Master in Computer Vision Barcelona

Module: 3D Vision
Lecture 2, part 2: Affine and metric rectification.
Homography estimation.
Lecturer: Coloma Ballester

Objects in the 3D world are transformed into image objects through a projective transformation.

Remember: Why is projective geometry useful?

- The image capture of the camera is described with a linear transformation.
- The intersection of two lines and the line that passes through two points is a linear operation.
- Points at infinity have a natural representation.
- 2D projective geometry:

Remember: Why is projective geometry useful?

- The image capture of the camera is described with a linear transformation.
- The intersection of two lines and the line that passes through two points is a linear operation.
- Points at infinity have a natural representation.
- 2D projective geometry:

Remember: Why is projective geometry useful?

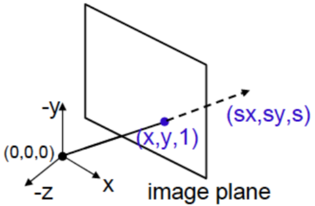
- The image capture of the camera is described with a linear transformation.
- The intersection of two lines and the line that passes through two points is a linear operation.
- Points at infinity have a natural representation.
- 2D projective geometry:

Remember: Why is projective geometry useful?

- The image capture of the camera is described with a linear transformation.
- The intersection of two lines and the line that passes through two points is a linear operation.
- Points at infinity have a natural representation.
- 2D projective geometry:

The projective plane

A point in the image is a **ray** in the projective space.



Each point (x, y) on the (image) plane is represented by a ray: (sx, sy, s), the **visual ray**, **visual direction** or **view direction**.

All points on the ray are equivalent: $(x, y, 1) \equiv (sx, sy, s)$

The Projective Plane $\mathbb{P}^2$

$$\mathbb{P}^2 = \underbrace{\{(x_1, x_2, x_3)^T \in \mathbb{R}^3 : x_3 \neq 0\}}_{\text{classic points}} \cup \underbrace{\Pi_0}_{\text{ideal points}} = \mathbb{R}^3 \setminus \{(0,0,0)\}$$

where

- Classic points ($x_3 \neq 0$): $(x_1, x_2, x_3) \equiv (sx_1, sx_2, sx_3)$, for $s \neq 0$.
  Homogeneous coordinates versus Cartesian coordinates.

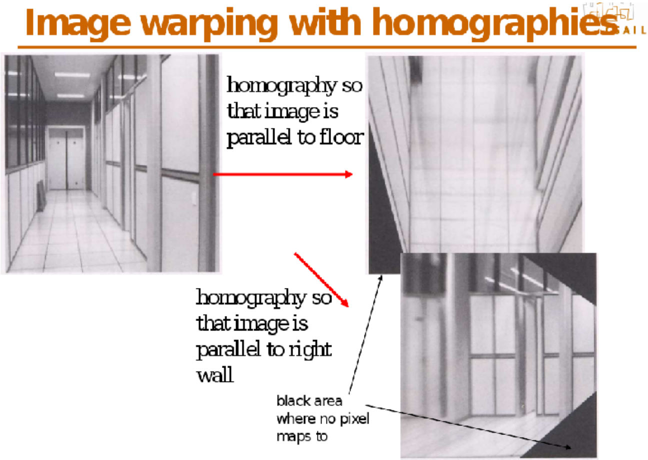
- Ideal points ($x_3 = 0$): $\Pi_0 = \{(x_1, x_2, 0)^T : (x_1, x_2) \neq (0,0)\}$,

  the line at infinity $\ell_\infty = (0,0,1)^T$: $< \ell_\infty, \mathbf{x} >= 0$ for $\mathbf{x} \in \Pi_0$.

- The intersection of any line $\ell = (a, b, c)^T$ and ℓ_∞ is
  $\mathbf{x} = (b, -a, 0)^T$: a point at infinity (or ideal point).
  In other words, $\mathbf{x} = (b, -a, 0)^T$ is the point at infinity on the line ℓ;
  $\mathbf{x} = (b, -a, 0)^T$ is a view direction that is in Π_0.

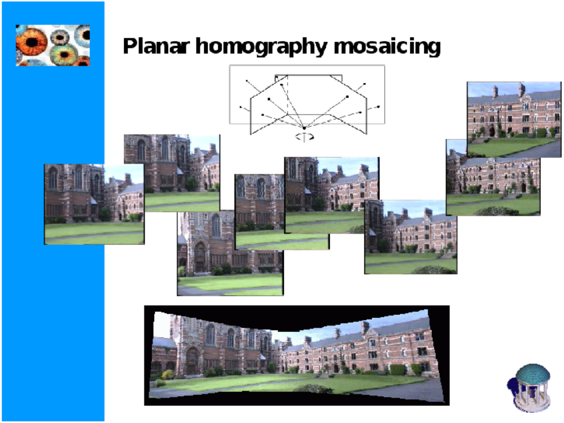
Now, let's see how 2D projective geometry allows

for instance,
to remove the projective distortion of flat objects
and build image mosaics (panoramas).



Now, let's see how 2D projective geometry allows

for instance,
to remove the projective distortion of flat objects
and build image mosaics (panoramas).



Lecture 2, part 2

Overview

1. Removing projective distortions:
   - Affine rectification.
   - Metric rectification.
2. Effectifely computing planar transformations:
   The Direct Linear Transformation (DLT) algorithm to compute a projective transformation that relates two images.

1. **Goal: To remove projective distortions.**

Affine and metric rectification

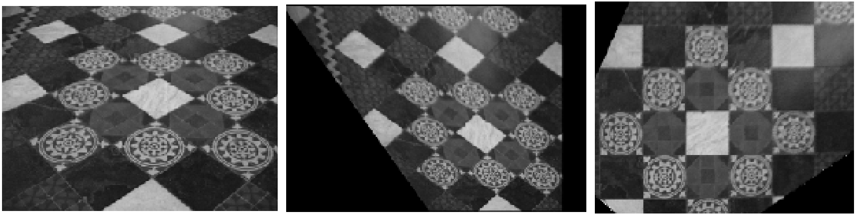


Image source: [Hartley Zisserman 2004]

Affine and metric rectification

- Suppose that a camera is observing a plane in the 3D scene.

Affine and metric rectification

- Suppose that a camera is observing a plane in the 3D scene.
- We have two planes: the plane of the scene Π and the plane of the image Π_{im}. We look at both planes as projective planes $\mathbb{P}^2$.

Affine and metric rectification

- Suppose that a camera is observing a plane in the 3D scene.

- We have two planes: the plane of the scene Π and the plane of the image $\Pi_{\rm im}$. We look at both planes as projective planes $\mathbb{P}^2$.

- Each ray from the origin of the camera to a point in the scene plane intersects the image plane at an image point. Since three aligned points in Π are projected to three aligned points in $\Pi_{\rm im}$, then **the projection is an homography H of** $\mathbb{P}^2$.

Affine and metric rectification

• Suppose that a camera is observing a plane in the 3D scene.

• We have two planes: the plane of the scene Π and the plane of the image Π_{im}. We look at both planes as projective planes $\mathbb{P}^2$.

• Each ray from the origin of the camera to a point in the scene plane intersects the image plane at an image point. Since three aligned points in Π are projected to three aligned points in Π_{im}, then **the projection is an homography H of** $\mathbb{P}^2$.

• The line at infinity ℓ_∞ in Π is projected into a line ℓ in Π_{im}.

