Master in Computer Vision Barcelona

Module: 3D Vision

Lecture 1: Introduction.
2D projective geometry.

Lecturer: Gloria Haro

Introduction

3D vision

What is 3D vision?

3D vision

What is 3D vision?

Subarea of **Computer Vision**, goals:

- Understanding/modeling geometric relations between:
  - images from different viewpoints
  - images and 3D world
- Extracting/reconstructing 3D information from:
  - images
  - depth sensors

3D vision

What is 3D vision?

Subarea of **Computer Vision**, goals:

- Understanding/modeling geometric relations between:
  - images from different viewpoints
  - images and 3D world
- Extracting/reconstructing 3D information from:
  - images
  - depth sensors

What is 3D vision not?

Generation of synthetic 3D content (**Computer Graphics**)

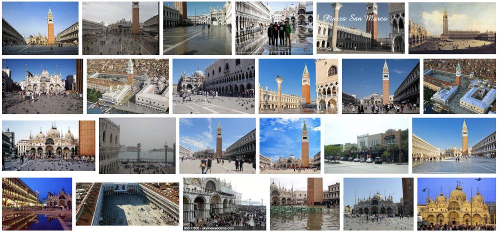
Multi-view systems and applications

Image source: [N. Campbell]

A world of images and cameras

Multi-view systems and applications

Large-scale 3D reconstruction



Building Rome in a Day project: https://grail.cs.washington.edu/rome/

N. Snavely, S. M. Seitz, and R. Szeliski. Photo Tourism: Exploring image collections in 3D. SIGGRAPH, 2006
S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, R. Szeliski. Building Rome in a Day. International Conference on Computer Vision, 2009

Dense reconstruction from unstructured image collections

J. L. Schnberger, E. Zheng, M. Pollefeys, J. Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. European Conference on Computer Vision, 2016

Multi-view systems and applications

Large-scale 3D reconstruction

Neural Radiance Fields for Unconstrained Photo Collections

(a) Photos (b) Renderings

Brandenburg Gate

Sacre Coeur

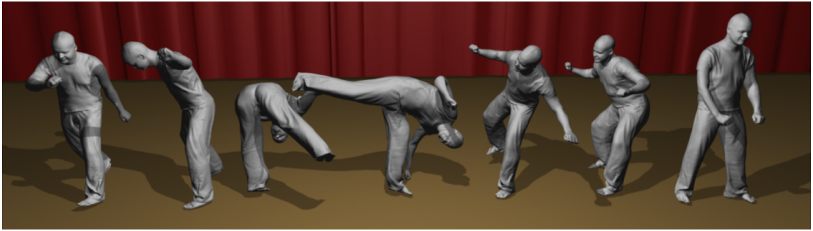
Trevi Fountain

NeRF in the Wild: https://nerf-w.github.io/

R. Martin-Brualla, N. Radwan, M.S.M. Sajjadi, J. T. Barron, A. Dosovitskiy, D. Duckworth. NeRF in the Wild. CVPR, 2021

Multi-view systems and applications

Motion capture



E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.P. Seidel, S. Thrun. Performance capture from sparse multi-view video, ACM Transactions on Graphics (TOG), 27(3), 2008

Facial expressions

More info in Image source

Multi-view systems and applications

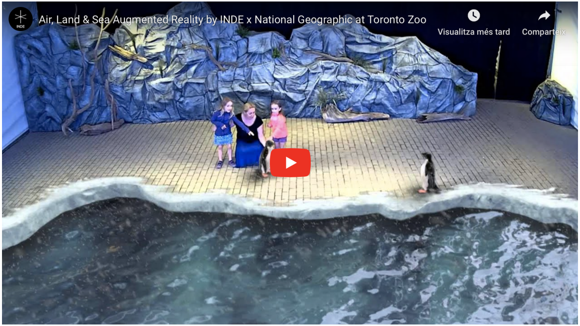
Bullet time effect

Multiple cameras

Single camera

Multi-view systems and applications

Augmented reality



Other examples here

Multi-view systems and applications

Match moving

Multi-view systems and applications

La fotografía, que muestra a familias junto a la playa de Barcelona el 26 de abril, fue cuestionada en redes sociales. EMILIO MORENATTI (AP)

