Computer Vision: Lecture 1

Christopher Zach

2023-10-30

Outline

- General course information
- What is computer vision & 3D reconstruction from images??
- Modelling cameras
- Homogeneous coordinates

General Course Information

- Lectures: $14 \times 2h$
  - Check TimeEdit for lecture hall
  - If I am late Thursday mornings, blame Hisingsbron...
- Lab sessions: $14 \times 2h$
- Assignments: 4 (*mandatory* and *optional* parts)
- Project (*mandatory* and *optional* parts)

Course requirements

- Passing the course: Completed assignments and project in time
  - You need 50% of mandatory theoretical exercises points for *each* assignment
  - You need to complete the mandatory computer exercises
- Grade 4: Same as above and at least 50% of all achievable *optional* points
- Grade 5: Same as above and oral exam

There is no written exam in the course

General Course Information

- Lectures: $14 \times 2h$
  - Check TimeEdit for lecture hall
  - If I am late Thursday mornings, blame Hisingsbron...
- Lab sessions: $14 \times 2h$
- Assignments: 4 (*mandatory* and *optional* parts)
- Project (*mandatory* and *optional* parts)

Course requirements

- Passing the course: Completed assignments and project in time
  - You need 50% of mandatory theoretical exercises points for *each* assignment
  - You need to complete the mandatory computer exercises
- Grade 4: Same as above and at least 50% of all achievable *optional* points
- Grade 5: Same as above and oral exam

There is no written exam in the course

Assignments and Exercises

- 4 assignments, which contain both theoretical and computer exercises
- You have the opportunity to work with the assignments and ask questions during the exercise sessions
- After completing an assignment, you submit a report in pdf together with your code in Canvas
- You have about 2 weeks for each assignment (and 1 month for the project) Exact dates are written in the assignments, alternatively check Canvas
- You may collaborate with another students for the computer exercises. Groups larger than two students are not allowed.
  - State clearly in the report who you collaborated with
- The reports should be done individually. (You should understand and be able to explain the solution you hand in.)

Assignments and Exercises

- 4 assignments, which contain both theoretical and computer exercises
- You have the opportunity to work with the assignments and ask questions during the exercise sessions
- After completing an assignment, you submit a report in pdf together with your code in Canvas
- You have about 2 weeks for each assignment (and 1 month for the project)
  Exact dates are written in the assignments, alternatively check Canvas
- You may collaborate with another students for the computer exercises. Groups larger than two students are not allowed.
  - State clearly in the report who you collaborated with
- The reports should be done individually.
  (You should understand and be able to explain the solution you hand in.)

Assignments and Exercises

Requirements to pass

- You need to obtain 50% of points for mandatory theory questions in each assignment
- You need to complete the mandatory computer exercises
  - Most of them will be used in the final project
- If you don't complete an assignment:
  - You may have the chance to submit a revised solution up to 1 week after getting feedback from the TA
  - Only applicable if your solutions require relatively minor corrections
- Late submission
  - You can use 5 late days in total
  - Downside: no optional points (for that assignment) is graded
- Use exercise sessions (but use them wisely)

Project

Requirements to pass

- Project: mandatory and optional parts
- Detailed description will be presented in late November / early December
  - Project outline: you will create software for 3D reconstruction from images
  - Details on the project, pitfalls and how to get optional points
- Deadline is early January

Cheating / Plagiarism

- Don't cheat
- Don't use repositories offering solutions from previous years
- Don't copy code from another student group
- Behave like a trustworthy 5th year student
- Ultimately passing this course should be easy

Low risk but high penalty!

Code development

- Matlab is the default programming language
  - TAs will focus on Matlab-related questions
  - Code provided by us will be in Matlab
  - Octave should work as well
- You may submit Python code
  - Restrict yourself to well-known libraries
    - NumPy for linear algebra
    - OpenCV for image loading and feature extracting & matching
    - Matplotlib / Open3D for visualization
- You cannot simply call a library function when you should implement an algorithm!
  - Exception: functions explicitly allowed or corresponding to supplied routines

Matlab	Python
`a = [1 0; 0 1]`	`a = np.array([[1, 0], [0, 1]])`
`b = [4 1; 2 2]`	`b = np.array([[4, 1], [2, 2]])`
`a*b`	`np.matmul(a, b)`

Literature

Literature

- Lecture and lecture slides are self-contained
- Optional: lecture notes
  - `https://www.maths.lth.se/matematiklth/personal/calle/datorseende19/index.html`
  - May be helpful
  - *Are neither necessary nor sufficient for this course*
- Some additional scientific papers

Additional Reading

- Hartley, Zisserman, Multiple View Geometry, 2004.
- Szeliski, Computer Vision Algorithms and Applications, 2010. Availible online; http://szeliski.org/Book/

Planned contents

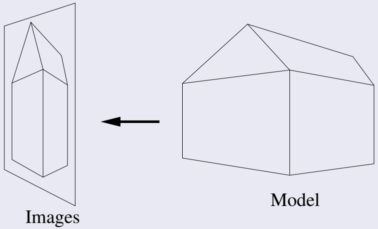
Week 1	Intro, camera model	Projective geometry
Week 2	Camera calibration, DLT I	DLT II, feature matching
Week 3	Two-view geometry I	Two-view geometry II
Week 4	Robust estimation	Minimal solvers
Week 5	MLE & Non-Linear opt., Factorization	Non-Seq. SfM, project pres.
Week 6	Bundle adjustment I, Uncertainty	Bundle adjustment II
Week 7	Non-rigid SfM (guest)	Dense reconstruction

Slides based on the ones by Carl Olsson & Fredrik Kahl
Lecture notes: Carl Olsson & Fredrik Kahl

What is Computer Vision?