Thus, planar transformations are one of the ingredients

Let $\mathbf{x}$ and $\mathbf{x}'$ be homogeneous coordinates.

$$\mathbf{x}' = H\mathbf{x} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix} \mathbf{x}$$

where A is an affine transformation of $\mathbb{R}^2$, $\vec{t} \in \mathbb{R}^2$ is a translation vector, $\vec{v}^T = (v_1, v_2)$ and where we normalized the $(3,3)$ entry of H to be $v = 1$.

Remember:

- H is called a **2D projective transformation** (or a **planar transformation**, or a **projective transformation of $\mathbb{P}^2$**, or a **2D homography**.)
- H is a non-singular matrix $\rightarrow$ represents an invertible mapping

Thus, planar transformations are one of the ingredients

Let $\mathbf{x}$ and $\mathbf{x}'$ be homogeneous coordinates.

$$\mathbf{x}' = H\mathbf{x} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix}\mathbf{x}$$

Key difference between a projective and affine transformation:

- As $\vec{v} = (v_1, v_2)^T \neq \vec{0}$ for a **projectivity** H, then H **maps an ideal point** $\begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix}$ **to a finite point** $\begin{pmatrix} A\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ v_1x_1 + v_2x_2 \end{pmatrix}$.
- However, if H is an **affinity**, the ideal point remains ideal (i.e. at infinity).

Even more: Characterization of affinities in the plane

Let H be a projective transformation of $\mathbb{P}^2$. Then,

H is an **affinity** if and only if **the line at infinity ℓ_∞ is a fixed line under H**.

Indeed:

Remember how does the map H act on lines. Indeed, if H transforms points as $\mathbf{x}' = H\mathbf{x}$, H transforms lines as $\ell' = H^{-T}\ell$.

If H is an affinity we can write $H = \begin{pmatrix} A & \vec{t} \\ 0 & 1 \end{pmatrix}$. Then $H^{-T} = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix}$.

Observe that

$$H^{-T}\ell_\infty = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \ell_\infty.$$

Conversely, let $H = (H_{ij})_{i,j=1}^3$ be a 2D homography. Assume that $H^{-T}\ell_\infty = \ell_\infty$. Let $\mathbf{x} = (1,0,0)^T \in \ell_\infty$. Since $H(1,0,0)^T \in \ell_\infty$ we deduce that $h_{31} = 0$. Similarly, from $H(0,1,0)^T \in \ell_\infty$ we deduce that $h_{32} = 0$. We conclude that H is an affinity.

Even more: Characterization of affinities in the plane

Let H be a projective transformation of $\mathbb{P}^2$. Then,

H is an **affinity** if and only if **the line at infinity ℓ_∞ is a fixed line under H**.

Indeed:

Remember how does the map H act on lines. Indeed, if H transforms points as $\mathbf{x}' = H\mathbf{x}$, H transforms lines as $\ell' = H^{-T}\ell$.

If H is an affinity we can write $H = \begin{pmatrix} A & \vec{t} \\ 0 & 1 \end{pmatrix}$. Then $H^{-T} = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix}$.

Observe that

$$H^{-T}\ell_\infty = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \ell_\infty.$$

Conversely, let $H = (H_{ij})_{i,j=1}^3$ be a 2D homography. Assume that $H^{-T}\ell_\infty = \ell_\infty$. Let $\mathbf{x} = (1,0,0)^T \in \ell_\infty$. Since $H(1,0,0)^T \in \ell_\infty$ we deduce that $h_{31} = 0$. Similarly, from $H(0,1,0)^T \in \ell_\infty$ we deduce that $h_{32} = 0$. We conclude that H is an affinity.

Even more: Characterization of affinities in the plane

Let H be a projective transformation of $\mathbb{P}^2$. Then,

H is an **affinity** if and only if **the line at infinity ℓ_∞ is a fixed line under H**.

Indeed:

Remember how does the map H act on lines. Indeed, if H transforms points as $\mathbf{x}' = H\mathbf{x}$, H transforms lines as $\ell' = H^{-T}\ell$.

If H is an affinity we can write $H = \begin{pmatrix} A & \vec{t} \\ 0 & 1 \end{pmatrix}$. Then $H^{-T} = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix}$.

Observe that

$$H^{-T}\ell_\infty = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \ell_\infty.$$

Conversely, let $H = (H_{ij})_{i,j=1}^3$ be a 2D homography. Assume that $H^{-T}\ell_\infty = \ell_\infty$. Let $\mathbf{x} = (1,0,0)^T \in \ell_\infty$. Since $H(1,0,0)^T \in \ell_\infty$ we deduce that $h_{31} = 0$. Similarly, from $H(0,1,0)^T \in \ell_\infty$ we deduce that $h_{32} = 0$. We conclude that H is an affinity.

Even more: Characterization of affinities in the plane

Let H be a projective transformation of $\mathbb{P}^2$. Then,

H is an **affinity** if and only if **the line at infinity ℓ_∞ is a fixed line under H**.

Indeed:

Remember how does the map H act on lines. Indeed, if H transforms points as $\mathbf{x}' = H\mathbf{x}$, H transforms lines as $\ell' = H^{-T}\ell$.

If H is an affinity we can write $H = \begin{pmatrix} A & \vec{t} \\ 0 & 1 \end{pmatrix}$. Then $H^{-T} = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix}$.

Observe that

$$H^{-T}\ell_\infty = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \ell_\infty.$$

Conversely, let $H = (H_{ij})_{i,j=1}^3$ be a 2D homography. Assume that $H^{-T}\ell_\infty = \ell_\infty$. Let $\mathbf{x} = (1,0,0)^T \in \ell_\infty$. Since $H(1,0,0)^T \in \ell_\infty$ we deduce that $h_{31} = 0$. Similarly, from $H(0,1,0)^T \in \ell_\infty$ we deduce that $h_{32} = 0$. We conclude that H is an affinity.

Even more: Characterization of affinities in the plane

Let H be a projective transformation of $\mathbb{P}^2$. Then,

H is an **affinity** if and only if **the line at infinity ℓ_∞ is a fixed line under H**.

Indeed:

Remember how does the map H act on lines. Indeed, if H transforms points as $\mathbf{x}' = H\mathbf{x}$, H transforms lines as $\ell' = H^{-T}\ell$.

If H is an affinity we can write $H = \begin{pmatrix} A & \vec{t} \\ 0 & 1 \end{pmatrix}$. Then $H^{-T} = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix}$.

Observe that

$$H^{-T}\ell_\infty = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \ell_\infty .$$

Conversely, let $H = (H_{ij})_{i,j=1}^3$ be a 2D homography. Assume that $H^{-T}\ell_\infty = \ell_\infty$. Let $\mathbf{x} = (1,0,0)^T \in \ell_\infty$. Since $H(1,0,0)^T \in \ell_\infty$ we deduce that $h_{31} = 0$. Similarly, from $H(0,1,0)^T \in \ell_\infty$ we deduce that $h_{32} = 0$. We conclude that H is an affinity.

Even more: Characterization of affinities in the plane

Let H be a projective transformation of $\mathbb{P}^2$. Then,

H is an **affinity** if and only if **the line at infinity ℓ_∞ is a fixed line under H**.

Indeed:

Remember how does the map H act on lines. Indeed, if H transforms points as $\mathbf{x}' = H\mathbf{x}$, H transforms lines as $\ell' = H^{-T}\ell$.

If H is an affinity we can write $H = \begin{pmatrix} A & \vec{t} \\ 0 & 1 \end{pmatrix}$. Then $H^{-T} = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix}$.

