

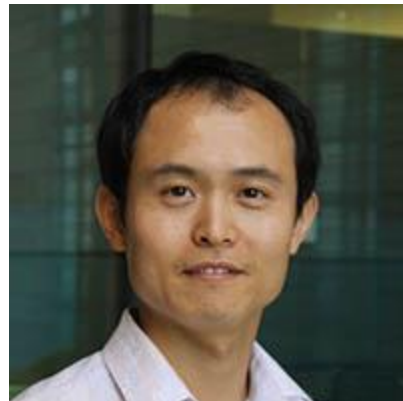
Lecture  
**Introduction**

Liangliang Nan

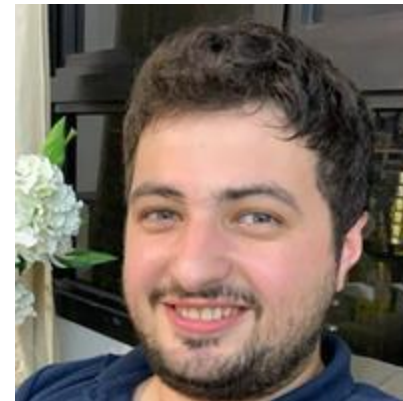
# Agenda

- The teacher and teaching assistant
- Overview of the course
  - What the course is about
  - What you will learn
  - Lectures
  - Assignments
  - Final exam
  - Communication method
- Review linear algebra basics

# Teacher & teaching assistants



[Liangliang Nan](#)  
LiangliangNan#0976



[Nail Ibrahimli](#)  
nibrahimli#5857

# What the course is about

- Photogrammetry
- Computer Vision
- 3D Computer Vision

What are the differences?



# What the course is about

- Photogrammetry
  - General
    - Obtaining info about objects or environment
      - Recording: digital image capturing
      - Measuring: generating 2D or 3D measurements
      - Interpreting: detecting interested objects
    - Output can be
      - A map
      - A drawing
      - 3D model
      - ...

# What the course is about

- Photogrammetry
  - General
  - Specific scopes in GEO1016
    - Extracting 3D geometry from images
    - Goal: generating 3D digital models of an object or a scene
    - Using well-defined photogrammetric methods
      - e.g., image matching, structure from motion, multi-view stereo, surface reconstruction

# What the course is about

- Photogrammetry
- Computer vision
  - A field of artificial intelligence
    - Training computers to interpret and understand the visual world
    - Using digital images or videos
  - Mimics the human visual system
    - Video tracking
    - Object detection/recognition
    - Scene reconstruction
    - ...

# What the course is about

- Photogrammetry
- Computer vision
- 3D computer vision
  - Scene reconstruction/modeling
  - 3D data processing (e.g., semantic segmentation, classification)





# What the course is about

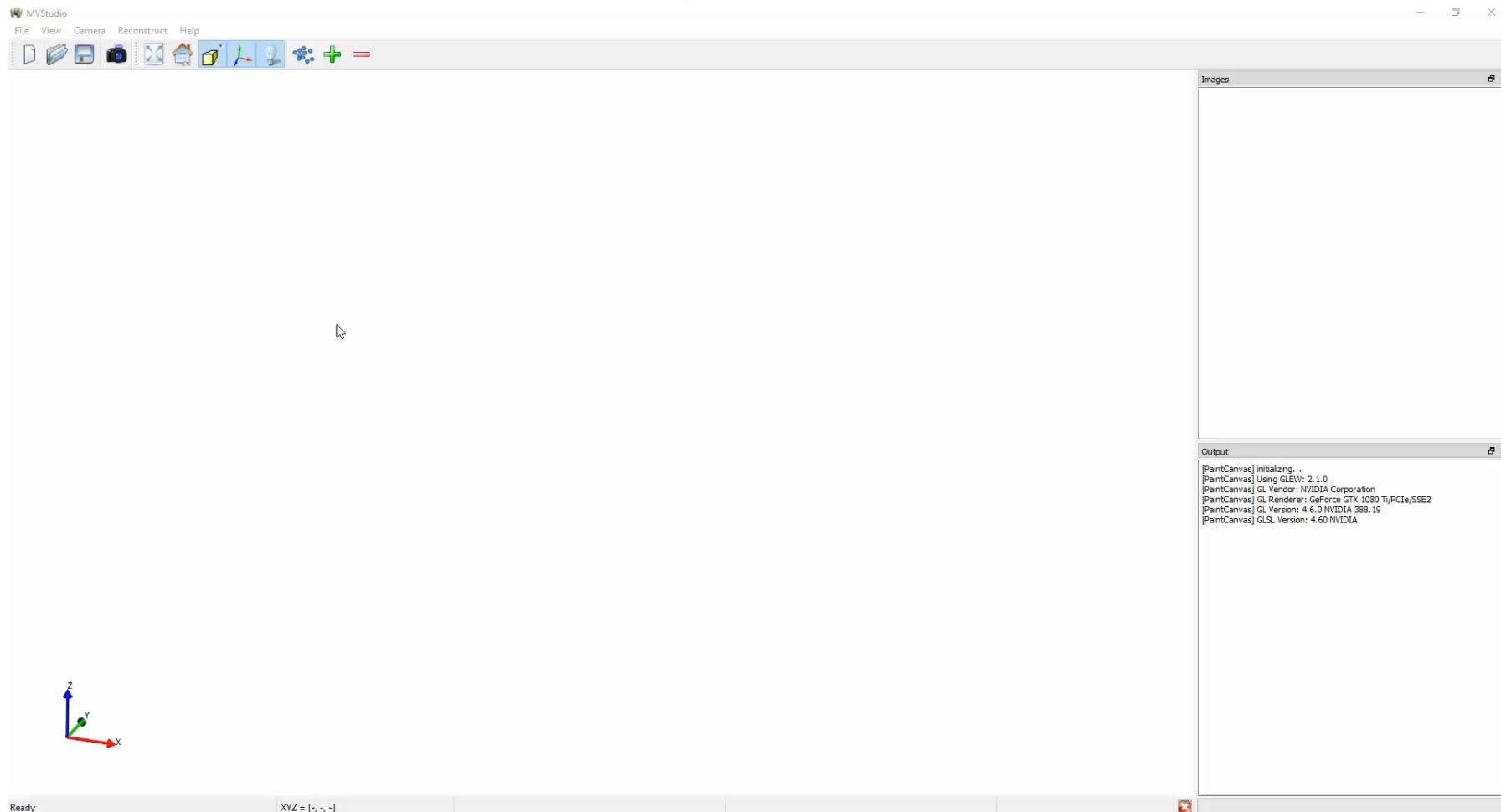
- Photogrammetry
- Computer vision
- 3D computer vision
  - Scene reconstruction/modeling
  - 3D data processing (e.g., semantic segmentation, classification)