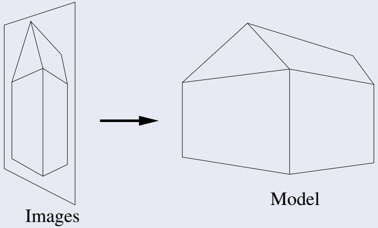
- 1970–2015: enabling software to understand images in 3D
- 2015–20xx: umbrella term for image analysis, comp. photography, pattern recognition etc.

Computer Graphics



Generate images from a 3D model.

Computer Vision



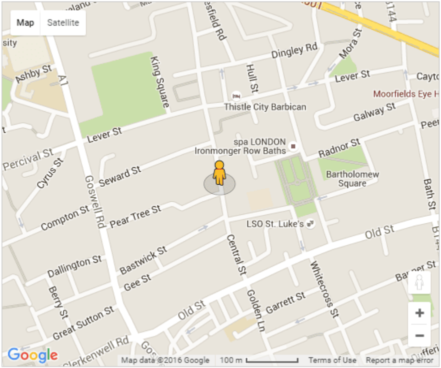
The inverse problem: Generate 3D model from images.

What can we do with 3D models?

Aerial photogrammetry: city planning, flood/disaster simulation

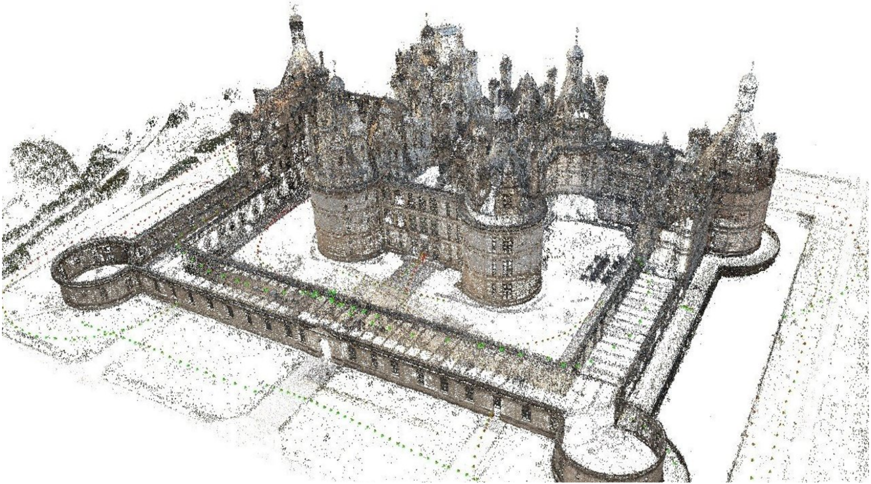
What can we do with 3D models?

Aerial + terrestrial photogrammetry: street view



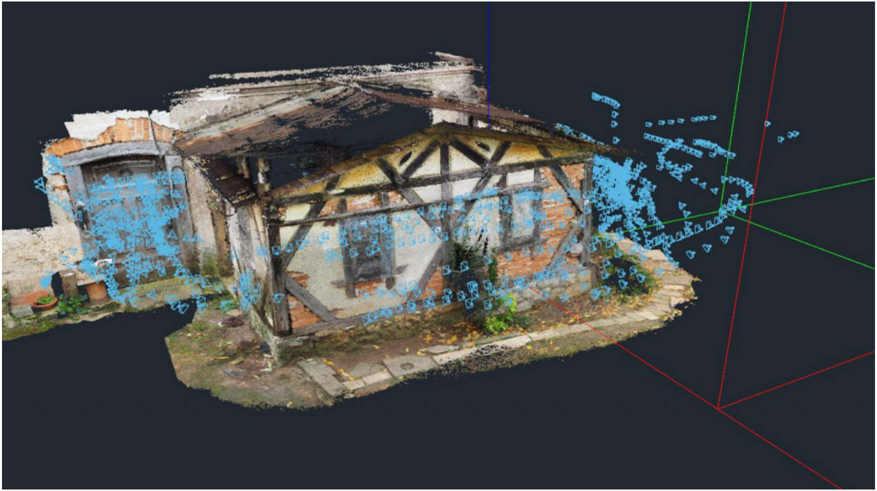
What can we do with 3D models?

Cultural heritage preservation



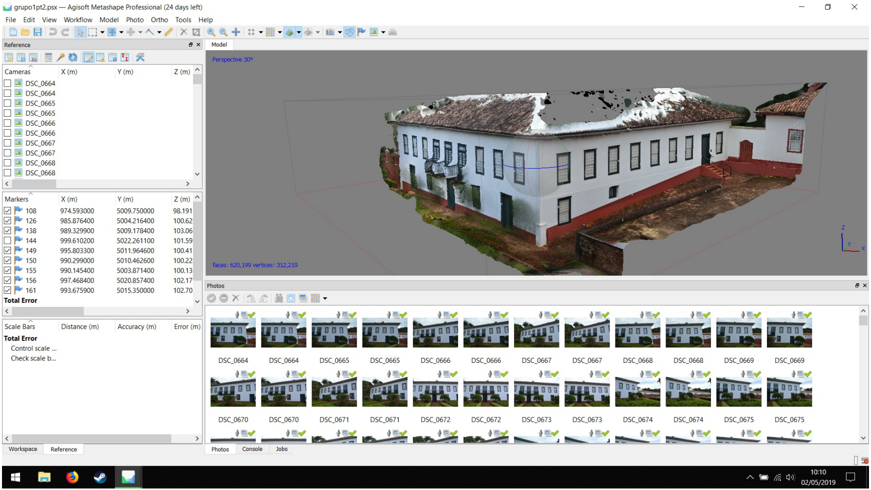
What can we do with 3D models?

Cultural heritage preservation



What can we do with 3D models?