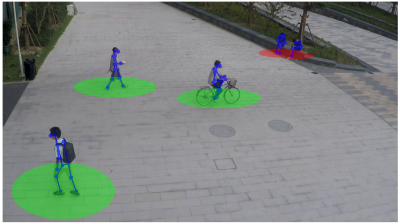
Multi-view systems and applications

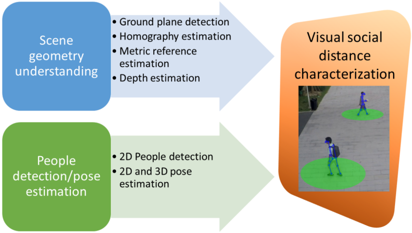
SERGIO RODRÍGUEZ

Multi-view systems and applications

The Visual Social Distancing Problem

Marco Cristani, *Member, IEEE,* Alessio Del Bue, *Member, IEEE,* Vittorio Murino, *Senior Member, IEEE,*
Francesco Setti, *Member, IEEE,* and Alessandro Vinciarelli, *Member, IEEE*





Calibration

1. Calibrated case: Multi-view stereo, Shape from X

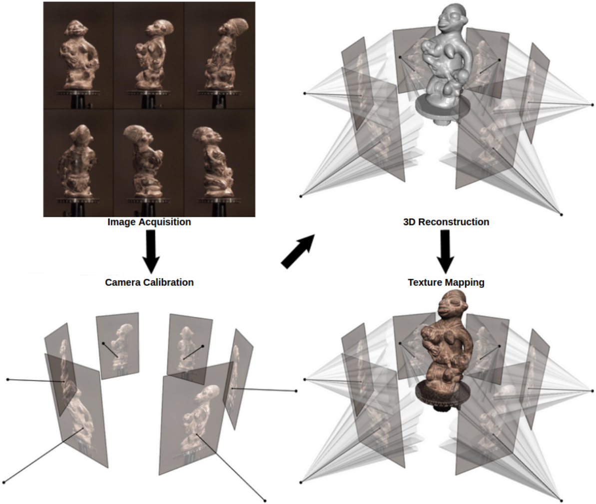


Image source: [C. Hernández]

Calibration

2. Non-calibrated case: Structure from motion (SfM), Simultaneous Localization and Mapping (SLAM).

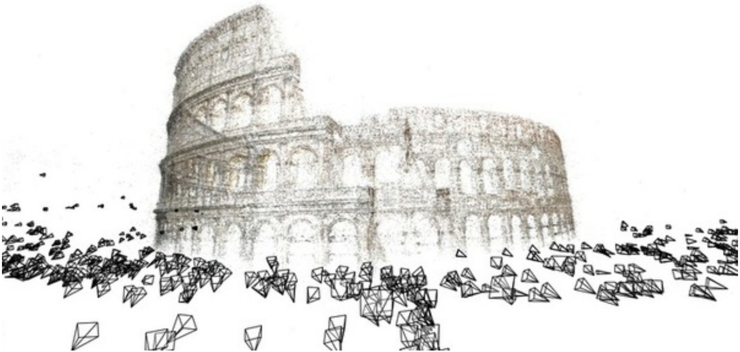
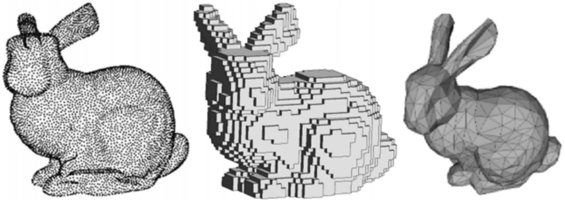


Image source: [Agarwal et al. 2010]

3D shape representation

Explicit representations



Point cloud Voxels Mesh

Image source: [L. Hoang et al. 2019]

3D shape representation

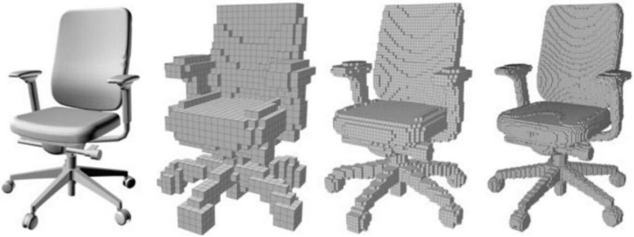
Explicit representations

Point cloud Colored mesh Mesh

Image source: [Furukawa and Ponce 2007]