Within this course:

**3D computer vision == Photogrammetry**

# What the course is about

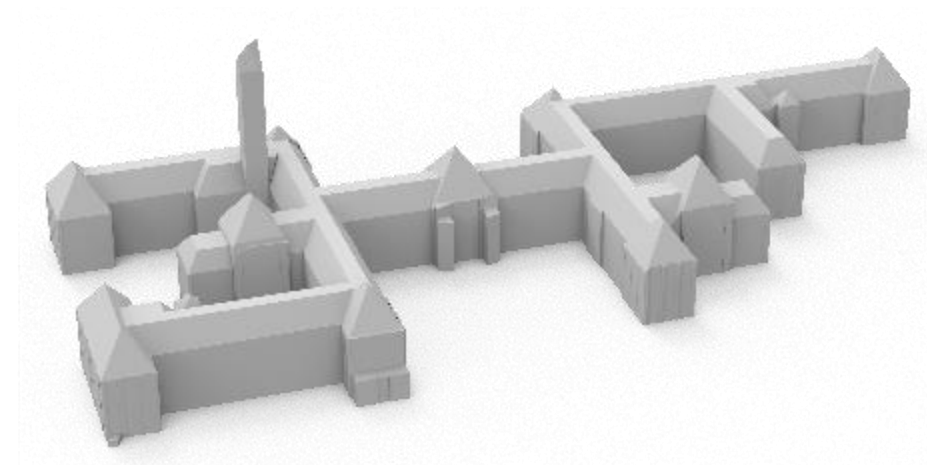
- Demo



Code & data: <https://github.com/LiangliangNan/MVStudio>

# What the course is about

- Photogrammetry
- Computer vision
- 3D computer vision
  - Scene reconstruction/modeling
  - 3D data processing (e.g., semantic segmentation, classification)
  - Urban objects



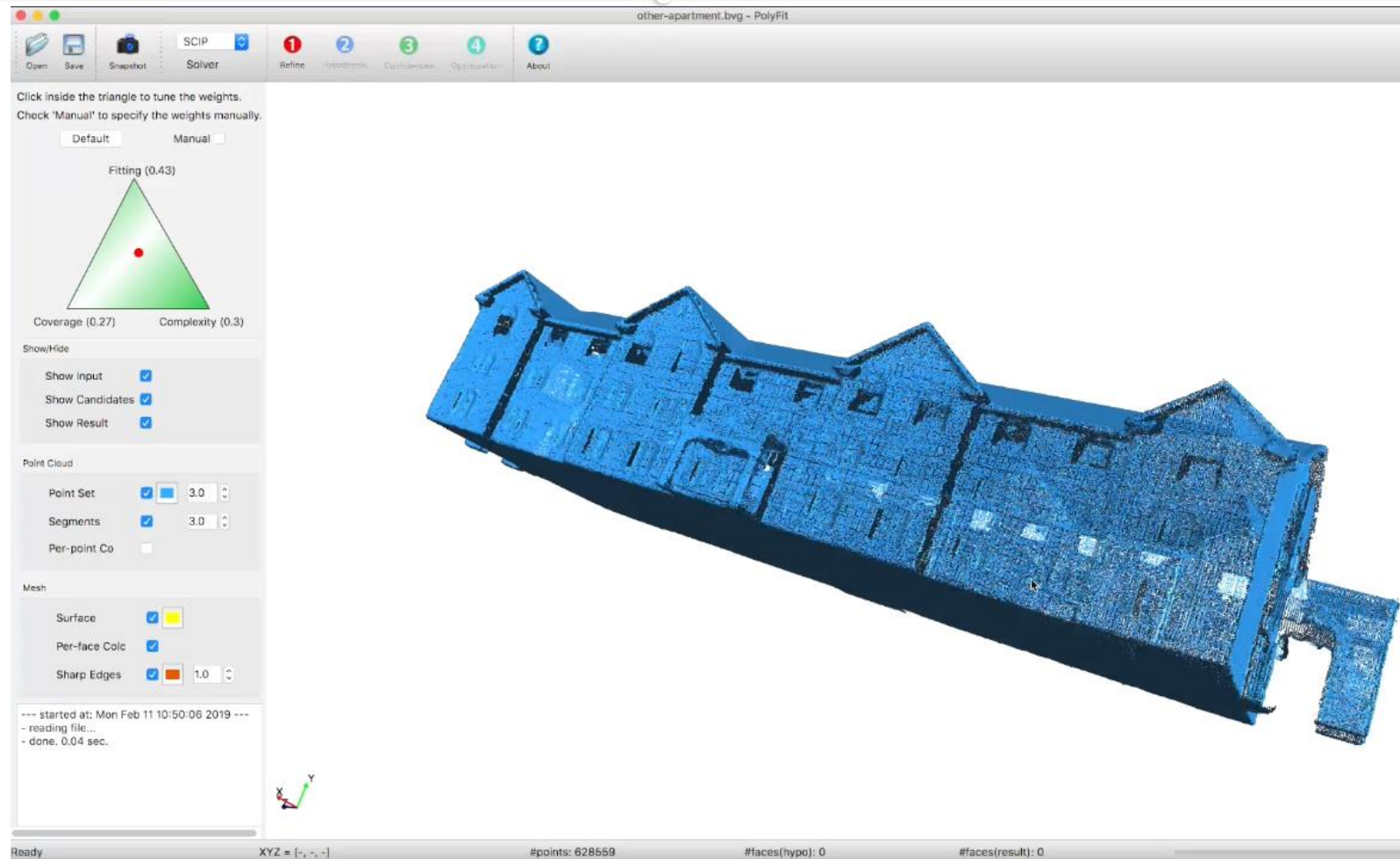
# What the course is about

- Photogrammetry
- Computer vision
- 3D computer vision
  - Scene reconstruction/modeling
  - 3D data processing (e.g., semantic segmentation, classification)
  - Urban objects
  - Point clouds vs. Surface models