Architectural planning



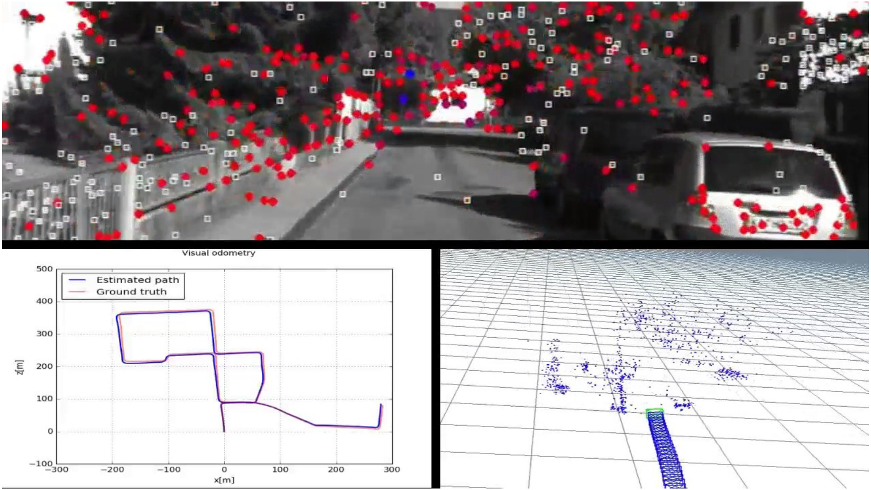
By NELAC - N.ELAC

What can we do with 3D models?

Augmented reality

What can we do with 3D models?

Autonomous vehicles & robotics

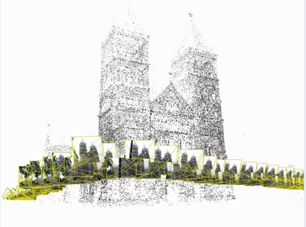


Main Goal of the Course: Multiview Reconstruction

Given Images

4 images out of a sequence with 435 images.

Compute 3D Model

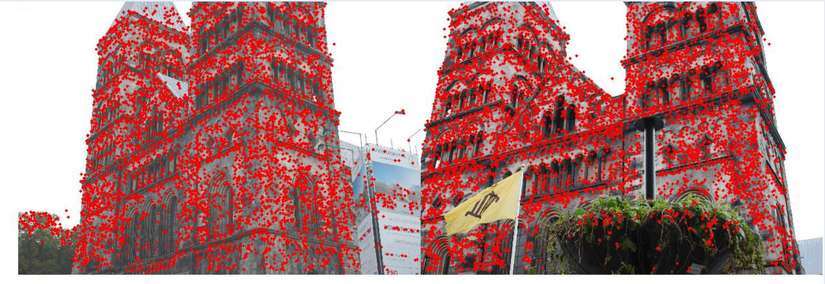


Multiview Reconstruction Pipeline

Point Detection and Matching

Multiview Reconstruction Pipeline

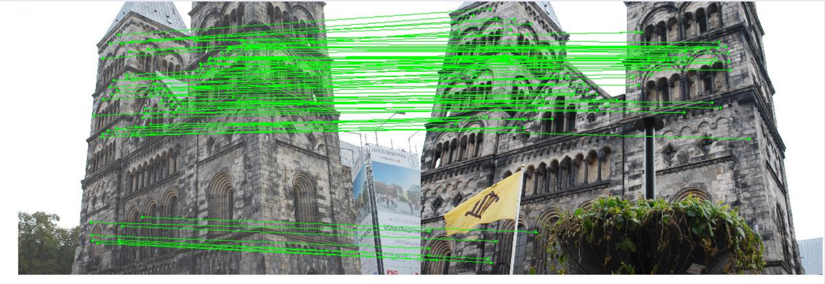
Point Detection and Matching



Detect interesting (descriptive) points in all images.

Multiview Reconstruction Pipeline

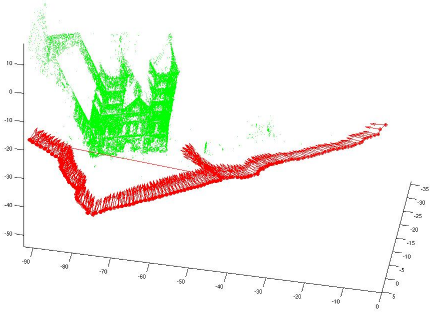
Point Detection and Matching



Match points between images.

Multiview Reconstruction Pipeline

Geometric Computations (main part of this course!)



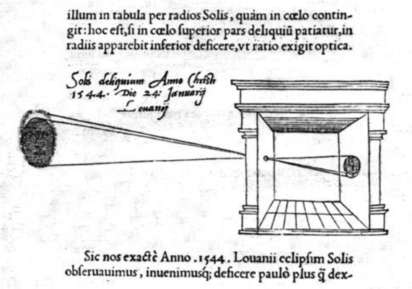
Compute 3D-positions of the matched points, position and orientation of the cameras.

Video

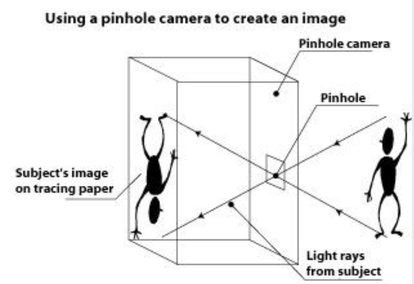
Live demo

Camera Model

The Pinhole Camera

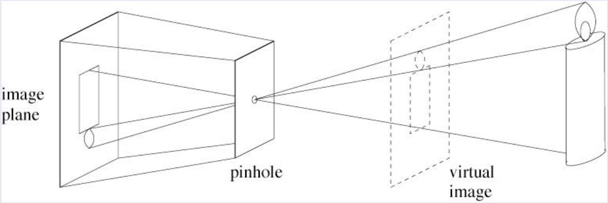


Reinerus Gemma-Frisus camera obscura from 1544.



