Analytical 3d - Introduction

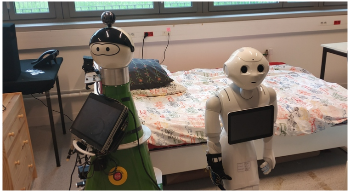
Antoine Manzanera

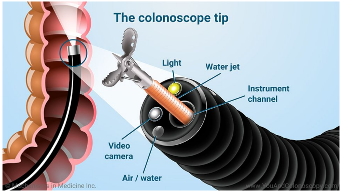
ENSTA Paris

ROB317 - 3d Computer Vision

Motivations: 3d Reconstruction from Videos

Reconstructing the scene geometry from videos is useful in many applications: Robot navigation (obstacle detection), Metrology, 3d Cartography, Medicine...





- \+ It is a cheap and flexible approach: One single passive camera, Adaptive baseline,...
- – It strongly relies on scene structure (texture) and precise camera positioning.

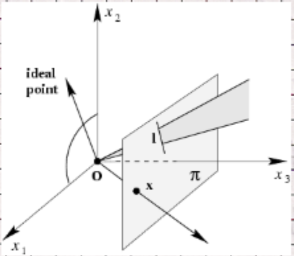
Presentation Outline

Presentation Outline

Projective Geometry in $\mathbb{P}^2$

- Homogeneous coordinates $\rightarrow$ additional component $\rightarrow$ non injective representation
- Affine transformations represented by linear functions $\rightarrow$ simpler operations
- Points and lines at infinity represented with finite coordinates

Projective Geometry in $\mathbb{P}^2$



$\mathbb{R}^2 \qquad\qquad \mathbb{P}^2$

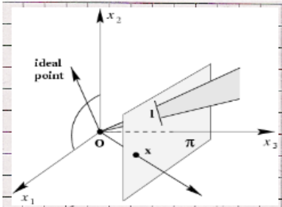
$$\begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} u/w \\ v/w \end{pmatrix} \longleftarrow \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

- Equivalence Classes: $\forall \lambda \neq 0 \quad \lambda x \equiv x$
- Duality point / line:

  $m = (x, y, 1)^t \qquad \ell = (a, b, c)^t$
- Ideal points: $(x, y, 0)^t$
- Line at infinity: $(0, 0, 1)^t$

Projective Geometry in $\mathbb{P}^2$



Point m belongs to line ℓ

$$\Updownarrow$$

$$\boxed{m^t \ell = 0}$$

Point m is at the intersection of lines ℓ and ℓ':

$$\boxed{\ell \times \ell' = m}$$

Line ℓ passes through points m and m':

$$\boxed{m \times m' = \ell}$$

Notation:

pre-vector product: $[U]_\times = \begin{pmatrix} 0 & -w & v \\ w & 0 & -u \\ -v & u & 0 \end{pmatrix}$

with $U = (u, v, w)^t$

Then: $\boxed{U \times U' = [U]_\times U'}$

Projective transformations

- A projective transformation h of the plane is characterized by the fact that: if three point m_1, m_2 and m_3 are aligned, $h(m_1)$, $h(m_2)$ and $h(m_3)$ are aligned too.
- A function $h : \mathbb{P}^2 \mapsto \mathbb{P}^2$ is a projective transformation if and only if there exists a non singular 3×3 matrix H such that $\forall m \in \mathbb{P}^2, h(m) = Hm$.

Projective transformations 1: Translations

$$H = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

- with $\mathbf{t} = (t_x \ t_y)^T$ translation vector
- 2 degrees of freedom

Projective transformations 2: Isometries

$$H = \begin{pmatrix} \cos(\theta) & -\varepsilon \sin(\theta) & t_x \\ \sin(\theta) & \varepsilon \cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

- with $\mathbf{t} = (t_x \ t_y)^T$ translation vector
- θ rotation angle
- $\varepsilon = \pm 1 \rightarrow$ direct / indirect isometry
- 3 degrees of freedom
- *preserves:* angles, lengths, areas

Projective transformations 3: Similarities

$$H = \begin{pmatrix} s\cos(\theta) & -s\sin(\theta) & t_x \\ s\sin(\theta) & s\cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