# What the course is about

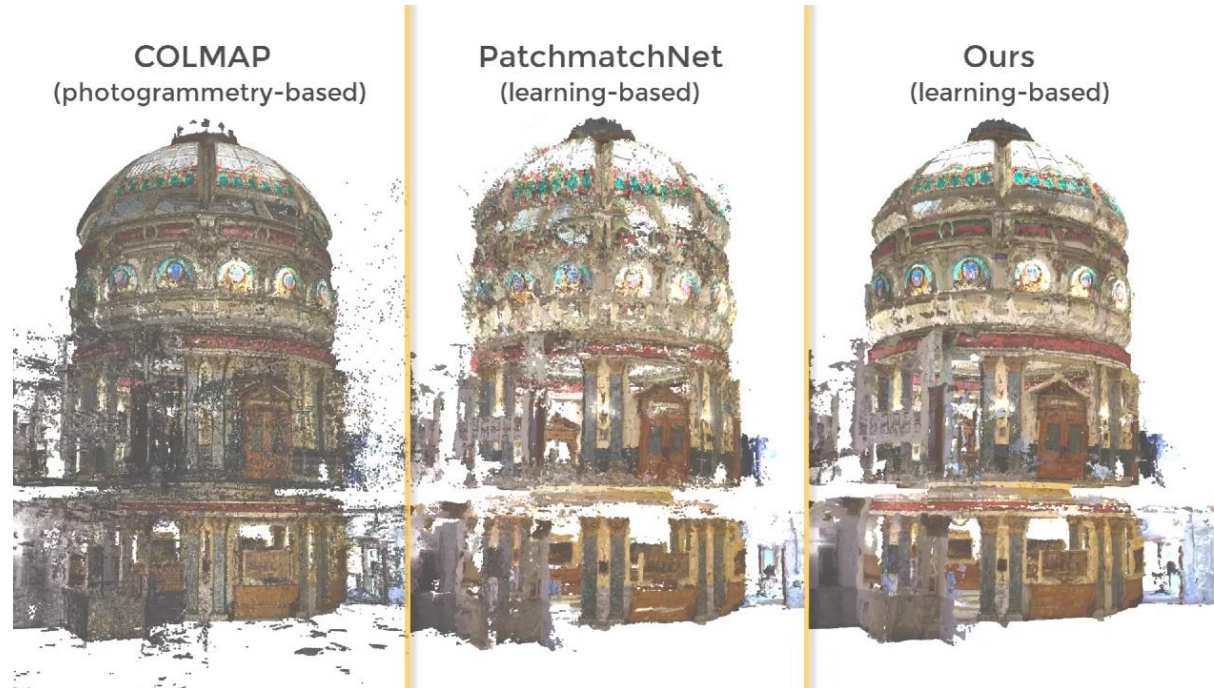
- Demo



Code & data: <https://github.com/LiangliangNan/PolyFit>

# Applications of 3DV

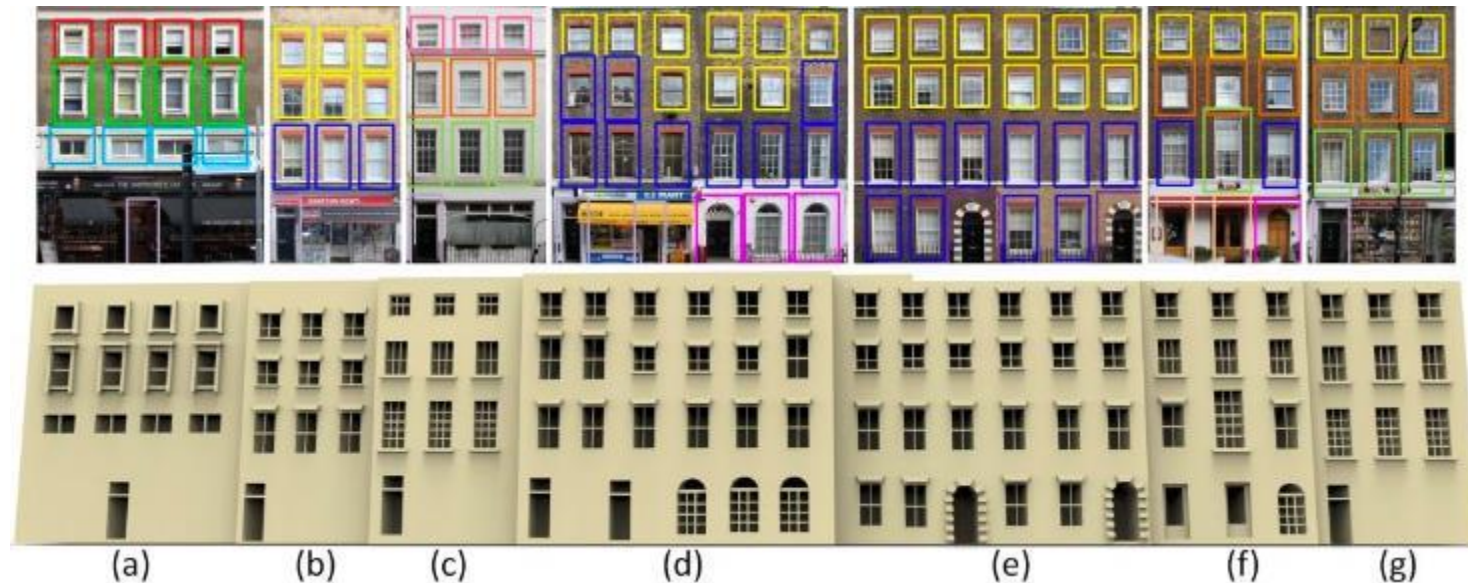
- 3D reconstruction from images





# Applications of 3DV

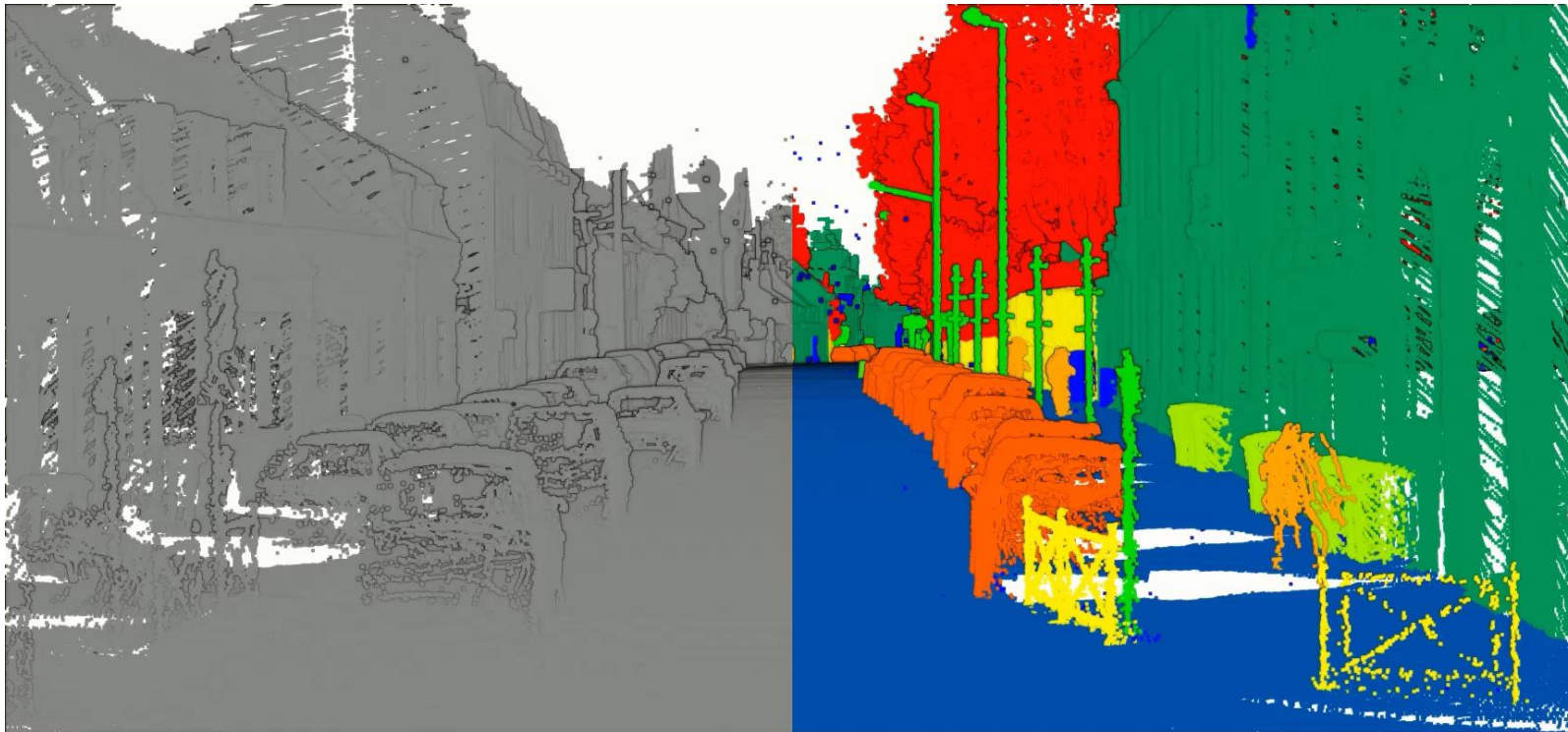
- Façade parsing and 3D modeling



Nan et al. Template Assembly for Detailed Urban Reconstruction. *Computer Graphics Forum*, Vol. 34, No. 2, 2015

# Applications of 3DV

- Semantic segmentation





# What you will learn

- Theory, methodology, and algorithms
- The complete pipeline for modeling real-world objects (mainly buildings) from images
  - Data acquisition
  - Processing
  - Reconstruction

# Topics/Lectures

- 1,2: Introduction & Linear algebra
- 3,4: Camera models
- 5,6: Camera calibration
- 7,8: Epipolar geometry
- 9,10: Triangulation & structure from motion
- 11,12: Image matching
- 13,14: Multi-view stereo [by Nail]
- 15,16: Surface reconstruction

# Learning activities

- Lectures
  - Mostly 2 hours per week
- Group work and lab sessions
  - Mostly 2 x 2-hour sessions per week
  - Teachers available at lab sessions
    - Data acquisition
    - Install/Use software
    - Programming
    - Debugging
    - Discussion with teammates and/or teachers

# Assessment

- 3 group assignments (40 %)
  - Group performance
  - Personal contribution/Peer reviews
- Final exam (60%):
  - Types of questions
    - Multiple-choice questions with **a single** correct answer
    - Multiple-choice questions with **at least 2** correct answers