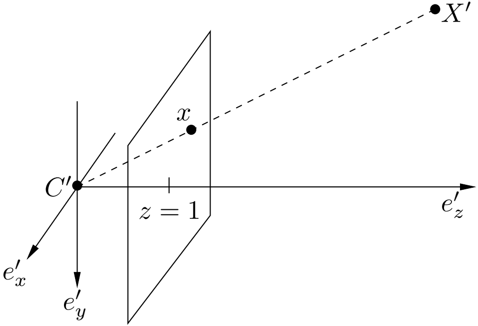
Camera Model

The Pinhole Camera



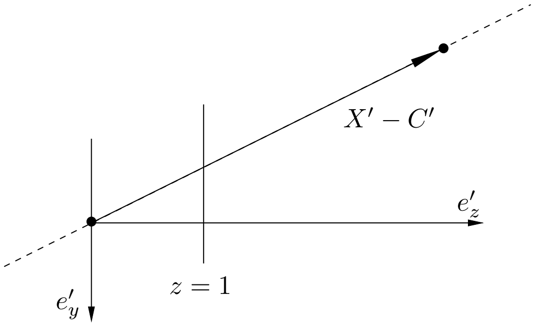
- Idealized model of how a real camera with a lens system works
- Approximately correct for modern consumer cameras

The Pinhole Camera Model



The mathematical model for a pinhole camera

The Pinhole Camera Model



The model viewed from the side. (The vector e'_x points out of the figure.)

The Pinhole Camera Model

Going from 3D points to image points involves 3 steps

- Transforming from world coordinates to camera coordinates

$$\mathbf{X}' = \mathtt{R}(\mathbf{X} - \mathbf{C}) = \mathtt{R}\mathbf{X} + \mathbf{T}$$

- Perspective projection: camera coordinates to image coordinates

$$\mathbf{x} = \frac{1}{X_3'} \begin{pmatrix} X_1' \\ X_2' \\ X_3' \end{pmatrix} = \begin{pmatrix} X_1'/X_3' \\ X_2'/X_3' \\ 1 \end{pmatrix}$$

- Calibration matrix: image coordinates to sensor coordinates

$$\mathbf{p} = \mathtt{K}\mathbf{x} = \begin{pmatrix} \gamma f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}$$

If $\mathtt{K}$ is given, then the camera is *calibrated*, otherwise *uncalibrated*. Modern cameras: $\gamma = 1$ and $s = 0$. More about that in lecture 3.

The Pinhole Camera Model

Going from 3D points to image points involves 3 steps

- Transforming from world coordinates to camera coordinates

$$\mathbf{X}' = \mathtt{R}(\mathbf{X} - \mathbf{C}) = \mathtt{R}\mathbf{X} + \mathbf{T}$$

- Perspective projection: camera coordinates to image coordinates

$$\mathbf{x} = \frac{1}{X_3'} \begin{pmatrix} X_1' \\ X_2' \\ X_3' \end{pmatrix} = \begin{pmatrix} X_1'/X_3' \\ X_2'/X_3' \\ 1 \end{pmatrix}$$

- Calibration matrix: image coordinates to sensor coordinates

$$\mathbf{p} = \mathtt{K}\mathbf{x} = \begin{pmatrix} \gamma f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}$$

If $\mathtt{K}$ is given, then the camera is *calibrated*, otherwise *uncalibrated*. Modern cameras: $\gamma = 1$ and $s = 0$. More about that in lecture 3.

The Pinhole Camera Model

Going from 3D points to image points involves 3 steps

- Transforming from world coordinates to camera coordinates

$$\mathbf{X}' = \mathtt{R}(\mathbf{X} - \mathbf{C}) = \mathtt{R}\mathbf{X} + \mathbf{T}$$

- Perspective projection: camera coordinates to image coordinates

$$\mathbf{x} = \frac{1}{X_3'} \begin{pmatrix} X_1' \\ X_2' \\ X_3' \end{pmatrix} = \begin{pmatrix} X_1'/X_3' \\ X_2'/X_3' \\ 1 \end{pmatrix}$$

- Calibration matrix: image coordinates to sensor coordinates

$$\mathbf{p} = \mathtt{K}\mathbf{x} = \begin{pmatrix} \gamma f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}$$

If $\mathtt{K}$ is given, then the camera is *calibrated*, otherwise *uncalibrated*.
Modern cameras: $\gamma = 1$ and $s = 0$. More about that in lecture 3.

The Pinhole Camera Model

Transforming from world coordinates to camera coordinates

- Euclidean transformation

$$\mathbf{X}' = \mathtt{R}(\mathbf{X} - \mathbf{C}) = \mathtt{R}\mathbf{X} + \mathbf{T}$$

- Rotates and shifts points (position vectors) in 3D
- Matrix $\mathtt{R} \in \mathbb{R}^{3\times 3}$ is a rotation matrix

$$\mathtt{R}^\top \mathtt{R} = \mathtt{I} \qquad \det(\mathtt{R}) = 1$$

- Camera center $\mathbf{C} \in \mathbb{R}^3$ and translation vector $\mathbf{T} \in \mathbb{R}^3$

$$\mathbf{C}' = \mathbf{0} = \mathtt{R}\mathbf{C} + \mathbf{T} \iff \mathbf{C} = -\mathtt{R}^\top \mathbf{T} \iff \mathbf{T} = -\mathtt{R}\mathbf{C}$$

The Pinhole Camera Model

Transforming from world coordinates to camera coordinates

- How do direction vectors transform?

$$\mathbf{d}' = \mathtt{R}\mathbf{d}$$

- How do planes transform?

$$\mathbf{X}' = \mathtt{R}\mathbf{X} + \mathbf{T} \qquad \begin{pmatrix} X_1' \\ X_2' \\ X_3' \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \mathtt{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix}}_{=:\mathtt{M}} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix}$$