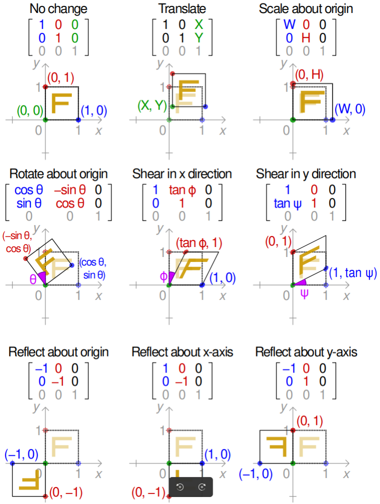
- with $\mathbf{t} = (t_x \ \ t_y)^T$ translation vector
- θ rotation angle
- s homothety factor
- 4 degrees of freedom
- *preserves:* angles, ratios of lengths/areas, parallel lines

Projective transformations 4: Affine transformations

$$H = \begin{pmatrix} a_1^1 & a_1^2 & t_x \\ a_2^1 & a_2^2 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

- 6 degrees of freedom
- *preserves:* ratios of areas, parallel lines

(Figure from Wikipedia)



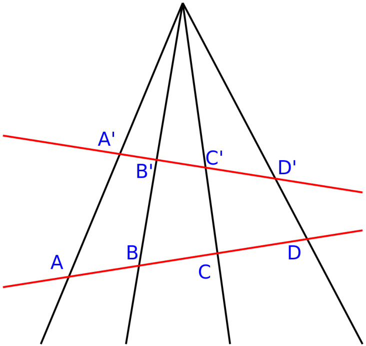
Projective transformations 5: Homographies

$$H = \begin{pmatrix} a_1^1 & a_1^2 & t_x \\ a_2^1 & a_2^2 & t_y \\ v_1 & v_2 & 1 \end{pmatrix}$$

- $\mathbf{v} = (v_1 \ v_2)^T$ relates to the action on points/lines at infinity
- 8 degrees of freedom
- *preserves:* cross-ratios of four points on a line:

$$\frac{AC \times BD}{AD \times BC} = \frac{A'C' \times B'D'}{A'D' \times B'C'}$$

(Figure from Wikipedia)



Homographies on points/lines at infinity

Consider a line (or a point!) at infinity $\mathbf{l}_\infty = (l_1 \ l_2 \ 0)^T$
When applied an affine transformation:

$$\begin{pmatrix} a_1^1 & a_1^2 & t_x \\ a_2^1 & a_2^2 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} l_1 \\ l_2 \\ 0 \end{pmatrix} = \begin{pmatrix} l_1 a_1^1 + l_2 a_1^2 \\ l_1 a_2^1 + l_2 a_2^2 \\ 0 \end{pmatrix}$$

A line at infinity remains at infinity!
When applied a general homography:

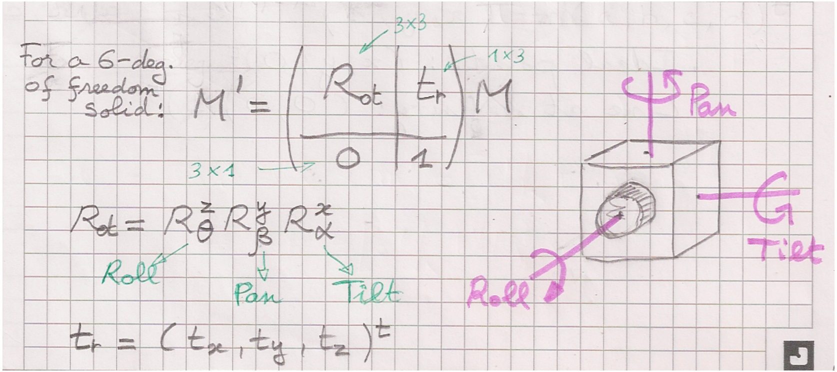
$$\begin{pmatrix} a_1^1 & a_1^2 & t_x \\ a_2^1 & a_2^2 & t_y \\ v_1 & v_2 & 1 \end{pmatrix} \begin{pmatrix} l_1 \\ l_2 \\ 0 \end{pmatrix} = \begin{pmatrix} l_1 a_1^1 + l_2 a_1^2 \\ l_1 a_2^1 + l_2 a_2^2 \\ l_1 v_1 + l_2 v_2 \end{pmatrix}$$

A line at infinity becomes finite!
This allows to observe vanishing points and horizon lines.

