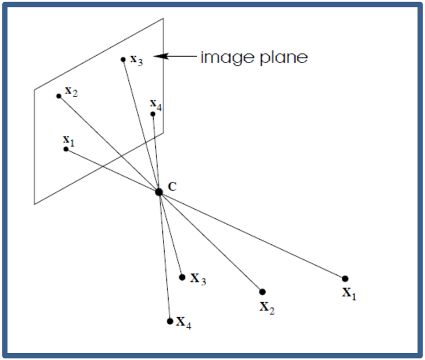
Master in
Computer Vision
Barcelona

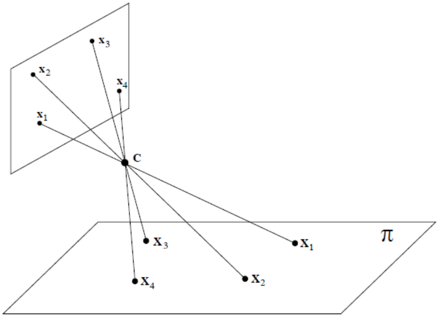
Module: 3D Vision
Lecture 3: Homography estimation / 3D projectivity / Camera Model

Federico Sukno / Pedro Cavestany

Planar transformations

Planar transformations



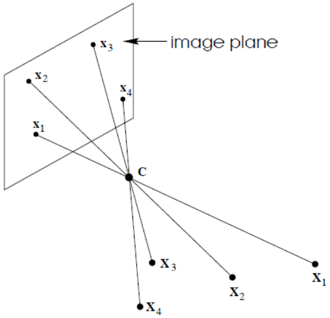


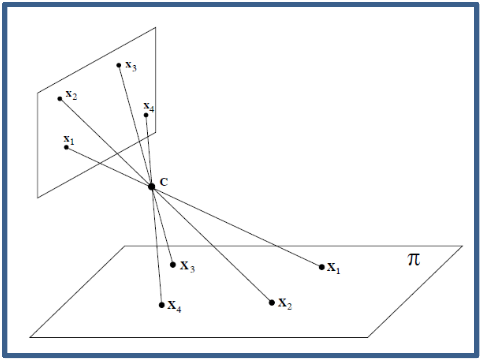
The action of a projective camera on a point in space may be expressed in terms of a linear mapping of homogeneous coordinates.

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \mathrm{P}_{3\times 4} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ \mathrm{T} \end{pmatrix}$$

Image source: [Hartley Zisserman 2004]

Planar transformations





If all the points lie on a plane, then the linear mapping reduces to:

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \mathrm{H}_{3\times 3} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{T} \end{pmatrix}$$

Image source: [Hartley Zisserman 2004]

Planar transformations

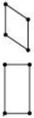
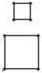
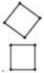
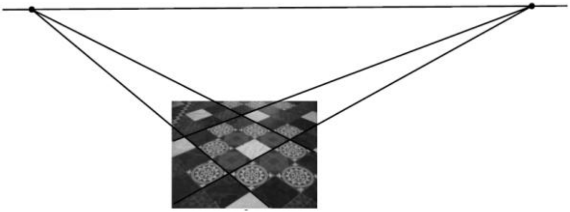
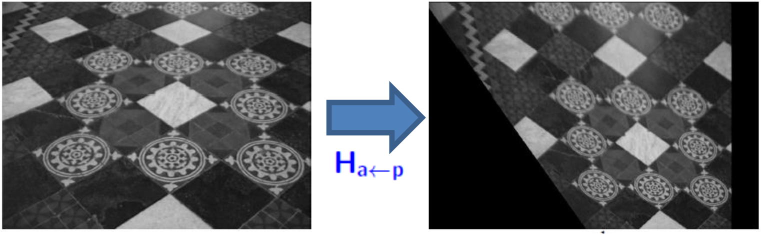
Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$.
Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle. The circular points, **I**, **J** (see section 2.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

Image source: [Hartley Zisserman 2004]

Affine Rectification from the Vanishing Line



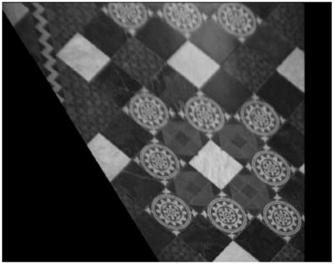


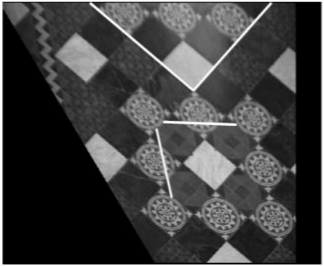
We apply $H_{a\leftarrow p}$ to the whole image to obtain the affine-rectified image.

Affine and Matric Rectification

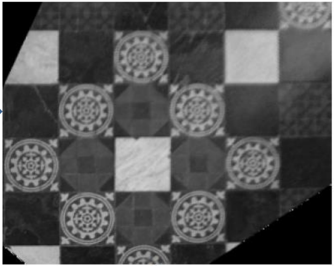
$\mathbf{H}_{a \leftarrow p}$

$$\begin{bmatrix} \mathrm{I} & \mathbf{0} \\ \mathbf{v}^{\mathsf{T}} & 1 \end{bmatrix}$$





$$\begin{bmatrix} \mathrm{K} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$$



Class 3 Outline

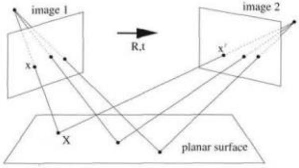
1. Homography Estimation
2. Projective geometry and transformations in 3D
3. Camera models.

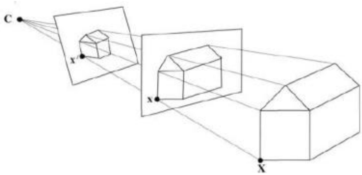
Master in Computer Vision Barcelona

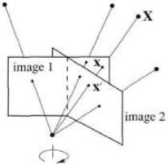
HOMOGRAPHY COMPUTATION

Homografies (or planar transformations)

Goal: Compute the homography that relates to images







A homography relates two images:

- of the same plane in the 3D scene;
- taken with a camera rotating about its centre;
- taken with the same static camera varying its focal length;
- the whole scene is far away from the camera.

Image source: [Hartley Zisserman 2004]

Computing Homographies

Consider a set of points $\mathbf{x}_i$ in $\mathbb{P}^2$ that **correspond** to points $\mathbf{x}'_i$ in $\mathbb{P}^2$, $i = 1, \ldots, n$.

We need **at least 4 points** in general position **(*)** to compute a projective transformation $H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$ of $\mathbb{P}^2$.
That is, we need $n \geq 4$.

(*) general position means that no three points are collinear.

The set of n equations to compute H are

$$\mathbf{x}'_i = H\mathbf{x}_i.$$

Let's compute the homography H.