Observe that

$$H^{-T}\ell_\infty = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \ell_\infty .$$

Conversely, let $H = (H_{ij})_{i,j=1}^3$ be a 2D homography. Assume that $H^{-T}\ell_\infty = \ell_\infty$. Let $\mathbf{x} = (1,0,0)^T \in \ell_\infty$. Since $H(1,0,0)^T \in \ell_\infty$ we deduce that $h_{31} = 0$. Similarly, from $H(0,1,0)^T \in \ell_\infty$ we deduce that $h_{32} = 0$. We conclude that H is an affinity.

Even more: Characterization of affinities in the plane

Let H be a projective transformation of $\mathbb{P}^2$. Then,

H is an **affinity** if and only if **the line at infinity ℓ_∞ is a fixed line under H**.

<u>Indeed:</u>

Remember how does the map H act on lines. Indeed, if H transforms points as $\mathbf{x}' = H\mathbf{x}$, H transforms lines as $\ell' = H^{-T}\ell$.

If H is an affinity we can write $H = \begin{pmatrix} A & \vec{t} \\ 0 & 1 \end{pmatrix}$. Then $H^{-T} = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix}$.

Observe that

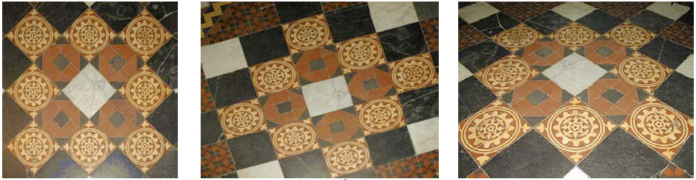
$$H^{-T}\ell_\infty = \begin{pmatrix} A^{-T} & 0 \\ -\vec{t}^T A^{-T} & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \ell_\infty .$$

Conversely, let $H = (H_{ij})_{i,j=1}^3$ be a 2D homography. Assume that $H^{-T}\ell_\infty = \ell_\infty$. Let $\mathbf{x} = (1,0,0)^T \in \ell_\infty$. Since $H(1,0,0)^T \in \ell_\infty$ we deduce that $h_{31} = 0$. Similarly, from $H(0,1,0)^T \in \ell_\infty$ we deduce that $h_{32} = 0$. We conclude that H is an affinity.

Affine and metric rectification

Let $\mathbf{x}$ and $\mathbf{x}'$ be homogeneous coordinates.

$$\mathbf{x}' = H\mathbf{x} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix} \mathbf{x}$$



- We are interested in computing projective distortions that go from world coordinates to image coordinates, **and the other way around**.
- For the moment since we work in $\mathbb{P}^2$ when we use world coordinates we mean Euclidean coordinates and when using image coordinates we refer to a general case with projective distortions.

Affine and metric rectification

$$\mathbf{x}' = H\mathbf{x} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix}\mathbf{x}.$$

- Let us decompose the general projective distortion into a hierarchy that goes from projective to similarity.

$$\mathbf{x} \equiv \mathbf{x}_p \longrightarrow \mathbf{x}_a \longrightarrow \mathbf{x}_s \longrightarrow \mathbf{x}_e \equiv \mathbf{x}'$$

H can decomposed as $H = H_{e\leftarrow s}H_{s\leftarrow a}H_{a\leftarrow p}$, where

$$H_{a\leftarrow p} = \begin{pmatrix} I & \vec{0} \\ \vec{v}^T & 1 \end{pmatrix}, \quad H_{s\leftarrow a} = \begin{pmatrix} K & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}, \quad H_{e\leftarrow s} = \begin{pmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix},$$

and $A = sRK + \vec{t}\vec{v}^T$, where R is a rotation, K upper-triangular matrix, $s > 0$, $\vec{t} \in \mathbb{R}^2$, $\vec{v}^T = (v_1, v_2)$.

Affine and metric rectification

$$\mathbf{x}' = H\mathbf{x} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix} \mathbf{x}.$$

• Let us decompose the general projective distortion into a hierarchy that goes from projective to similarity.

$$\mathbf{x} \equiv \mathbf{x}_p \longrightarrow \mathbf{x}_a \longrightarrow \mathbf{x}_s \longrightarrow \mathbf{x}_e \equiv \mathbf{x}'$$

H can decomposed as $H = H_{e\leftarrow s} H_{s\leftarrow a} H_{a\leftarrow p}$, where

$$H_{a\leftarrow p} = \begin{pmatrix} I & \vec{0} \\ \vec{v}^T & 1 \end{pmatrix}, \qquad H_{s\leftarrow a} = \begin{pmatrix} K & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}, \qquad H_{e\leftarrow s} = \begin{pmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix},$$

and $A = sRK + \vec{t}\vec{v}^T$, where R is a rotation, K upper-triangular matrix, $s > 0$, $\vec{t} \in \mathbb{R}^2$, $\vec{v}^T = (v_1, v_2)$.

Affine and metric rectification

$$\mathbf{x}' = H\mathbf{x} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix} \mathbf{x}.$$

• Let us decompose the general projective distortion into a hierarchy that goes from projective to similarity.

$$\mathbf{x} \equiv \mathbf{x}_p \longrightarrow \mathbf{x}_a \longrightarrow \mathbf{x}_s \longrightarrow \mathbf{x}_e \equiv \mathbf{x}'$$

H can decomposed as $H = H_{e \leftarrow s} H_{s \leftarrow a} H_{a \leftarrow p}$, where

$$H_{a \leftarrow p} = \begin{pmatrix} I & \vec{0} \\ \vec{v}^T & 1 \end{pmatrix}, \qquad H_{s \leftarrow a} = \begin{pmatrix} K & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}, \qquad H_{e \leftarrow s} = \begin{pmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix},$$

and $A = sRK + \vec{t}\vec{v}^T$, where R is a rotation, K upper-triangular matrix, $s > 0$, $\vec{t} \in \mathbb{R}^2$, $\vec{v}^T = (v_1, v_2)$.

Affine and metric rectification

How to use these results?

$$\mathbf{x}' = H\mathbf{x} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix} \mathbf{x}.$$

$H = H_{e \leftarrow s} H_{s \leftarrow a} H_{a \leftarrow p}$, where $A = sRK + \vec{t}\vec{v}^T$ and

$$H_{a \leftarrow p} = \begin{pmatrix} I & \vec{0} \\ \vec{v}^T & 1 \end{pmatrix}, \qquad H_{s \leftarrow a} = \begin{pmatrix} K & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}, \qquad H_{e \leftarrow s} = \begin{pmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix}.$$

- **If we know** H, the decomposition can be easily computed. Indeed, $\vec{t}, \vec{v}$ are immediately given. Then $sRK = A - \vec{t}\vec{v}^T$ and this can be obtained by decomposing $A - \vec{t}\vec{v}^T$ in the product of a rotation and an upper triangular matrix (for instance, a QR decomposition).