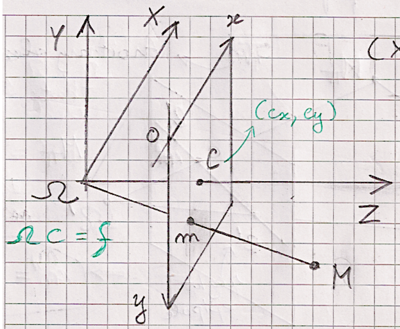
Projective Geometry in $\mathbb{P}^3$

- $\mathbb{R}^3 \leftrightarrow \mathbb{P}^3$: $(X,Y,Z) \rightarrow (X,Y,Z,1)$; $(u/h, v/h, w/h) \leftarrow (u,v,w,h)$
- Duality point / plane: $M = (X,Y,Z,1)^t$ / $\Pi = (a,b,c,d)$.
- Lines are defined from 2 points or from 2 planes!

$\mathbb{P}^3$ allows to express linearly affine transformations:



Camera (Calibration) Matrix: Intrinsics



$$(X, Y, Z)^t \longmapsto \underbrace{\left(\frac{fX}{Z} + c_x, \frac{fY}{Z} + c_y\right)^t}_{m}$$

$$K = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

Calibration (intrinsics) Matrix

$$\underbrace{\tilde{m}}_{\in \mathbb{P}^2} = \begin{pmatrix} fX + Zc_x \\ fY + Zc_y \\ Z \end{pmatrix} = \underbrace{(\underbrace{K}_{3\times 3} \mid \underbrace{0_3}_{1\times 3})}_{P} \underbrace{\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}}_{\in \mathbb{P}^3}$$

Projection and Back-Projection Matrices

$M = (X, Y, Z)^t \in \mathbb{R}^3$
$m = (x, y)^t \in \mathbb{R}^2$, and $\tilde{m} = (x, y, 1)^t \in \mathbb{P}^2$

Camera (Projection) Matrix

$$m = \pi(M) = \left(f\frac{X}{Z} + c_x, f\frac{X}{Z} + c_x\right)$$

Equivalent to:

$$\tilde{m} = KM$$

with: $K = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}$

Back-Projection Matrix

$$M = \pi^{-1}(m, Z) = \left(Z\frac{x-c_x}{f}, Z\frac{y-c_y}{f}, Z\right)$$

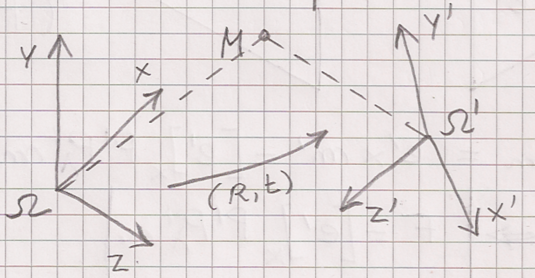
Equivalent to:

$$M = \underbrace{Z}_{\text{Depth}}\underbrace{K^{-1}\tilde{m}}_{\text{Direction}}$$

with: $K^{-1} = \begin{pmatrix} \frac{1}{f} & 0 & -\frac{c_x}{f} \\ 0 & \frac{1}{f} & -\frac{c_y}{f} \\ 0 & 0 & 1 \end{pmatrix}$

Displacement Matrix: Extrinsics

$$\tilde{m}' = \underbrace{\left(K \,|\, 0_3\right) \left(\begin{array}{c|c} R & -Rt \\ \hline 0_3^t & 1 \end{array}\right)}_{P'} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$



$m = PM$

$m' = P'M$

Presentation Outline

Homographies

- **Homography** $\rightarrow$ Most general case of 2d projective transformation

$$\tilde{m}' = H\tilde{m}$$

- 8 degrees of freedom $\rightarrow$ At least four non colinear 2d points!
- Corresponds to 2 particular cases of image pairs:
  - 3d scene viewed under pure rotation around the optical centre ($\mathbf{t} = O_3$).
  - Same plane viewed under two different 3d poses.

