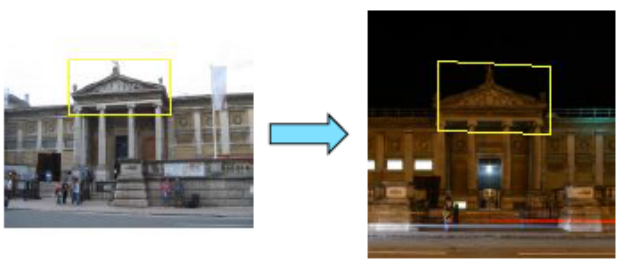
Why is it difficult?

- Want to establish correspondence despite possibly large chan ges in scale, viewpoint, lighting and partial occlusion

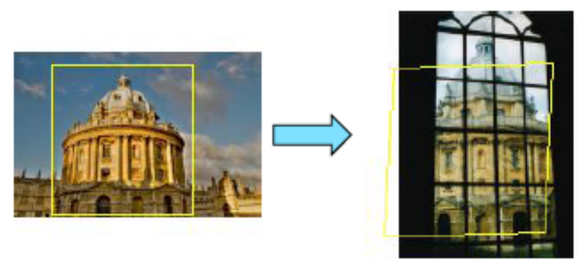
Scale



Viewpoint



Lighting



Occlusion

… and the image collection can be very large (e.g. 1M images)

Standard approach to image matching

- 0. Pre-processing:
  - Detect local features.
  - Extract descriptor for each feature.

- 1. Matching: Establish tentative (putative) correspondences based on local appearance of individual features (descriptors).

- 2. Verification: Verify matches based on semi-local / global geometric relations. (= Fitting)

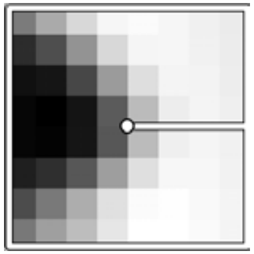
Challenges and Strategy

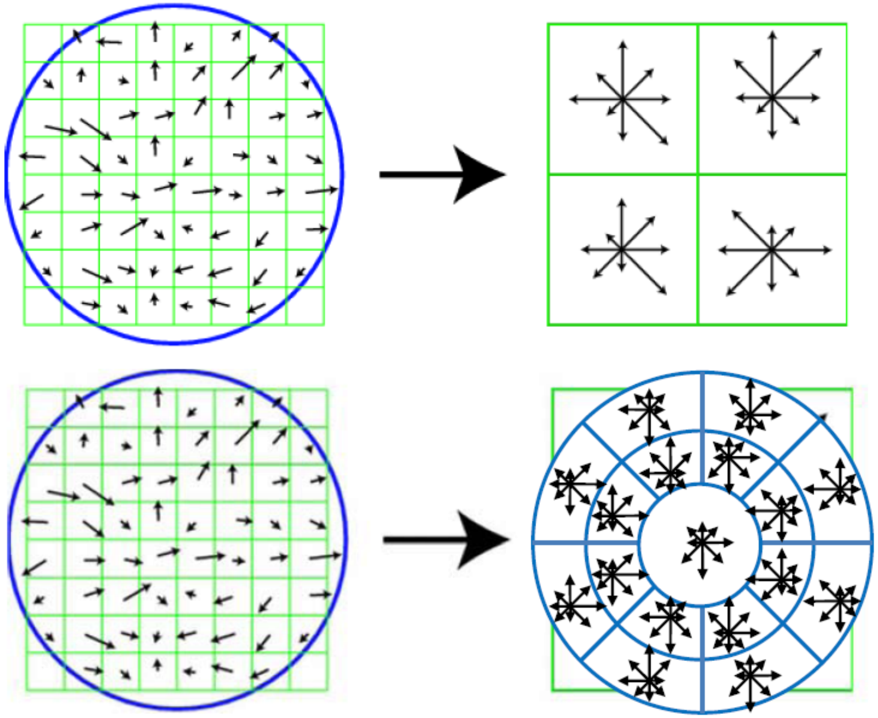
- Challenges in matching and fitting
  - Noises in data
  - Outliers
  - Missing data
- Strategy to robust matching and fitting

  We hope to handle noises, outliers and missing data effectively.
  - Extract features
  - Compute putative matches
  - Loop:
    - *Hypothesize* transformation T
    - *Verify* the transformation (search for other matches consistent with T)

Feature Descriptors

- Examples of feature descriptor
  - Template
  - SIFT, PCA-SIFT
  - HOG
  - Bias and gain normalization (MOPS)
  - Speeded Up Robust Features (SURF)
  - Gradient location-orientation histogram (GLOH)





Comparison between Features

- Feature representation
  - Typically, by a (normalized) vector

$$\boldsymbol{u} = (u_1, u_2, \dots, u_n)^{\mathrm{T}} \qquad \boldsymbol{v} = (v_1, v_2, \dots, v_n)^{\mathrm{T}}$$

- Distance between two feature descriptors
  - Various distance measures are available.
  - Sum of Squared Differences (SSD)

$$\mathrm{SSD}(\boldsymbol{u}, \boldsymbol{v}) = \|\boldsymbol{u} - \boldsymbol{v}\|^2 = \sum_i (u_i - v_i)^2$$

  - Cosine measure (inner-product)

$$\rho(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|} = \frac{\sum_i u_i v_i}{\sqrt{\sum_i u_i^2}\sqrt{\sum_i v_i^2}}$$

Comparison between Features

- Distance between two feature descriptors (cont'd)
  - Normalized correlation

$$\rho(\boldsymbol{u},\boldsymbol{v}) = \frac{(\boldsymbol{u}-\bar{\boldsymbol{u}})^{\mathrm{T}}(\boldsymbol{v}-\bar{\boldsymbol{v}})}{\|\boldsymbol{u}-\bar{\boldsymbol{u}}\|\|\boldsymbol{v}-\bar{\boldsymbol{v}}\|} = \frac{\sum_i (u_i-\bar{u}_i)(v_i-\bar{v}_i)}{\sqrt{\sum_i (u_i-\bar{u}_i)^2}\sqrt{\sum_i (v_i-\bar{v}_i)^2}}$$

  - Bhattacharyya distance

$$\rho(\boldsymbol{u},\boldsymbol{v}) = \left(1-\sum_i \sqrt{u_i v_i}\right)^{1/2}$$

$\boldsymbol{u},\boldsymbol{v}$: normalized histogram

  - Equivalence between Bhattacharyya coefficient and cosine measure

$$\sum_i \sqrt{u_i v_i} = \frac{\sum_i \sqrt{u_i v_i}}{\sqrt{\sum_i \left(\sqrt{u_i}\right)^2}\sqrt{\sum_i \left(\sqrt{v_i}\right)^2}} = \frac{\sqrt{\boldsymbol{u}}^{\mathrm{T}}\sqrt{\boldsymbol{v}}}{\|\sqrt{\boldsymbol{u}}\|\|\sqrt{\boldsymbol{v}}\|}$$

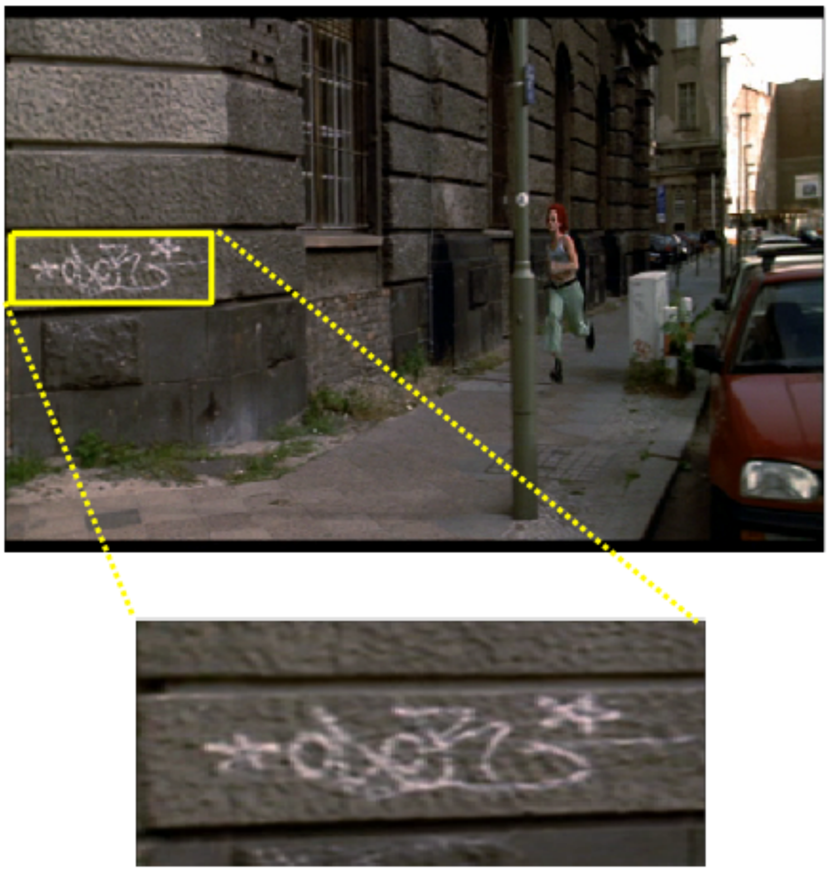
How to Match Feature Points in Two Views?

- Global threshold
  - Find the feature points closer than a certain threshold in the other view
  - Drawback
    - Difficult to find the optimal global threshold
    - Non-unique matching
- Nearest neighbor
  - Find the closest feature point in the other view.
  - Drawback
    - Many false positives
- Nearest neighbor distance ratio
  - Ratio between the distance to the nearest neighbor and the second nearest neighbor

There will be a lot of errors in matching, so you need to reject outliers effectively.

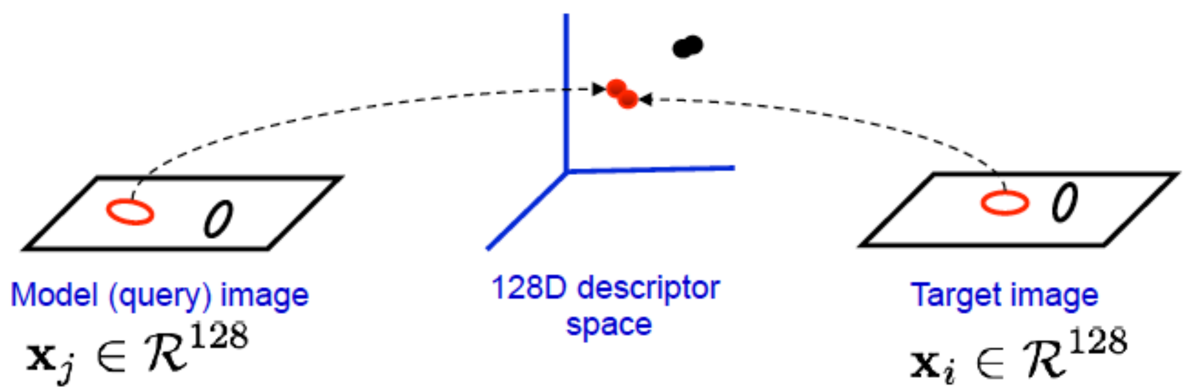
Image Matching

- Example I: Two images -"Where is the Graffiti?"



