3D Vision an introduction

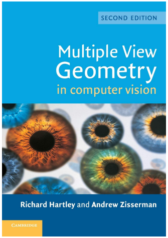
ENSTA Paris

Gianni Franchi

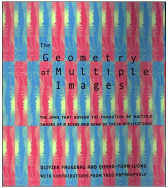
03/12/2021

Plan

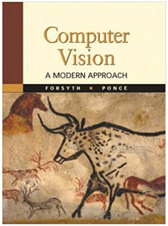
Some references



(a)



(b)

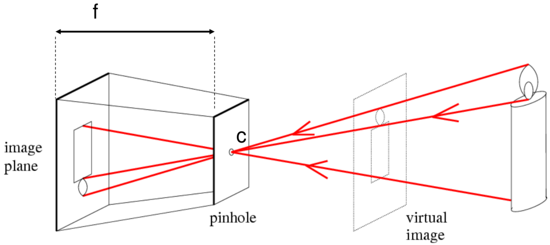


(c)

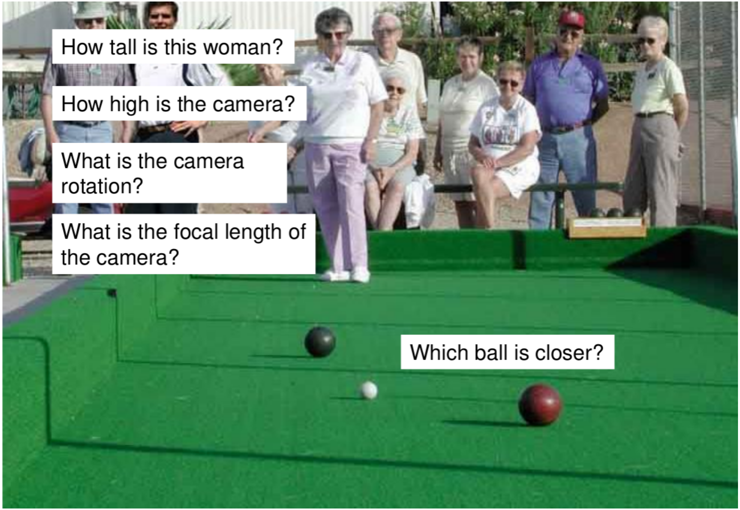
(a) :Hartley, Richard, and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 200
(b) : Luong, Quang-Tuan, and O. D. Faugeras. "The geometry of multiple images." MIT Press, Boston 2.3 (2001): 4-5.
(c) : Forsyth, David A., and Jean Ponce. Computer vision: a modern approach. Prentice Hall Professional Technical Reference, 2002.
Most of these slides comes from Marc Pollefeys course :
https://www.cs.unc.edu/ marc/mvg/slides.html

Pinhole camera

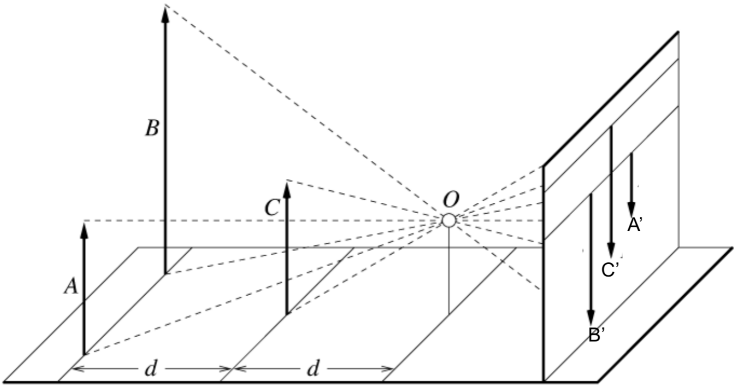
- A box with an infinitesimal small hole
- Camera center is the intersection of the rays



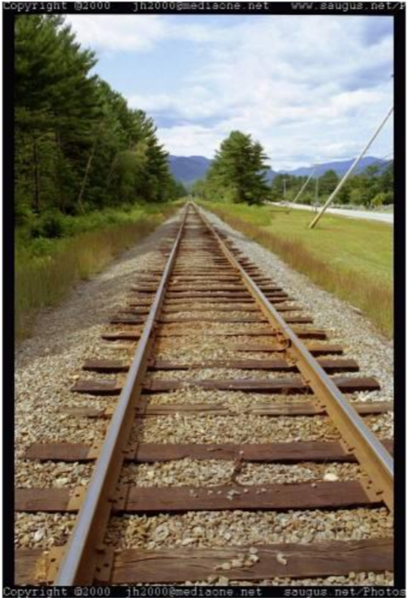
Camera and World Geometry



Length and area are not preserved



Vanishing points and lines



Camera and World Geometry

- **Lines** on the real world are **preserved** in the image
- **Angles** on the real world are **not preserved** in the image
- The **size of the object** on the real world are **not preserved** in the image since they depends on their position

Information loss caused by the camera projection

- A camera projects from the 3D world to a 2D image
- this causes a loss of information
- 3D information can only be recovered if additional information is available (multiple images, details about the camera, known size of objects, ...)