Rotation around the optical centre

In the case of a pure rotation around the optical centre ($\mathbf{t} = O_3$), the projected image transformation is a homography:

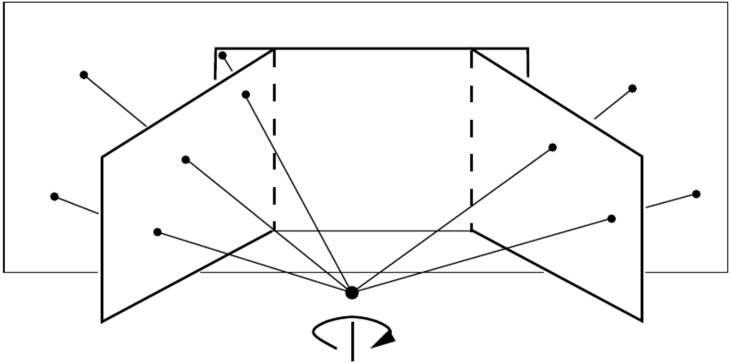


Figure from [**Hartley and Zisserman 2004**]

Rotation around the optical centre

Since $\mathbf{t} = O_3$ we get:

$$\tilde{m} = \left(\begin{array}{c|c} K & O_3 \end{array} \right) \tilde{M}$$

$$\tilde{m}' = \left(\begin{array}{c|c} K & O_3 \end{array} \right) \left(\begin{array}{c|c} R & \color{red}{O_3} \\ \hline O_3^t & 1 \end{array} \right) \tilde{M}$$

which can be written more simply:

$$\begin{array}{rcl} \tilde{m} & = & KM \\ \tilde{m}' & = & KRM = \underbrace{KRK^{-1}}_{H} \tilde{m} \end{array}$$

Rotation around the optical centre

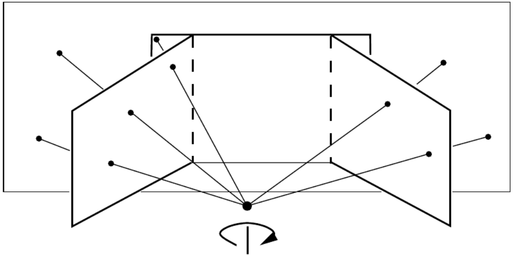
Note the difference between rotation around the optical centre ((a) to (b)), and translation ((a) to (c)):

a b c

Images from [**Hartley and Zisserman 2004**]

Rotation around the optical centre

Since there is no parallax, the images can be stitched to form a mosaic:

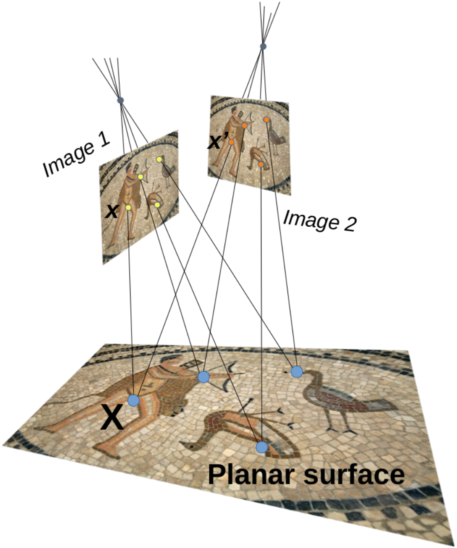


Plane viewed from different poses

$$\tilde{x} = H_{\pi,1} X$$

$$\tilde{x}' = H_{\pi,2} X$$

$$\tilde{x}' = H_{\pi,2} H_{\pi,1}^{-1} \tilde{x} = H_{\pi} \tilde{x}$$



Plane viewed from different poses

Let us first assume that $K = I_3$ (i.e. $f = 1, c_x = c_y = 0$). Then if the pose of the right camera is given by rotation matrix R and translation vector $\mathbf{t}$, we get:

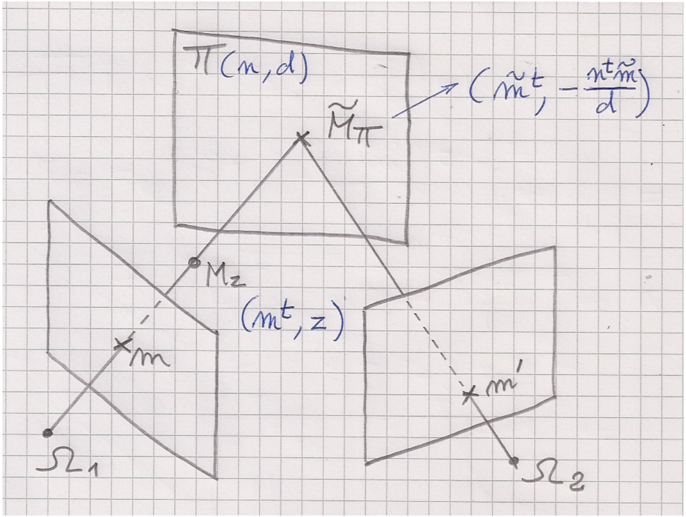
$$\tilde{m} = P\tilde{M} = \left(\begin{array}{c|c} I_3 & O_3 \end{array} \right) \tilde{M}$$

$$\tilde{m}' = P'\tilde{M} = \left(\begin{array}{c|c} R & \mathbf{t} \end{array} \right) \tilde{M}$$

Every point on the ray $M_z = (m^t, z)$ (parameterized by z) projects on m.
If the point M_z is on the plane π, it must satisfy: $\pi^t.\tilde{M}_z = 0$.
If the coordinates of the plane are given as $\pi = (\mathbf{n}^t, d)^t$, so that for points M on the plane, we have: $\mathbf{n}^t M + d = 0$,
then the point of the ray backprojected from m and intersecting plane π is:

$$\tilde{M}_\pi = \left(\tilde{m}^t, -\frac{\mathbf{n}^t \tilde{m}}{d} \right)^t$$

Plane viewed from different poses



Plane viewed from different poses

The point of the ray backprojected from m and intersecting plane π is:

$$\tilde{M}_\pi = \left(\tilde{m}^t, -\frac{\mathbf{n}^t\tilde{m}}{d}\right)^t$$

And then:

$$\begin{aligned}
\tilde{m}' &= P'\tilde{M}_\pi = \left(\begin{array}{c|c} R & \mathbf{t} \end{array}\right)\tilde{M}_\pi \\
&= R\tilde{m} - \tfrac{\mathbf{t}\mathbf{n}^t}{d}\tilde{m} \\
&= \underbrace{\left(R - \frac{\mathbf{t}\mathbf{n}^t}{d}\right)}_{H_\pi}\tilde{m}
\end{aligned}$$

Finally, by considering the internal parameter matrix K of a single camera moved with rotation R and translation $\mathbf{t}$, the homography related to the plane $\pi = (\mathbf{n}^t, d)^t$ is given by:

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

Presentation Outline

Estimation of a Homography

Now we wish to estimate the parameters of a homography using a set of correspondances from a pair of images:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_1^1 & a_1^2 & t_x \\ a_2^1 & a_2^2 & t_y \\ v_1 & v_2 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- In the following practical session we will use a the Direct Linear Transform (DLT) resolved by Singular Values Decomposition (SVD).
- The next slides are adapted from **Gianni Franchi**'s 2022 course.

Estimation by Direct Linear Transformation (DLT)

Let us rearrange the equation

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

we use auxiliary 1×3 vectors $\mathbf{h_1}$, $\mathbf{h_2}$ and $\mathbf{h_3}$:

$$\mathbf{x}' = \begin{bmatrix} \mathbf{h_1} \\ \mathbf{h_2} \\ \mathbf{h_3} \end{bmatrix} \mathbf{x}$$

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \mathbf{h_1 x} \\ \mathbf{h_2 x} \\ \mathbf{h_3 x} \end{bmatrix}$$

Estimation by Direct Linear Transformation (DLT)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \color{red}{\mathbf{h_1}}\mathbf{x} \\ \color{blue}{\mathbf{h_2}}\mathbf{x} \\ \color{magenta}{\mathbf{h_3}}\mathbf{x} \end{bmatrix}$$

$$x' = \frac{u'}{w'} = \frac{\mathbf{h_1 x}}{\mathbf{h_3 x}}$$

$$y' = \frac{v'}{w'} = \frac{\mathbf{h_2 x}}{\mathbf{h_3 x}}$$

Estimation by Direct Linear Transformation (DLT)