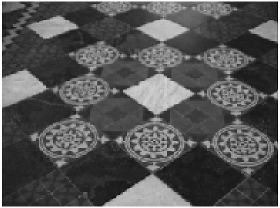
Affine and metric rectification

$$\mathbf{x}' = H\mathbf{x} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix} \mathbf{x}.$$

$H = H_{e\leftarrow s} H_{s\leftarrow a} H_{a\leftarrow p}$, where $A = sRK + \vec{t}\vec{v}^T$ and

$$H_{a\leftarrow p} = \begin{pmatrix} I & \vec{0} \\ \vec{v}^T & 1 \end{pmatrix}, \qquad H_{s\leftarrow a} = \begin{pmatrix} K & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}, \qquad H_{e\leftarrow s} = \begin{pmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix}.$$

- But, in general, **we don't know** H**: In general, we only have the image of a plane in the 3D world:**



Affine and metric rectification

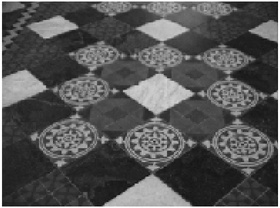
$$\mathbf{x}' = H\mathbf{x} = \begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix}\mathbf{x}.$$

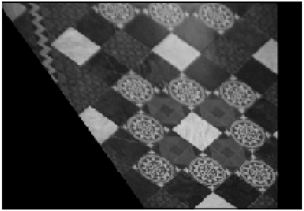
$H = H_{e\leftarrow s}H_{s\leftarrow a}H_{a\leftarrow p}$, where $A = sRK + \vec{t}\vec{v}^T$ and

$$H_{a\leftarrow p} = \begin{pmatrix} I & \vec{0} \\ \vec{v}^T & 1 \end{pmatrix}, \qquad H_{s\leftarrow a} = \begin{pmatrix} K & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}, \qquad H_{e\leftarrow s} = \begin{pmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix}.$$

- But, in general, **we don't know** H**: In general, we only have the image of a plane in the 3D world:**

From projective to affine: Compute $\mathbf{H}_{\mathbf{a}\leftarrow\mathbf{p}}$





Affine rectification

From projective to affine: Compute $\mathbf{H_{a\leftarrow p}}$

- Π plane of the 3D scene and Π_{im}, plane of the image. Π is a plane in the Euclidean space and the line at infinity is described by $\ell_\infty = (0,0,1)^T$

- Suppose that we are able to compute the line at infinity $\ell = (l_1, l_2, l_3)^T$ on the image Π_{im}. Assume that $l_3 \neq 0$.

- **The family of projective transformations of $\mathbb{P}^2$ that map ℓ to $\ell_\infty = (0,0,1)^T$** can be written as

$$\mathbf{H_{a\leftarrow p}} = H_a \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{pmatrix},$$

where H_a is <u>any</u> affine transformation (thus, the image, through H_a, of the line ℓ_∞ is ℓ_∞).

- Indeed, $H_{a\leftarrow p}^{-T}(l_1, l_2, l_3)^T = (0,0,1)^T$.

Affine rectification via the vanishing line

- A **vanishing point** of a world line is the image of its point at infinity.

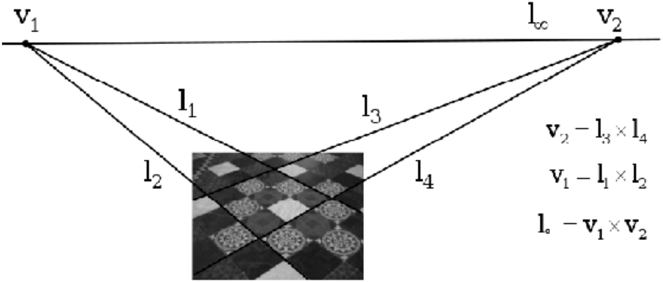
- Two parallel lines in the image of a plane Π (they are the image of parallel lines in the scene) intersect at its vanishing point.

- The **vanishing line** ℓ of Π is the image of its line at infinity.

Algorithm to compute the vanishing line ℓ and an homography that maps ℓ to ℓ_∞ :

1. Take two sets of two parallel lines in the image of a plane.
2. Each one provides a vanishing point, which can be computed from the cross product.
3. From these two points (which are on the vanishing line), compute the vanishing line ℓ.
4. Compute $H_{a \leftarrow p}$.

Affine rectification via the vanishing line



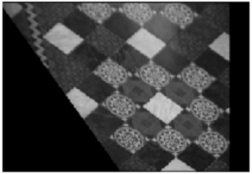


Image source: [Hartley Zisserman 2004]

Affine rectification

Finally we can affinely rectify the image u which contains the line ℓ by defining

$$u_{\text{affrect}}(\vec{x}_a) = u([H_{a \leftarrow p}^{-1} \mathbf{x}_a]),$$

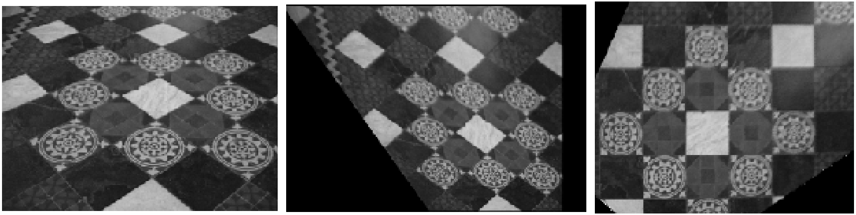
where $\mathbf{x}_a = (\vec{x}_a, 1)^T$ and $[\cdot]$ denotes$[(p_1, p_2, p_3)] = (p_1/p_3, p_2/p_3)$.

(Notice that $H_{a \leftarrow p}^{-1} \mathbf{x}_a$ are coordinates in the projective reference where u is defined.)

Metric rectification

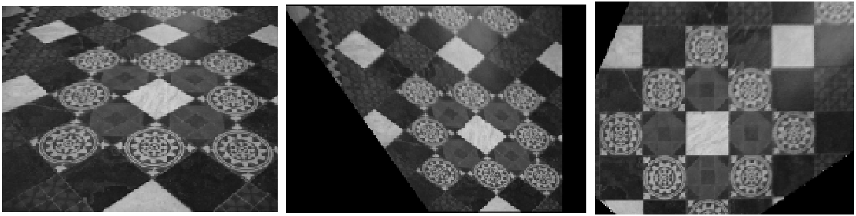
Metric rectification

Goal: to compute $\mathbf{H}_{s \leftarrow a}$ to **recover the metric properties.**



Metric rectification

Goal: to compute $\mathbf{H}_{s\leftarrow a}$ to recover the metric properties.



Remark. This is a stratified metric rectification method: we start from the (previous) affine-rectified image.

Image source: [Hartley Zisserman 2004]

Metric rectification

Main ingredient: Angle between two lines

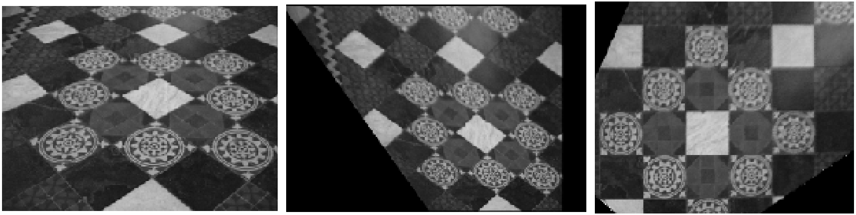
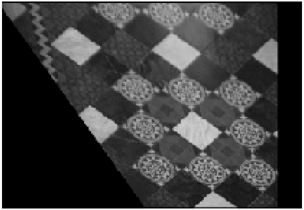


Image source: [Hartley Zisserman 2004]

Metric rectification

Suppose that the camera is observing a plane Π in the 3D scene. We have two planes: the plane of the scene Π and the plane of the image Π_{im}. They are related by a homography.

Assume that **the image has been affinely rectified**, so that the line at infinity in the image is $\ell_\infty = (0, 0, 1)^T$.



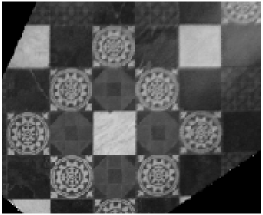


Image source: [Hartley Zisserman 2004]

Metric rectification

Goal: to compute $H_{s \leftarrow a}$ to recover the metric properties.