Motivations

- The Euclidean geometry has difficulties to describe an infinity point at a give direction
- The Euclidean geometry is subotimal to describe central projection
- The mathematic of Euclidean geometry can get complicated

Projective geometry is an alternative

Homogeneous Coordinates (HC)

- The Homogeneous Coordinates are a system of coordinates used in Projective geometry
- The formulas involving Homogeneous Coordinates are often simpler
- The point at infinity can be represented in the Homogeneous Coordinates with finite coordinates

Notations

Point $\mathcal{X}$

- in Homogeneous coordinates $\mathbf{x}$
- in Euclidean coordinates x

Line $\mathcal{L}$

- in Homogeneous coordinates $\mathbf{l}$

Plane $\mathcal{A}$

- in Homogeneous coordinates $\mathbf{A}$

Homogeneous coordinates 2D points

Definition:
The representation $\mathbf{x}$ of a geometric object is **homogeneous** if and only if $\mathbf{x}$ and $\lambda\mathbf{x}$ represent the same object for $\lambda \neq 0$.
Example:

$$\underbrace{\mathbf{x} = \lambda\mathbf{x}}_{\text{Homogeneous}} \quad \underbrace{x \neq \lambda x}_{\text{Euclidean}}$$

Homogeneous coordinates 2D points

- Homogeneous coordinates use a n+1 dimensional vector to represent the same (n-dim. point)
- Example $\mathbb{R}^2 \rightarrow \mathbb{P}^2$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

Homogeneous coordinates 2D points

Definition:

Homogeneous coordinates of a point $\mathcal{X}$ in the plane $\mathbb{R}^2$ is a 3 dimension vector different of $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

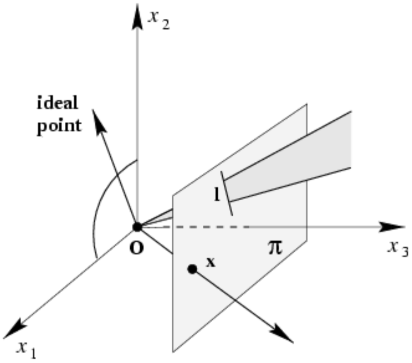
Definition:

to represent a point at infinity we add zero on the last coordinate $\begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}$ except $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$.

Definition:

The origin of the Euclidean Coordinate in the Homogeneous coordinates are given by $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.

Homogeneous coordinates



Homogeneous coordinates 3D points

Analogous for point in 3D Euclidian space $\mathbb{R}^3$

$$\underbrace{\begin{bmatrix} u \\ v \\ w \\ t \end{bmatrix} = \begin{bmatrix} u/t \\ v/t \\ w/t \\ 1 \end{bmatrix}}_{\text{Homogeneous}} \rightarrow \underbrace{\begin{bmatrix} u/t \\ v/t \\ w/t \end{bmatrix}}_{\text{Euclidean}}$$

Homogeneous coordinates 2D Lines

In the Euclidian coordinates we represent line by :

- Hesse normal form (angle ϕ, distance d):

$$x \cos(\phi) + y \sin(\phi) - d = 0$$

- Standard (Implicit) form :

$$xa + yb + c = 0$$

- ...

Homogeneous coordinates 2D Lines

In the Homogeneous coordinates we represent line by :

Point

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

line

$$\mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Equation of the line is:

$$\mathbf{l}^t\mathbf{x} = \mathbf{x}^t\mathbf{l} = 0$$

Homogeneous coordinates 2D Lines

Definition:
Homogeneous Coordinates of a line $\mathcal{L}$ in a plane is a 3 dimensional vector

$$\mathcal{L} : \mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} \text{ with } ||\mathbf{l}||^2 = l_1^2 + l_2^2 + l_3^2 \neq 0$$

It correponds to the Euclidian equation :

$$l_1 x + l_2 y + l_3 = 0$$

How can we easily check if a point is in a line?

Homogeneous coordinates 2D Lines

To check if $\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ intesect with the line $\mathcal{L} : \mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$ we just eveluate if the inner product equal zero. $\mathbf{x}^t \mathbf{l} = 0$.

How can we find the intesection of two lines?