3D shape representation

Explicit representations

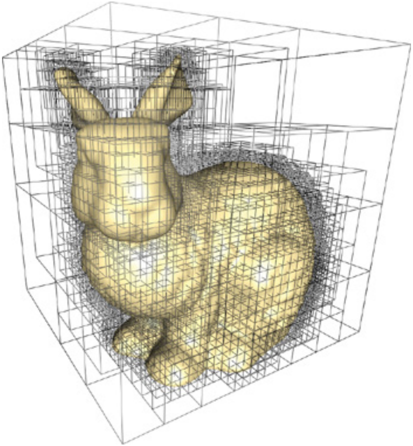


Voxel grid representation with an increasing voxel resolution

Image source: [Y. Li et al. 2016]

3D shape representation

Explicit representations

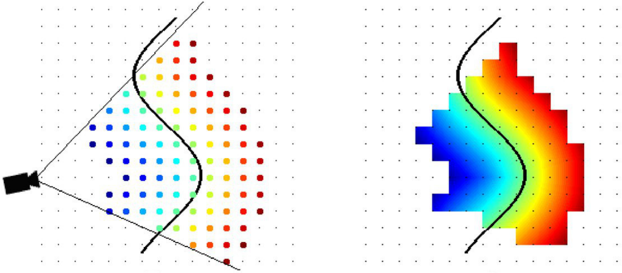


Octree

Image source: [S. Lefebvre, S. Hornus, F. Neyret]

3D shape representation

Implicit representations

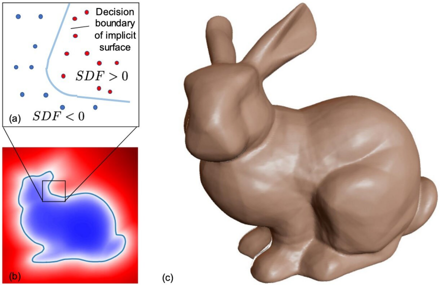


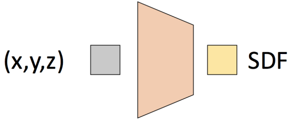
Signed distance function
(voxel-based, discrete)

Image source: [J.P. Morgan and R.L. Tutwiler, 2014]

3D shape representation

Implicit representations

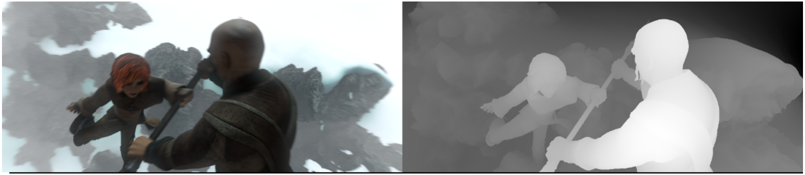




Signed Distance Function
(learned, continuous)

Images source: [J.J. Park et al., 2019]

3D shape representation



Depth map

Image source: [Butler et al. 2012]

Mathematical tools

- Linear Algebra
- Projective geometry
- Optimization
- Deep learning

Course contents and lecturers

Introduction. Projective geometry.	Gloria Haro
Image transformations and appliactions	Federico Sukno
Camera models, the geometry of one view	Federico Sukno
Camera calibration. Pose estimation	Gloria Haro
The geometry of two views	Javier Ruiz
Structure and depth computation. New view synthesis	Gloria Haro
Multi-view stereo. Structure from motion	Antonio Agudo
Autocalibration. Bundle adjustment	Antonio Agudo
3D sensors and point clouds	Josep Ramon Casas
Point cloud processing	Federico Sukno

Projective geometry

Projective geometry

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

Projective geometry

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

Geometric properties not preserved: lengths (distances), angles, distance ratio.

Projective geometry

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

Geometric properties not preserved: lengths (distances), angles, distance ratio.
What is preserved? straight lines (collinearity), cross ratio (ratio of ratio of lengths).

Projective geometry

School of Athens. Raphael, 1509-1511.

Image source: [Wikipedia]

Projective geometry

Why projective geometry?

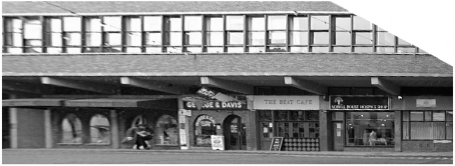
- The image capture of the camera is described with a linear transformation.
- The intersection of two lines and the line that passes through two points is a linear operation.
- Points at infinity have a natural representation.