Angles between lines on the projective plane

Let $\ell = (\ell_1, \ell_2, \ell_3)^T$ and $m = (m_1, m_2, m_3)^T$ be two lines.
In an **Euclidean frame**, their angle is computed from the dot product of their normals:

Writing $\ell = (\mathbf{n}(\ell), 1)^T$, where $\mathbf{n}(\ell) = (\frac{\ell_1}{\ell_3}, \frac{\ell_2}{\ell_3})$ is the **normal vector to** ℓ, the **angle θ between ℓ and m** satisfies

$$\cos\theta = \frac{\langle \mathbf{n}(\ell), \mathbf{n}(m) \rangle}{\sqrt{\langle \mathbf{n}(\ell), \mathbf{n}(\ell) \rangle}\sqrt{\langle \mathbf{n}(m), \mathbf{n}(m) \rangle}} = \frac{\langle \Omega_\infty \ell, m \rangle}{\sqrt{\langle \Omega_\infty \ell, \ell \rangle}\sqrt{\langle \Omega_\infty m, m \rangle}}.$$

Metric rectification

Goal: to compute $H_{s \leftarrow a}$ to recover the metric properties.

Angles between lines on the projective plane

Let $\ell = (\ell_1, \ell_2, \ell_3)^T$ and $m = (m_1, m_2, m_3)^T$ be two lines.
In an **Euclidean frame**, their angle is computed from the dot product of their normals:

Writing $\ell = (\mathbf{n}(\ell), 1)^T$, where $\mathbf{n}(\ell) = (\frac{\ell_1}{\ell_3}, \frac{\ell_2}{\ell_3})$ is the **normal vector to** ℓ, the **angle θ between ℓ and m** satisfies

$$\cos\theta = \frac{\langle \mathbf{n}(\ell), \mathbf{n}(m) \rangle}{\sqrt{\langle \mathbf{n}(\ell), \mathbf{n}(\ell) \rangle}\sqrt{\langle \mathbf{n}(m), \mathbf{n}(m) \rangle}} = \frac{\langle \Omega_\infty \ell, m \rangle}{\sqrt{\langle \Omega_\infty \ell, \ell \rangle}\sqrt{\langle \Omega_\infty m, m \rangle}}.$$

Metric rectification

Goal: to compute $H_{s \leftarrow a}$ to recover the metric properties.

Angles between lines on the projective plane

Let $\ell = (\ell_1, \ell_2, \ell_3)^T$ and $m = (m_1, m_2, m_3)^T$ be two lines.
In an **Euclidean frame**, their angle is computed from the dot product of their normals:

Writing $\ell = (\mathbf{n}(\ell), 1)^T$, where $\mathbf{n}(\ell) = (\frac{\ell_1}{\ell_3}, \frac{\ell_2}{\ell_3})$ is the **normal vector to** ℓ, the **angle θ between ℓ and m** satisfies

$$\cos\theta = \frac{\langle \mathbf{n}(\ell), \mathbf{n}(m) \rangle}{\sqrt{\langle \mathbf{n}(\ell), \mathbf{n}(\ell) \rangle}\sqrt{\langle \mathbf{n}(m), \mathbf{n}(m) \rangle}} = \frac{\langle \Omega_\infty \ell, m \rangle}{\sqrt{\langle \Omega_\infty \ell, \ell \rangle}\sqrt{\langle \Omega_\infty m, m \rangle}}.$$

This expression cannot be applied after an affine or projective transformation of the plane.

Metric rectification

Goal: to compute $H_{s \leftarrow a}$ to recover the metric properties.

Angles between lines on the projective plane

Let $\ell = (\ell_1, \ell_2, \ell_3)^T$ and $m = (m_1, m_2, m_3)^T$ be two lines.
In an **Euclidean frame**, their angle is computed from the dot product of their normals:

Writing $\ell = (\mathbf{n}(\ell), 1)^T$, where $\mathbf{n}(\ell) = (\frac{\ell_1}{\ell_3}, \frac{\ell_2}{\ell_3})$ is the **normal vector to** ℓ, the **angle θ between ℓ and m** satisfies

$$\cos\theta = \frac{\langle \mathbf{n}(\ell), \mathbf{n}(m)\rangle}{\sqrt{\langle \mathbf{n}(\ell), \mathbf{n}(\ell)\rangle}\sqrt{\langle \mathbf{n}(m), \mathbf{n}(m)\rangle}} = \frac{\langle \Omega_\infty \ell, m\rangle}{\sqrt{\langle \Omega_\infty \ell, \ell\rangle}\sqrt{\langle \Omega_\infty m, m\rangle}}.$$

This expression cannot be applied after an affine or projective transformation of the plane. But, we can write it as an expression that is **invariant under projective transformations**

Metric rectification

Goal: to compute $H_{s\leftarrow a}$ to recover the metric properties.

Angles between lines on the projective plane

Let $\ell = (\ell_1, \ell_2, \ell_3)^T$ and $m = (m_1, m_2, m_3)^T$ be two lines.
In an **Euclidean frame**, their angle is computed from the dot product of their normals:

Writing $\ell = (\mathbf{n}(\ell), 1)^T$, where $\mathbf{n}(\ell) = (\frac{\ell_1}{\ell_3}, \frac{\ell_2}{\ell_3})$ is the **normal vector to** ℓ, the **angle θ between ℓ and m** satisfies

$$\cos\theta = \frac{\langle \mathbf{n}(\ell), \mathbf{n}(m)\rangle}{\sqrt{\langle \mathbf{n}(\ell), \mathbf{n}(\ell)\rangle}\sqrt{\langle \mathbf{n}(m), \mathbf{n}(m)\rangle}} = \frac{\langle \Omega_\infty \ell, m\rangle}{\sqrt{\langle \Omega_\infty \ell, \ell\rangle}\sqrt{\langle \Omega_\infty m, m\rangle}}.$$

where $\Omega_\infty = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$.

- ℓ and m are **orthogonal** if $\langle \Omega_\infty \ell, m\rangle = 0$.

Angles on the projective plane

Key property: The previous expression is invariant under a point projective transformation $\mathbf{x}' = H\mathbf{x}$.

Given a general change of projective coordinates $\mathbf{x}' = H\mathbf{x}$, where H is a homography.

If ℓ', m' are the image of the lines ℓ, m in this new homogeneous coordinates, the angle θ can be computed as

$$\cos\theta = \frac{\langle \Omega_\infty \ell, m\rangle}{\sqrt{\langle \Omega_\infty \ell, \ell\rangle}\sqrt{\langle \Omega_\infty m, m\rangle}} = \frac{\langle M\ell', m'\rangle}{\sqrt{\langle M\ell', \ell'\rangle}\sqrt{\langle Mm', m'\rangle}}.$$

where $M = H\Omega_\infty H^T$.

Application: Consider the image of a wall of a building. Assume that the image is not fronto-parallel. Suppose that there are two lines in the wall and you can observe them in the image. How do you compute using only the image the angle that these two lines make in the real world?

- ℓ and m are orthogonal if $\langle M\ell', m'\rangle = 0$ $\quad (= \langle \Omega_\infty \ell, m\rangle = 0)$.

Angles on the projective plane

Key property: The previous expression is invariant under a point projective transformation $\mathbf{x}' = H\mathbf{x}$.

Given a general change of projective coordinates $\mathbf{x}' = H\mathbf{x}$, where H is a homography.

If ℓ', m' are the image of the lines ℓ, m in this new homogeneous coordinates, the angle θ can be computed as

$$\cos\theta = \frac{\langle \Omega_\infty \ell, m\rangle}{\sqrt{\langle \Omega_\infty \ell, \ell\rangle}\sqrt{\langle \Omega_\infty m, m\rangle}} = \frac{\langle M\ell', m'\rangle}{\sqrt{\langle M\ell', \ell'\rangle}\sqrt{\langle Mm', m'\rangle}}.$$

where $M = H\Omega_\infty H^T$.