We can rewrite the equations:

$$\begin{cases} -\mathbf{h_1 x} & & +x'\mathbf{h_3 x} = 0 \\ & -\mathbf{h_2 x} & +y'\mathbf{h_3 x} = 0 \end{cases}$$

we want to estimate $\mathbf{h_1}$, $\mathbf{h_2}$ and $\mathbf{h_3}$

Estimation by Direct Linear Transformation (DLT)

Let us write $\mathbf{h} = \begin{bmatrix} \mathbf{h_1} & \mathbf{h_2} & \mathbf{h_3} \end{bmatrix}^t$. $\mathbf{h}$ is a vector of size 9×1.
We can rewrite the previous system with $\mathbf{h}$, as follows:

$$\begin{cases} \mathbf{a}_x^t \mathbf{h} = 0 \\ \mathbf{a}_y^t \mathbf{h} = 0 \end{cases}$$

with

$$\mathbf{a}_x^t = \begin{bmatrix} -\mathbf{x}^t & \mathbf{0}_3^t & x'\mathbf{x}^t \end{bmatrix}$$

$$\mathbf{a}_x^t = \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \end{bmatrix}$$

$$\mathbf{a}_y^t = \begin{bmatrix} \mathbf{0}_3^t & -\mathbf{x}^t & y'\mathbf{x}^t \end{bmatrix}$$

$$\mathbf{a}_y^t = \begin{bmatrix} 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{bmatrix}$$

Estimation by Direct Linear Transformation (DLT)

Now let us consider that we have multiple pairs of points indexed by i:

$$\mathbf{a}_{x_i}^t = \begin{bmatrix} -\mathbf{x}_i^t & \mathbf{0}^t & x_i'\mathbf{x}_i^t \end{bmatrix}$$

$$\mathbf{a}_{y_i}^t = \begin{bmatrix} \mathbf{0}^t & -\mathbf{x}_i^t & y_i'\mathbf{x}_i^t \end{bmatrix}$$

We can rewrite the previous system for the N pairs of points:

$$\begin{cases} \mathbf{a}_{x_1}^t \mathbf{h} = 0 \\ \mathbf{a}_{y_1}^t \mathbf{h} = 0 \\ \vdots \\ \mathbf{a}_{x_N}^t \mathbf{h} = 0 \\ \mathbf{a}_{y_N}^t \mathbf{h} = 0 \end{cases}$$

Collecting everything together we have:

$$\underbrace{\mathbf{A}}_{2N\times 9}\underbrace{\mathbf{h}}_{9\times 1} = \underbrace{\mathbf{0}}_{9\times 1}$$

Estimation by Direct Linear Transformation (DLT)

- if we use $N = 4$ then we have an exact solution
- if we use $N > 4$ then we have an **over-determined solution**. There are no exact solution, hence we need to find approximate solution.
- Additional constraint is needed to avoid 0, e.g. $\|\mathbf{h}\|_2^2 = 1$

Estimation of $\mathbf{h}$: Minimisation

In the case of redundant observations we get inconsistencies (due to the noise).
Let us write $\mathbf{A}\mathbf{h} = \mathbf{w}$.
Our goal is to find $\mathbf{h}$ such that:

$$\hat{\mathbf{h}} = \arg\min_{\mathbf{h}} \mathbf{w}^t\mathbf{w}$$

$$\hat{\mathbf{h}} = \arg\min_{\mathbf{h}} \mathbf{h}^t\mathbf{A}^t\mathbf{A}\mathbf{h}$$

with $\|h\|_2^2 = 1$
How do we minimize the loss?

Estimation of **h**: Singular Value Decomposition

The eigenvector belonging to the smallest eigenvalue of $\mathbf{A}^t\mathbf{A}$ provides the solution of the over-determined, constrained system of linear equations:

$$\underbrace{\mathbf{A}}_{2N\times 9} = \underbrace{\mathbf{U}}_{2N\times 9}\underbrace{\mathbf{S}}_{9\times 9}\underbrace{\mathbf{V}}_{9\times 9} = \sum_{i=1}^{9} s_i \mathbf{u}_i \mathbf{v}_i^t$$

with $\mathbf{U}^t\mathbf{U} = \mathbf{I}_9$ and $\mathbf{V}^t\mathbf{V} = \mathbf{I}_9$
The vector v_i are orthonormal since

$$v_i v_j^t = \begin{cases} 0 \text{ if } i \neq j \\ 1 \text{ if } i = j \end{cases}$$

So, **h** is equal to v_9, with s_9 the smallest eigen value.