We will study:

- **2D projective geometry:** It allows to remove the projective distortion of flat objects and build image mosaics (panoramas).
- **3D projective geometry:** It models the camera projection and allows the 3D reconstruction, the calibration and auto-calibration.

Projective geometry

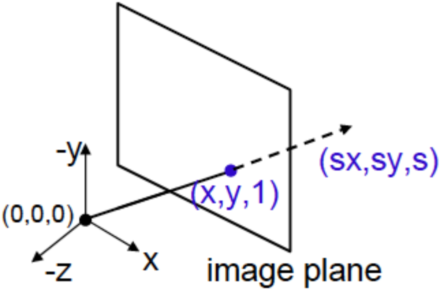
Removal of projective distortion



Panorama

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: $(x, y, 1) \equiv (sx, sy, s)$
$(s \in \mathbb{R},\ s \neq 0!)$

Image source: [S. Seitz]

The projective plane

Notation: $\mathbb{P}^2$, **projective space of dimension 2** (the projective plane)

1. Representation of points:

- Points in the 2D Euclidean space $\mathbb{R}^2$:

$$(x, y)^T = \begin{pmatrix} x \\ y \end{pmatrix} \quad \textbf{cartesian coordinates}$$

- Points in the 2D projective space $\mathbb{P}^2$:

$$\mathbf{x} = (x_1, x_2, x_3)^T = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \textbf{homogeneous coordinates}$$

The projective plane

Notation: $\mathbb{P}^2$, **projective space of dimension 2** (the projective plane)

1. Representation of points:

- Points in the 2D Euclidean space $\mathbb{R}^2$:

$$(x, y)^T = \begin{pmatrix} x \\ y \end{pmatrix} \text{ cartesian coordinates}$$

- Points in the 2D projective space $\mathbb{P}^2$:

$$\mathbf{x} = (x_1, x_2, x_3)^T = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ homogeneous coordinates}$$

Given $\mathbf{x}$ and $\mathbf{x}' \in \mathbb{P}^2$, we define the **equivalence relationship** as:

$$\mathbf{x} \equiv \mathbf{x}' \text{ (or } \mathbf{x} \sim \mathbf{x}' \text{), if } \exists\, \lambda \neq 0 \text{ such that } \mathbf{x} = \lambda \mathbf{x}'.$$

The projective plane

Notation: $\mathbb{P}^2$, **projective space of dimension 2** (the projective plane)

1. Representation of points:

- Points in the 2D Euclidean space $\mathbb{R}^2$:

$$(x, y)^T = \begin{pmatrix} x \\ y \end{pmatrix} \text{ cartesian coordinates}$$

- Points in the 2D projective space $\mathbb{P}^2$:

$$\mathbf{x} = (x_1, x_2, x_3)^T = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ homogeneous coordinates}$$

Given $\mathbf{x}$ and $\mathbf{x}' \in \mathbb{P}^2$, we define the **equivalence relationship** as:

$$\mathbf{x} \equiv \mathbf{x}' \text{ (or } \mathbf{x} \sim \mathbf{x}' \text{), if } \exists\, \lambda \neq 0 \text{ such that } \mathbf{x} = \lambda \mathbf{x}'.$$

We represent a point of the plane $\mathbb{R}^2$ by a vector $\mathbf{x}$ in $\mathbb{R}^3 \setminus \{(0,0,0)\}$.

The projective plane

How do we transform from cartesian to homogeneous coord. ($\mathbb{R}^2 \rightarrow \mathbb{P}^2$)?

$$(x, y)^T \rightarrow (x, y, 1)^T$$

The projective plane

How do we transform from cartesian to homogeneous coord. $(\mathbb{R}^2 \rightarrow \mathbb{P}^2)$?

$$(x, y)^T \rightarrow (x, y, 1)^T$$

How do we transform from homogeneous to cartesian coord. $(\mathbb{P}^2 \rightarrow \mathbb{R}^2)$?

$$(x_1, x_2, x_3)^T \rightarrow \underbrace{\left(\frac{x_1}{x_3}, \frac{x_2}{x_3}\right)^T}_{\text{we need } x_3 \neq 0!} = \underbrace{\left(\frac{sx}{s}, \frac{sy}{s}\right)^T}_{(x_1,x_2,x_3)^T=(sx,sy,s)^T} = (x, y)^T$$

The projective plane

How do we transform from cartesian to homogeneous coord. $(\mathbb{R}^2 \to \mathbb{P}^2)$?