Application: Consider the image of a wall of a building. Assume that the image is not fronto-parallel. Suppose that there are two lines in the wall and you can observe them in the image. How do you compute using only the image the angle that these two lines make in the real world?

- ℓ and m are orthogonal if $\langle M\ell', m'\rangle = 0$ $\quad (= \langle \Omega_\infty \ell, m\rangle = 0)$.

Metric rectification

Assume that the image has been affinely rectified. The **goal** now is to compute $\mathbf{H_{s \leftarrow a}}$ to **recover the metric properties.**

Fact 1: Let H be a projective transformation of $\mathbb{P}^2$. Then H is a similarity if and only if $H\Omega_\infty H^T = \Omega_\infty$.

Fact 2: In our affinely rectified image, there is still an affine undeterminacy to represent the image, which is represented by a matrix
$H_{\mathrm{a}} = \begin{pmatrix} A & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}$. (which fixes $\ell_\infty = (0,0,1)^T$)

Fact 3: For any $H_{\mathrm{a}} = \begin{pmatrix} A & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}$, $M = H_{\mathrm{a}}\Omega_\infty H_{\mathrm{a}}^T = \begin{pmatrix} AA^T & \vec{0} \\ \vec{0}^T & 0 \end{pmatrix}$.
(in other words, $= \begin{pmatrix} S & \vec{0} \\ \vec{0}^T & 0 \end{pmatrix}$, where $S := AA^T$ is a 2×2 symmetric matrix).

- From constraints on lines, one can compute $H_{s \leftarrow a}$ using the following algorithm:

Algorithm

In our affinely rectified image, there is still an affine undeterminacy to represent the image, which is represented by a matrix $H_a = \begin{pmatrix} A & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}$. From Fact 3, $M = H_a \Omega_\infty H_a^T = \begin{pmatrix} S & \vec{0} \\ \vec{0}^T & 0 \end{pmatrix}$, with $S = AA^T$.

1. Let $\mathbf{l} = (l_1, l_2, l_3)$ and $\mathbf{m} = (m_1, m_2, m_3)$ be the image of two lines that are orthogonal in the world. Then $\mathbf{l}^T M \mathbf{m} = 0$. This equation writes
$$(l_1 m_1, l_1 m_2 + l_2 m_1, l_2 m_2)\vec{s} = 0,$$
where $\vec{s}^T = (s_1, s_2, s_3)^T$ is the vector with the entries of the 2×2 symmetric matrix $S = AA^T = \begin{pmatrix} s_1 & s_2 \\ s_2 & s_3 \end{pmatrix}$.
2. **The image of two such orthogonal pairs provide two equations and they permit to compute** (s_1, s_2, s_3) as its null vector (is an homogenous system of two equations with three unknowns).
3. **Knowing** S**, use the Cholesky decomposition** of S to compute **an upper triangular matrix** K such that $S = KK^T$.
4. **The matrix** K **is a possible matrix** A **that can be used to metrically rectify the image**.

Metric rectification via orthogonal lines

5. Define $H_{a \leftarrow s} = \begin{pmatrix} K & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}$ and $H = H_{\mathrm{a}\leftarrow\mathrm{s}}^{-1} H_{\mathrm{a}}$, where

$$H_{\mathrm{a}\leftarrow\mathrm{s}}^{-1} = H_{\mathrm{s}\leftarrow\mathrm{a}} = \begin{pmatrix} K^{-1} & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}$$

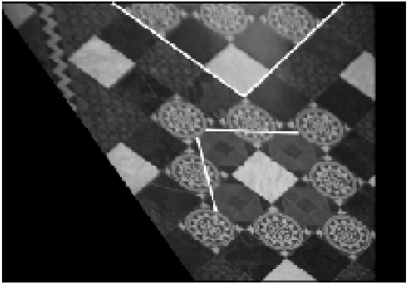
   Then, $H \Omega_\infty H^T = \Omega_\infty$. and, from Fact 1, we know that H is a similarity. Let's denote it by H_{s}. That is, $H_{\mathrm{a}} = H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}}$.

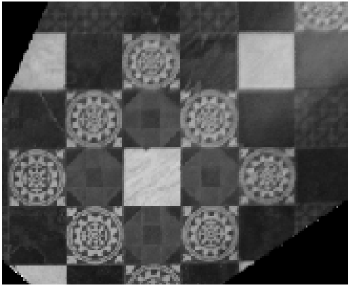
6. The rectified image can be defined by

$$u_{\mathrm{metrect}}(\vec{x}_s) = u_{\mathrm{affrect}}([H_{\mathrm{a}\leftarrow\mathrm{s}} \mathbf{x}_s]),$$

   where $\mathbf{x}_s = (\vec{x}_s, 1)$ and $[\cdot]$ denotes $[(x_1, x_2, x_3)] = (x_1/x_3, x_2/x_3)$ when $x_3 \neq 0$.

Metric rectification





Metric rectification via orthogonal lines.

Image source: [Hartley Zisserman 2004]

Additional: Metric rectification of an image with a single step

- We start with the original perspective image of the plane (not affinely rectified): Consider the image of a wall of a building by an (unknown) homography H.

- Assume that $\mathbf{l}_1$ and $\mathbf{m}_1$ are two lines in the wall, and $\mathbf{l}'_1$ and $\mathbf{m}'_1$ are the image of the lines $\mathbf{l}_1$ and $\mathbf{m}_1$ on the image Π_{im}.

- If $\mathbf{l}_1$ and $\mathbf{m}_1$ are orthogonal, $\langle \Omega_\infty \mathbf{l}_1, \mathbf{m}_1 \rangle = \langle M\mathbf{l}'_1, \mathbf{m}'_1 \rangle = 0$ where $M = H\Omega_\infty H^T$.

- This gives one linear equation for the entries of M. Since M is a symmetric matrix, it has 6 unknowns but we count 5 using homogeneous coordinates. Thus to construct 5 linear equations we need **5 line pairs which are images of orthogonal line pairs** in the world: $\mathbf{l}_i, \mathbf{m}_i$, $i = 1, \ldots, 5$.

- Notice that, if $H = H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}}$, then

$$M = H\Omega_\infty H^T = H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}} \Omega_\infty (H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}})^T = \begin{pmatrix} KK^T & KK^T\vec{v} \\ \vec{v}^T KK^T & \vec{v}^T KK^T \vec{v} \end{pmatrix}$$

That is, if we know M, we can compute K and $\vec{v}$ and transform the projective distortions into metric ones.

Additional: Metric rectification of an image with a single step

- We start with the original perspective image of the plane (not affinely rectified): Consider the image of a wall of a building by an (unknown) homography H.

- Assume that $\mathbf{l}_1$ and $\mathbf{m}_1$ are two lines in the wall, and $\mathbf{l}'_1$ and $\mathbf{m}'_1$ are the image of the lines $\mathbf{l}_1$ and $\mathbf{m}_1$ on the image Π_{im}.

- If $\mathbf{l}_1$ and $\mathbf{m}_1$ are orthogonal, $\langle \Omega_\infty \mathbf{l}_1, \mathbf{m}_1 \rangle = \langle M\mathbf{l}'_1, \mathbf{m}'_1 \rangle = 0$ where $M = H\Omega_\infty H^T$.