    - Open questions
  - Lectures, handouts, assignments, lab exercises

# Assessment

- Pass (on a 100-point scale)?
  - Assignments  $\geq 55\%$
  - Exam  $\geq 55\%$
  - Total  $\geq 57.5\%$
- Repair an assignment: if lower than 55%
  - **Only one chance**
  - Only be assessed with a 57.5 %
- Retake the exam: if lower than 55%
  - **Only one chance**
  - An entirely new examination

# Assignments

- Three mandatory assignments, each released after the lecture
  - Camera calibration
  - Triangulation
  - Reconstruction
- Programming: implementation and experiment 3DV algorithms

# Assignments

- C++ source code framework provided
- If you fail one of the assignments?
  - Depending code and data will be provided
- Work in groups (3 students per group)

# Assignments

- What to submit?
  - Report
    - Max 3 pages (excluding figures, tables, references)
    - Individual contribution

## Isaac Newton (75 %)

- Compared the reconstruction results from method [1] and method [2];
- Implemented the function `reorient_normals()`;
- Came up with a novel reconstruction method and implemented it in function `reconstruct()`;
- Wrote the “Methodology” section of the report.

## Albert Einstein (20 %)

- Preparing and pre-processing of the point clouds, i.e., taking photos, run SfM and MVS, cropping the buildings from the messy point clouds, and normal estimation;
- Wrote the “Implementation Details” section of the report.

## Thomas Edison (5 %)

- Wrote the “Abstract” section of the report.