Computing Homographies

Let $\mathbf{x}_i = (x_i, y_i, w_i)$, $\mathbf{x}'_i = (x'_i, y'_i, w'_i)$ (e.g., $w_i = w'_i = 1$.)
Let us write

$$H = \begin{pmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{pmatrix}.$$

That is $\mathbf{h}_k^T = (h_{i1}, h_{k2}, h_{k3})$ is the k row of H $(k = 1, 2, 3)$. Then

$$\begin{pmatrix} x'_i \\ y'_i \\ w'_i \end{pmatrix} = \mathbf{x}'_i = H\mathbf{x}_i = \begin{pmatrix} \mathbf{h}_1^T\mathbf{x}_i \\ \mathbf{h}_2^T\mathbf{x}_i \\ \mathbf{h}_3^T\mathbf{x}_i \end{pmatrix}.$$

In homogeneous coordinates we have the equations

$$\frac{x'_i}{w'_i} = \frac{\mathbf{h}_1^T\mathbf{x}_i}{\mathbf{h}_3^T\mathbf{x}_i},$$

$$\frac{y'_i}{w'_i} = \frac{\mathbf{h}_2^T\mathbf{x}_i}{\mathbf{h}_3^T\mathbf{x}_i}.$$

Computing Homographies

$$x_i' \mathbf{h}_3^T \mathbf{x}_i - w_i' \mathbf{h}_1^T \mathbf{x}_i = 0,$$
$$y_i' \mathbf{h}_3^T \mathbf{x}_i - w_i' \mathbf{h}_2^T \mathbf{x}_i = 0.$$

Thus, each correspondence $\mathbf{x}_i \longleftrightarrow \mathbf{x}_i'$ produces two equations for the 9 unknowns $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3$. In matricial form:

$$\begin{pmatrix} \mathbf{0}^T & -w_i'\mathbf{x}_i^T & y_i'\mathbf{x}_i^T \\ w_i'\mathbf{x}_i^T & \mathbf{0}^T & -x_i'\mathbf{x}_i^T \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = \mathbf{0} \in \mathbb{R}^2.$$

That is,

$$\mathbf{A}_i \mathbf{h} = \mathbf{0}.$$

(where $\mathbf{A}_i$ is 2×9 and $\mathbf{h}$ is 9×1).

Remark 1. These equations can also be obtained taking into account that from $\mathbf{x}_i' = H\mathbf{x}_i$ we have $\mathbf{x}_i' \times H\mathbf{x}_i = \mathbf{0}$.
Remark 2. Actually we aim to compute H (i.e., $\mathbf{h}$) such that the so-called **algebraic distance** (or **algebraic error**) $d_{\mathrm{alg}}(\mathbf{x}_i', H\mathbf{x}_i) = \|\mathrm{proj}_2(\mathbf{x}_i' \times H\mathbf{x}_i)\|$ be 0 (where $\mathrm{proj}_2(\mathbf{x}_i' \times H\mathbf{x}_i)$ denotes the first two coordinates of $\mathbf{x}_i' \times H\mathbf{x}_i$).

Computing Homographies

We have obtained the 2 equations $\mathbf{A}_i\mathbf{h} = \mathbf{0}$ from a given correspondence $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$.

If we have $n \geq 4$ correspondences $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$ we have $2n \geq 8$ homogeneous equations.

If $\mathbf{A}$ denotes the matrix obtained by $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \dots \\ \mathbf{A}_n \end{pmatrix}$.

then the system of equations is

$$\mathbf{A}\mathbf{h} = \mathbf{0} \in \mathbb{R}^{2n}.$$

The vector h is in the null space of A.

– If $n = 4 \rightarrow$ dim Ker A$\geq 1 \rightarrow$ there is a exact solution.

– If $n > 4$, there is exact solution to the overdetermined system iff rank(A)< 9.

Otherwise, to avoid the zero solution $\rightarrow$ approximate solution $\rightarrow$ SVD: minimize $||\mathbf{A}\mathbf{h}||$ subject to $||\mathbf{h}|| = 1$.

Algorithm 1: Direct Linear Transform (DLT)

Objective: Given n 2D to 2D point correspondences $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$, determine the 2D homography matrix H such that $\mathbf{x}'_i = H\mathbf{x}_i$ for all $i = 1, \ldots, n$.

Algorithm:

1. For each correspondence $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$, compute $\mathbf{A}_i$.
2. Assemble the n matrices $\mathbf{A}_i$ to form the $2n \times 9$ matrix $\mathbf{A}$.
3. Compute the SVD decomposition of $\mathbf{A}$. Then, if $\mathbf{A} = UDV^T$ is the SVD of $\mathbf{A}$ with D a diagonal matrix with positive diagonal entries arranged in descending order, then $\mathbf{h}$ is the last column of V.
4. The matrix H is

$$\begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}.$$

where $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)^T$.

Non-invariance of the DLT algorithm

- The result of the DLT algorithm for computing 2D homographies depends on the coordinate system in which points are expressed.

- It is not invariant to similarity transformations of the image.
Thus some coordinate systems are in some way better than others for computing a 2D homography.

- Solution to this problem: to apply a method of normalization of the data (consisting of translation and scaling of image coordinates) before applying the DLT algorithm.
Subsequently an appropriate correction to the result expresses the computed H with respect to the original coordinate system.

- This normalizing transformation will diminish the effect of the arbitrary selection of origin and scale in the coordinate frame of the image, and will mean that the combined algorithm is invariant to a similarity transformation of the image.

Why do we need normalization in DLT?

Remark: the DLT Algorithm minimizes an **algebraic error**, that is, the Euclidean error

$$\|\mathbf{Ah}\|^2 = \sum_{i=1}^{n} \|\mathbf{A}_i\mathbf{h}\|^2 = \sum_{i=1}^{n} \|\mathrm{proj}_2(\mathbf{x}_i' \times H\mathbf{x}_i)\|^2$$

where $\mathrm{proj}_2(\mathbf{x}_i' \times H\mathbf{x}_i)$ denotes the first two coordinates of $\mathbf{x}_i' \times H\mathbf{x}_i$.