Step 1. Establish tentative correspondence

Establish tentative correspondences between object model image and target image by nearest neighbour matching on SIFT vectors



Need to solve some variant of the "nearest neighbor problem" for all feature vectors, $\mathbf{x}_j \in \mathcal{R}^{128}$, in the query image:

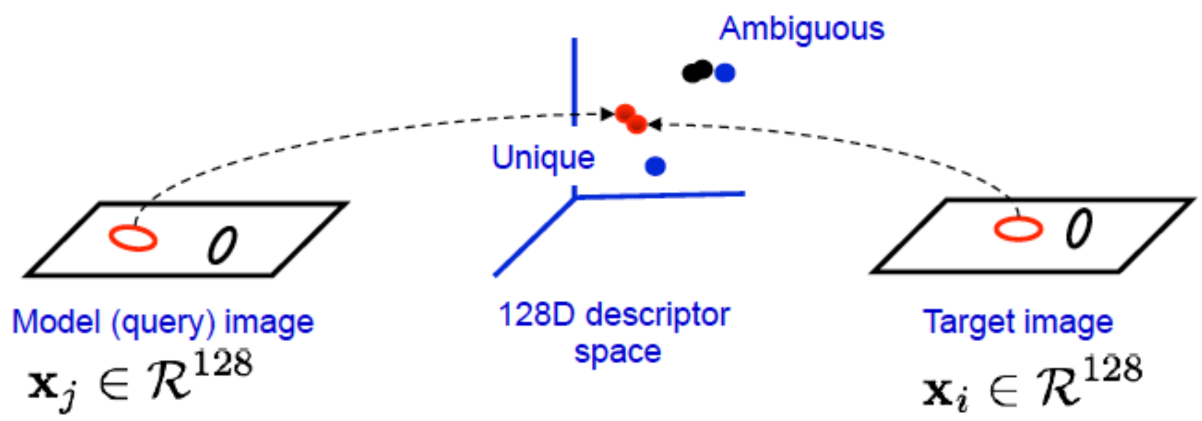
$$\forall j \; NN(j) = \arg\min_i ||\mathbf{x}_i - \mathbf{x}_j||,$$

where, $\mathbf{x}_i \in \mathcal{R}^{128}$, are features in the target image.

Can take a long time if many target images are considered.

Step 1. Establish tentative correspondence

Examine the distance to the 2nd nearest neighbour [Lowe, IJCV 2004]



If the 2nd nearest neighbour is much further than the 1st nearest neighbour
Match is more "unique" or discriminative.

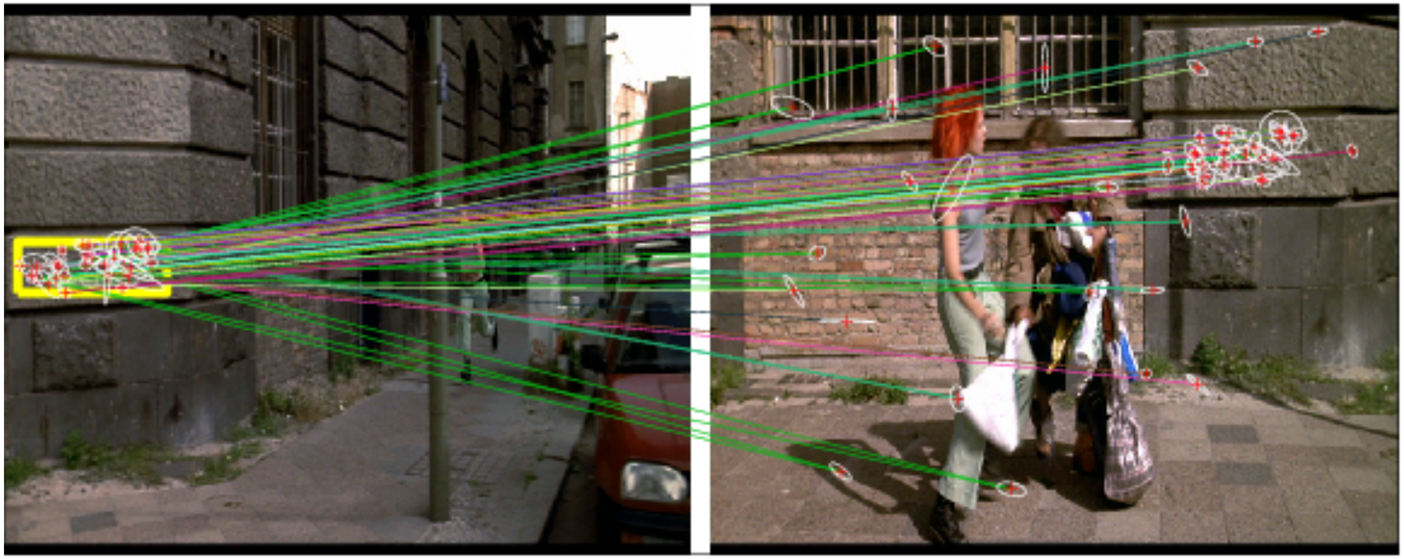
Measure this by the ratio: $r = d_{1NN} / d_{2NN}$

r is between 0 and 1
r is small the match is more unique.

Works very well in practice.

Image Matching

- Problem with matching on local descriptors alone



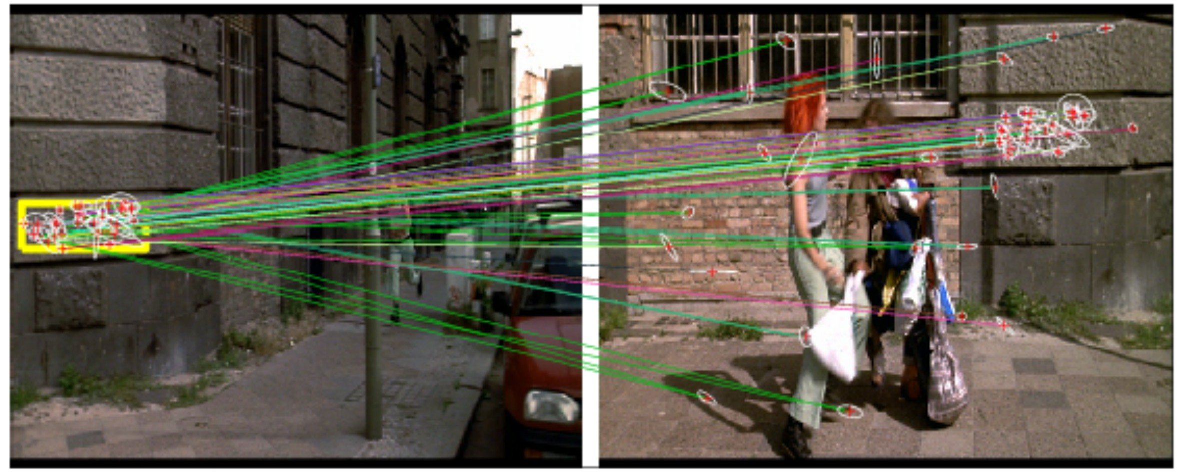
  - Too much individual invariance
  - Affine adaptation of regions is not perfect
  - Locally appearance can be ambiguous

- Solution: semi-local/global spatial relations to verify matches.

Image Matching

Initial matches

Nearest-neighbor search based on appearance descriptors alone.



After spatial verification

Step 2: Spatial verification

- Semi-local constraints
  - Constraints on spatially close-by matches

- Global geometric relations
  - Require a consistent global relationship between all matches

Step 2: Spatial verification

- Semi-local constraints:
  - Example I. – neighbourhood consensus

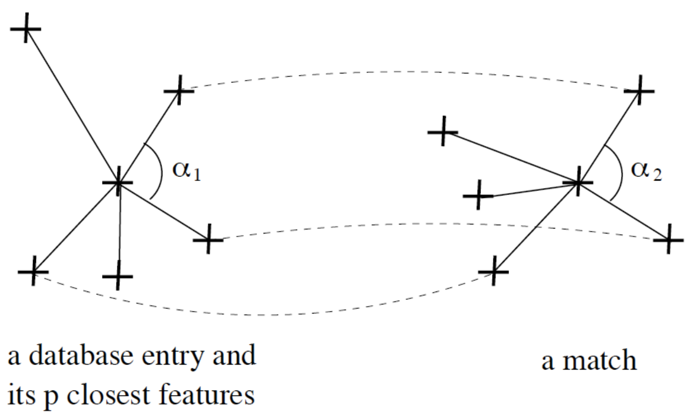
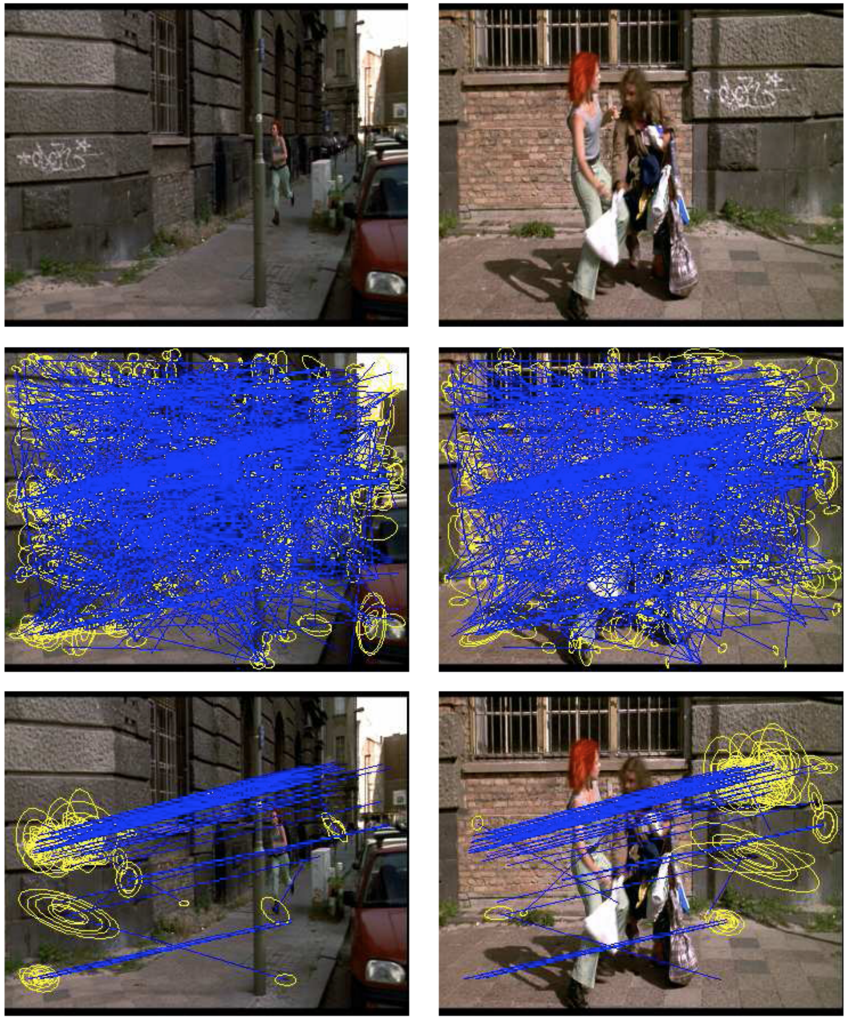


Fig. 4. Semi-local constraints : neighbours of the point have to match and angles have to correspond. Note that not all neighbours have to be matched correctly.