Homogeneous coordinates 2D Lines

Let us define two lines $\mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$ and $\mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}$

A point $\mathbf{x}$ intersects if

$$\begin{bmatrix} \mathbf{l}^t\mathbf{x} \\ \mathbf{m}^t\mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This can be rewritten :

$$\begin{bmatrix} l_1 x + l_2 y \\ m_1 x + m_2 y \end{bmatrix} = \begin{bmatrix} -l_3 \\ -m_3 \end{bmatrix}$$

Homogeneous coordinates 2D Lines

A system of linear equations can solved via Cramer's rule. If we have:

$$A\mathbf{x} = \mathbf{b}$$

then the coordinate i of x is

$$x_i = \frac{\det(A_i)}{\det(A)}$$

with A_i being matrix A where the ith column replaced by b.

Homogeneous coordinates 2D Lines

The Solution of

$$\begin{bmatrix} l_1 x + l_2 y \\ m_1 x + m_2 y \end{bmatrix} = \begin{bmatrix} -l_3 \\ -m_3 \end{bmatrix}$$

is

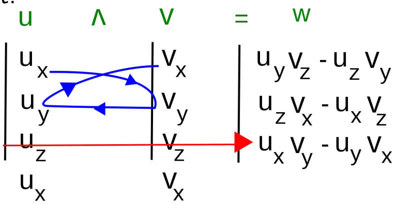
$$x = \frac{l_2 m_3 - l_3 m_2}{l_1 m_2 - l_2 m_1} \; y = \frac{l_3 m_1 - l_1 m_3}{l_1 m_2 - l_2 m_1}$$

Homogeneous coordinates 2D Lines

the cross product $\mathbf{u} \wedge \mathbf{v}$ of two vectors $\mathbf{u} = \left[u_x, u_y, u_z\right]^t$ and $\mathbf{v} = \left[v_x, v_y, v_z\right]^t$ is a vector :

$$\mathbf{u} \wedge \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \sin(\theta) \mathbf{n}$$

with $\mathbf{n}$ a unit vector perpendicular to the plane that contains $\mathbf{v}$ and $\mathbf{u}$, and θ the angle between $\mathbf{u}$ and $\mathbf{v}$.
Trick to calculate it:



Homogeneous coordinates 2D Lines

The Solution of

$$\begin{bmatrix} l_1 x + l_2 y \\ m_1 x + m_2 y \end{bmatrix} = \begin{bmatrix} -l_3 \\ -m_3 \end{bmatrix}$$

is

$$x = \frac{l_2 m_3 - l_3 m_2}{l_1 m_2 - l_2 m_1} \; y = \frac{l_3 m_1 - l_1 m_3}{l_1 m_2 - l_2 m_1}$$

Then using the cross product we have :

$$\mathbf{x} = \mathbf{l} \wedge \mathbf{m}$$

A simple way for computing the intersection of 2 lines using HC

Homogeneous coordinates 2D Lines

Let us define two points $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$

How can we find the lines that go in these two points?

Homogeneous coordinates 2D Lines

Let us definetwo points $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$

A line $\mathbf{l}$ intersects the 2 points if

$$\begin{bmatrix} \mathbf{l}^t \mathbf{x} \\ \mathbf{l}^t \mathbf{y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This can be rewritten :

$$\begin{bmatrix} l_1 x_1 + l_2 x_2 \\ l_1 y_1 + l_2 y_2 \end{bmatrix} = \begin{bmatrix} -l_3 x 3 \\ -l_3 y 3 \end{bmatrix}$$

Homogeneous coordinates 2D Lines

Based on the following system :

$$\begin{bmatrix} l_1 x_1 + l_2 x_2 \\ l_1 y_1 + l_2 y_2 \end{bmatrix} = \begin{bmatrix} -l_3 x3 \\ -l_3 y3 \end{bmatrix}$$

we can apply the Cramer's rule:

$$l_1 = \frac{l_3(x_2 y_3 - y_2 x_3)}{x_1 y_2 - x_2 y_1} \; l_2 = \frac{l_3(x_3 y_1 - y_3 x_1)}{x_1 y_2 - x_2 y_1}$$

Then using the cross product we have :

$$\mathbf{l} = \mathbf{x} \wedge \mathbf{y}$$

Summary

- A point lies on a line if :

$$\mathbf{x}^t \mathbf{l} = 0$$

- A point of the intersection of two lines is :

$$\mathbf{x} = \mathbf{l} \wedge \mathbf{m}$$

- A lines through two give points is :

$$\mathbf{l} = \mathbf{x} \wedge \mathbf{y}$$

Points at Infinity

- It is possible to **explicity** model infinitively distant points with finite coordinates :

$$\mathbf{x}_\infty = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}$$

- We can keep the **direction** of one infinitively distant point

Intersection at Infinity

- All lines $\mathbf{l}$ pass through a point at infinity $\mathbf{x}_\infty$ if : $\mathbf{x}_\infty^t \mathbf{l} = 0$
- This hold for any lines $\mathbf{l} = \left[l_1, l_2, *\right]^t$

What conclusion comes from these slides?