Result 4.3. *Let* $\mathtt{T}'$ *be a similarity transformation with scale factor* s*, and let* $\mathtt{T}$ *be an arbitrary projective transformation. Further, suppose* $\mathtt{H}$ *is any 2D homography and let* $\tilde{\mathtt{H}}$ *be defined by* $\tilde{\mathtt{H}} = \mathtt{T}'\mathtt{H}\mathtt{T}^{-1}$*. Then* $\|\tilde{\mathtt{A}}\tilde{\mathbf{h}}\| = s\|\mathtt{A}\mathbf{h}\|$ *where* $\mathbf{h}$ *and* $\tilde{\mathbf{h}}$ *are the vectors of entries of* $\mathtt{H}$ *and* $\tilde{\mathtt{H}}$.

$$d_{\mathrm{alg}}(\tilde{\mathbf{x}}_i', \tilde{\mathtt{H}}\tilde{\mathbf{x}}_i) = s\,d_{\mathrm{alg}}(\mathbf{x}_i', \mathtt{H}\mathbf{x}_i)$$

Why do we need normalization in DLT?

$$\tilde{\mathtt{A}}_i\tilde{\mathbf{h}} = (\tilde{\epsilon}_{i1}, \tilde{\epsilon}_{i2})^{\mathsf{T}} = s\mathtt{R}(\epsilon_{i1}, \epsilon_{i2})^{\mathsf{T}} = s\mathtt{R}\mathtt{A}_i\mathbf{h}.$$

Thus, there is a one-to-one correspondence between $\mathtt{H}$ and $\tilde{\mathtt{H}}$ giving rise to the same error, except for constant scale. It may appear therefore that the matrices $\mathtt{H}$ and $\tilde{\mathtt{H}}$ minimizing the algebraic error will be related by the formula $\tilde{\mathtt{H}} = \mathtt{T}'\mathtt{H}\mathtt{T}^{-1}$, and hence one may retrieve $\mathtt{H}$ as the product $\mathtt{T}'^{-1}\tilde{\mathtt{H}}\mathtt{T}$. This conclusion is **false** however. For, although $\mathtt{H}$ and $\tilde{\mathtt{H}}$ so defined give rise to the same error ϵ, the condition $\|\mathtt{H}\| = 1$, imposed as a constraint on the solution, is not equivalent to the condition $\|\tilde{\mathtt{H}}\| = 1$.

$$\begin{aligned} & \text{minimize } \sum_i d_{\text{alg}}(\mathbf{x}'_i, \mathtt{H}\mathbf{x}_i)^2 \text{ subject to } \|\mathtt{H}\| = 1 \\ \Leftrightarrow\ & \text{minimize } \sum_i d_{\text{alg}}(\tilde{\mathbf{x}}'_i, \tilde{\mathtt{H}}\tilde{\mathbf{x}}_i)^2 \text{ subject to } \|\mathtt{H}\| = 1 \\ \not\Leftrightarrow\ & \text{minimize } \sum_i d_{\text{alg}}(\tilde{\mathbf{x}}'_i, \tilde{\mathtt{H}}\tilde{\mathbf{x}}_i)^2 \text{ subject to } \|\tilde{\mathtt{H}}\| = 1 \end{aligned}$$

Why do we need normalization in DLT?

$$\mathbf{x}' = \mathtt{H}_{\mathrm{P}}\mathbf{x} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^{\mathsf{T}} & v \end{bmatrix} \mathbf{x}$$

$$d_{\mathrm{alg}}(\mathbf{x}'_i, \mathtt{H}\mathbf{x}_i)^2 = \|\boldsymbol{\epsilon}_i\|^2 = \left\| \begin{bmatrix} \mathbf{0}^{\mathsf{T}} & -w'_i\mathbf{x}_i^{\mathsf{T}} & y'_i\mathbf{x}_i^{\mathsf{T}} \\ w'_i\mathbf{x}_i^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} & -x'_i\mathbf{x}_i^{\mathsf{T}} \end{bmatrix} \mathbf{h} \right\|^2$$

Algorithm 2: Normalized DLT

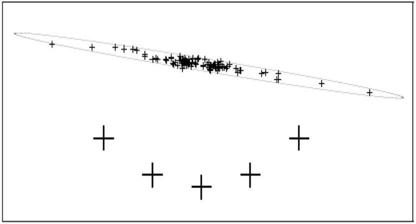
Objective: Given n 2D to 2D point correspondences $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$, determine the 2D homography matrix H such that $\mathbf{x}'_i = H\mathbf{x}_i$ for all i.

Algorithm:

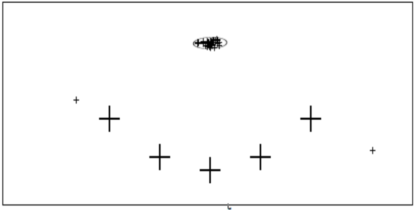
1. **Normalization of x:** Compute a similarity transformation $\mathcal{T}$, consisting of a translation and scaling, that takes points $\mathbf{x}_i$ to a new set of points $\widetilde{\mathbf{x}}_i$ such that the centroid of the points $\widetilde{\mathbf{x}}_i$ is the coordinate origin $(0,0)^T$, and their average distance from the origin is $\sqrt{2}$.
2. **Normalization of x′ :** Compute a similar transformation $\mathcal{T}'$ for the points in the second image, transforming points $\mathbf{x}'_i$ to a new set of points $\widetilde{\mathbf{x}}'_i$.
3. **Apply the DLT algorithm** above to the correspondences $\widetilde{\mathbf{x}}_i \longleftrightarrow \widetilde{\mathbf{x}}'_i$ and obtain an homography $\widetilde{H}$.
4. **Denormalization:** Set $H = \mathcal{T}'^{-1}\widetilde{H}\mathcal{T}$.

How important is normalization in DLT?

Examples of homographies computed from 5 corresponding points (+), repeated 1000 times with small perturbations. The 6^{th} point (on top) is mapped using the resulting homographies



DLT



Normalized DLT

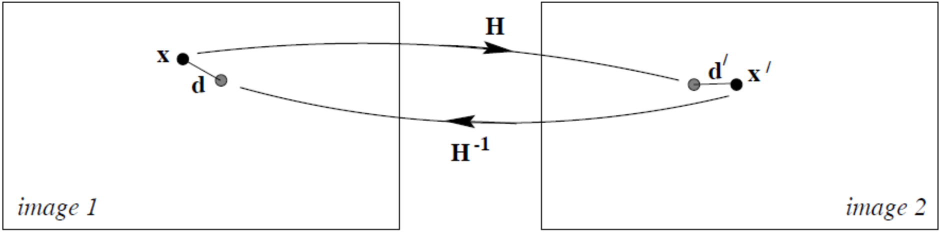
Geometric Errors

Transfer Error

$$\sum_i d(\mathbf{x}'_i, \mathrm{H}\bar{\mathbf{x}}_i)^2$$

Symmetric Transfer Error

$$\sum_i d(\mathbf{x}_i, \mathrm{H}^{-1}\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, \mathrm{H}\mathbf{x}_i)^2$$



Reprojection Error

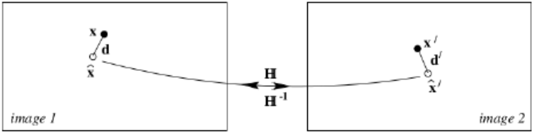
$$\min_{H,\hat{\mathbf{x}}_i,\hat{\mathbf{x}}'_i} \sum_{i=1}^{n} \|[\mathbf{x}_i] - [\hat{\mathbf{x}}_i]\|^2 + \|[\mathbf{x}'_i] - [\hat{\mathbf{x}}'_i]\|^2 \quad \text{such that } \hat{\mathbf{x}}'_i = H\hat{\mathbf{x}}_i, \ \forall i$$

where $[(x_1, x_2, x_3)] = (x_1/x_3, x_2/x_3)$ when $x_3 \neq 0$. Also written

$$\min_{H,\hat{\mathbf{x}}_i,\hat{\mathbf{x}}'_i} \sum_{i=1}^{n} d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \quad \text{such that } \hat{\mathbf{x}}'_i = H\hat{\mathbf{x}}_i, \ \forall i$$

where $d(\mathbf{a}, \mathbf{b}) = (a_1/a_3 - b_1/b_3)^2 + (a_2/a_3 - b_2/b_3)^2$ is the so-called geometric distance (the Euclidean distance between the inhomogeneous points represented by **a** and **b**).