- See [an example](#) on the Course Webpage



# Assignments

- What to submit?
  - Report
  - Code
    - Highly recommended: Collaboration using GitHub
    - [optional] Include the link to the GitHub repository in the report
    - Reproduce the results
      - **Doesn't compile:** -10%
      - **Doesn't reproduce the result:** -40%

# Assignments

- What to submit?
- We allow multiple submissions
  - Incorporating comments from teachers/peers
  - Evaluation based on 1<sup>st</sup> submission + 5% maximum

Example:

First submission 60%, then final mark will be  $\leq 65\%$


# Assignments

- What to submit?
- We allow multiple submissions
- Strict deadlines
  - Late submission
    - 10% deducted per day late
    - Not acceptable after 3 days late

# Assignments

- What to submit?
- We allow multiple submissions
- Strict deadline
- **Teamwork**
  - **Everyone active in coding/discussion/report**
  - **We strongly discourage**
    - report writing to one person and code writing to another
    - one person working on course A and another on course B
    - strategically provide perfectly equal individual contributions

# Assignments



- 
- Copy from others/internet, or use AI (e.g., ChatGPT, DeepSeek)
    - Code
    - Sentences
    - Figures
    - ...
  - Submit to BrightSpace [plagiarism check turned on]

# Assignments: start earlier




# Communication method

- Course website
  - <https://3d.bk.tudelft.nl/courses/geo1016/>



Photogrammetry and 3D Computer Vision - Course year 2025

Home News Schedule Lectures Assignments Resources Discussion



- Feb. 5. The first lecture/meeting will be on Feb. 12. Check out more details on the [course calendar](#).
- Jan. 10. The course website for the 2025 year is online.
- [All news...](#)

### About

Photogrammetry and 3D Computer Vision (**3DV**) aim at recovering the structure of real-world objects/scenes from images. This course is about the theories, methodologies, and techniques of 3D computer vision for the built environment. In the term of this course, you will learn the basic knowledge and algorithms in 3D computer vision through a series of lectures, reading materials, lab exercises, and group assignments.

### Contents

The topics of this course cover the whole pipeline of reconstructing 3D models from images:

- Cameras models and camera calibration: how a point from the real world gets projected onto the image plane and how to recover the camera parameters from a set of observations;
- Epipolar geometry: the geometric relations between 3D points and their image points; the constraints between the image points;
- Image matching: define and match image features (SIFT) to establish correspondences between images;
- Structure from motion: recover/refine geometry and camera parameters from a set of images;
- Multi-view stereo for recovering dense geometry (e.g., point clouds) from images;
- Surface reconstruction: obtain 3D surface models of real-world objects from point clouds.

### Goals

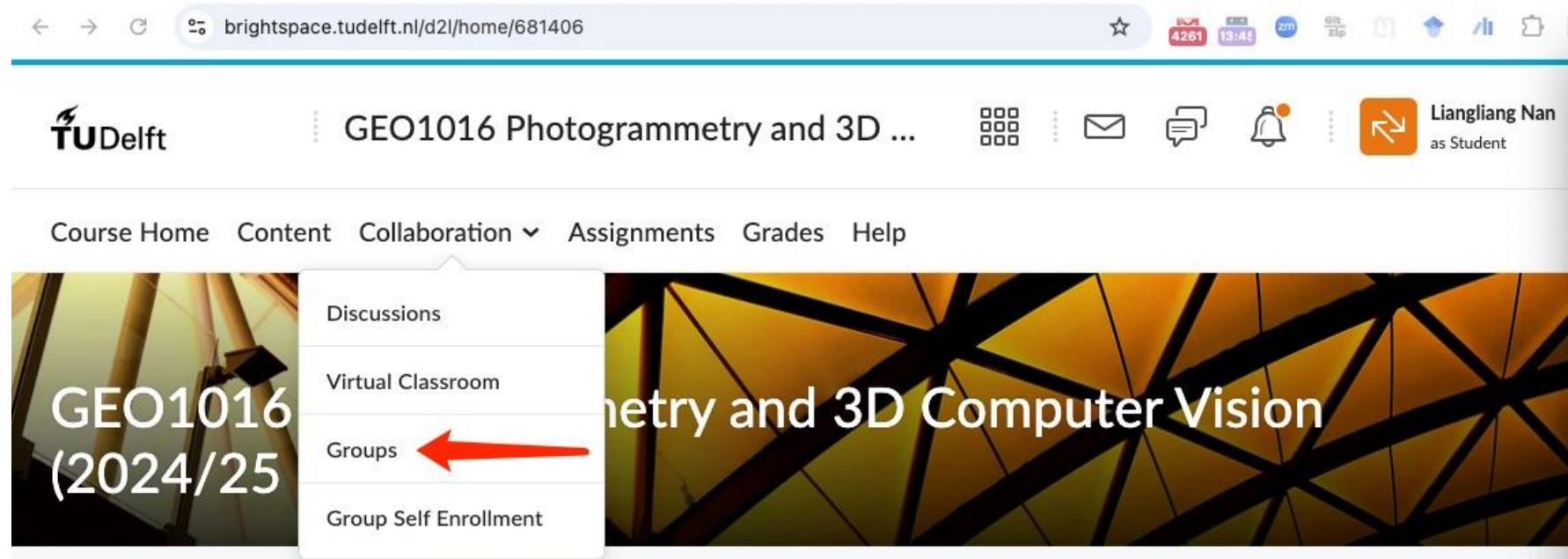
# Communication method

- Course website
  - <https://3d.bk.tudelft.nl/courses/geo1016/>
- Discussion
  - Lab/Lecture hours
  - Discord channel: <https://discord.gg/zmaZDNkjjH>



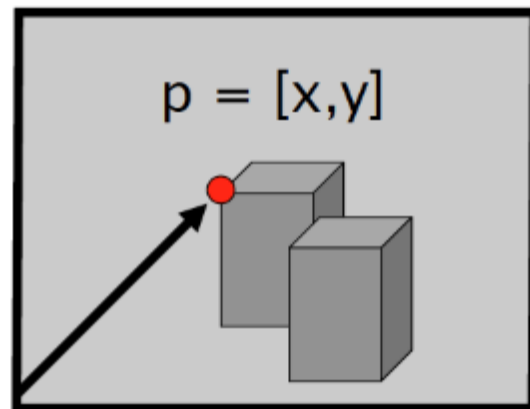
# Work in groups

- Find your teammates ...
  - 3 students per team/group
  - Form your team on BrightSpace:

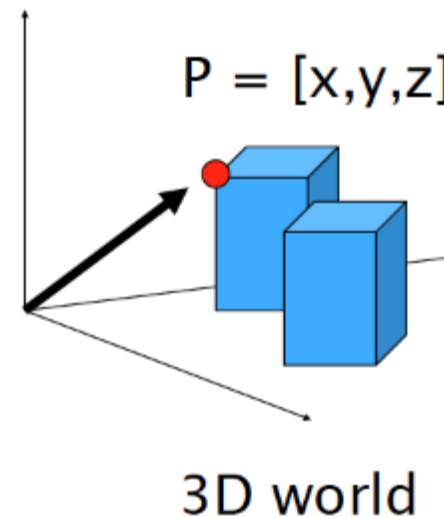


# Review linear algebra

# Vectors (i.e., 2D and 3D vectors)



Image

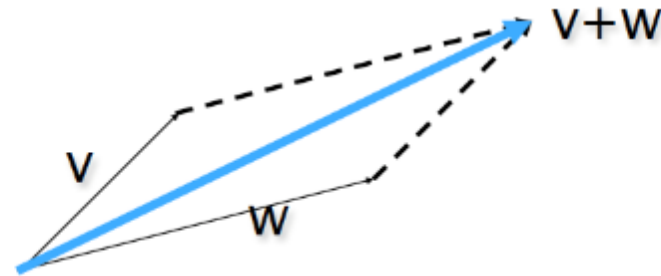


3D world

# Vector arithmetic

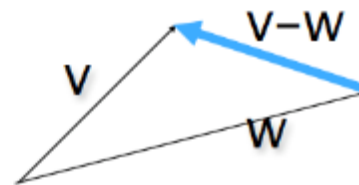
- Addition

$$\mathbf{v} + \mathbf{w} = (x_1, x_2) + (y_1, y_2) = (x_1 + y_1, x_2 + y_2)$$



- Subtraction

$$\mathbf{v} - \mathbf{w} = (x_1, x_2) - (y_1, y_2) = (x_1 - y_1, x_2 - y_2)$$



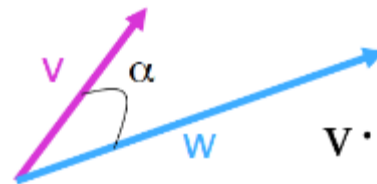
# Vector arithmetic

- Scalar product

$$a\mathbf{v} = a(x_1, x_2) = (ax_1, ax_2)$$



- Dot (inner) product



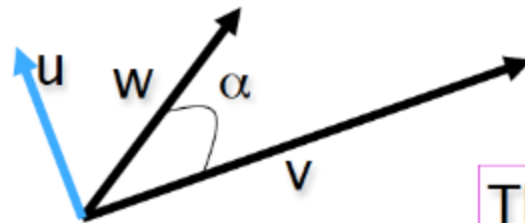
$$\mathbf{v} \cdot \mathbf{w} = (x_1, x_2) \cdot (y_1, y_2) = x_1y_1 + x_2y_2$$

The inner product is a **SCALAR!**

$$\mathbf{v} \cdot \mathbf{w} = (x_1, x_2) \cdot (y_1, y_2) = \|\mathbf{v}\| \cdot \|\mathbf{w}\| \cos\alpha$$