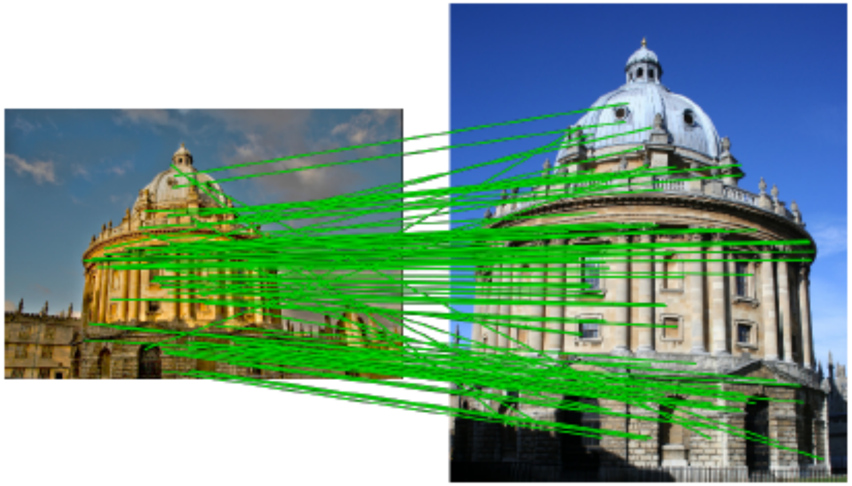
[Schmid&Mohr, PAMI 1997]



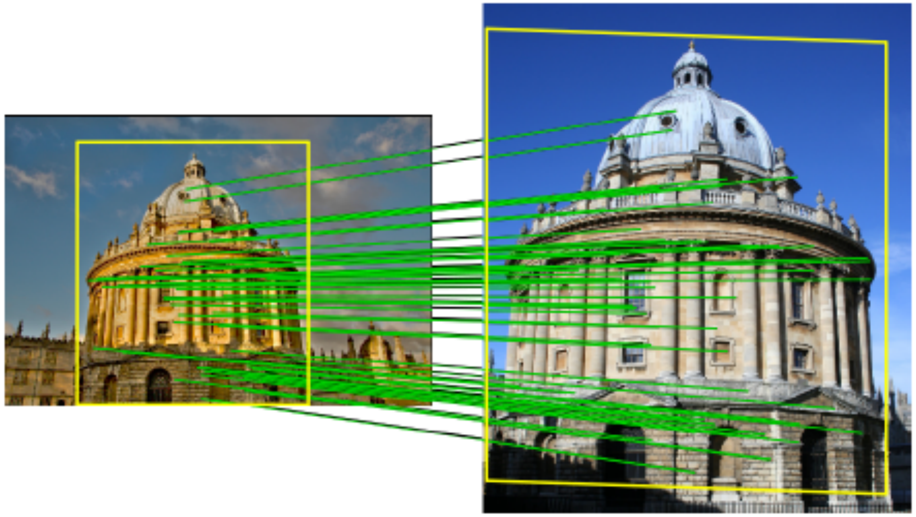
[Schaffalitzky & Zisserman, CIVR 2004]

Geometric verification with global constraints

- All matches must be consistent with a global geometric relatio n / transformation.
- Need to simultaneously (i) estimate the geometric relation / tr ansformation and (ii) the set of consistent matches



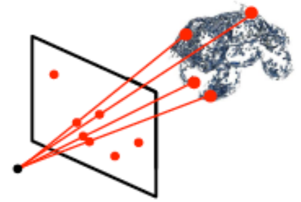
Tentative matches



Matches consistent with an affine transformation

Geometric verification with global constraints

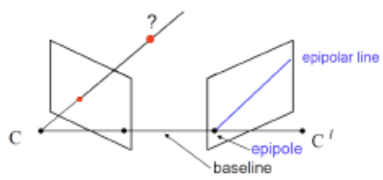
Examples of global constraints

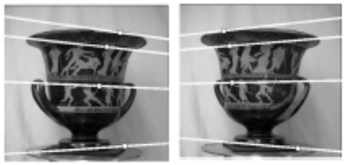


1 view and known 3D model.

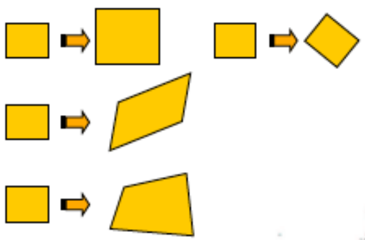
- Consistency with a (known) 3D model.

2 views





- Epipolar constraint
- **2D transformations**
  - Similarity transformation
  - Affine transformation
  - Projective transformation



N-views

Are images consistent with a 3D model?

Geometric verification with global constraints

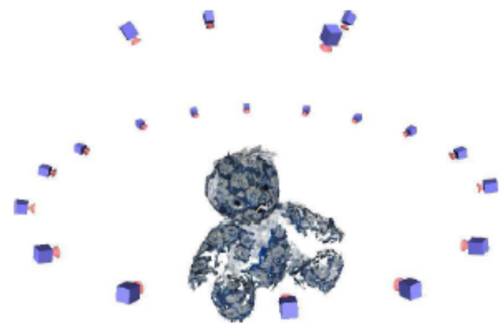
3D constraint: example

- Matches must be consistent with a 3D model

Offline: Build a 3D model

3 (out of 20) images used to build the 3D model

(a)



Recovered 3D model

At test time:

Object recognized in a previously unseen pose

Recovered pose

(d)

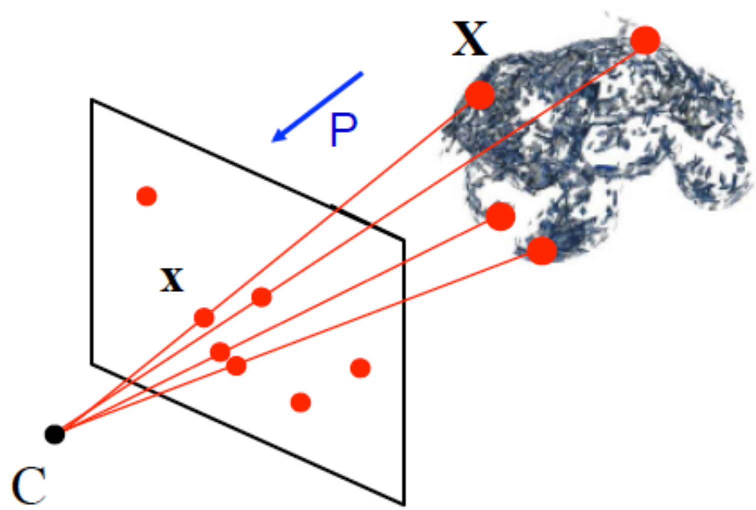
[Lazebnik, Rothganger, Schmid, Ponce, CVPR'03]

Geometric verification with global constraints

3D constraint: example

Given 3D model (set of known 3D points X's) and a set of measured 2D image points x,

find camera matrix P and a set of geometrically consistent correspondences x↔X.



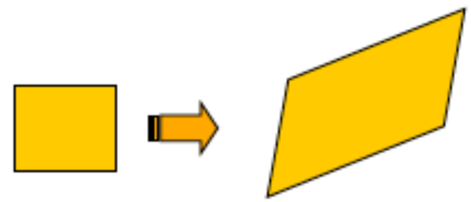
$$\mathbf{x} = \mathbf{P}\mathbf{X}$$

P : 3×4 matrix
$\mathbf{X}$: 4-vector
$\mathbf{x}$: 3-vector

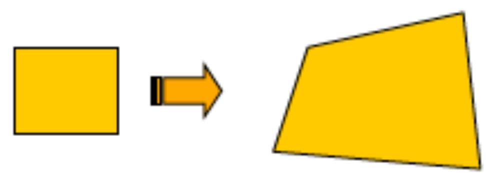
Geometric Relations

- Similarity
  - (translation, scale, rotation)

- Affine



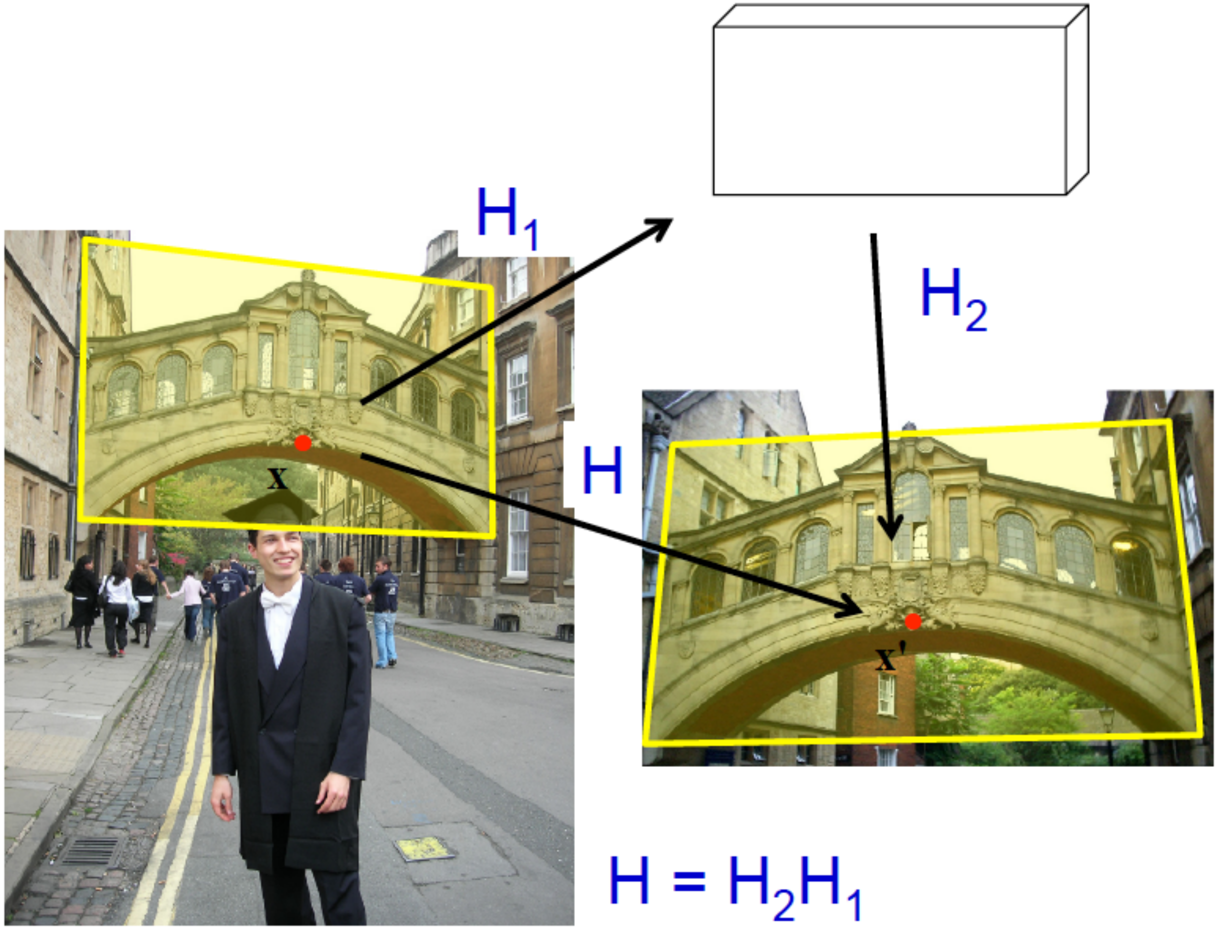
- Projective
  - (homography)



- Why are 2D planar transformations important?