Interpretation: The measured correspondences $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$ can be noisy (and/or n can be > 4). Thus, not perfectly verify $\mathbf{x}'_i = H\mathbf{x}_i$, nor $\mathbf{x}_i = H^{-1}\mathbf{x}'_i$. How much it is necessary to correct the measurements in each of the two images in order to obtain a perfectly matched set of image points $\hat{\mathbf{x}}'_i = H\hat{\mathbf{x}}_i$?



Reprojection vs Transfer Error

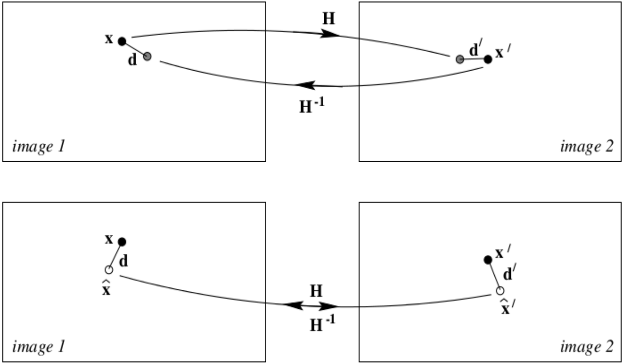


Fig. 4.2. *A comparison between symmetric transfer error (upper) and reprojection error (lower) when estimating a homography. The points* $\mathbf{x}$ *and* $\mathbf{x}'$ *are the measured (noisy) points. Under the estimated homography the points* $\mathbf{x}'$ *and* $\mathtt{H}\mathbf{x}$ *do* not *correspond perfectly (and neither do the points* $\mathbf{x}$ *and* $\mathtt{H}^{-1}\mathbf{x}'$*). However, the estimated points,* $\hat{\mathbf{x}}$ *and* $\hat{\mathbf{x}}'$*, do correspond perfectly by the homography* $\hat{\mathbf{x}}' = \mathtt{H}\hat{\mathbf{x}}$*. Using the notation* $d(\mathbf{x}, \mathbf{y})$ *for the Euclidean image distance between* $\mathbf{x}$ *and* $\mathbf{y}$*, the symmetric transfer error is* $d(\mathbf{x}, \mathtt{H}^{-1}\mathbf{x}')^2 + d(\mathbf{x}', \mathtt{H}\mathbf{x})^2$*; the reprojection error is* $d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$.

Robust Estimation

- Image correspondences often contain noise
  - Therefore we typically aim for more than 4 point correspondences
- The accuracy of correspondences is variable
  - Some correspondences might be considerably less accurate than others (outliers)
  - Estimation can be seriously affected by outliers
- In robust estimation we need to solve two problems simultaneously
  - Estimate the quantity (or transformation in our case) of interest
  - Determine what points (or correspondences) are inliers and which ones are outliers

Image correspondences

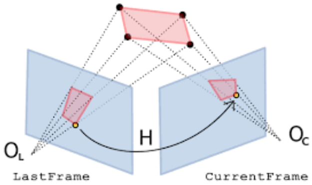


Image correspondences

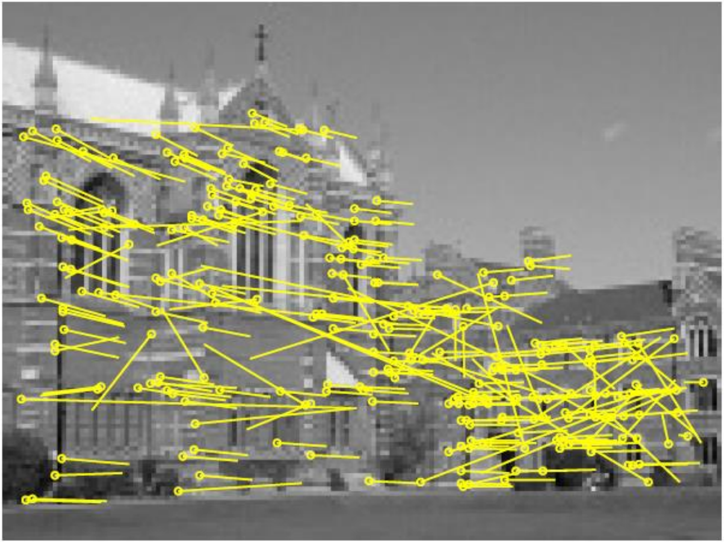
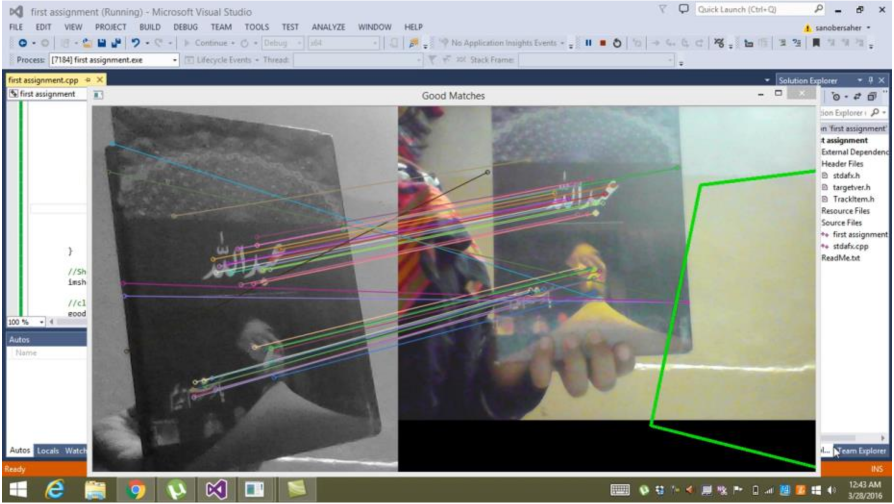


Image correspondences



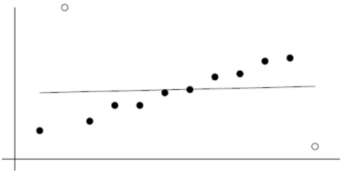
RANdom Sample Consensus (RANSAC)

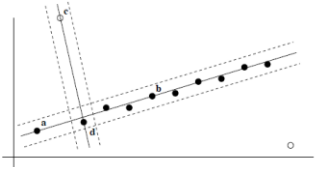
Objective

Robust fit of a model to a data set S which contains outliers.