Intersection at Infinity

- All lines $\mathbf{l}$ pass through a point at infinity $\mathbf{x}_\infty$ if : $\mathbf{x}_\infty^t \mathbf{l} = 0$
- this hold for any lines $\mathbf{l} = \left[l_1, l_2, * \right]^t$

What conclusion comes from these slides?

$\rightarrow$ **All parallel lines meet at one point at infinity!**

$$\mathbf{x} = \begin{bmatrix} a \\ b \\ c_1 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c_2 \end{bmatrix} = \begin{bmatrix} bc_2 - bc_1 \\ ac_1 - ac_2 \\ 0 \end{bmatrix}$$

Points and lines at Infinity

- Infinitively distant point

$$\mathbf{x}_\infty = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}$$

- Infinitively distant line is :

$$\mathbf{l}_\infty = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- all the points at infinity lie at $\mathbf{l}_\infty$
- $\mathbf{l}_\infty$ can be interpreted as the horizon

Projective Transformations

Definition Projective Transformations:
A projectivity is an invertible mapping h from P^2 to itself such that three points x_1, x_2, x_3 lie on the same line if and only if $h(x_1), h(x_2), h(x_3)$ do.
Theorem of Projective Geometry :
A mapping $h : P^2 \rightarrow P^2$ is a projectivity if and only if there exist a non-singular 3×3 matrix H such that for any point in P^2 reprented by a vector x it is true that $h(x) = Hx$.

Projective Transformations : Translation

- General projective mapping:

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

- Translation: 2 Parameters

$$H = \lambda \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0}^t & 1 \end{bmatrix}$$

with $\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$ and $\mathbf{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

2D Transformations : Isometries (Class I)

(iso=same, metric=measure)

$$\mathbf{x'} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \epsilon \cos(\theta) & -\sin(\theta) & t_x \\ \epsilon \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ with } \epsilon = \pm 1$$

orientation preserving: $\epsilon = 1$
orientation reversing: $\epsilon = -1$
3DOF (1 rotation, 2 translation)
special cases: pure rotation, pure translation
Invariants: length, angle, area

2D Transformations : Similarities (Class II)

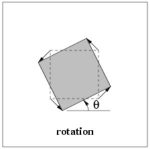
(isometry + scale)

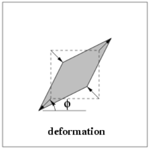
$$\mathbf{x'} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos(\theta) & -s\sin(\theta) & t_x \\ s\sin(\theta) & s\cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

also know as equi - form (shape preserving)
4DOF (1 scale, 1 rotation, 2 translation)
Invariants: ratios of length, angle, ratios of areas, parallel lines

2D Transformations : Affine transformations (Class III)

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



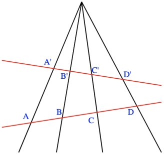


6DOF (2 scale, 2 rotation, 2 translation)

Invariants: parallel lines, ratios of parallel lengths, ratios of areas

2D Transformations : Projective transformations (Class IV)

$$\mathbf{x'} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ v_1 & v_2 & \nu \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



8DOF (2 scale, 2 rotation, 2 translation, 2 line at infinity)

Action non-homogeneous over the plane

Invariants: cross-ratio of four points on a line (ratio of ratio)

Action of affinities and projectivities on line at infinity

Let us consider an infinite line $\mathbf{l}_\infty = \begin{bmatrix} l_1 & l_2 & 0 \end{bmatrix}^t$
Let us apply an Affine transformation on $\mathbf{l}_\infty$

$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & \nu \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ 0 \end{bmatrix} = \begin{bmatrix} l_1 a_{11} + l_2 a_{12} \\ l_1 a_{21} + l_2 a_{22} \\ 0 \end{bmatrix}$$

Hence, a line at infinity stays at infinity, but points move along line.
Let us apply an Projective transformation on $\mathbf{l}_\infty$

$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ v_1 & v_2 & \nu \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ 0 \end{bmatrix} = \begin{bmatrix} l_1 a_{11} + l_2 a_{12} \\ l_1 a_{21} + l_2 a_{22} \\ l_1 v_1 + l_2 v_2 \end{bmatrix}$$

Hence, a line at infinity becomes finite, allows to observe vanishing points, horizon,

Decomposition of projective transformations

Let us consider that we can decompose the transformation into H_S a similarity, H_A and affine projection, and projective transformations H_P, then the final transformation can calculated:

$$H = H_S \times H_A \times H_P$$

Parameters estimation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_x \\ v_1 & v_2 & \nu \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

How many points are needed?

Parameters estimation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_x \\ v_1 & v_2 & \nu \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

How many points are needed?