Geometric Relations

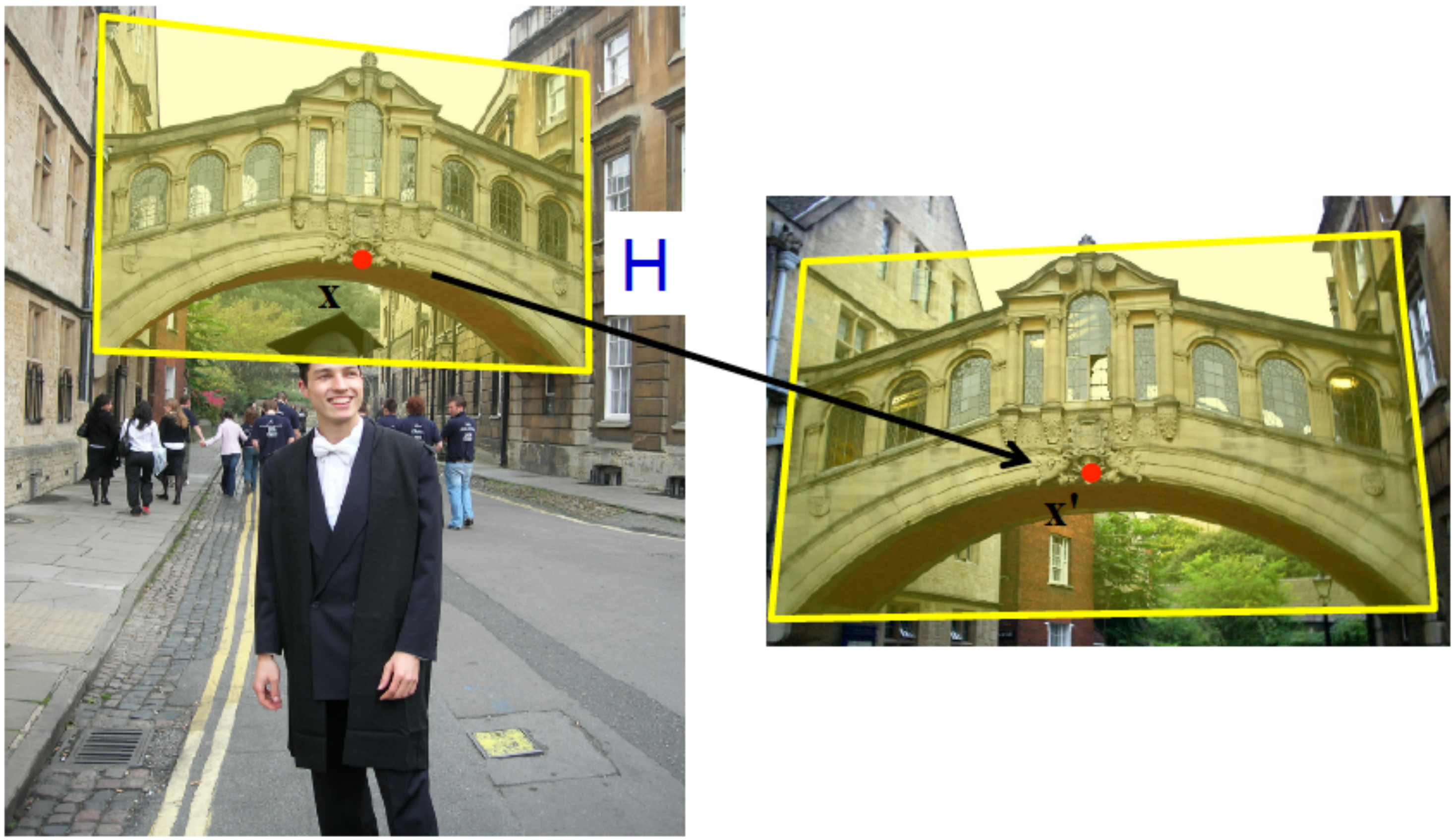
Planes in the scene induce *homographies*



$H = H_2 H_1$

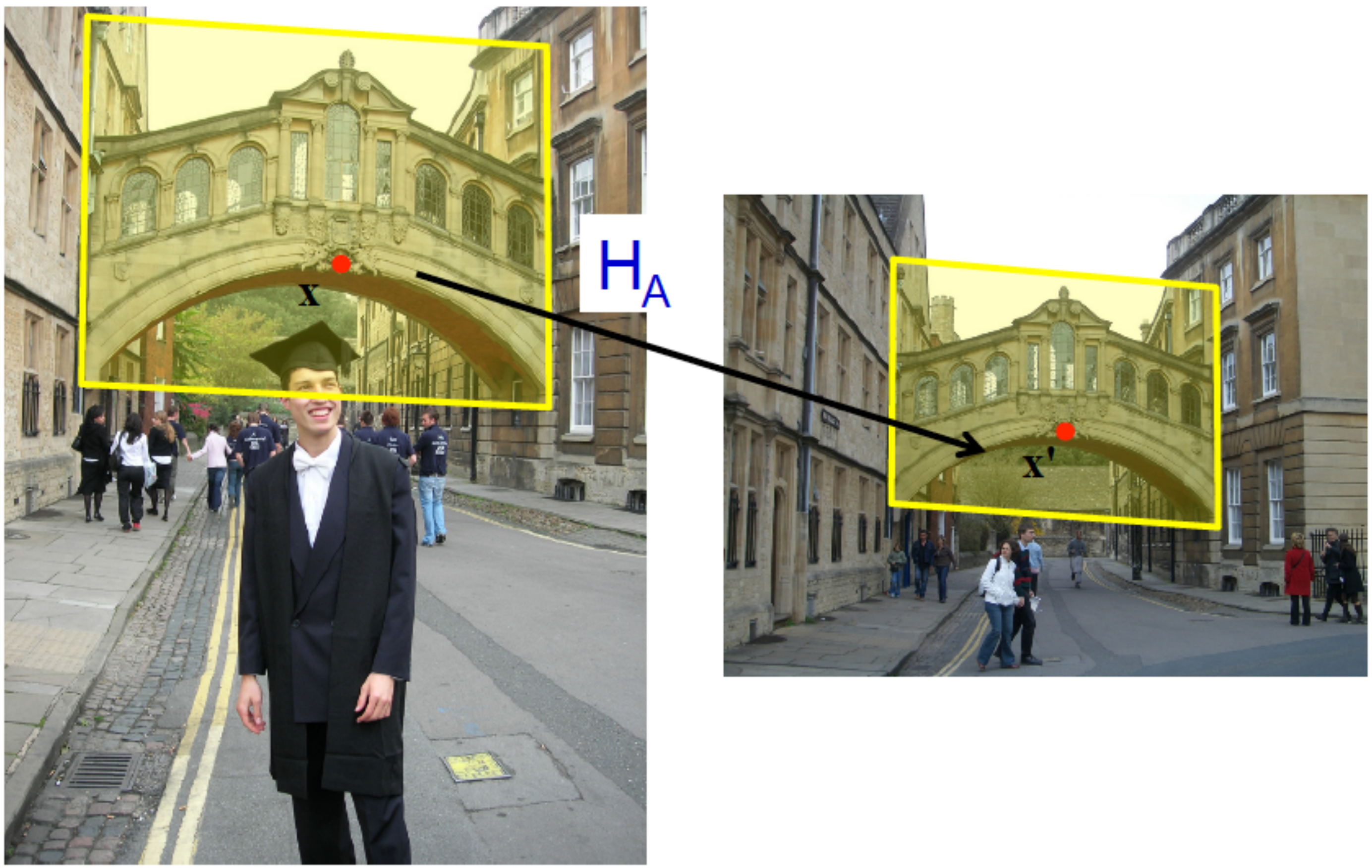
Geometric Relations

Points on the plane transform as x' = H x, where x and x' are image points (in homogeneous coordinates), and H is a 3x3 matrix.



Geometric Relations

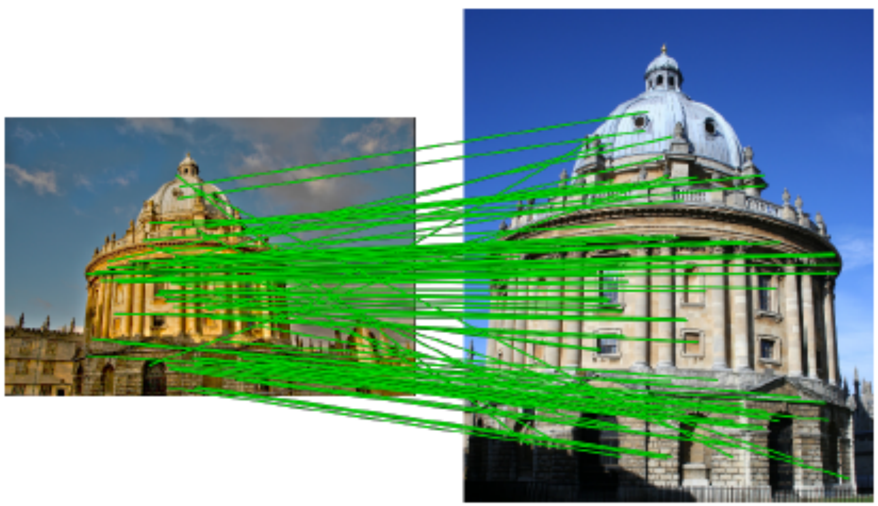
Homography is often approximated well by 2D affine geometric transformation