Algorithm

(i) Randomly select a sample of s data points from S and instantiate the model from this subset.
(ii) Determine the set of data points S_i which are within a distance threshold t of the model. The set S_i is the consensus set of the sample and defines the inliers of S.
(iii) If the size of S_i (the number of inliers) is greater than some threshold T, re-estimate the model using all the points in S_i and terminate.
(iv) If the size of S_i is less than T, select a new subset and repeat the above.
(v) After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i.





Algorithm 3: Robust estimation of homographies

1. **Interest points:** Compute interest points in each image, $\{\mathbf{x}_i\}, \{\mathbf{x}'_j\}$.
2. **Putative correspondences:** Compute a set of interest point matches using SIFT or based on proximity and similarity of their intensity neighbourhood. $\{\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i\}$.
3. **RANSAC robust estimation:** Repeat for N samples:

- Select a **random sample of 4 correspondences** and compute the fundamental matrix H using the **normalized DLT algorithm**.
- Calculate either the previous **geometric distance d** or the **distance $\mathbf{d}_\perp$** defined below for each putative correspondence.

$$\mathbf{d}_\perp{}^2 = \|[\mathbf{x}'_i] - [H\mathbf{x}_i]\|^2 + \|[H^{-1}\mathbf{x}'_i] - [\mathbf{x}_i]\|^2$$

- Compute the number of inliers consistent with H by the number of correspondences for which $d_\perp < t$ pixels.
- Repeat.

In practice, $t = \sqrt{5.99}\sigma$.

Choose the H with the largest number of inliers. In the case of tie choose the solution that has the lowest standard deviation of inliers.

Algorithm 3: Robust estimation of homographies

1. **Interest points:** Compute interest points in each image, $\{\mathbf{x}_i\}, \{\mathbf{x}'_j\}$.
2. **Putative correspondences:** Compute a set of interest point matches using SIFT or based on proximity and similarity of their intensity neighbourhood. $\{\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i\}$.
3. **RANSAC robust estimation:** Repeat for N samples:

   Choose the H with the largest number of inliers. In the case of tie choose the solution that has the lowest standard deviation of inliers.
4. **Re-estimation:** re-estimate H from all correspondences classified as inliers using the normalized DLT algorithm.
5. **Guided matching:** Further interest point correspondences are now determined using the estimated H to define a search region about the transferred point position (optional).

Points 4 and 5 can be iterated until convergence

Algorithm 3: Robust estimation of homographies

1. **Interest points:** Compute interest points in each image, $\{\mathbf{x}_i\}, \{\mathbf{x}'_j\}$.
2. **Putative correspondences:** Compute a set of interest point matches using SIFT or based on proximity and similarity of their intensity neighbourhood. $\{\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i\}$.
3. **RANSAC robust estimation:** Repeat for N samples:

   Choose the H with the largest number of inliers. In the case of tie choose the solution that has the lowest standard deviation of inliers.
4. **Re-estimation:** re-estimate H from all correspondences classified as inliers using the normalized DLT algorithm.
5. **Guided matching:** Further interest point correspondences are now determined using the estimated H to define a search region about the transferred point position (optional).

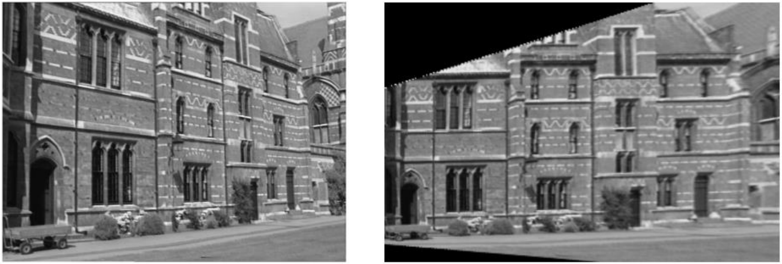
One can replace previous step 4 by

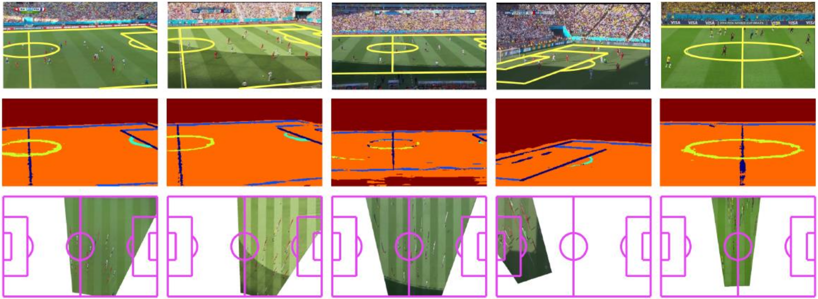
(4′) Non-linear estimation: re-estimate H from all correspondences classified as inliers by minimizing a geometric cost function using an optimization method like Newton or the Levenberg-Marquardt algorithm.

Example of robust homography estimation

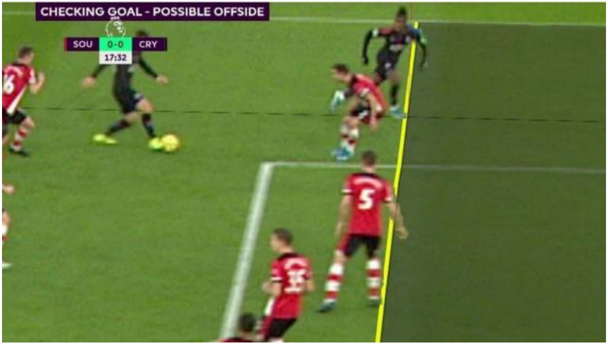
Some applications

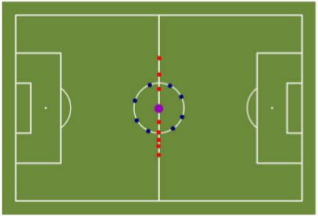
Some applications





Some applications



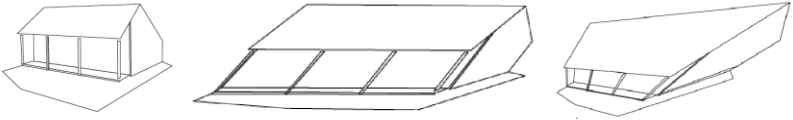


Class 3 Outline

1. Homography Estimation
2. Projective geometry and transformations in 3D
3. Camera models.

3D Projective geometry and transformations

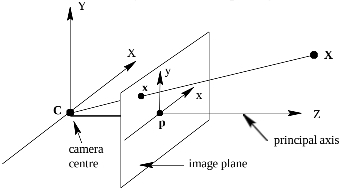
Now, **3D projective geometry: It models the camera projection and allows the 3D reconstruction, the calibration and auto-calibration.**

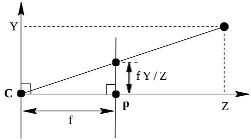


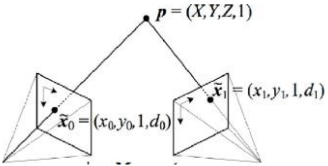
3D Transformations

3D projective geometry and transformations

Why is 3D projective geometry useful?







