Visual Odometry

Morten R. Hannemose, mohan@dtu.dk

April 25, 2025

02504 Computer vision course lectures,
DTU Compute, Kgs. Lyngby 2800, Denmark

This lecture is being livestreamed and recorded

(hopefully)

Two feedback persons

Learning objectives

After this lecture you should be able to:

- choose the correct decomposition of the essential matrix
- explain why the scale of the translation is unknown
- explain the Perspective-n-Point problem
- implement a simple visual odometry algorithm

Presentation topics

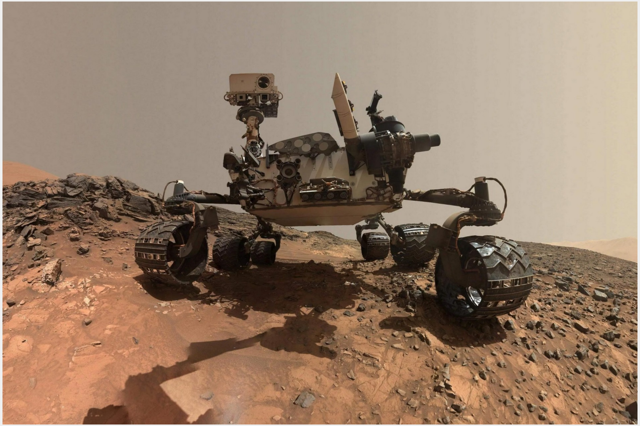
Decomposing the Essential Matrix

Perspective-n-Point

Putting it all together

Motivation

Let's say you're a rover on Mars...



Motivation

SLAM answers the questions:

- How is this camera moving through the world?
- What is the shape of the world?

Many applications:

- Drones
- Robotic vacuum cleaners
- Virtual reality headsets
- Augmented reality
- Autonomous cars

Many similar and related concepts

- Visual Odometry
- SLAM (Simultaneous Localization and Mapping)
- SfM (Structure from motion)

They all deal with some form of estimating a 3D map of the world and camera poses, but have emphasis on different parts.

Multiple “cameras”

The unknown scale of t – Mathematical argument

$$\boldsymbol{E} = \boldsymbol{R}[\boldsymbol{t}]_\times$$
$$0 = \boldsymbol{E}(s\boldsymbol{t}) = \boldsymbol{R}[\boldsymbol{t}]_\times(s\boldsymbol{t})$$

We can see that $\boldsymbol{t}$ lies in the null space of $\boldsymbol{E}$ but also that it can be arbitrarily scaled

The unknown scale of t – Conceptual reason

Decomposing the Essential Matrix

Essential matrix

You have matched features between two cameras, and want to make it robust.

Estimate $\boldsymbol{F}$ or $\boldsymbol{E}$ with RANSAC.

Estimating E

- How many points are required?
- Ask yourself:
  - How many degrees of freedom does it have?
  - How many degrees of freedom does a single point fix?

Estimating E

- How many points are required?
- Ask yourself:
  - How many degrees of freedom does it have?
  - How many degrees of freedom does a single point fix?
- Five.
- Not possible with linear algorithm from five points.
- Typically estimated using Nister's five-point algorithm
  - Involves solving tenth degree polynomial
  - Is implemented in OpenCV.

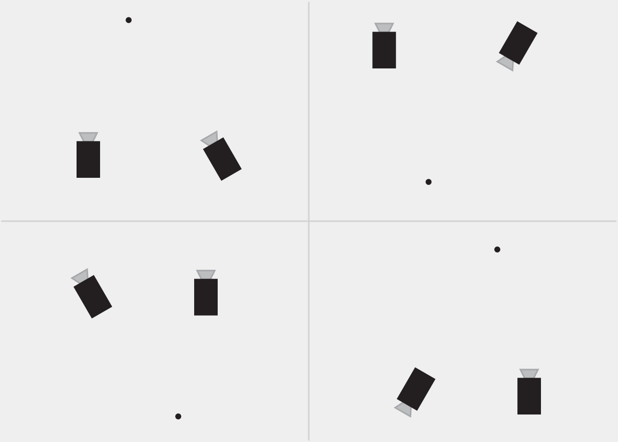
Decomposing the Essential Matrix

- The essential matrix can be computed from $\boldsymbol{R}$ and $\boldsymbol{t}$.
  - Can we recover them from $\boldsymbol{E}$?