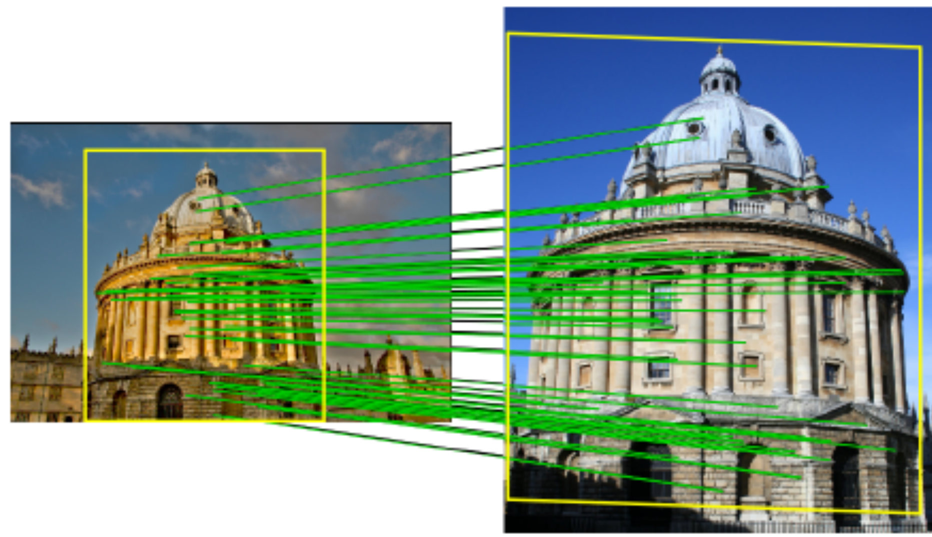
Geometric Relations

- Homography is often approximated well by 2D affine geometri c transformation

Two images with similar camera viewpoint



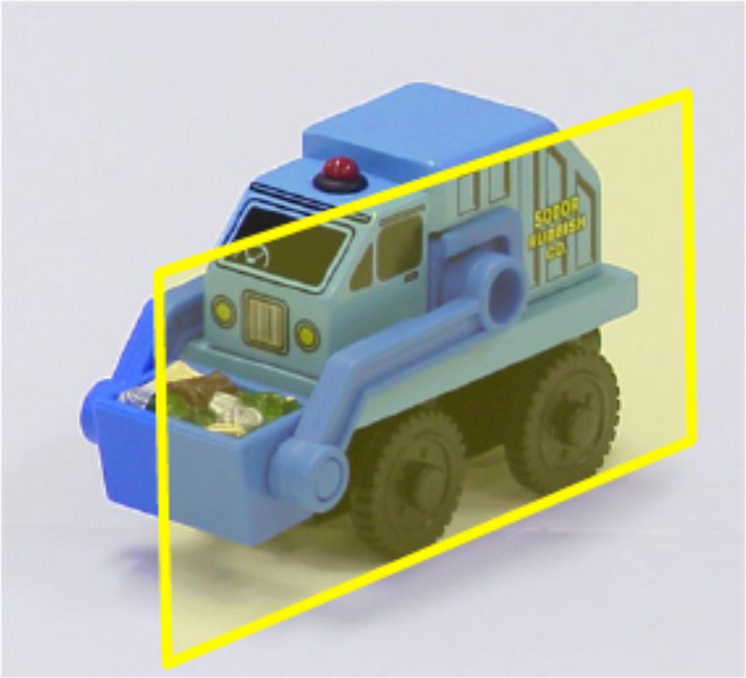
Tentative matches



Matches consistent with an affine transformation

Geometric Relations

- Estimating 2D affine transformation
  - Approximates viewpoint changes for roughly planar objects and roughl y orthographic cameras
  - Can be used to initialize fitting for more complex models



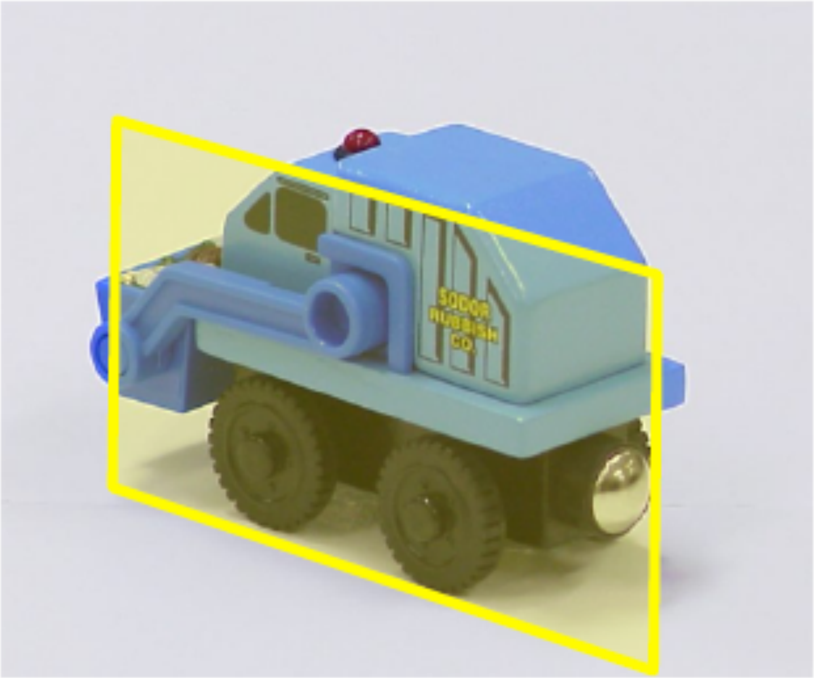


Image Stitching

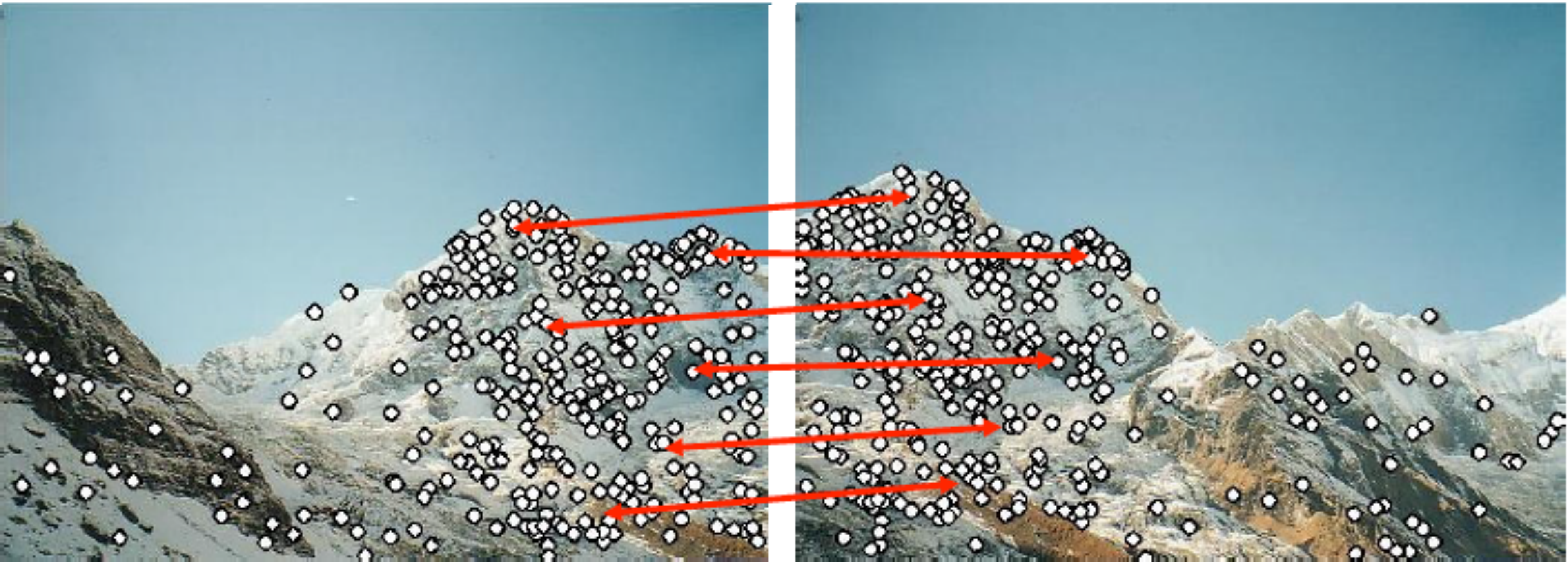
Estimation of homography

Image Stitching

- Motivation: panorama stitching
- We have two images – how do we combine them?

Image Stitching

- Motivation: panorama stitching
- We have two images – how do we combine them?



- Extract & match features

Image Stitching

- Motivation: panorama stitching
- We have two images – how do we combine them?

- Align images

Image Stitching

Preliminaries for Geometric Image Matching

- **Point representation**
- **Transformation**
- **Camera model & 3D-2D projection**
- Geometric models between images
- Model fitting schemes

Homogeneous Coordinates

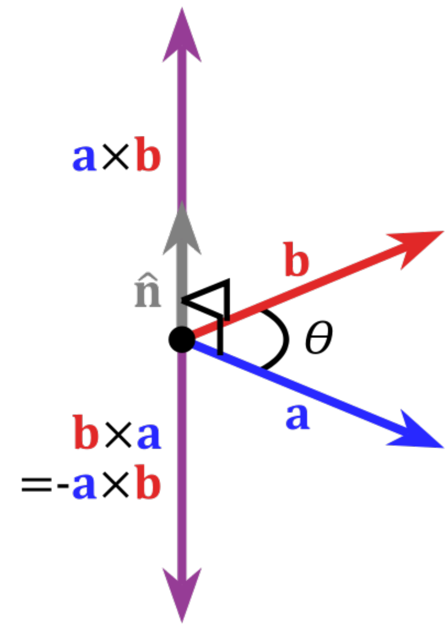
- Overall scaling is NOT important.

$$(x, y, 1)^{\mathrm{T}} \Leftrightarrow (\alpha x, \alpha y, \alpha)^{\mathrm{T}} \qquad \alpha \neq 0$$

- It is useful in projective geometry for
  - Basic representation of geometric entities: points, lines, planes, etc.
  - Representation of point and line at infinity
  - Transformations
  - Others…

Cross Product

- Binary operation on two vectors
  - Perpendicular to both of the two vectors
  - The magnitude of the product equals the area of a parallelogram with the vectors for sides



$$\boldsymbol{a}_1 \times \boldsymbol{a}_2 = [\boldsymbol{a}_1]_\times \boldsymbol{a}_2 = \begin{pmatrix} 0 & -c_1 & b_1 \\ c_1 & 0 & -a_1 \\ -b_1 & a_1 & 0 \end{pmatrix} \begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix}$$

$$\boldsymbol{a}_1 \times \boldsymbol{a}_2 = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix} = \begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix} \boldsymbol{i} - \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix} \boldsymbol{j} + \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \boldsymbol{k}$$

$$= b_1 c_2 \boldsymbol{i} + a_2 c_1 \boldsymbol{j} + a_1 b_2 \boldsymbol{k} - b_2 c_1 \boldsymbol{i} - a_1 c_2 \boldsymbol{j} - a_2 b_1 \boldsymbol{k}$$

Representation of Line and Point in 2D

- Homogeneous representation of line

$$ax + by + c = 0 \Leftrightarrow (a, b, c)^{\mathrm{T}}$$

- Homogeneous representation of point
  - Point in 2D: $(x, y, 1)^{\mathrm{T}}$
- Special cases
  - Ideal point (or point at infinity)
    - Point that two parallel lines meet
    - Does not corresponds to any finite point in $\Re^2$.

$$\boldsymbol{x}_{ideal} = (x, y, 0)^{\mathrm{T}} \qquad (x, y)\text{: direction of the lines}$$

  - Line at infinity: line that Ideal points meet

$$\boldsymbol{l}_{\infty} = (0,0,1)^{\mathrm{T}}$$

  - Ideal point is on line at infinity.

$$\boldsymbol{x}_{ideal}^{\mathrm{T}} \boldsymbol{l}_{\infty} = 0$$

Relationship between Lines and Points

- Lines vs. points
  - A point $\boldsymbol{x} = (x, y, 1)^{\mathrm{T}}$ lies on the line $\boldsymbol{l} = (a, b, c)^{\mathrm{T}}$

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{l} = \boldsymbol{l}^{\mathrm{T}}\boldsymbol{x} = 0 \Leftrightarrow \boldsymbol{x} \cdot \boldsymbol{l} = \boldsymbol{l} \cdot \boldsymbol{x} = 0$$

  - Intersection of two lines $\boldsymbol{l} = (a, b, c)^{\mathrm{T}}$ and $\boldsymbol{l}' = (a', b', c')^{\mathrm{T}}$

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{l} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{l}' = 0 \Longrightarrow \boldsymbol{x} = \boldsymbol{l} \times \boldsymbol{l}'$$