At least as many independent equations as degrees of freedom required

2 independent equations / point

We need at least 4 points.

8 degrees of freedom

Parameters estimation

- **Minimal solution :** 4 points yield an exact solution for H
- **More points :**
    - No exact solution, because measurements are inexact ("noise")
    - Search for "best" according to some cost function
    - Algebraic or geometric/statistical cost

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 variables A,B and C.

$$\mathbf{x'} = \begin{bmatrix} \mathbf{A}^t \\ \mathbf{B}^t \\ \mathbf{C}^t \end{bmatrix} \mathbf{x}$$

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \mathbf{A}^t\mathbf{x} \\ \mathbf{B}^t\mathbf{x} \\ \mathbf{C}^t\mathbf{x} \end{bmatrix}$$

Estimation by Direct Linear Transformation (DLT)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \mathbf{A}^t\mathbf{x} \\ \mathbf{B}^t\mathbf{x} \\ \mathbf{C}^t\mathbf{x} \end{bmatrix}$$

$$x' = \frac{u'}{w'} = \frac{\mathbf{A}^t\mathbf{x}}{\mathbf{C}^t\mathbf{x}}$$

$$y' = \frac{v'}{w'} = \frac{\mathbf{B}^t\mathbf{x}}{\mathbf{C}^t\mathbf{x}}$$

Estimation by Direct Linear Transformation (DLT)

We can rewrite the equations:

$$\begin{cases} -\mathbf{A}^t\mathbf{x} & & +x'\mathbf{x}\mathbf{C}^t = 0 \\ & -\mathbf{B}^t\mathbf{x} & +y'\mathbf{x}\mathbf{C}^t = 0 \end{cases}$$

we want to estimate A, B and C

Estimation by Direct Linear Transformation (DLT)

Let us write $\mathbf{P} = \begin{bmatrix}\mathbf{A} & \mathbf{B} & \mathbf{C}\end{bmatrix}^t$. P is a vector of size 9×1.
We can rewrite the previous system with $\mathbf{P}$.

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

with

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

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

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

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

Estimation by Direct Linear Transformation (DLT)

Now let us consider that we have multiple paires 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 paires of points :

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

Collecting everything together we have:

$$\underbrace{\mathbf{M}}_{2N\times 9}\underbrace{\mathbf{P}}_{9\times 1} = \underbrace{\mathbf{0}}_{9\times 1}$$

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 needed to avoid 0, e.g. $\|P\|_2^2 = 1$

Estimation of P

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

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

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} \mathbf{P}^t\mathbf{M}^t\mathbf{MP}$$

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

Estimation of P

The eigenvector belonging to the smallest eigenvalue of M solves the system of linear equations.

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

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, $\mathbf{P}$ is equal to v_9 with s_9 the smallest eigen value.

Estimation of P

The estimate of **P** i given by

$$\hat{\mathbf{P}} = \begin{bmatrix} \hat{\mathbf{A}} \\ \hat{\mathbf{B}} \\ \hat{\mathbf{C}} \end{bmatrix} = v_9$$

This leads to the estimated projection matrix.
No solution if all point x are on a line.

DLT algorithm

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

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

Estimation of P

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

$$10^2 \quad 10^2 \quad 1 \quad 10^2 \quad 10^2 \quad 1 \quad 10^4 \quad 10^4 \quad 10^2$$

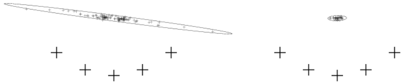
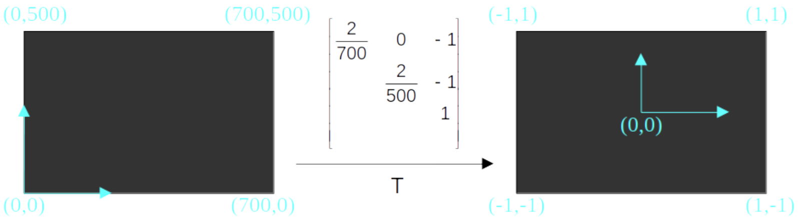


Illustration of distributed errors whose repartition respectively depends, and does depends on the dimensions on the left and the right image. How do we transform all the coordinates so that the coordinate are between $[-1, 1]$?