$$(x, y)^T \to (x, y, 1)^T$$

How do we transform from homogeneous to cartesian coord. $(\mathbb{P}^2 \to \mathbb{R}^2)$?

$$(x_1, x_2, x_3)^T \to \underbrace{\left(\frac{x_1}{x_3}, \frac{x_2}{x_3}\right)^T}_{\text{we need } x_3 \neq 0!} = \underbrace{\left(\frac{sx}{s}, \frac{sy}{s}\right)^T}_{(x_1, x_2, x_3)^T = (sx, sy, s)^T} = (x, y)^T$$

We require $x_3 \neq 0$
$\Rightarrow$ points in $\mathbb{P}^2$ with $x_3 = 0$ **do not have an equivalent** in $\mathbb{R}^2$, these represent **points at infinity**.

The projective plane

2. Representation of lines:

Given $(x, y)^T$ a point in the plane $\mathbb{R}^2$, the equation of the line that passes through the point is:

$$ax + by + c = 0, \quad a, b, c \in \mathbb{R}.$$

The projective plane

2. Representation of lines:

Given $(x, y)^T$ a point in the plane $\mathbb{R}^2$, the equation of the line that passes through the point is:

$$ax + by + c = 0, \quad a, b, c \in \mathbb{R}.$$

If $\mathbf{x} = (x, y, 1)^T$ and $\ell = (a, b, c)^T$ we have that the line equation can be writen as:

$$\langle \ell, \mathbf{x} \rangle = 0, \text{ or } \ell^T \mathbf{x} = 0, \text{ or } \mathbf{x}^T \ell = 0.$$

(Note: it holds also for $\mathbf{x}' = s\mathbf{x}$)

$\ell = (a, b, c)^T \in \mathbb{R}^3 \setminus \{(0, 0, 0)\}$ represents a line in $\mathbb{P}^2$,

line in homogeneous coordinates.

The projective plane

2. Representation of lines:

Given $(x, y)^T$ a point in the plane $\mathbb{R}^2$, the equation of the line that passes through the point is:

$$ax + by + c = 0, \quad a, b, c \in \mathbb{R}.$$

If $\mathbf{x} = (x, y, 1)^T$ and $\ell = (a, b, c)^T$ we have that the line equation can be writen as:

$$\langle \ell, \mathbf{x} \rangle = 0, \text{ or } \ell^T \mathbf{x} = 0, \text{ or } \mathbf{x}^T \ell = 0.$$

(Note: it holds also for $\mathbf{x}' = s\mathbf{x}$)

$\ell = (a, b, c)^T \in \mathbb{R}^3 \setminus \{(0, 0, 0)\}$ represents a line in $\mathbb{P}^2$,
line in homogeneous coordinates.

Equivalent representations:

$$\ell \equiv \ell' \text{ if } \exists k \neq 0 \text{ such that } \ell = k\ell'.$$

The projective plane

Consider the line $\ell = (a, b, c)^T$ $(ax + by + c = 0)$,
the **point at infinity of the line** is $(-b, a, 0)^T$.

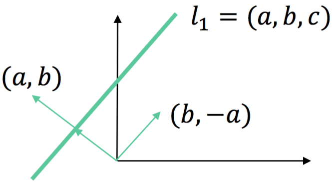
Image source: [M. Pollefeys]

The projective plane

Consider the line $\ell = (a, b, c)^T$ $(ax + by + c = 0)$,
the **point at infinity of the line** is $(-b, a, 0)^T$.

Points of the type $(x_1, x_2, 0)^T$ represent points at infinity: they are called **ideal points**.

- Ideal points correspond to *directions*



Ideal point $\begin{pmatrix} b \\ -a \\ 0 \end{pmatrix}$

Image source: [M. Pollefeys]