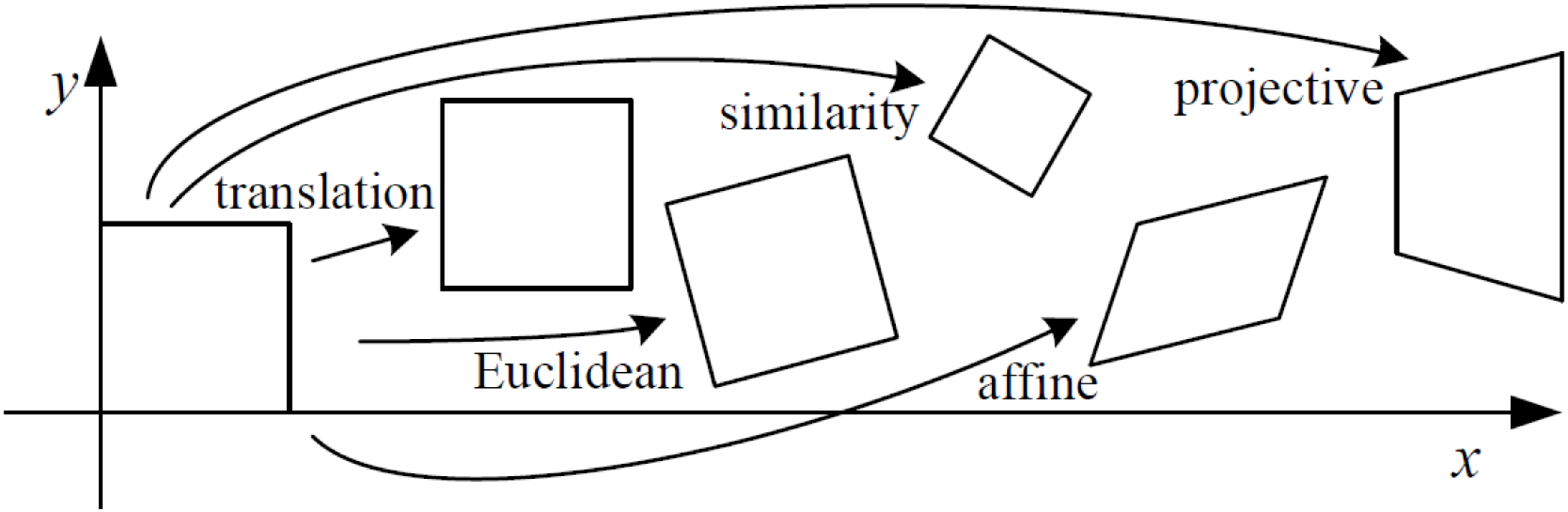
  - Line through two points $\boldsymbol{x} = (x, y, 1)^{\mathrm{T}}$ and $\boldsymbol{x}' = (x', y', 1)^{\mathrm{T}}$

$$\boldsymbol{l}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{l}^{\mathrm{T}}\boldsymbol{x}' = 0 \Longrightarrow \boldsymbol{l} = \boldsymbol{x} \times \boldsymbol{x}'$$

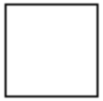
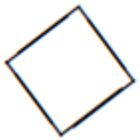
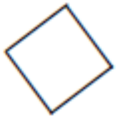
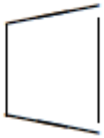
  - Duality of point and line: Points and lines can be swapped.

2D Transformation

- Schematic view of 2D planar transformation



Hierarchy of 2D Transformation

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	3	lengths	
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	4	angles	
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array}\right]_{2\times 3}$	6	parallelism	
projective	$\left[\begin{array}{c} \tilde{\boldsymbol{H}} \end{array}\right]_{3\times 3}$	8	straight lines	

2D Transformation

- Translation: $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t} \Leftrightarrow \boldsymbol{x}' = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix} \boldsymbol{x}$
- Euclidean (rigid): $\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t} \Leftrightarrow \boldsymbol{x}' = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix} \boldsymbol{x}$
- Similarity: $\boldsymbol{x}' = s\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t} \Leftrightarrow \boldsymbol{x}' = \begin{pmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix} \boldsymbol{x}$
- Affine: $\boldsymbol{x}' = \boldsymbol{A}\boldsymbol{x} \Leftrightarrow \boldsymbol{x}' = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}$
- Projective: $\boldsymbol{x}' = \boldsymbol{H}\boldsymbol{x} \Leftrightarrow \boldsymbol{x}' = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \boldsymbol{x}$

Hierarchy of 2D Transformation (Class 1)

- Isometry (distance-preserving map between metric spaces)

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

  - Preserve Euclidean distances and angles
  - 3 degrees of freedom: translation and rotation

$$\boldsymbol{x}' = \boldsymbol{H}_E \boldsymbol{x} \Leftrightarrow \boldsymbol{x}' = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix} \boldsymbol{x} \qquad \boldsymbol{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Hierarchy of Transformations (Class 2)

- Similarity transformation
  - Isometry composed with an isotropic scaling

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

  - 4 degrees of freedom: $\{t_x, t_y, \theta, s\}$
  - preserve the "shape" (equi-form)
  - invariants: angle, parallel line, ratio of length

$$\boldsymbol{x}' = \boldsymbol{H}_S \boldsymbol{x} \Leftrightarrow \boldsymbol{x}' = \begin{pmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix} \boldsymbol{x}$$

Hierarchy of Transformations (Class 3)

- Affine transform
  - Non-singular linear transformation followed by a translation

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

  - Include non-isotropic scaling
  - 6 degrees of freedom
  - Invariants: parallel lines, ratio of length of parallel line segments, ratio of areas

$$\boldsymbol{x}' = \boldsymbol{H}_A \boldsymbol{x} \Leftrightarrow \boldsymbol{x}' = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix} \boldsymbol{x}$$

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}} = (\boldsymbol{U}\boldsymbol{V}^{T})(\boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}) \\ &= \boldsymbol{R}(\theta)\boldsymbol{R}(-\phi)\boldsymbol{S}\boldsymbol{R}(\phi) \end{aligned} \qquad \boldsymbol{S} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Hierarchy of Transformations (Class 4)

- Projective transformation

$$\boldsymbol{x}' = \boldsymbol{H}_P \boldsymbol{x} \Leftrightarrow \boldsymbol{x}' = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{t} \\ \boldsymbol{v}^{\mathrm{T}} & v \end{pmatrix} \boldsymbol{x}$$

  - General non-singular linear transformation of homogeneous coordinates
  - An invertible mapping such that three points lie on the same line if and only if the images of the points do.
  - 8 degrees of freedom
  - Invariants
    - Cross ratio of four collinear points: $(|\overline{\boldsymbol{x}_1\boldsymbol{x}_2}||\overline{\boldsymbol{x}_3\boldsymbol{x}_4}|)/(|\overline{\boldsymbol{x}_1\boldsymbol{x}_3}||\overline{\boldsymbol{x}_2\boldsymbol{x}_4}|)$
    - Concurrency: multiple lines' meeting at a single point
    - Collinearity
  - Parallel lines are no more parallel in projective transformation.
    - Ideal point (point at infinity) -> finite point

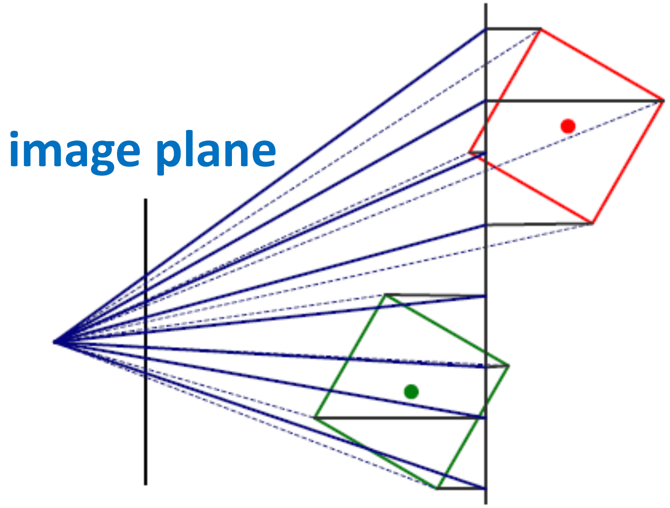
3D Transformations

Transformation	Preservation	Matrix
Translation		$\begin{pmatrix} \boldsymbol{I} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix}$
Euclidean	• Volume	$\begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix}$
Similarity	• Angles • Absolute conic	$\begin{pmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix}$
Affine	• Parallelism • Volume ratio • Plain at infinity	$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix}$
Projective	• Straight lines • Intersection • Tangency of surfaces	$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{t} \\ \boldsymbol{v}^{\mathrm{T}} & v \end{pmatrix}$

3D to 2D Projections

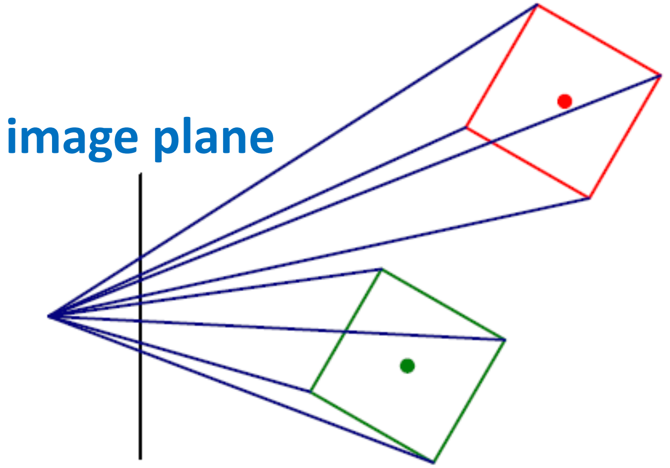
- Orthographic projection

$$\boldsymbol{x}_{2d} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}_{3d}$$



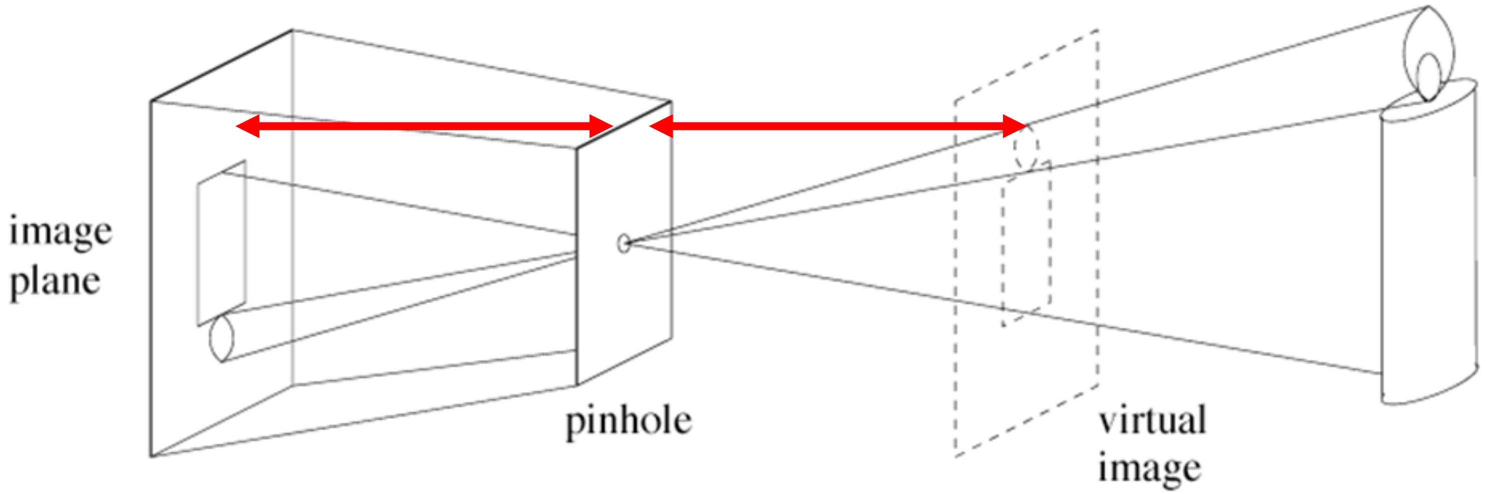
- Perspective projection

$$\boldsymbol{x}_{2d} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \boldsymbol{x}_{3d}$$