Plane equation $0 = \Pi^\top \binom{\mathbf{X}}{1}$

$$0 = \Pi^\top \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \Pi^\top \mathtt{M}^{-1}\mathtt{M} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \left(\mathtt{M}^{-\top}\Pi\right)^\top \begin{pmatrix} \mathbf{X}' \\ 1 \end{pmatrix}$$

$$\mathtt{M}^{-1} = \begin{pmatrix} \mathtt{R}^\top & -\mathtt{R}^\top\mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix} \qquad \mathtt{M}^{-\top} = \begin{pmatrix} \mathtt{R} & \mathbf{0} \\ -\mathbf{T}^\top\mathtt{R} & 1 \end{pmatrix}$$

The Pinhole Camera Model

Transforming from world coordinates to camera coordinates

- How do direction vectors transform?

$$\mathbf{d}' = \mathtt{R}\mathbf{d}$$

- How do planes transform?

$$\mathbf{X}' = \mathtt{R}\mathbf{X} + \mathbf{T} \qquad \begin{pmatrix} X_1' \\ X_2' \\ X_3' \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \mathtt{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix}}_{=:\mathtt{M}} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix}$$

Plane equation $0 = \Pi^\top \binom{\mathbf{X}}{1}$

$$0 = \Pi^\top \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \Pi^\top \mathtt{M}^{-1}\mathtt{M} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \left(\mathtt{M}^{-\top}\Pi\right)^\top \begin{pmatrix} \mathbf{X}' \\ 1 \end{pmatrix}$$

$$\mathtt{M}^{-1} = \begin{pmatrix} \mathtt{R}^\top & -\mathtt{R}^\top \mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix} \qquad \mathtt{M}^{-\top} = \begin{pmatrix} \mathtt{R} & \mathbf{0} \\ -\mathbf{T}^\top \mathtt{R} & 1 \end{pmatrix}$$

The Pinhole Camera Model

Transforming from world coordinates to camera coordinates

- How do direction vectors transform?

$$\mathbf{d}' = \mathtt{R}\mathbf{d}$$

- How do planes transform?

$$\mathbf{X}' = \mathtt{R}\mathbf{X} + \mathbf{T} \qquad \begin{pmatrix} X_1' \\ X_2' \\ X_3' \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \mathtt{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix}}_{=:\mathtt{M}} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix}$$

Plane equation $0 = \Pi^\top \binom{\mathbf{X}}{1}$

$$0 = \Pi^\top \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \Pi^\top \mathtt{M}^{-1}\mathtt{M} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \left(\mathtt{M}^{-\top}\Pi\right)^\top \begin{pmatrix} \mathbf{X}' \\ 1 \end{pmatrix}$$

$$\mathtt{M}^{-1} = \begin{pmatrix} \mathtt{R}^\top & -\mathtt{R}^\top\mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix} \qquad \mathtt{M}^{-\top} = \begin{pmatrix} \mathtt{R} & \mathbf{0} \\ -\mathbf{T}^\top\mathtt{R} & 1 \end{pmatrix}$$

The Pinhole Camera Model

Transforming from world coordinates to camera coordinates

- How do direction vectors transform?

$$\mathbf{d}' = \mathtt{R}\mathbf{d}$$

- How do planes transform?

$$\mathbf{X}' = \mathtt{R}\mathbf{X} + \mathbf{T} \qquad \begin{pmatrix} X_1' \\ X_2' \\ X_3' \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \mathtt{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix}}_{=:\mathtt{M}} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix}$$

Plane equation $0 = \Pi^\top \binom{\mathbf{X}}{1}$

$$0 = \Pi^\top \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \Pi^\top \mathtt{M}^{-1}\mathtt{M} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \left(\mathtt{M}^{-\top}\Pi\right)^\top \begin{pmatrix} \mathbf{X}' \\ 1 \end{pmatrix}$$

$$\mathtt{M}^{-1} = \begin{pmatrix} \mathtt{R}^\top & -\mathtt{R}^\top\mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix} \qquad \mathtt{M}^{-\top} = \begin{pmatrix} \mathtt{R} & \mathbf{0} \\ -\mathbf{T}^\top\mathtt{R} & 1 \end{pmatrix}$$

The Pinhole Camera Model

Transforming from world coordinates to camera coordinates

- How do direction vectors transform?

$$\mathbf{d}' = \mathtt{R}\mathbf{d}$$

- How do planes transform?

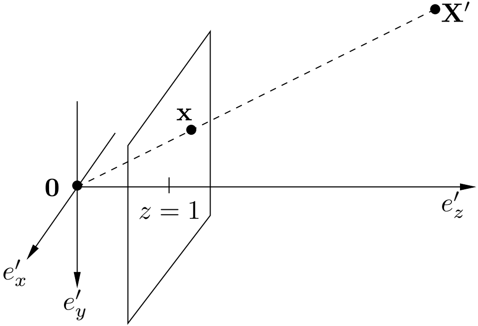
$$\mathbf{X}' = \mathtt{R}\mathbf{X} + \mathbf{T} \qquad \begin{pmatrix} X_1' \\ X_2' \\ X_3' \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \mathtt{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix}}_{=:\mathtt{M}} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix}$$

Plane equation $0 = \Pi^\top \binom{\mathbf{X}}{1}$

$$0 = \Pi^\top \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \Pi^\top \mathtt{M}^{-1}\mathtt{M} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \left(\mathtt{M}^{-\top}\Pi\right)^\top \begin{pmatrix} \mathbf{X}' \\ 1 \end{pmatrix}$$

$$\mathtt{M}^{-1} = \begin{pmatrix} \mathtt{R}^\top & -\mathtt{R}^\top\mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix} \qquad \mathtt{M}^{-\top} = \begin{pmatrix} \mathtt{R} & \mathbf{0} \\ -\mathbf{T}^\top\mathtt{R} & 1 \end{pmatrix}$$

The Pinhole Camera Model



The mathematical model for a pinhole camera

The Pinhole Camera Model

Perspective projection: from camera coordinates to image coordinates

- Non-linear mapping $\pi : \mathbb{R}^3 \to \mathbb{R}^3$

$$\mathbf{x} = \pi(\mathbf{X}') = \frac{1}{X_3'} \begin{pmatrix} X_1' \\ X_2' \\ X_3' \end{pmatrix} = \begin{pmatrix} X_1'/X_3' \\ X_2'/X_3' \\ 1 \end{pmatrix}$$

- $\pi(s\mathbf{X}') = \pi(\mathbf{X}')$ for all $s \neq 0$

$$\pi^{-1}(\mathbf{x}) = \{s\mathbf{x} : s \neq 0\}$$

- X_3' is the *depth* of $\mathbf{X}'$: point-plane distance!