# Vector arithmetic

- Cross (vector) product

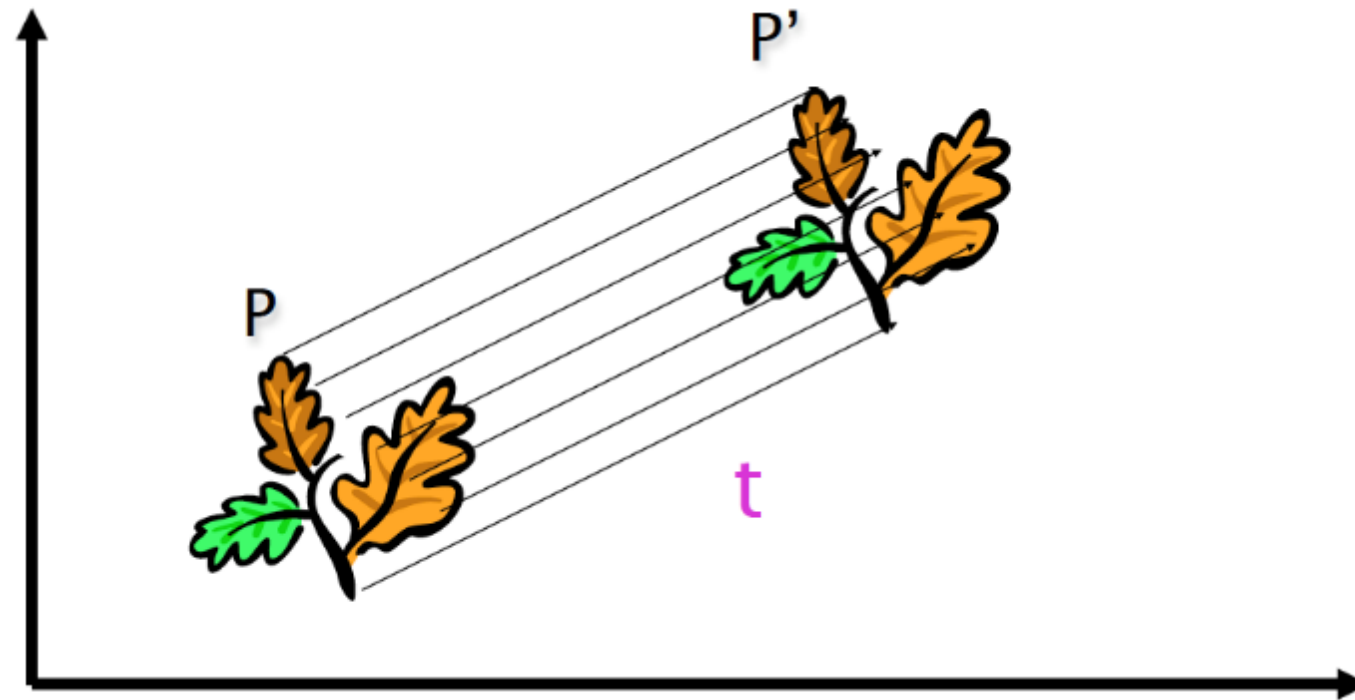


$$u = v \times w$$

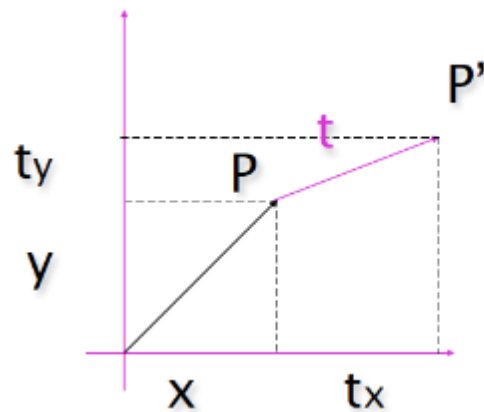
The cross product is a **VECTOR!**

Magnitude:  $\|u\| = \|v \times w\| = \|v\| \|w\| \sin \alpha$

# Translation



# Translation



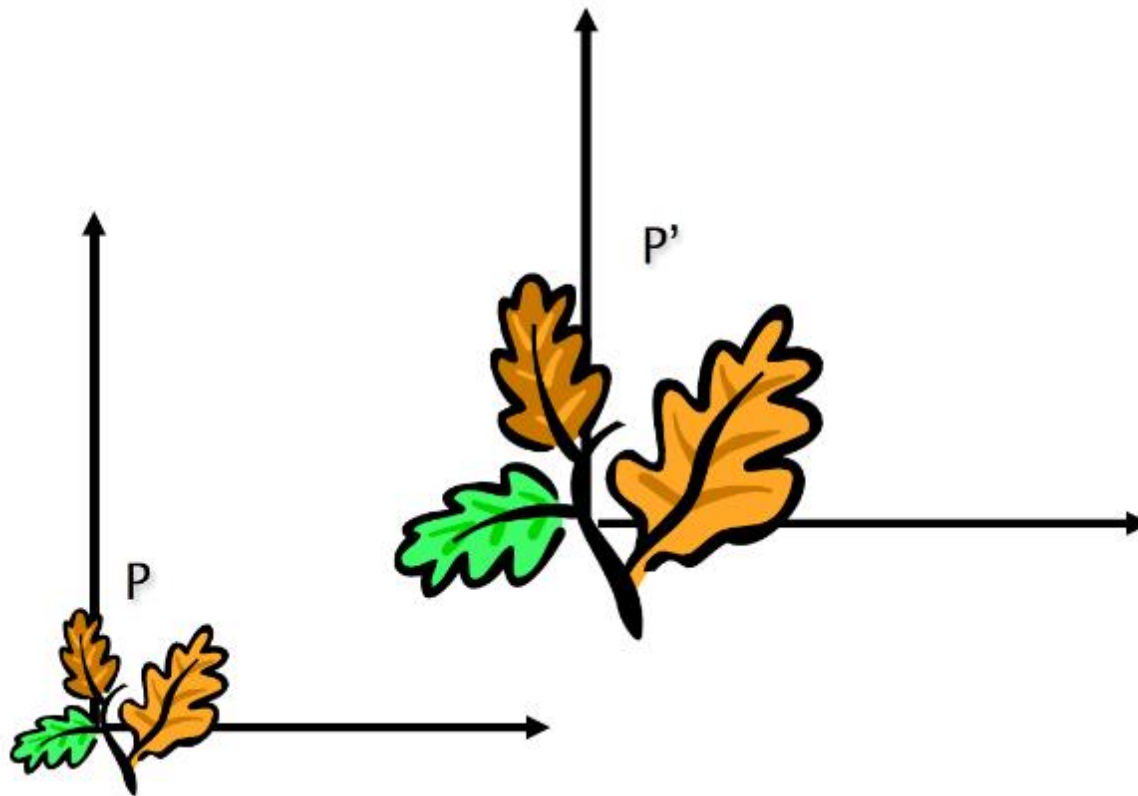
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{t} = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

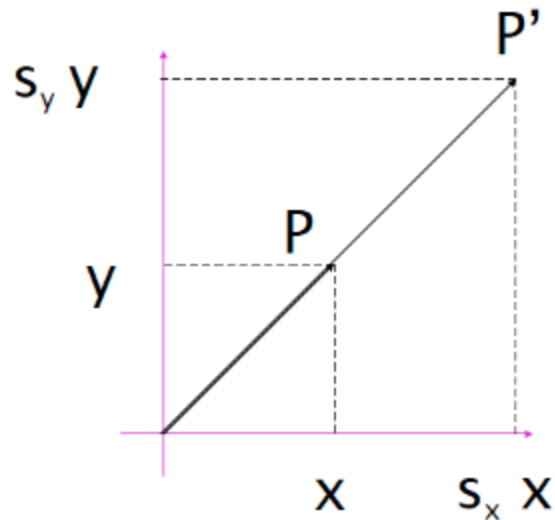
$$\begin{aligned} \mathbf{P}' &\rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \cdot \mathbf{P} = \mathbf{T} \cdot \mathbf{P} \end{aligned}$$