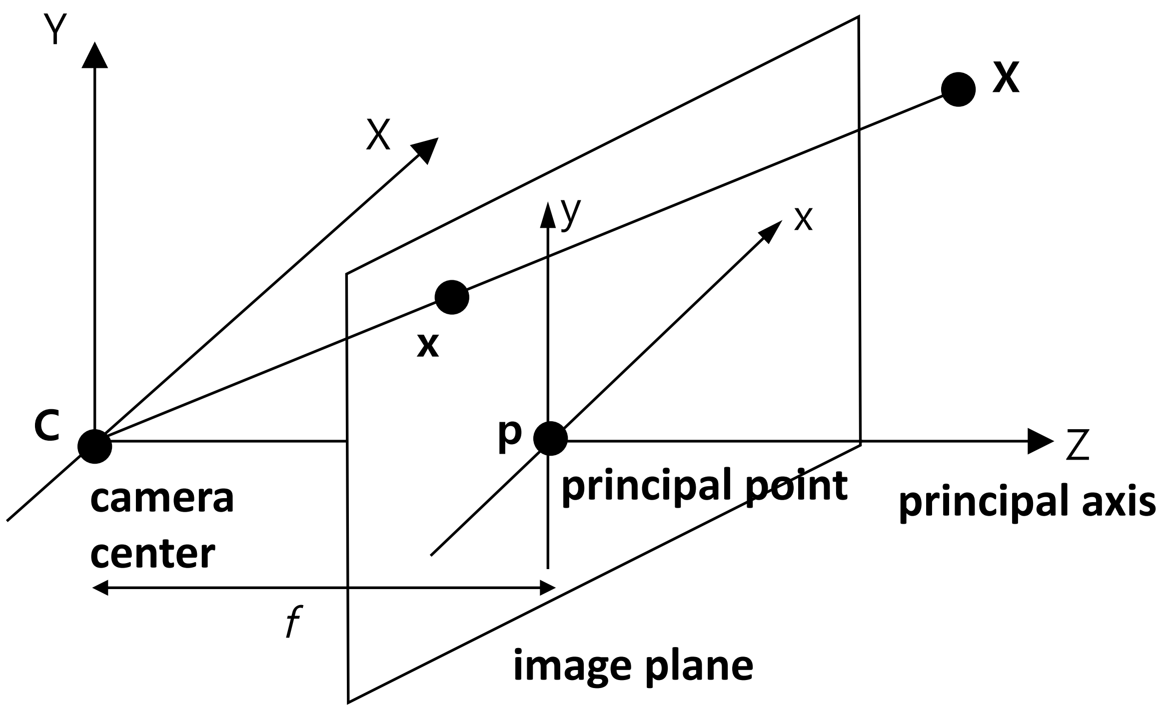


Pinhole Camera

- What is pinhole camera?
  - An abstract camera model: box with a small hole and translucent plate
  - One ray would pass through each point in the plane.
  - Inverted image is observed in image plane.
  - Virtual image: plane in front of pinhole with same distance to the image plane



Basic Pinhole Model

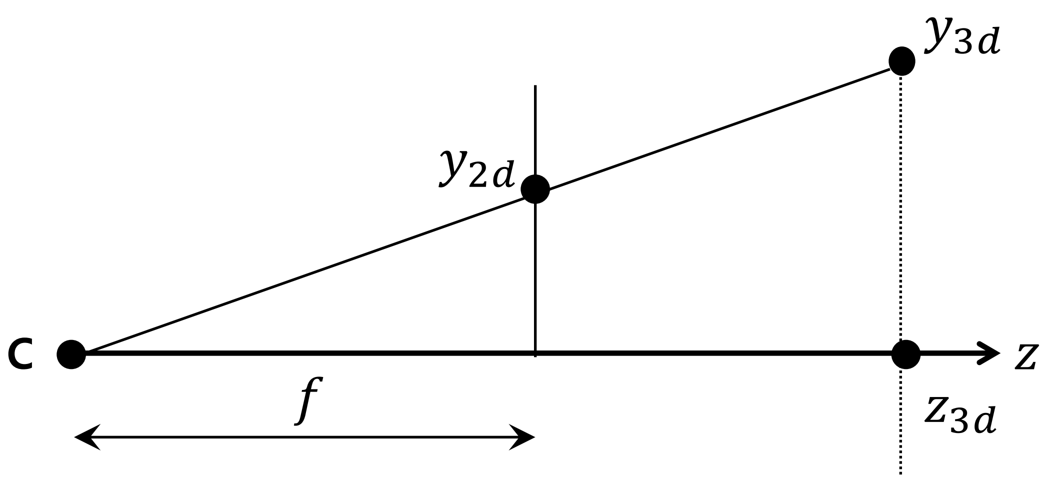


- Mapping from Euclidean 3-space $\Re^3$ to Euclidean 2-space $\Re^2$

$$(x, y, z)^{\mathrm{T}} \longmapsto \left(\frac{fx}{z}, \frac{fy}{z}\right)^{\mathrm{T}}$$

Projection with Homogeneous Coordinates

- Central projection using homogenous coordinates



$$y_{2d} = \frac{f y_{3d}}{z_{3d}}$$

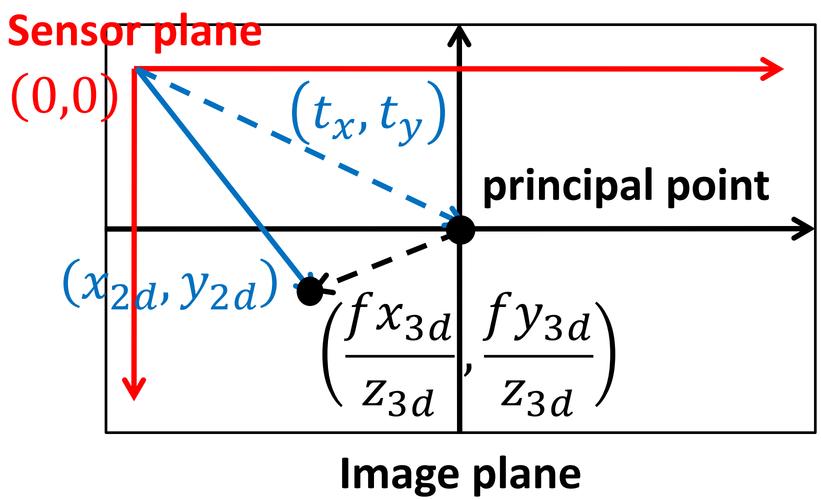
$$\begin{pmatrix} x_{3d} \\ y_{3d} \\ z_{3d} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f x_{3d} \\ f y_{3d} \\ z_{3d} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_{3d} \\ y_{3d} \\ z_{3d} \\ 1 \end{pmatrix}$$

$$\boldsymbol{x}_{2d} = \boldsymbol{P}\boldsymbol{x}_{3d}$$

$$\boldsymbol{P} = \mathrm{diag}([f, f, 1])\ [\boldsymbol{I}|\boldsymbol{0}]$$

Image Plane vs. Sensor Plane

- Principal point offset



Point in image plane

Point in camera coordinate system

$$\boldsymbol{x}_{2d} = \boldsymbol{K}[\boldsymbol{I}|\boldsymbol{0}]\hat{\boldsymbol{x}}_{3d}$$

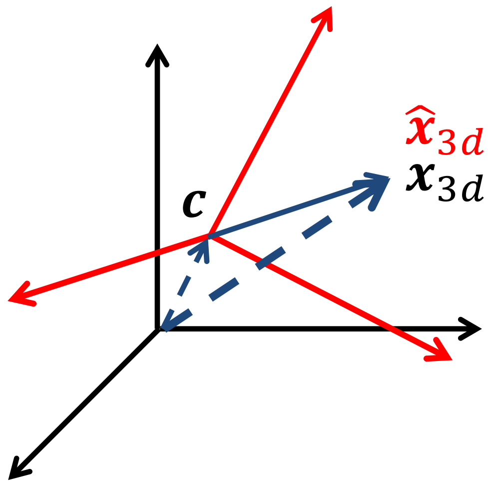
$$\begin{pmatrix} x_{3d} \\ y_{3d} \\ z_{3d} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f x_{3d} + z_{3d} t_x \\ f y_{3d} + z_{3d} t_y \\ z_{3d} \end{pmatrix} = \begin{pmatrix} f & 0 & t_x & 0 \\ 0 & f & t_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_{3d} \\ y_{3d} \\ z_{3d} \\ 1 \end{pmatrix}$$

$$\boldsymbol{K} = \begin{pmatrix} f & 0 & t_x \\ 0 & f & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

Camera calibration matrix

Camera Rotation and Translation

- Camera rotation and translation



$$\widehat{\boldsymbol{x}}^*_{3d} = \boldsymbol{R}^*(\boldsymbol{x}^*_{3d} - \boldsymbol{c}^*)$$

In non-homogeneous coordinate system

$$\widehat{\boldsymbol{x}}_{3d} = \begin{pmatrix} \boldsymbol{R} & -\boldsymbol{R}\boldsymbol{c} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix} \boldsymbol{x}_{3d}$$

$$\begin{aligned} \boldsymbol{x}_{2d} &= \boldsymbol{K}[\boldsymbol{I}|\boldsymbol{0}]\widehat{\boldsymbol{x}}_{3d} \\ &= \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I}|-\boldsymbol{c}]\boldsymbol{x}_{3d} \end{aligned}$$

$$\boldsymbol{P} = \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I}|-\boldsymbol{c}] = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}] \quad \text{where } \boldsymbol{t} = -\boldsymbol{R}\boldsymbol{c}$$

camera matrix

  - 9 DOF: 3 for $\boldsymbol{K}$, 3 for $\boldsymbol{R}$, 3 for $\boldsymbol{c}$
  - Internal (intrinsic) camera parameters: $\boldsymbol{K}$
  - External (extrinsic) camera parameters: $\boldsymbol{R}$ and $\boldsymbol{c}$