Discussion

- What does it mean if the depth $X_3' < 0$? If $X_3' < 1$?
- What happens if $X_3' \to \infty$? If $X_3' = 0$?
- Isn't $\mathbf{x} \in \mathbb{R}^3$, but image/sensor coordinates should be 2D?

The Pinhole Camera Model

Perspective projection: from camera coordinates to image coordinates

- Non-linear mapping $\pi : \mathbb{R}^3 \to \mathbb{R}^3$

$$\mathbf{x} = \pi(\mathbf{X}') = \frac{1}{X_3'} \begin{pmatrix} X_1' \\ X_2' \\ X_3' \end{pmatrix} = \begin{pmatrix} X_1'/X_3' \\ X_2'/X_3' \\ 1 \end{pmatrix}$$

- $\pi(s\mathbf{X}') = \pi(\mathbf{X}')$ for all $s \neq 0$

$$\pi^{-1}(\mathbf{x}) = \{s\mathbf{x} : s \neq 0\}$$

- X_3' is the *depth* of $\mathbf{X}'$: point-plane distance!

Discussion

- What does it mean if the depth $X_3' < 0$? If $X_3' < 1$?
- What happens if $X_3' \to \infty$? If $X_3' = 0$?
- Isn't $\mathbf{x} \in \mathbb{R}^3$, but image/sensor coordinates should be 2D?

Homogeneous coordinates and $\mathbb{P}^2$

Projective space $\mathbb{P}^2$

If there exists a $\lambda \neq 0$ such that $\mathbf{x} = \lambda \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$, then we write $\mathbf{x} \sim \mathbf{y}$. An equivalence relation $\sim$ induces equivalence classes

$$[\mathbf{x}] := \{\mathbf{y} \in \mathbb{R}^3 : \mathbf{y} \sim \mathbf{x}\}$$

$\mathbb{P}^2$ is the quotient space of $\mathbb{R}^3$ by $\sim$: $\mathbb{P}^2 = \{[\mathbf{x}] : \mathbf{x} \in \mathbb{R}^3\} = \mathbb{R}^3 / \sim$
By definition, we exclude the origin $\mathbf{0} = (0,0,0)^\top$ from $\mathbb{P}^2$

- Straightforward to extend to $\mathbb{P}^k = \mathbb{R}^{k+1} / \sim$
- If $\mathbf{x} = (x_1, x_2, x_3)^\top$ with $x_3 \neq 0$, then $\begin{pmatrix} x_1/x_3 \\ x_2/x_3 \\ 1 \end{pmatrix}$ is a representative for $[\mathbf{x}]$
- Bijection between $(x_1, x_2, 1)^\top \in [\mathbf{x}] \in \mathbb{P}^2$ and $(x_1, x_2)^\top \in \mathbb{R}^2$
- We can move between $\mathbb{R}^2$ and $\mathbb{P}^2$ as needed (if $x_3 \neq 0$)!

$(x_1, x_2, x_3)^\top$ are homogeneous coordinates corresponding to $(x_1/x_3, x_2/x_3)^\top$

Homogeneous coordinates and $\mathbb{P}^2$

Projective space $\mathbb{P}^2$

If there exists a $\lambda \neq 0$ such that $\mathbf{x} = \lambda \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$, then we write $\mathbf{x} \sim \mathbf{y}$. An equivalence relation $\sim$ induces equivalence classes

$$[\mathbf{x}] := \{\mathbf{y} \in \mathbb{R}^3 : \mathbf{y} \sim \mathbf{x}\}$$

$\mathbb{P}^2$ is the quotient space of $\mathbb{R}^3$ by $\sim$: $\mathbb{P}^2 = \{[\mathbf{x}] : \mathbf{x} \in \mathbb{R}^3\} = \mathbb{R}^3 / \sim$
By definition, we exclude the origin $\mathbf{0} = (0,0,0)^\top$ from $\mathbb{P}^2$

- Straightforward to extend to $\mathbb{P}^k = \mathbb{R}^{k+1} / \sim$
- If $\mathbf{x} = (x_1, x_2, x_3)^\top$ with $x_3 \neq 0$, then $\begin{pmatrix} x_1/x_3 \\ x_2/x_3 \\ 1 \end{pmatrix}$ is a representative for $[\mathbf{x}]$
- Bijection between $(x_1, x_2, 1)^\top \in [\mathbf{x}] \in \mathbb{P}^2$ and $(x_1, x_2)^\top \in \mathbb{R}^2$
- We can move between $\mathbb{R}^2$ and $\mathbb{P}^2$ as needed (if $x_3 \neq 0$)!