Estimation of P



Normalized DLT algorithm

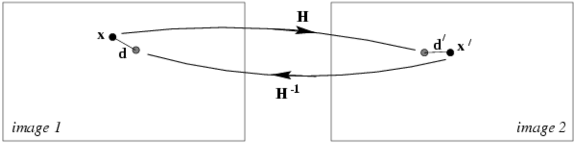
Objective:
Given $N \geq 4$ 2D to 2D point correspondences $(\mathbf{x}_i, \mathbf{x}'_i)$, determine the 2D homography matrix H such that $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$.
Algorithm:

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

Homography estimation different cost function

- Algebraic Distance : This is a metric can minimized by DLT.
- Geometric Distance : The goal is to minimize the Reprojection error:

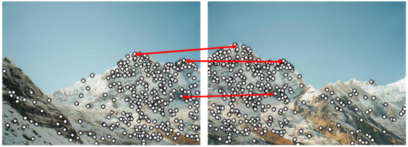
$$D(\mathbf{x}, \mathbf{H}^{-1}\mathbf{x}')^2 + D(\mathbf{x}', \mathbf{H}\mathbf{x})^2$$



Estimation of homography by Ransac

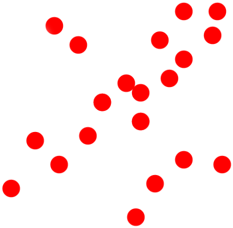
What if set of matches contains outliers?
We need to perform a Robust estimation.

Estimation of homography by Ransac



- Extract features
- Compute putative matches e.g. "closest descriptor"
- Loop:
  - Hypothesize transformation T from some matches
  - Verify transformation (search for other matches consistent with T)
- apply T

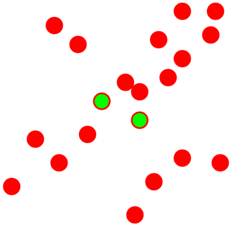
Estimation of homography by Ransac



Algorithm:

- Sample(randomly) the number of points required to fit the model
- Solve for model parameters using sample
- Score by the fraction of inliers within a present threshold of the model
- Repeat 1-3 until the best model is found with high confidence

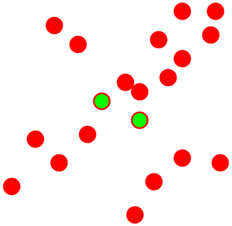
Estimation of homography by Ransac



Algorithm:

- Sample(randomly) the number of points required to fit the model
- Solve for model parameters using sample
- Score by the fraction of inliers within a present threshold of the model
- Repeat 1-3 until the best model is found with high confidence

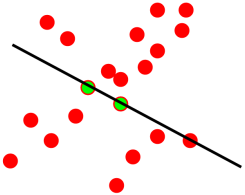
Estimation of homography by Ransac



Algorithm:

- Sample(randomly) the number of points required to fit the model
- Solve for model parameters using sample
- Score by the fraction of inliers within a present threshold of the model
- Repeat 1-3 until the best model is found with high confidence

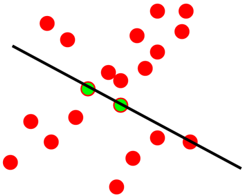
Estimation of homography by Ransac



Algorithm:

- Sample(randomly) the number of points required to fit the model
- Solve for model parameters using sample
- Score by the fraction of inliers within a present threshold of the model
- Repeat 1-3 until the best model is found with high confidence

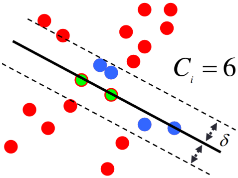
Estimation of homography by Ransac



Algorithm:

- Sample(randomly) the number of points required to fit the model
- Solve for model parameters using sample
- Score by the fraction of inliers within a present threshold of the model
- Repeat 1-3 until the best model is found with high confidence

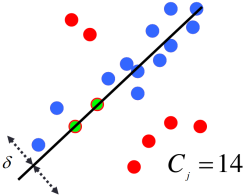
Estimation of homography by Ransac



Algorithm:

- Sample(randomly) the number of points required to fit the model
- Solve for model parameters using sample
- Score by the fraction of inliers within a present threshold of the model
- Repeat 1-3 until the best model is found with high confidence

Estimation of homography by Ransac



Algorithm:

- Sample(randomly) the number of points required to fit the model
- Solve for model parameters using sample
- Score by the fraction of inliers within a present threshold of the model
- **Repeat 1-3 until the best model is found with high confidence**

More support implies better fit

Ransac [1] algorithm

Objective:
Robust fit of model to data set S which contains outliers
Algorithm:

- Randomly select a sample of s data points from S and instantiate the model from this subset.
- 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 samples and defines the inliers of S.
- If the subset of S_i is greater than some threshold T, reestimate the model using all the points in S_i and terminate
- If the size of S_i is less than T, select a new subset and repeat the above.
- After N trials the largest consensus set S_i is selected, and the model is reestimated using all the points in the subset S_i

[1] Martin A. Fischler and Robert C. Bolles, " Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography"

Ransac Parameters

- The size of the subset (minimum size to define the model)
- Typically minimum number needed to fit the model
- Error tolerence threshold δ
- Minimum consensus Threshold w
- Number of iteration k

It is important to know the proportion of inlier w. The Error tolerence threshold δ is related to the gaussian error in inliers.