# Scaling



# Scaling



$$\mathbf{P} = (x, y) \rightarrow \mathbf{P}' = (s_x x, s_y y)$$

$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

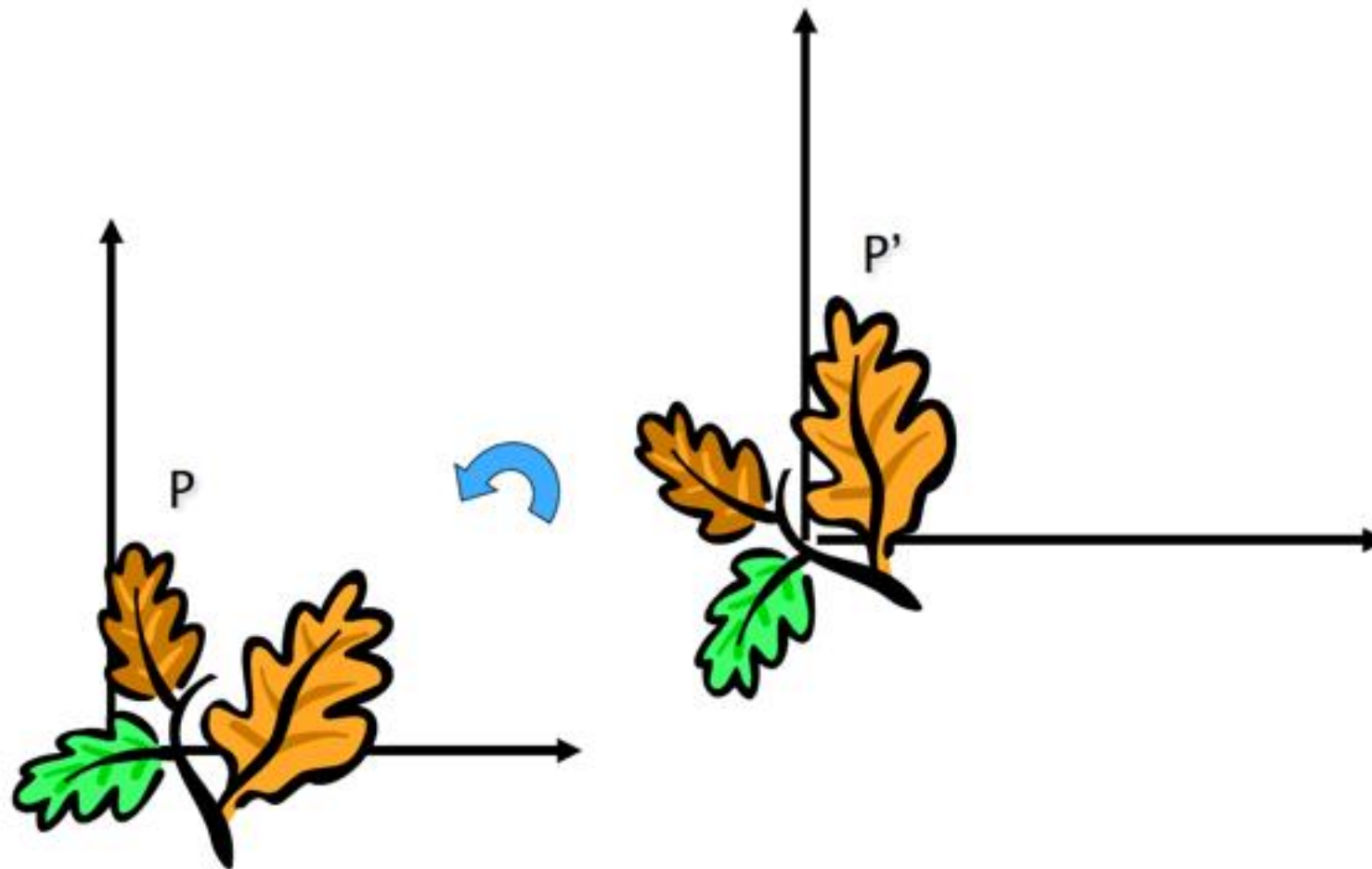
$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}' & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \mathbf{P} = \mathbf{S} \cdot \mathbf{P}$$

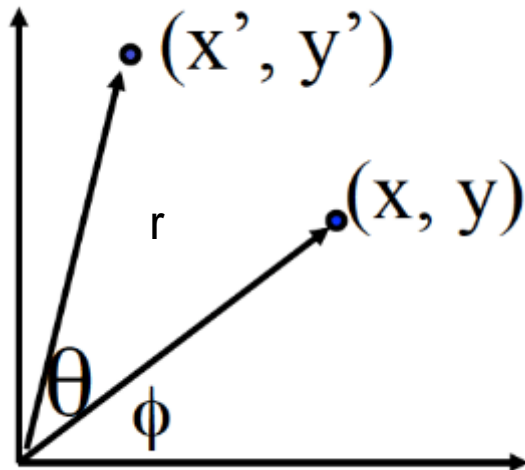
# Scaling & Translation

$$\begin{aligned}\mathbf{P}'' &= \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \\ &= \underbrace{\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}}_A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\end{aligned}$$

# Rotation



# Rotation

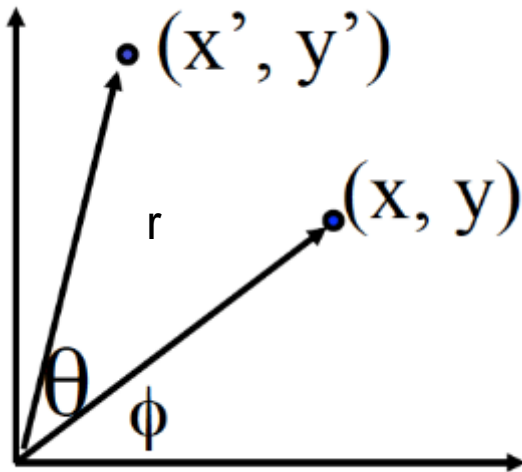


$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} x \\ y \end{bmatrix}$$

What is the inverse transformation



# Rotation



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} x \\ y \end{bmatrix}$$

What is the inverse transformation

- Rotation by  $-\theta$

$\mathbf{R}$  has many interesting properties:

$$\mathbf{R}^{-1} = \mathbf{R}^T \quad \mathbf{R} \cdot \mathbf{R}^T = \mathbf{R}^T \cdot \mathbf{R} = \mathbf{I} \quad \det(\mathbf{R}) = 1$$

# Translation + Rotation + Scaling

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \mathbf{R}' & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{S} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \boxed{\begin{bmatrix} \mathbf{R}' \mathbf{S} & \mathbf{t} \\ 0 & 1 \end{bmatrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

If  $s_x = s_y$ , this is a similarity transformation

# Next lecture

- Camera models

$$\mathbf{p} = M\mathbf{P}$$
$$= \boxed{K} \boxed{\begin{bmatrix} R & \mathbf{t} \end{bmatrix}} \mathbf{P}$$

Internal (intrinsic) parameters

External (extrinsic) parameters