Basic example: Pinhole camera

Written in projective coordinates of P^3 and P^2, it is a linear map :
Central projection in homogeneous coordinates.

3D projective geometry

We will study the geometric properties and objects of the 3D projective space, denoted by $\mathbb{P}^3$. Many of the concepts are generalizations of the 2D projective space, the projective plane $\mathbb{P}^2$.

- For instance, the points of $\mathbb{P}^3$ increase the 3D Euclidean space $\mathbb{R}^3$ by adding a set of ideal points, which are on a plane at infinity $\mathbf{\Pi}_\infty$ (analogous to l_∞ in $\mathbb{P}^2$; $\mathbb{P}^2$ was constructed by adding to $\mathbb{R}^2$ the points on l_∞).
- The parallel lines and now also the parallel planes meet at $\mathbf{\Pi}_\infty$.
- The points are represented in homogeneous coordinates which increase in one component the cartesian coordinates in 3D.
- On the other hand, there appear new properties thanks to the new dimension that is added.

3D projective space

- Euclidean space $\mathbb{R}^3$: $(X, Y, Z)^T$ **cartesian coordinates** (inhomogeneous).
- Projective space $\mathbb{P}^3$: $\mathbf{X} = (x_1, x_2, x_3, x_4)^T$ **homogeneous coordinates** (or projective), except the point $(0, 0, 0, 0)^T$.

  If $x_4 \neq 0$, $\mathbf{X}$ represents the point $(X, Y, Z)^T = (\frac{x_1}{x_4}, \frac{x_2}{x_4}, \frac{x_3}{x_4})^T$ in $\mathbb{R}^3$.
- Notation:
  $P^3 := \{(x_1, x_2, x_3, x_4)^T \in \mathbb{R}^4 : x_4 \neq 0\} \cup \Pi_0(\mathbb{R}^4) = \mathbb{R}^4 \setminus \{(0, 0, 0, 0),$

  where $\Pi_0(\mathbb{R}^4) = \{(x_1, x_2, x_3, 0)^T : (x_1, x_2, x_3)^T \in \mathbb{R}^3 \setminus \{(0, 0, 0)\}\}$.

  We say that $P^3 = \{\text{clasic points}\} \cup \{\text{ideal points (at infinity)}\}$.
- Given $\mathbf{X}, \mathbf{X}' \in P^3$, we define the equivalence relation

$$\mathbf{X} \equiv \mathbf{X}' \text{ if there exists } \lambda \neq 0 \text{ such that } \mathbf{X} = \lambda \mathbf{X}'.$$

3D projective space

Definition: **Projective camera**

$$\mathbf{x}_{\mathrm{im}} = \mathbf{P}\mathbf{X}_{\mathrm{w}}.$$

where $\mathbf{X}_{\mathrm{w}} \in \mathbb{P}^3$ represents the coordinates of a point in the world coordinate frame, $\mathbf{x}_{\mathrm{im}} \in \mathbb{P}^2$ are image coordinates, and

Points at ∞

Homogeneous points $(x_1, x_2, x_3, 0)$ represent points at infinity (ideal points).

They form a plane: $x_4 = 0$ (the plane at infinity $\mathbf{\Pi}_\infty$).

Equation: $(0, 0, 0, 1) \cdot (x_1, x_2, x_3, x_4)^T = 0$,

or $\mathbf{\Pi}_\infty^T \mathbf{X} = 0$, where $\mathbf{\Pi}_\infty = (0, 0, 0, 1)^T$.

Planes in 3D Projective space

A plane in $\mathbb{P}^3$ can be written as

$$\pi_1 x_1 + \pi_2 x_2 + \pi_3 x_3 + \pi_4 x_4 = 0,$$

or in short as

$$\mathbf{\Pi}^T \mathbf{X} = 0,$$

where $\mathbf{\Pi} = (\pi_1, \pi_2, \pi_3, \pi_4)^T$.

The first three coordinates of $\mathbf{\Pi}$ correspond to the normal $\mathbf{n} \in \mathbb{R}^3$ of Euclidean geometry.

Using inhomogeneous coordinates we can write $\mathbf{X} = (\widetilde{\mathbf{X}}, 1)^T$ where $\widetilde{\mathbf{X}} \in \mathbb{R}^3$, and $\mathbf{\Pi} = (\mathbf{n}, d)^T$ with $\|\mathbf{n}\| = 1$ and d = distance to the origin, and we may write the plane equation as

$$\mathbf{n} \cdot \widetilde{\mathbf{X}} + d = 0.$$

Quadrics and Absolute Conic

- A **quadric** is a surface of $\mathbb{P}^3$ given by the equation

$$\mathbf{X}^T Q \mathbf{X} = 0, \quad \text{where } Q \text{ is a symmetric } 4 \times 4 \text{ matrix.}$$

  It is the analogous object to a conic in $\mathbb{P}^2$. Many of the properties of quadrics follow directly from those of conics.

- Degrees of freedom $= 9 = 10 - 1 =$ the 10 parameters of a symmetric 4×4 matrix minus one for the scale factor.

- The plane $\mathbf{\Pi} = Q\mathbf{X}$ is called the **polar plane** of $\mathbf{X}$ with respect to Q^*.
- **Dual quadric:** The dual of a quadric is also a quadric: $\mathbf{\Pi}^T Q^* \mathbf{\Pi} = 0$
- The **absolute conic** Ω_∞: a conic on $\mathbf{\Pi}_\infty$ that satisfies $X_1^2 + X_2^2 + X_3^2 = 0$ and $X_4 = 0$
- Orthogonality condition:

$$\cos\theta = \frac{\mathbf{d}_1^T \Omega_\infty \mathbf{d}_2}{\sqrt{(\mathbf{d}_1^T \Omega_\infty \mathbf{d}_1)(\mathbf{d}_2^T \Omega_\infty \mathbf{d}_2)}}$$

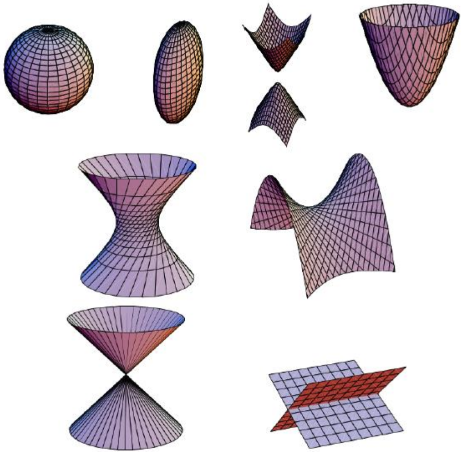
Lines

$$\cos\theta = \frac{\boldsymbol{\pi}_1^\mathsf{T} \mathsf{Q}_\infty^* \boldsymbol{\pi}_2}{\sqrt{(\boldsymbol{\pi}_1^\mathsf{T} \mathsf{Q}_\infty^* \boldsymbol{\pi}_1)(\boldsymbol{\pi}_2^\mathsf{T} \mathsf{Q}_\infty^* \boldsymbol{\pi}_2)}}.$$

Planes

Quadrics

Some examples:



(we refer to [Hartley Zisserman 2004], 3.2.4, for details)

3D Projective Transformations

A projective transformation (also called homography) of $\mathbb{R}^3$ is given by a non-singular 4×4 matrix H.