number of iteration k

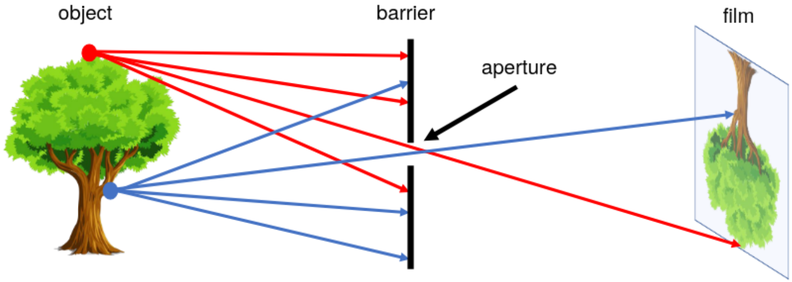
Let us write $\mathrm{P}(\textbf{inlier}) = \mathrm{w}$ the probability of choosing an inlier.
Then for Subset of size n : $\mathrm{P}(\textbf{a Subset with no outlier}) = \mathrm{w}^{\mathrm{n}}$.
So, a subset of size n has a probability
$\mathrm{P}(\textbf{a Subset with outlier(s)}\ ) = 1 - \mathrm{w}^{\mathrm{n}}$.
So, the probability of choosing a subset with outliers in all k repetitions is : $\mathrm{P}(\textbf{k Subset with outlier(s)}\ ) = (1 - \mathrm{w}^{\mathrm{n}})^{\mathrm{k}}$.
So the probability of successful run is $\mathrm{P}(\textbf{sucess}) = 1 - (1 - \mathrm{w}^{\mathrm{n}})^{\mathrm{k}}$.

$$k = \frac{\log(1 - P(\text{sucess}))}{\log(1 - w^n)}$$

number of iteration k with $P(\text{sucess}) = 0.99$

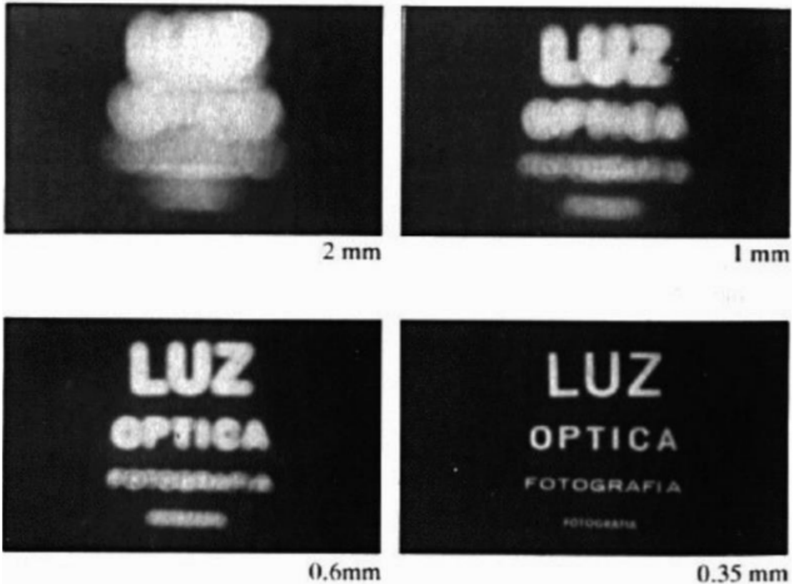
Sampl e size n	Proportion of outliers						
	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Pinhole cameras



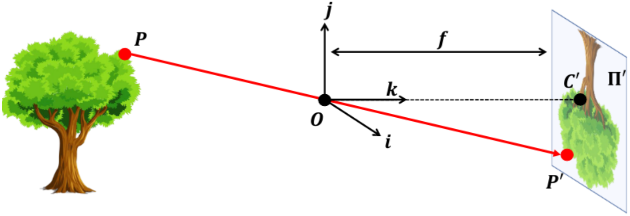
Each point on the 3D object emits multiple rays of light outwards. Without a barrier in place, every point on the film will be influenced by light rays emitted from every point on the 3D object.

Pinhole cameras



Pinhole cameras

The aperture is referred to as the pinhole O or center of the camera
The distance between the image plane and the pinhole O is the focal length f.