$(x_1, x_2, x_3)^\top$ are homogeneous coordinates corresponding to $(x_1/x_3, x_2/x_3)^\top$

Homogeneous coordinates and $\mathbb{P}^2$

Projective space $\mathbb{P}^2$

If there exists a $\lambda \neq 0$ such that $\mathbf{x} = \lambda \mathbf{y}$ for $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^3$, then we write $\mathbf{x} \sim \mathbf{y}$. An equivalence relation $\sim$ induces equivalence classes

$$[\mathbf{x}] := \{\mathbf{y} \in \mathbb{R}^3 : \mathbf{y} \sim \mathbf{x}\}$$

$\mathbb{P}^2$ is the quotient space of $\mathbb{R}^3$ by $\sim$: $\mathbb{P}^2 = \{[\mathbf{x}] : \mathbf{x} \in \mathbb{R}^3\} = \mathbb{R}^3 / \sim$
By definition, we exclude the origin $\mathbf{0} = (0, 0, 0)^\top$ from $\mathbb{P}^2$

- Straightforward to extend to $\mathbb{P}^k = \mathbb{R}^{k+1} / \sim$
- If $\mathbf{x} = (x_1, x_2, x_3)^\top$ with $x_3 \neq 0$, then $\begin{pmatrix} x_1/x_3 \\ x_2/x_3 \\ 1 \end{pmatrix}$ is a representative for $[\mathbf{x}]$
- Bijection between $(x_1, x_2, 1)^\top \in [\mathbf{x}] \in \mathbb{P}^2$ and $(x_1, x_2)^\top \in \mathbb{R}^2$
- We can move between $\mathbb{R}^2$ and $\mathbb{P}^2$ as needed (if $x_3 \neq 0$)!

$(x_1, x_2, x_3)^\top$ are homogeneous coordinates corresponding to $(x_1/x_3, x_2/x_3)^\top$

Homogeneous coordinates and $\mathbb{P}^2$

Projective space $\mathbb{P}^2$

If there exists a $\lambda \neq 0$ such that $\mathbf{x} = \lambda \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$, then we write $\mathbf{x} \sim \mathbf{y}$. An equivalence relation $\sim$ induces equivalence classes

$$[\mathbf{x}] := \{\mathbf{y} \in \mathbb{R}^3 : \mathbf{y} \sim \mathbf{x}\}$$

$\mathbb{P}^2$ is the quotient space of $\mathbb{R}^3$ by $\sim$: $\mathbb{P}^2 = \{[\mathbf{x}] : \mathbf{x} \in \mathbb{R}^3\} = \mathbb{R}^3 / \sim$
By definition, we exclude the origin $\mathbf{0} = (0,0,0)^\top$ from $\mathbb{P}^2$

- Straightforward to extend to $\mathbb{P}^k = \mathbb{R}^{k+1} / \sim$
- If $\mathbf{x} = (x_1, x_2, x_3)^\top$ with $x_3 \neq 0$, then $\begin{pmatrix} x_1/x_3 \\ x_2/x_3 \\ 1 \end{pmatrix}$ is a representative for $[\mathbf{x}]$
- Bijection between $(x_1, x_2, 1)^\top \in [\mathbf{x}] \in \mathbb{P}^2$ and $(x_1, x_2)^\top \in \mathbb{R}^2$
- We can move between $\mathbb{R}^2$ and $\mathbb{P}^2$ as needed (if $x_3 \neq 0$)!

$(x_1, x_2, x_3)^\top$ are homogeneous coordinates corresponding to $(x_1/x_3, x_2/x_3)^\top$

Homogeneous coordinates and $\mathbb{P}^2$

- Converting to homogeneous coordinates ("homogenizing")

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 \rightsquigarrow \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} \in \left[\begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} \right] \in \mathbb{P}^2$$

  Representative $(x_1, x_2, 1)^\top$ is element of $\mathbb{R}^3$

- Converting to Cartesian coordinates ("de-homogenizing")

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \mathbb{P}^2 \rightsquigarrow \begin{pmatrix} x_1/x_3 \\ x_2/x_3 \end{pmatrix} \in \mathbb{R}^2$$

  if $x_3 \neq 0$

Homogeneous coordinates and $\mathbb{P}^k$

Why are homogeneous coordinates useful?

- Recall Euclidean transformation

$$\mathbf{X}' = \mathtt{R}\mathbf{X} + \mathbf{T} \qquad\qquad \mathbf{X}, \mathbf{X}' \in \mathbb{R}^3$$

  Affine expression (linear + offset)

- Interpret $\mathbf{X} \in \mathbb{P}^3$:

$$\mathbf{X}' = \underbrace{\begin{pmatrix} \mathtt{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix}}_{\in \mathbb{R}^{4x4}} \mathbf{X} \qquad\qquad \mathbf{X}, \mathbf{X}' \in \mathbb{P}^3$$

  Purely linear expression

Homogeneous coordinates and $\mathbb{P}^k$

Why are homogeneous coordinates useful?

- Recall Euclidean transformation

$$\mathbf{X}' = \mathtt{R}\mathbf{X} + \mathbf{T} \qquad\qquad \mathbf{X}, \mathbf{X}' \in \mathbb{R}^3$$

  Affine expression (linear + offset)

- Interpret $\mathbf{X} \in \mathbb{P}^3$:

$$\mathbf{X}' = \underbrace{\begin{pmatrix} \mathtt{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{pmatrix}}_{\in \mathbb{R}^{4x4}} \mathbf{X} \qquad\qquad \mathbf{X}, \mathbf{X}' \in \mathbb{P}^3$$

  Purely linear expression

Homogeneous coordinates and $\mathbb{P}^k$

Why are homogeneous coordinates useful?