The projective plane

Line equation $\boxed{ax+by+c=0}$

if $b \neq 0$, also: $y = -\frac{a}{b}x - \frac{c}{b}$

slope $(y=mx+c)$

Parametrization of a line

$(x(t), y(t)) = (x_0, y_0) + t(v_1, v_2)$ direction vector, $t \in \mathbb{R}$

point on line

$$\left.\begin{aligned} x &= x_0 + v_1 t \\ y &= y_0 + v_2 t \end{aligned}\right] \xrightarrow[\text{if } v_1 \neq 0,\ v_2 \neq 0]{} \frac{x-x_0}{v_1} = \frac{y-y_0}{v_2}$$

$$\underbrace{-v_2}_{a} x + \underbrace{v_1}_{b} y + \underbrace{x_0 v_2 - y_0 v_1}_{} = 0$$

direction vector: $(v_1, v_2) = (b, -a)$

normal vectors: (a, b) and $(-a, -b)$

$\perp$

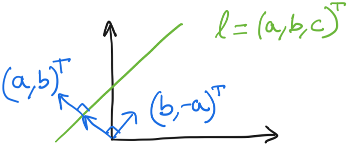
The projective plane

Parametrization: $(x(t), y(t))^T = (x_0, y_0)^T + t\,(v_1, v_2)^T \quad t \in \mathbb{R}$

$$= (x_0 + t v_1,\ y_0 + t v_2)^T$$

$$\begin{pmatrix} x_0 + t v_1 \\ y_0 + t v_2 \end{pmatrix} \in \mathbb{R}^2 \longleftrightarrow \begin{pmatrix} x_0 + t v_1 \\ y_0 + t v_2 \\ 1 \end{pmatrix} \in \mathbb{P}^2$$

$$\begin{pmatrix} x_0 + t v_1 \\ y_0 + t v_2 \\ 1 \end{pmatrix} \underset{\substack{\text{assume} \\ t \neq 0}}{\equiv} \begin{pmatrix} x_0/t + v_1 \\ y_0/t + v_2 \\ 1/t \end{pmatrix} \xrightarrow[\lim_{t\to\infty}]{} \underbrace{\begin{pmatrix} v_1 \\ v_2 \\ 0 \end{pmatrix} = \begin{pmatrix} b \\ -a \\ 0 \end{pmatrix} \equiv \begin{pmatrix} -b \\ a \\ 0 \end{pmatrix}}_{\text{point at infinity of line } (a,b,c)}$$



ideal point

The projective plane

Intersection of two lines

Consider the lines $\ell = (a, b, c)^T$ and $\ell' = (a', b', c')^T$, their **intersection point** $\mathbf{x}$ is:

$$\mathbf{x} = \ell \times \ell'.$$

Recall: the cross product

$$\ell \times \ell' = \begin{vmatrix} i & j & k \\ a & b & c \\ a' & b' & c' \end{vmatrix} = (bc' - b'c, a'c - ac', ab' - a'b).$$

The projective plane

The intersection of two parallel lines

The lines $\ell = (a, b, c)^T$ and $\ell' = (a, b, c')^T$ are parallel, they have the same slope $-a/b$.

Their intersection point is

$$\ell \times \ell' = (bc' - bc, ac - ac', 0)^T \Rightarrow \text{ a point at infinity.}$$

The projective plane

The intersection of two parallel lines

The lines $\ell = (a, b, c)^T$ and $\ell' = (a, b, c')^T$ are parallel, they have the same slope $-a/b$.

Their intersection point is

$$\ell \times \ell' = (bc' - bc, ac - ac', 0)^T \Rightarrow \text{ a point at infinity.}$$

If $c \neq c'$ then $(bc' - bc, ac - ac', 0)^T \equiv (-b, a, 0)^T$

The projective plane

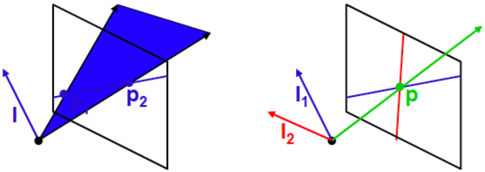
The line that joins two points

Consider the points $\mathbf{x} = (x_1, x_2, x_3)^T$ and $\mathbf{x}' = (x'_1, x'_2, x'_3)^T$, the **line that passes through the two points** is:

$$\ell = \mathbf{x} \times \mathbf{x}'.$$

Point and line duality

A line ℓ is perpendicular to every point (ray) $\mathbf{p}$ on the line: $\ell^T \mathbf{p} = 0$



The line ℓ spanned by rays $\mathbf{p_1}$ and $\mathbf{p_2}$: $\Rightarrow \ell = \mathbf{p_1} \times \mathbf{p_2}$

- ℓ is perpendicular to $\mathbf{p_1}$ and $\mathbf{p_2}$
- ℓ is the plane normal

The point $\mathbf{p}$ intersection of two lines ℓ_1 and ℓ_2:

- $\mathbf{p}$ is perpendicular to ℓ_1 and ℓ_2: $\Rightarrow \mathbf{p} = \ell_1 \times \ell_2$

Points and lines are **dual** in the projective space: given any formula, the meanings of points and lines can be switched to get another formula.