Estimation of **h**: Singular Value Decomposition

The estimate of **h** is given by

$$\hat{\mathbf{h}} = \begin{bmatrix} \hat{\mathbf{h_1}} & \hat{\mathbf{h_2}} & \hat{\mathbf{h_3}} \end{bmatrix}^t = v_9$$

This leads to the estimated projection matrix.
No solution if too many points $\mathbf{x}_i$ are on a line.

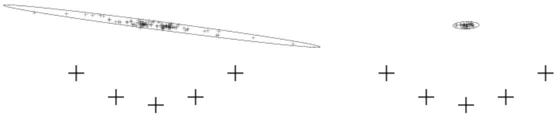
DLT + SVD algorithm

Objective:
Given $N \geq 4$ 2d to 2d point correspondences $(\mathbf{x}_i, \mathbf{x}'_i)$, determine the 2d homography matrix $\mathbf{H}$ such that $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$.
Algorithm:

- For each correspondence $(\mathbf{x}_i, \mathbf{x}'_i)$ compute $\mathbf{A}_i$.
- Assemble N 2×9 matrices $\mathbf{A}_i$ into a single $2N \times 9$ matrix $\mathbf{A}$
- Obtain SVD of $\mathbf{A}$. Solution for $\mathbf{h}$ is the last row of $\mathbf{V}$
- Determine $\mathbf{H}$ from $\mathbf{h}$

Estimation of $\mathbf{h}$: Data ranges

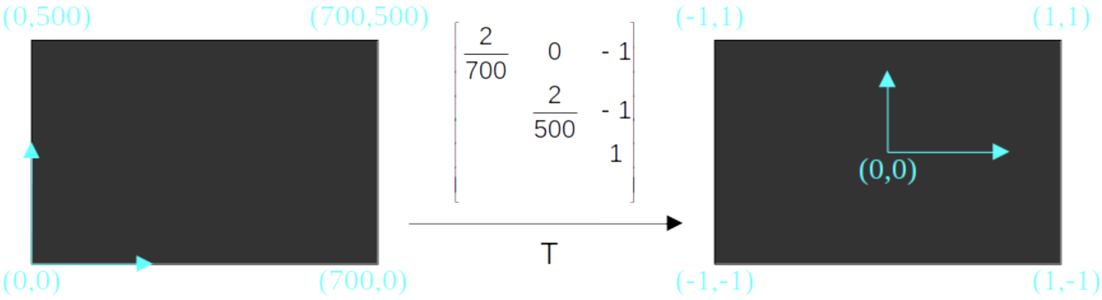
$$\begin{array}{c}\mathbf{A}_i^t = \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{bmatrix} \\ \quad\quad 10^2 \quad 10^2 \quad 1 \quad 10^2 \quad 10^2 \quad 1 \quad 10^4 \quad 10^4 \quad 10^2 \end{array}$$



Dependence of error distribution on the dimensions of images.

How to transform them so that the coordinates are within $[-1, 1]$?

Estimation of **h**: Data normalisation



Normalised DLT algorithm

Objective:
Given $N \geq 4$ 2d to 2d point correspondences $(\mathbf{x}_i, \mathbf{x}'_i)$, determine the 2d homography matrix $\mathbf{H}$ such that $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$.
Algorithm:

- Apply the normalisation $\tilde{\mathbf{x}}_i = \mathbf{T}_{\text{norm}}\mathbf{x}_i$ and $\tilde{\mathbf{x}}'_i = \mathbf{T}_{\text{norm}}\mathbf{x}'_i$
- Apply DLT with $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_i)$
- Denormalise the homography: $\mathbf{H} = \mathbf{T}_{\text{norm}}^{-1}\tilde{\mathbf{H}}\mathbf{T}_{\text{norm}}$