- This gives one linear equation for the entries of M. Since M is a symmetric matrix, it has 6 unknowns but we count 5 using homogeneous coordinates. Thus to construct 5 linear equations we need **5 line pairs which are images of orthogonal line pairs** in the world: $\mathbf{l}_i, \mathbf{m}_i$, $i = 1, \ldots, 5$.

- Notice that, if $H = H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}}$, then

$$M = H\Omega_\infty H^T = H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}} \Omega_\infty (H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}})^T = \begin{pmatrix} KK^T & KK^T\vec{v} \\ \vec{v}^T KK^T & \vec{v}^T KK^T \vec{v} \end{pmatrix}$$

That is, if we know M, we can compute K and $\vec{v}$ and transform the projective distortions into metric ones.

Additional: Metric rectification of an image with a single step

- We start with the original perspective image of the plane (not affinely rectified): Consider the image of a wall of a building by an (unknown) homography H.

- Assume that $\mathbf{l}_1$ and $\mathbf{m}_1$ are two lines in the wall, and $\mathbf{l}'_1$ and $\mathbf{m}'_1$ are the image of the lines $\mathbf{l}_1$ and $\mathbf{m}_1$ on the image Π_{im}.

- If $\mathbf{l}_1$ and $\mathbf{m}_1$ are orthogonal, $\langle \Omega_\infty \mathbf{l}_1, \mathbf{m}_1 \rangle = \langle M\mathbf{l}'_1, \mathbf{m}'_1 \rangle = 0$ where $M = H\Omega_\infty H^T$.

- This gives one linear equation for the entries of M. Since M is a symmetric matrix, it has 6 unknowns but we count 5 using homogeneous coordinates. Thus to construct 5 linear equations we need **5 line pairs which are images of orthogonal line pairs** in the world: $\mathbf{l}_i, \mathbf{m}_i$, $i = 1, \ldots, 5$.

- Notice that, if $H = H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}}$, then

$$M = H\Omega_\infty H^T = H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}} \Omega_\infty (H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}})^T = \begin{pmatrix} KK^T & KK^T\vec{v} \\ \vec{v}^T KK^T & \vec{v}^T KK^T \vec{v} \end{pmatrix}$$

That is, if we know M, we can compute K and $\vec{v}$ and transform the projective distortions into metric ones.

Additional: Metric rectification of an image with a single step

- We start with the original perspective image of the plane (not affinely rectified): Consider the image of a wall of a building by an (unknown) homography H.

- Assume that $\mathbf{l}_1$ and $\mathbf{m}_1$ are two lines in the wall, and $\mathbf{l}_1'$ and $\mathbf{m}_1'$ are the image of the lines $\mathbf{l}_1$ and $\mathbf{m}_1$ on the image Π_{im}.

- If $\mathbf{l}_1$ and $\mathbf{m}_1$ are orthogonal, $\langle \Omega_\infty \mathbf{l}_1, \mathbf{m}_1 \rangle = \langle M\mathbf{l}_1', \mathbf{m}_1' \rangle = 0$ where $M = H\Omega_\infty H^T$.

- This gives one linear equation for the entries of M. Since M is a symmetric matrix, it has 6 unknowns but we count 5 using homogeneous coordinates. Thus to construct 5 linear equations we need **5 line pairs which are images of orthogonal line pairs** in the world: $\mathbf{l}_i, \mathbf{m}_i$, $i = 1, \ldots, 5$.

Additional: Metric rectification of an image with a single step

• We start with the original perspective image of the plane (not affinely rectified): Consider the image of a wall of a building by an (unknown) homography H.

• Assume that $\mathbf{l}_1$ and $\mathbf{m}_1$ are two lines in the wall, and $\mathbf{l}'_1$ and $\mathbf{m}'_1$ are the image of the lines $\mathbf{l}_1$ and $\mathbf{m}_1$ on the image Π_{im}.

• If $\mathbf{l}_1$ and $\mathbf{m}_1$ are orthogonal, $\langle \Omega_\infty \mathbf{l}_1, \mathbf{m}_1 \rangle = \langle M\mathbf{l}'_1, \mathbf{m}'_1 \rangle = 0$ where $M = H\Omega_\infty H^T$.

• This gives one linear equation for the entries of M. Since M is a symmetric matrix, it has 6 unknowns but we count 5 using homogeneous coordinates. Thus to construct 5 linear equations we need **5 line pairs which are images of orthogonal line pairs** in the world: $\mathbf{l}_i, \mathbf{m}_i$, $i = 1, \ldots, 5$.

• Notice that, if $H = H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}}$, then

$$M = H\Omega_\infty H^T = H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}} \Omega_\infty (H_{\mathrm{p}\leftarrow\mathrm{a}} H_{\mathrm{a}\leftarrow\mathrm{s}} H_{\mathrm{s}\leftarrow\mathrm{e}})^T = \begin{pmatrix} KK^T & KK^T\vec{v} \\ \vec{v}^T KK^T & \vec{v}^T KK^T \vec{v} \end{pmatrix}$$

That is, if we know M, we can compute K and $\vec{v}$ and transform the projective distortions into metric ones.

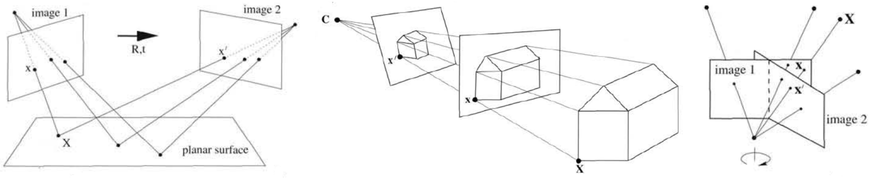
2. Homography computation

2. Homography computation

Goal: **Compute the homography that relates to images**

- DLT algorithm (algebraic method).
- Robust normalized DLT algorithm (algebraic method).
- Gold-Standard algorithm (geometric method).

Homografies (or planar transformations)



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]

Let's compute a 2D projective transformation

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

In a practical situation, the points $\mathbf{x}_i$ and $\mathbf{x}'_i$ are points in two images, each image being considered as a projective plane $\mathbb{P}^2$.

Let's compute a 2D projective transformation

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

In a practical situation, the points $\mathbf{x}_i$ and $\mathbf{x}'_i$ are points in two images, each image being considered as a projective plane $\mathbb{P}^{\mathbf{2}}$.

Let's compute a 2D projective transformation

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.

Let's compute a 2D projective transformation

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.

Let's compute a 2D projective transformation

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

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

That is,

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

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

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 $\underline{\mathbf{A}_i \text{ is } 2 \times 9}$ and $\underline{\mathbf{h} \text{ is } 9 \times 1}$).

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**) $d_{\text{alg}}(\mathbf{x}_i', H\mathbf{x}_i) = \|\text{proj}_2(\mathbf{x}_i' \times H\mathbf{x}_i)\|$ be 0 (where $\text{proj}_2(\mathbf{x}_i' \times H\mathbf{x}_i)$ 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

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

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 $||\mathbf{Ah}||$ subject to $||\mathbf{h}|| = 1$.

Direct Linear Transformation (DLT) algorithm

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 $\mathbf{A}$. The singular vector of modulus 1 associated to the smallest singular value of $\mathbf{A}$ is the solution $\mathbf{h}$. More precisely, 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 $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)$.

Direct Linear Transformation (DLT) algorithm

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 $\mathbf{A}$. The singular vector of modulus 1 associated to the smallest singular value of $\mathbf{A}$ is the solution $\mathbf{h}$. More precisely, 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 $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)$.

Direct Linear Transformation (DLT) algorithm

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 $\mathbf{A}$. The singular vector of modulus 1 associated to the smallest singular value of $\mathbf{A}$ is the solution $\mathbf{h}$. More precisely, 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 $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)$.