Image source: [S. Seitz]

The projective plane

Duality

Given any formula, the meaning of points and lines can be switched to get another valid formula.

Ex: $\mathbf{x} = \ell \times \ell'$ and $\ell = \mathbf{x} \times \mathbf{x}'$

Ex: Exists a single line that joins two points, and
exists a single point intersecting two lines.

The projective plane

The line at infinity

$$\ell_\infty = (0, 0, 1)^T$$

The projective plane

The line at infinity

$$\ell_\infty = (0, 0, 1)^T$$

Comments:

- It is formed by the points of the plane $\pi_0 = \{(x_1, x_2, 0)^T : (x_1, x_2) \neq (0, 0)\}$.
- It satisfies $\langle (0, 0, 1)^T, (x_1, x_2, 0)^T \rangle = 0$.
- Intersection of ℓ_∞ and $\ell = (a, b, c)^T$: $\mathbf{x} = (b, -a, 0)^T$, point at infinity.

The projective plane

The line at infinity

$$\ell_\infty = (0,0,1)^T$$

Application: image rectification (removal of projective distortion)

The projective plane

Conics are curves formed by the intersection of a cone with planes at different angles: the conic sections. These curves are: circles, ellipses, parabolas, and hyperbolas.

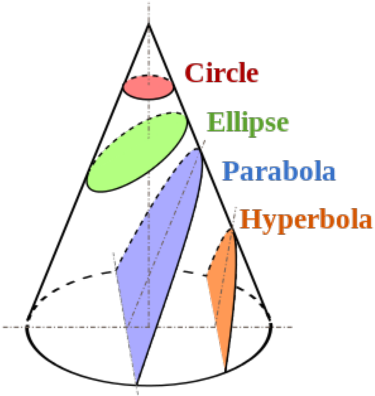


Image source: [Wikipedia]

The projective plane

3. Representation of conics:

Equation of a conic in the plane:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0, \quad a, b, c, d, e, f \in \mathbb{R}.$$

Consider the symmetric matrix:

$$C = \begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix}$$

For $\mathbf{x} = (x, y, 1)^T$, the conic equation writes:

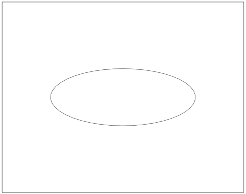
$$\langle C\mathbf{x}, \mathbf{x} \rangle = 0, \text{ or } \mathbf{x}^T C\mathbf{x} = 0$$

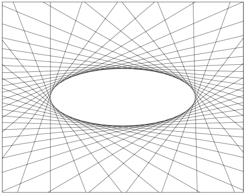
(Note: it holds also for $\mathbf{x}' = s\mathbf{x}$)
Equivalent representations: $C \equiv C'$, if $\exists k \neq 0$ such that $C' = kC$.

The projective plane

3. Representation of conics:

Dual conic (or line conic) There exists a conic that defines an equation for lines (duality points-lines).





Denoted by C^*. It verifies:

$$\langle C^*\ell, \ell\rangle = 0, \text{ or } \ell^T C^*\ell = 0$$

(Note: it holds also for $\ell' = s\ell$)
Equivalent representations: $C^* \equiv C^{*'}$, if $\exists k \neq 0$ such that $C^{*'} = kC^*$.
If C is invertible, then $C^* \equiv C^{-1}$

Image source: [Hartley and Zisserman 2004]

The projective plane

Conics

Applications:

- Image rectification (removal of projective distortion)
- Measuring real angles from image projections
- Camera calibration
- Auto-calibration

Summary

We have seen:

- Introduction to 3D vision and applications
- Definition of 2D projective space, $\mathbb{P}^2$
- Homogeneous (and cartesian) coordinates
- Points, lines and conics in $\mathbb{P}^2$

Summary

We have seen:

- Introduction to 3D vision and applications
- Definition of 2D projective space, $\mathbb{P}^2$
- Homogeneous (and cartesian) coordinates
- Points, lines and conics in $\mathbb{P}^2$

Next class:

- 2D transformations in $\mathbb{P}^2$
- Homographies and practical applications
- Homography estimation
- Affine and metric rectification.

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.