Other Projective Cameras

- CCD cameras
  - Having non-square pixels
  - 10 DOF

$$K = \begin{pmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{matrix} \alpha_x = f m_x \\ \alpha_y = f m_y \end{matrix}$$

m_x, m_y: # of pixels in unit distance

- Finite projective camera
  - Add skew parameter: 11 DOF
  - $\boldsymbol{K}$ is homogeneous.
  - $\boldsymbol{KR}$ is non-singular.
  - $\boldsymbol{KR}$ can be decomposed by QR factorization

$$K = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

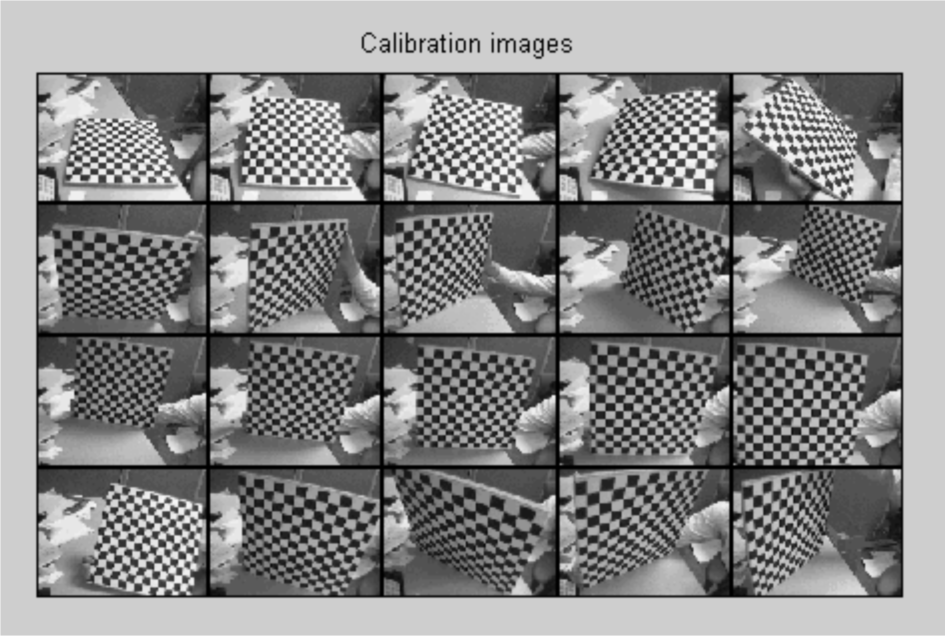
$$\boldsymbol{P} = [\boldsymbol{KR} | -\boldsymbol{c}]$$

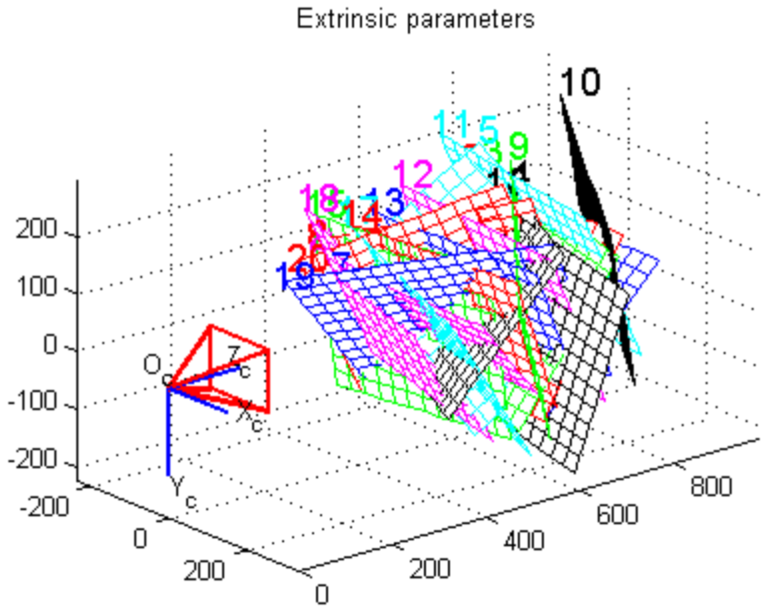
- General projective camera
  - Arbitrary rank-3 3x4 matrix
  - No non-singularity restriction on the left hand 3x3 submatrix
  - 11 DOF

$$\boldsymbol{P} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & 1 \end{pmatrix}$$

Camera Calibration

- Finding extrinsic and intrinsic parameters of a camera





Camera Calibration

- Finding intrinsic and extrinsic camera parameters

$$\boldsymbol{P} = \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I}|-\boldsymbol{c}]$$

  - Intrinsic parameters: $\boldsymbol{K}$
  - Extrinsic parameters: $\boldsymbol{R}$ and $\boldsymbol{c}$

- 3D to 2D correspondences

$$\boldsymbol{x}_{2d} = \boldsymbol{P}\boldsymbol{x}_{3d} \qquad \boldsymbol{x}_{2d} \times \boldsymbol{P}\boldsymbol{x}_{3d} = \boldsymbol{0}$$

$$\boldsymbol{x}_{2d} = (x, y, w)^{\mathrm{T}}$$

$$\boldsymbol{x}_{2d} \times \boldsymbol{P}\boldsymbol{x}_{3d} = \underbrace{\begin{pmatrix} \boldsymbol{0}^{\mathrm{T}} & -w\boldsymbol{x}_{3d}^{\mathrm{T}} & y\boldsymbol{x}_{3d}^{\mathrm{T}} \\ w\boldsymbol{x}_{3d}^{\mathrm{T}} & \boldsymbol{0}^{\mathrm{T}} & -x\boldsymbol{x}_{3d}^{\mathrm{T}} \\ -y\boldsymbol{x}_{3d}^{\mathrm{T}} & x\boldsymbol{x}_{3d}^{\mathrm{T}} & \boldsymbol{0}^{\mathrm{T}} \end{pmatrix}}_{\boldsymbol{A}_i} \underbrace{\begin{pmatrix} \boldsymbol{p}^1 \\ \boldsymbol{p}^2 \\ \boldsymbol{p}^3 \end{pmatrix}}_{\boldsymbol{p}} = \boldsymbol{0}$$

3x12 rank-2 matrix **11 DOF**

Camera Calibration

- Solution

$$\boldsymbol{Ap} = \mathbf{0} \qquad \boldsymbol{A} = [\boldsymbol{A}_1^{\mathrm{T}} \quad \boldsymbol{A}_2^{\mathrm{T}} \quad \cdots \quad \boldsymbol{A}_n^{\mathrm{T}}]^{\mathrm{T}}$$

  - We need at least 5.5 correspondences to solve the linear system.
  - Note that $\boldsymbol{P}$ is a homogeneous matrix.
  - When $\boldsymbol{A}$ is an 11x12 matrix, $\boldsymbol{p}$ is 1-dimensional null-space.
  - When $\boldsymbol{A}$ is larger, it is an over-determined problem.

$$\boldsymbol{p}^* = \underset{\boldsymbol{p}}{\operatorname{argmin}} \|\boldsymbol{Ap}\| \quad \text{subject to } \|\boldsymbol{p}\| = 1$$

$\boldsymbol{p}^*$ is the unit singular vector of $\boldsymbol{A}$ corresponding to the least singular value.

$$\|\boldsymbol{Ap}\| = \|\boldsymbol{UDV}^{\mathrm{T}}\boldsymbol{p}\| = \|\boldsymbol{DV}^{\mathrm{T}}\boldsymbol{p}\| \qquad \|\boldsymbol{y}\| = \|\boldsymbol{V}^{\mathrm{T}}\boldsymbol{p}\|$$

$$\min\|\boldsymbol{Ap}\| = \min\|\boldsymbol{Dy}\| \quad \text{subject to } \|\boldsymbol{y}\| = 1$$

$$\text{solution: } \boldsymbol{y} = (0 \quad 0 \quad \cdots \quad 1)^{\mathrm{T}} \qquad \boldsymbol{p} = \boldsymbol{Vy}$$

Camera Calibration

- After finding camera matrix
  - We got

$$\boldsymbol{P} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

  - Recall

$$\boldsymbol{P} = \boldsymbol{KR}[\boldsymbol{I}|-\boldsymbol{c}] = [\boldsymbol{KR}| - \boldsymbol{KRc}]$$

$$\boldsymbol{K} = \begin{pmatrix} f & s & t_x \\ 0 & f & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

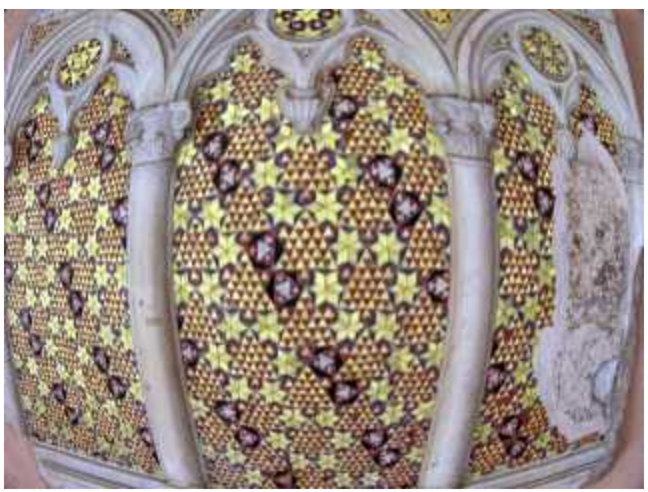
$\boldsymbol{R}$: orthonormal matrix

  - Camera parameters can be obtained by QR (RQ) factorization and solving a simple linear system

Lens Distortions

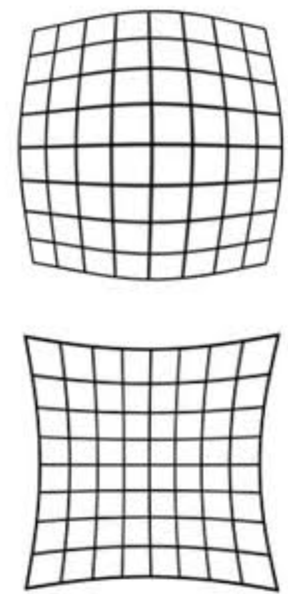
- Linear camera model
  - Image point and optical center are collinear.
  - Not realistic in the real (non-pinhole) lenses
- Radial distortion
  - Non-linear error
  - More significant as the focal length of the lens decrease

$\Leftrightarrow$ **as the lens is cheaper**



barrel distortion

pincushion distortion



Lens Distortions

- Radial distortion
  - Note: Lens distortion takes place during the initial projection of the world onto the image plane

$$(\tilde{x}_{2d}, \tilde{y}_{2d}, 1)^{\mathrm{T}} = \boldsymbol{x}_{2d} = \boldsymbol{K}[\boldsymbol{I}|\boldsymbol{0}]\widehat{\boldsymbol{x}}_{3d}$$

Ideal projection

$$\begin{pmatrix} x_{2d} \\ y_{2d} \end{pmatrix} = L(\tilde{r}) \begin{pmatrix} \tilde{x}_{2d} \\ \tilde{y}_{2d} \end{pmatrix}$$

Actual projection

Distortion factor

  - Correction: using (low-order) polynomials

$$\hat{x} = x_c + L(r)(x - x_c)$$
$$\hat{y} = y_c + L(r)(y - y_c)$$
$$L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \cdots$$
$$r = (x - x_c)^2 + (y - y_c)^2$$

(x, y): measured
(x_c, y_c): distortion center
$(\hat{x}, \hat{y})$: corrected

Reading Assignment

- Chapter 4 of Szeliski's book
- Chapter 1 of Forsyth & Ponce's book

- If you are more interested, refer to Chapter 6 & 7 of Hartley & Zisserman's book.