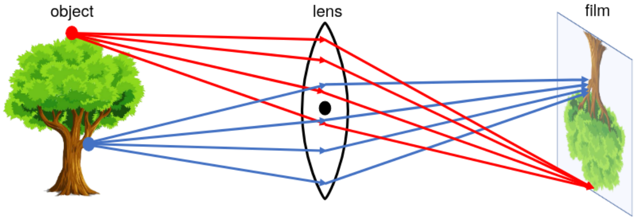


Let us write $P = \begin{bmatrix} x & y & z \end{bmatrix}^t$ a point in the 3D world.
Let $P' = \begin{bmatrix} x' & y' \end{bmatrix}^t$ be its projection on Π'. Using geometry we have:

$$P' = \begin{bmatrix} f\frac{x}{z} & f\frac{y}{z} \end{bmatrix}^t$$

Modern cameras

Thanks to the lens all rays of light that are emitted by some point P are refracted by the lens such that they converge to a single point P'



Let us write $P = \begin{bmatrix} x & y & z \end{bmatrix}^t$ a point in the 3D world.
Let $\mathbf{P}' = \begin{bmatrix} x' & y' \end{bmatrix}^t$ be its projection on Π'. Using geometrical optics we have:

$$\mathbf{P}' = \begin{bmatrix} z'\frac{x}{z} & z'\frac{y}{z} \end{bmatrix}^t$$

with z' the distance between the lens and the film.

Digital image space

This $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ mapping is referred to as a projective transformation.
Note that the image coordinates have their origin C' at the image center where the k axis intersects the image plane.
Digital images typically have their origin at the **lower-left corner** of the image. We need to do a translation with the vector $\begin{bmatrix} c_x & c_x \end{bmatrix}^t$. So the new coordinate are:

$$\mathbf{P}' = \begin{bmatrix} z'\frac{x}{z} + c_x \\ z'\frac{y}{z} + c_y \end{bmatrix}^t$$

Digital image space

The next effect we must account for is that the points in digital images are expressed in **pixels**, while points in image plane are represented in **physical measurements (e.g. centimeters)**.
So we need to change of the units in the two axes of the image plane.
So the new coordinate are:

$$\mathbf{P}' = \begin{bmatrix} z' k \frac{x}{z} + c_x \\ z' l \frac{y}{z} + c_y \end{bmatrix}^t = \begin{bmatrix} \alpha \frac{x}{z} + c_x \\ \beta \frac{y}{z} + c_y \end{bmatrix}^t$$

k and l may be different because the aspect ratio of the unit element is not guaranteed to be one

Digital image space

Using homogeneous coordinates, we can formulate :

$$\mathbf{P}' = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{P} = \begin{bmatrix} \alpha & 0 & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3,1} \end{bmatrix} \mathbf{P}$$

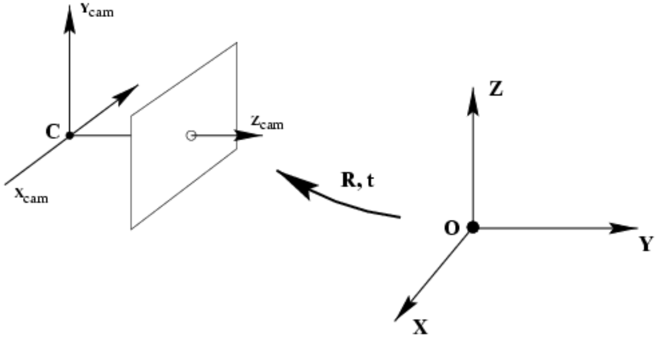
So

$$\mathbf{P}' = K \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3,1} \end{bmatrix} \mathbf{P}$$

The matrix K is often referred to as the camera matrix. To parameters are missing : **skewness** and **distortion**. But it goes beyond the scope of this course. **K** is linked to intrinsic parameters.

Digital image space

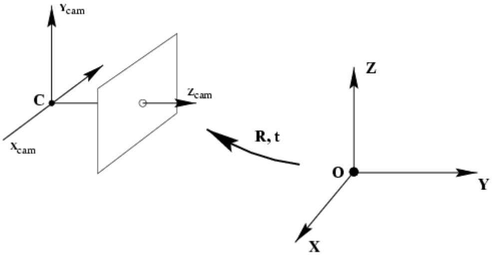
So far, we have described a mapping between a point **P** in the 3D camera reference system to a point**P**′ in the 2D image plane. **But what if the information about the 3D world is available in a different coordinate system?**



Digital image space

Then, we need to include an additional transformation that relates points from the world reference $\mathbf{P}_w$ system to the camera reference system $\mathbf{P}$. This transformation is captured by a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$.

$$\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \mathbf{P}_w$$



$\mathbf{R}$ and $\mathbf{T}$ are the extrinsic parameters.

Digital image space

$$\mathbf{P}' = K \begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix} \mathbf{P}_w = \mathbf{M}\mathbf{P}_w$$

The extrinsic paramters include the rotation and translation, which do not depend on the camera's build.
$\mathbf{M}$ is a 3×4 projection matrix with 11 degrees of freedom: 5 from the intrinsic camera matrix, 3 from extrinsic rotation, and 3 from extrinsic translation.