- Recall projection onto the image plane, followed by calibration matrix ($s = 0$)

$$\mathbf{p} = \mathtt{K} \cdot \pi(\mathbf{X}) = \begin{pmatrix} \gamma f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_1/X_3 \\ X_2/X_3 \\ 1 \end{pmatrix} = \begin{pmatrix} \gamma f X_1/X_3 + x_0 \\ f X_2/X_3 + x_1 \\ 1 \end{pmatrix} \in \mathbb{P}^2$$

- Observe that

$$\pi(\mathtt{K}\mathbf{X}) = \frac{1}{X_3} \begin{pmatrix} \gamma f X_1 + x_0 X_3 \\ f X_2 + x_1 X_3 \\ X_3 \end{pmatrix} = \mathtt{K} \cdot \pi(\mathbf{X}) \in [\mathtt{K}\mathbf{X}]$$

- We can apply the perspective division as final step

$$\underbrace{[\mathtt{K}(\mathtt{R}\ \mathbf{T})\mathbf{X}]}_{\mathbf{X} \in \mathbb{P}^3} = \pi(\underbrace{\mathtt{K}(\mathtt{R}\mathbf{X} + \mathbf{T})}_{= \mathtt{P}\mathbf{X}})$$

Projection matrix $\mathtt{P} = \mathtt{K}(\mathtt{R} \mid \mathbf{T}) \in \mathbb{R}^{3 \times 4}$

Homogeneous coordinates and $\mathbb{P}^k$

Why are homogeneous coordinates useful?

- Recall projection onto the image plane, followed by calibration matrix $(s = 0)$

$$\mathbf{p} = \mathtt{K} \cdot \pi(\mathbf{X}) = \begin{pmatrix} \gamma f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_1/X_3 \\ X_2/X_3 \\ 1 \end{pmatrix} = \begin{pmatrix} \gamma f X_1/X_3 + x_0 \\ f X_2/X_3 + x_1 \\ 1 \end{pmatrix} \in \mathbb{P}^2$$

- Observe that

$$\pi(\mathtt{K}\mathbf{X}) = \frac{1}{X_3} \begin{pmatrix} \gamma f X_1 + x_0 X_3 \\ f X_2 + x_1 X_3 \\ X_3 \end{pmatrix} = \mathtt{K} \cdot \pi(\mathbf{X}) \in [\mathtt{K}\mathbf{X}]$$

- We can apply the perspective division as final step

$$\underbrace{[\mathtt{K}(\mathtt{R}\ \mathbf{T})\mathbf{X}]}_{\mathbf{X} \in \mathbb{P}^3} = \pi(\underbrace{\mathtt{K}(\mathtt{R}\mathbf{X} + \mathbf{T})}_{= \mathtt{P}\mathbf{X}})$$

Projection matrix $\mathtt{P} = \mathtt{K}(\mathtt{R} \mid \mathbf{T}) \in \mathbb{R}^{3 \times 4}$

Homogeneous coordinates and $\mathbb{P}^k$

Why are homogeneous coordinates useful?

- Recall projection onto the image plane, followed by calibration matrix ($s = 0$)

$$\mathbf{p} = \mathtt{K} \cdot \pi(\mathbf{X}) = \begin{pmatrix} \gamma f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_1/X_3 \\ X_2/X_3 \\ 1 \end{pmatrix} = \begin{pmatrix} \gamma f X_1/X_3 + x_0 \\ f X_2/X_3 + x_1 \\ 1 \end{pmatrix} \in \mathbb{P}^2$$

- Observe that

$$\pi(\mathtt{K}\mathbf{X}) = \frac{1}{X_3} \begin{pmatrix} \gamma f X_1 + x_0 X_3 \\ f X_2 + x_1 X_3 \\ X_3 \end{pmatrix} = \mathtt{K} \cdot \pi(\mathbf{X}) \in [\mathtt{K}\mathbf{X}]$$

- We can apply the perspective division as final step

$$\underbrace{[\mathtt{K}(\mathtt{R}\ \mathbf{T})\mathbf{X}]}_{\mathbf{X} \in \mathbb{P}^3} \equiv \pi(\underbrace{\mathtt{K}(\mathtt{R}\mathbf{X} + \mathbf{T})}_{= \mathtt{P}\mathbf{X}})$$

Projection matrix $\mathtt{P} = \mathtt{K}(\mathtt{R} \mid \mathbf{T}) \in \mathbb{R}^{3\times 4}$

Homogeneous coordinates and $\mathbb{P}^k$

Why are homogeneous coordinates useful?

- Allowing to work with quantities such as $[(x_1, x_2, 0)^\top]$
- Points, lines and planes at infinity
- More on that in the next lecture

The Pinhole Camera Model

Attention

Different conventions used in the literature/software

- Computer vision community mostly uses
  - Camera is looking in z direction
  - World coordinates to camera coordinates: $\mathbf{X}' = \mathtt{R}\mathbf{X} + \mathbf{T}$
- Computer graphics (OpenGL)
  - Camera is looking in $-z$ direction
- "Position-centric" parametrization
  - World coordinates to camera coordinates: $\mathbf{X}' = \mathtt{R}(\mathbf{X} \pm \mathbf{C})$
  - $\pm\mathbf{C}$ might be called "translation"
- Inverse parametrization
  - World coordinates to camera coordinates: $\mathbf{X}' = \mathtt{R}^\top(\mathbf{X} \pm \mathbf{C})$
  - "Look at" parametrization
- Lots of ambiguities about image coordinates (mm) and sensor coordinates (pixels)

At least the right-handed coordinate system convention is generally accepted.

The Pinhole Camera Model: Summary

Summary

- Model for a pinhole camera

$$\mathbf{x} \sim \underbrace{\mathtt{K}(\mathtt{R} \mid \mathbf{T})}_{=\mathtt{P}} \mathbf{X} = \mathtt{P}\mathbf{X} \qquad \mathbf{X} \in \mathbb{P}^3$$

- Camera represented by its projection matrix (or camera matrix) $\mathtt{P} \in \mathbb{R}^{3\times 4}$
  - $\mathtt{P}$ has 11 d.o.f.
  - If $\mathtt{K}$ is known, then the camera is said to be calibrated
- Camera center
  - in camera coordinates: $\mathbf{0}$
  - in world coordinates: $-\mathtt{R}^\top \mathbf{T}$

Next Time

- Lab session today in ES61, ES62, ES63 and via Zoom

Projective Geometry

- Projective space
- Projective transformations