If $\mathbf{X}$ and $\mathbf{X}'$ are homogeneous coordinates in $\mathbb{P}^3$, it can be written as

$$\mathbf{X}' = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{pmatrix} \mathbf{X}.$$

In short,

$$\mathbf{X}' = H\mathbf{X}.$$

Degrees of freedom: 15 ($\rightarrow$ 4×4 elements - 1 multiplicative factor).

As in the case of planar projective transformations, lines are mapped to lines, it preserves incidence relations such as the intersection point of a line with a plane, and order of contact.

3D Isometry

Let $\mathbf{X}$ and $\mathbf{X}'$ be homogeneous coordinates in $\mathbb{P}^3$.

$$\mathbf{X}' = H_e \mathbf{X} = \begin{pmatrix} R & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix} \mathbf{X}.$$

where H_e is a 4×4 matrix,
R is a rotation 3×3 matrix (orthogonal matrix) and
$\vec{t}$ is a 3×1 translation vector.

Degrees of freedom: 6
$\rightarrow$ 3 for the rotation angles + 3 for the translation coefficients

Invariants: lengths, angles.

Remark: Isometries are transformations of $\mathbb{R}^3$ that preserve the Euclidean distance; i.e., $\|\mathcal{T}\vec{X} - \mathcal{T}\vec{Y}\|_2 = \|\vec{X} - \vec{Y}\|_2$ for all $\vec{X}, \vec{Y} \in \mathbb{R}^3$. An isometry $\mathcal{T} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is $\mathcal{T}\vec{X} = R\vec{X} + \vec{t}$. In homogeneous coordinates, $\mathbf{X}' = H_e \mathbf{X}$, where $\mathbf{X} = (\vec{X}, 1)$.

3D Similarity

Let $\mathbf{X}$ and $\mathbf{X}'$ be homogeneous coordinates in $\mathbb{P}^3$.

$$\mathbf{X}' = H_s\mathbf{X} = \begin{pmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix}\mathbf{X}.$$

where H_e is a 4×4 matrix,

s is an isotropic scaling factor,

R is a rotation 3×3 matrix (orthogonal matrix) and

$\vec{t}$ is a translation vector.

Degrees of freedom: 7

$\rightarrow$ 3 for the rotation angles + 3 for the translation coefficients + 1 for scaling factor

Invariants: angles, ratios, the absolute conic Ω_∞.

3D Affine Transformation

Let $\mathbf{X}$ and $\mathbf{X}'$ be homogeneous coordinates in $\mathbb{P}^3$.

$$\mathbf{X}' = H_a \mathbf{X} = \begin{pmatrix} A & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix} \mathbf{X}.$$

where A is a non-singular 3×3 matrix ,
$\vec{t}$ is a translation vector.

Degrees of freedom: 12
$\rightarrow$ 9 for A + 3 for the translation coefficients

Invariants: parallelism, ratio of two volumes, the plane at infinity $\mathbf{\Pi}_\infty$.

3D Projective Transformation

Let $\mathbf{X}$ and $\mathbf{X}'$ be homogeneous coordinates in $\mathbb{P}^3$.

$$\mathbf{X}' = H_p \mathbf{X} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & v \end{pmatrix} \mathbf{X}.$$

where H_p is a homography

$$H_p = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{pmatrix}$$

Degrees of freedom: 15
$\rightarrow 4 \times 4$ elements - 1 multiplicative factor

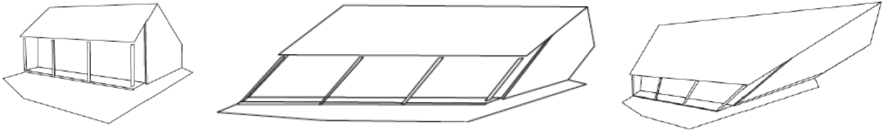
Invariants: concurrency, collinearity, order of contact, cross ratio.

3D Projective Transformation

Let $\mathbf{X}$ and $\mathbf{X}'$ be homogeneous coordinates in $\mathbb{P}^3$.

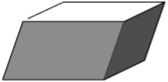
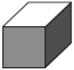
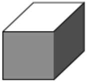
$$\mathbf{X}' = H_p \mathbf{X} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & v \end{pmatrix} \mathbf{X}.$$

where H_p is a homography



Similarity Affine Projective

Hierarchy of 3D Transformations

Group	Matrix	Distortion	Invariant properties
Projective 15 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^{\mathsf{T}} & v \end{bmatrix}$		Intersection and tangency of surfaces in contact. Sign of Gaussian curvature.
Affine 12 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids. The plane at infinity, π_∞, (see section 3.5).
Similarity 7 dof	$\begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$		The absolute conic, Ω_∞, (see section 3.6).
Euclidean 6 dof	$\begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$		Volume.

Recovering affinity and similarity in 3D

- The plane at infinity has the canonical position $\mathbf{\Pi}_\infty = (0,0,0,1)^T$ in affine 3-space. (analogous to $\ell_\infty = (0,0,1)^T$ in $\mathbb{P}^2$)
- It contains the points (and directions) at infinity $(x_1, x_2, x_3, 0)^T$.
- **$\mathbf{\Pi}_\infty$ enables the identification of affine properties such as parallelism**:
  - Two planes are parallel if and only if they intersect at a line in $\mathbf{\Pi}_\infty$.
  - Two lines are parallel (also, a line is parallel to a plane), if they intersect at a point in $\mathbf{\Pi}_\infty$.
- In $\mathbb{P}^3$, any pair of planes intersect in a line. When the planes are parallel, they intersect in a line at $\mathbf{\Pi}_\infty$.
- An affinity keeps fixed the plane $\mathbf{\Pi}_\infty$ at infinity . That is, $\mathbf{\Pi}_\infty$ is a fixed plane under the projective transformation H if, and only if H is an affinity.
- A similarity keeps fixed the absolute conic $\mathbf{\Omega}_\infty$ at infinity. That is, $\mathbf{\Omega}_\infty$ is a fixed conic under the projective transformation H if, and only if H is a similarity.