Direct Linear Transformation (DLT) algorithm

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 $\mathbf{A}$. The singular vector of modulus 1 associated to the smallest singular value of $\mathbf{A}$ is the solution $\mathbf{h}$. More precisely, 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 $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)$.

Direct Linear Transformation (DLT) algorithm

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 $\mathbf{A}$. The singular vector of modulus 1 associated to the smallest singular value of $\mathbf{A}$ is the solution $\mathbf{h}$. More precisely, 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 $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)$.

Direct Linear Transformation (DLT) algorithm

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 $\mathbf{A}$. The singular vector of modulus 1 associated to the smallest singular value of $\mathbf{A}$ is the solution $\mathbf{h}$. More precisely, 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 $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)$.

Direct Linear Transformation (DLT) algorithm

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 $\mathbf{A}$. The singular vector of modulus 1 associated to the smallest singular value of $\mathbf{A}$ is the solution $\mathbf{h}$. More precisely, 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 $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)$.

Warning: Data normalization is an essential step in 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.

Normalized DLT algorithm

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 $H = \mathcal{T}'^{-1}\widetilde{H}\mathcal{T}$.

Normalized DLT algorithm

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 $H = \mathcal{T}'^{-1}\widetilde{H}\mathcal{T}$.

Normalized DLT algorithm

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 $H = \mathcal{T}'^{-1}\widetilde{H}\mathcal{T}$.

The (normalized) DLT algorithm minimizes an error cost:

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

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

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

We call $d_{\mathrm{alg}}(\mathbf{a}, \mathbf{b}) = \|\mathrm{proj}_2(\mathbf{a} \times \mathbf{b})\|_2 = (a_2 b_3 - a_3 b_2)^2 + (a_1 b_3 - a_3 b_1)^2$ the algebraic distance of $\mathbf{a}$ and $\mathbf{b}$. We call $\|\mathrm{proj}_2(\mathbf{x}'_i \times H\mathbf{x}_i)\|_2$ the algebraic error.

Another posibility: Minimize a geometric distance and error

Geometric algorithm: it minimizes the reprojection error

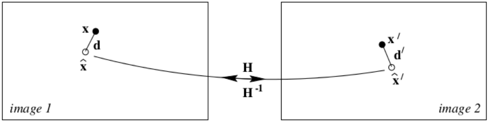
$$\min_{H,\hat{\mathbf{x}}_i,\hat{\mathbf{x}}'_i} \sum_{i=1}^{n} \|[\mathbf{x}_i] - [\hat{\mathbf{x}}_i]\|^2 + \|[\mathbf{x}'_i] - [\hat{\mathbf{x}}'_i]\|^2 \quad \text{such that } \hat{\mathbf{x}}'_i = H\hat{\mathbf{x}}_i,\ \forall i$$

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 $\hat{\mathbf{x}}'_i = H\hat{\mathbf{x}}_i$?



Gold Standard algorithm (in order to minimize it)

- The Gold Standard algorithm allows a robust computation of the 2D homography between two images.

- It uses RANSAC which is a robust estimation algorithm.

Algorithm for a robust computation of the 2D homography between two images

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

   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).

The last two steps can be iterated until the number of correspondences is stable.

Given a correspondence $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$, the **distance $\mathbf{d}_\perp$** is

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

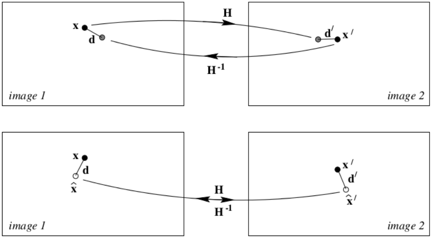


Fig. 4.2. *A comparison between symmetric transfer error (upper) and reprojection error (lower) when estimating a homography. The points* $\mathbf{x}$ *and* $\mathbf{x}'$ *are the measured (noisy) points. Under the estimated homography the points* $\mathbf{x}'$ *and* $\mathtt{H}\mathbf{x}$ *do* not *correspond 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* $d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$*.*

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

Image source: [Hartley Zisserman 2004]

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.

Scenario raising a different problem

,

Scenario raising a different problem

,

Scenario raising a different problem

- Lets consider the problem of camera calibration in sport event scenarios.

,

Scenario raising a different problem

- Lets consider the problem of camera calibration in sport event scenarios.

- One can approach the calibration of this kind of planar view scenarios by, first, **computing the homography from the projected playing field in the image and a reference playing field given by its actual dimensions.**

,

Scenario raising a different problem

- Lets consider the problem of camera calibration in sport event scenarios.

- One can approach the calibration of this kind of planar view scenarios by, first, **computing the homography from the projected playing field in the image and a reference playing field given by its actual dimensions.**

- But, In sport scenarios like football or basketball, we often deal with **central views** where only the central circle and some additional primitives like the central line and the central point or a touch line are visible.,

Scenario raising a different problem

- Lets consider the problem of camera calibration in sport event scenarios.

- One can approach the calibration of this kind of planar view scenarios by, first, **computing the homography from the projected playing field in the image and a reference playing field given by its actual dimensions.**

- But, In sport scenarios like football or basketball, we often deal with **central views** where only the central circle and some additional primitives like the central line and the central point or a touch line are visible.,

- Can we identify (at least) 4 pair correspondences?

How to estimate the plane homography of a central view of the playing field?

HOMOGRAPHY ESTIMATION USING ONE ELLIPSE CORRESPONDENCE AND MINIMAL ADDITIONAL INFORMATION

Luis Alvarez

CTIM: Center of Image Technologies
Departamento de Informática y Sistemas
Univ. de Las Palmas de Gran Canaria (Spain)

Vicent Caselles

Departament de Tecnologies de
la Informació i les Comunicacions
Universitat Pompeu Fabra (Spain)

ABSTRACT

In sport scenarios like football or basketball, we often deal with central views where only the central circle and some additional primitives like the central line and the central point or a touch line are visible. In this paper we first characterize, from a mathematical point of view, the set of homographies that project a given ellipse into the unit circle, next, using some extra minimal additional information like the knowledge of the position in the image of the central line and central point or a touch line we show a method to fully determine the plane homography. We present some experiments in sport

Proposed solution

Homography estimation problem for central view scenarios based on **characterizing the general form of the homographies transforming a given ellipse into a circle**.

Proposed solution

Homography estimation problem for central view scenarios based on **characterizing the general form of the homographies transforming a given ellipse into a circle**.

+ additional information such as the projection of the central line, and the central point or a touch line:

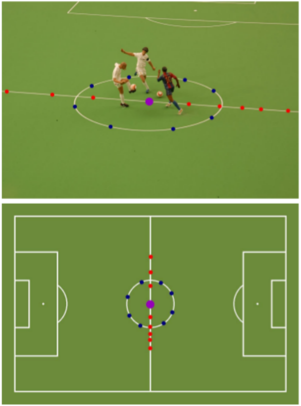


Fig. 2. Top: original Image with selected primitive points. Down: reference playing field where the original primitive

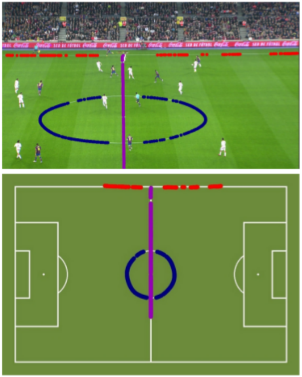


Fig. 3. Top: original Image with selected primitive points. Down: reference playing field where the original primitive points are projected using the estimated homography.

References

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

[Szeliski 2010] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.