Recovering affinity and similarity in 3D

- **Π_∞ allows to remove the projective distortion in a 3D reconstruction from N cameras** and therefore allows to identify parallel lines.

- **Ω_∞ allows to remove the projective distortion in a 3D reconstruction from N cameras** and therefore allows to identify orthogonal lines.

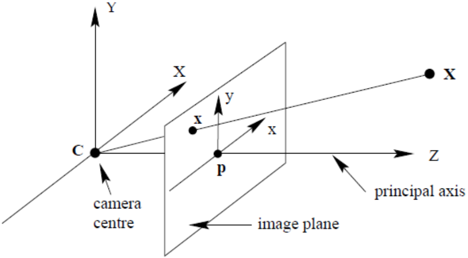
- When we transform a 3D reconstruction with a transformation fixing the plane at infinity to $(0,0,0,1)^T$, **the 3D reconstruction is related to the 3D scene by an affine transformation**.

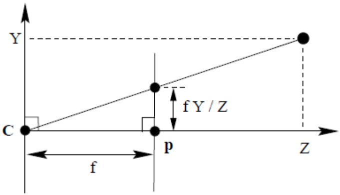
- When we transform a 3D reconstruction with a transformation fixing the absolute conic at its canonical position ($X_1^2 + X_2^2 + X_3^2 = 0$ and $X_4 = 0$) **the 3D reconstruction is related to the 3D scene by a similarity transformation**.

Outline

1. Homography Estimation
2. Projective geometry and transformations in 3D
3. Camera models.

The pinhole camera model

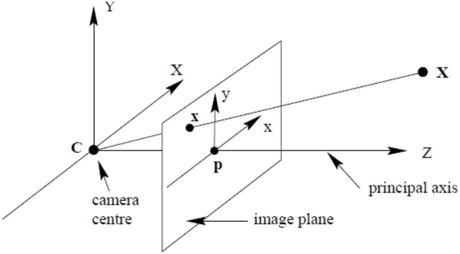


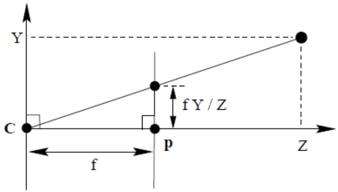


Assuming that the optic is perfect and the camera realizes a central projection from the world to the image plane, consider a reference frame where its origin is the center of projection and the image plane is given by $Z = f$ where f is the focal length of the camera. A point $(X, Y, Z)^T$ in the world is mapped to the point $(fX/Z, fY/Z, f)^T$ in the image plane. This gives a map from $\mathbb{R}^3$ to $\mathbb{R}^2$ written as:

$$(X, Y, Z)^T \to (fX/Z, fY/Z)^T.$$

Pinhole camera

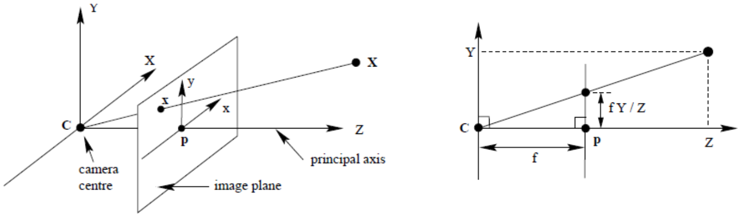




Camera anatomy:

Camera centre (or optical centre): the centre of projection.
Principal axis (or principal ray): the lne from the camera centre perpendicular to the **image plane**.
Principal point P: where the principal axis meets the image plane.
Principal plane: the plane through the camera centre parallel to the image plane.

Pinhole camera



$$(X, Y, Z)^T \to (fX/Z, fY/Z)^T.$$

Notice that this map is non-linear in (X, Y, Z). It will be linear if we write it in projective coordinates of $\mathbb{P}^3$ and $\mathbb{P}^2$:

$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

Image and camera coordinate systems

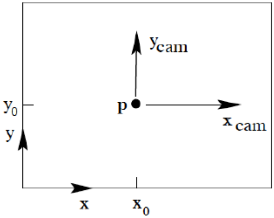


Fig. 6.2. *Image* (x, y) *and camera* (x_{cam}, y_{cam}) *coordinate systems.*

If the origin of coordinates in the image plane is not at the principal point:

$$(X, Y, Z)^T \rightarrow (fX/Z + p_x, fY/Z + p_y)^T$$

and

$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Image and camera coordinate systems

We define the intrinsic matrix K

$$K = \begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

and therefore

$$\mathbf{x} = K[\mathrm{I}|0]\mathbf{X}_{cam}$$

K is also called camera calibration matrix. Note that $\mathbf{X}_{cam}$ is in the coordinate frame of the camera.
The most general configuration of K (CCD cameras):

$$K = \begin{pmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

$s = 0$ pretty much always.

Transformation between the world and camera reference systems

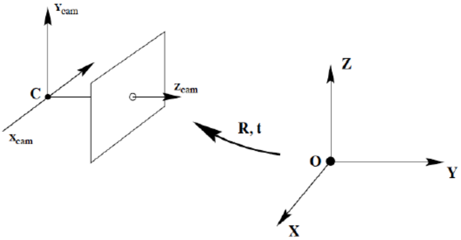


Fig. 6.3. *The Euclidean transformation between the world and camera coordinate frames.*

If points are expressed in terms of the world coordinate frame $\mathbf{O}$, $\tilde{\mathbf{X}}_{cam} = R(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$, which in homogeneous coordinates is:

$$\mathbf{X}_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0^T & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\tilde{C} \\ 0^T & 1 \end{bmatrix} \mathbf{X}$$

Transformation between the world and camera reference systems

In previous equation, if we take into account that

$$\mathbf{x} = K[\mathrm{I}|0]\mathbf{X}_{cam}$$

Then we can write

$$\mathbf{x} = KR[\,\mathrm{I}\,|-\tilde{C}]\mathbf{X}$$

Normally, $\tilde{\mathbf{X}}_{cam}$ is expressed as $\tilde{\mathbf{X}}_{cam} = R(\tilde{\mathbf{X}} + \mathbf{t})$ and the transformation is $P = K[R|\mathbf{t}]$ and the expression becomes

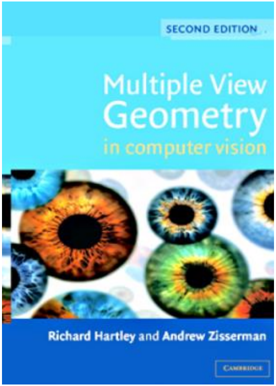
$$\mathbf{x} = P\mathbf{X}$$

where $\mathbf{t} = -R\tilde{C}$ and P is the **camera matrix**

K is the internal orientation of P. R and $\mathbf{t}$ are the external orientation of P. The calibration of a camera is the process of finding the internal and external orientations.

References

[Hartley and Zisserman 2004] R.I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.



References

[Hartley and Zisserman 2004] R.I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.