Decomposing the Essential Matrix

- The essential matrix can be computed from $\boldsymbol{R}$ and $\boldsymbol{t}$.
  - Can we recover them from $\boldsymbol{E}$?
- Decomposing the essential matrix is ill posed
  - Two possible rotations
  - The sign of the translation is unknown.
- A total of four possible poses for the second camera
  - Check all four combinations
  - Choose the one with the most points in front of both cameras.

Decomposing the Essential Matrix



Back to visual odometry

We can find the pose of two cameras relative to each other

- How can we estimate the pose of a third camera?
- Using the essential matrix again will give us a new arbitrarily scaled translation

Back to visual odometry

We can find the pose of two cameras relative to each other

- How can we estimate the pose of a third camera?
- Using the essential matrix again will give us a new arbitrarily scaled translation
- Idea: Use the translation between the first two cameras to fix the scale.
- Triangulate points using the first two cameras
  - Use these 3D points to find the pose of the third camera

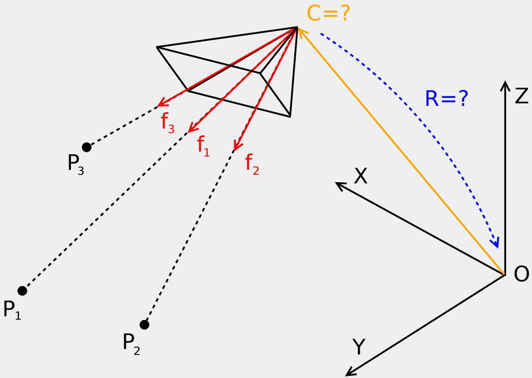
Short break

Perspective-n-Point

Perspective-n-Point (PnP)

The Perspective-n-Point (PnP) problem.

Estimating the pose of a calibrated camera from n corresponding 3D-2D correspondences.



Kneip, Laurent, Davide Scaramuzza, and Roland Siegwart. "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation." CVPR 2011. IEEE, 2011.

PnP vs camera resectioning

In week 4 you did this for an uncalibrated camera (pest).

For an uncalibrated camera it is also called camera resectioning.

Naïve solution

- Estimate the projection matrix ($\tilde{\boldsymbol{P}}$)
- Compute $\boldsymbol{K}^{-1}\tilde{\boldsymbol{P}}$
- $\boldsymbol{K}^{-1}\tilde{\boldsymbol{P}} \approx \begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\end{bmatrix}$
    - $\boldsymbol{R}$ is likely not a proper rotation matrix
    - Requires many points

Perspective-n-Point (PnP)

- How many points are required?
- Ask yourself:
  - How many degrees of freedom does it have?
  - How many degrees of freedom does a single point fix?

Perspective-n-Point (PnP)

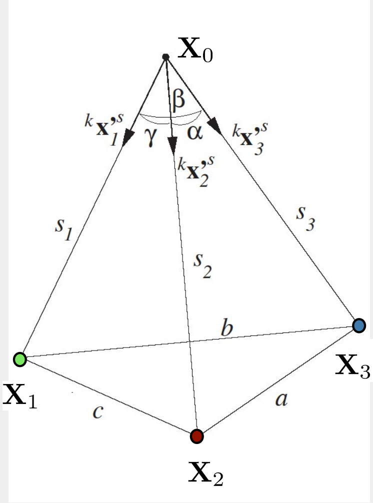
- How many points are required?
- Ask yourself:
  - How many degrees of freedom does it have?
  - How many degrees of freedom does a single point fix?
- Three correspondences are required
- This minimal case is therefore also known as P3P

P3P – Geometry

- The three 2D points give three pairwise angles
- The distances between the three 3D points give three pairwise distances

P3P – Geometry

- The three 2D points give three pairwise angles
- The distances between the three 3D points give three pairwise distances



From slides by Cyrill Stachniss

PnP in summary

- Three correspondences generate four possible solutions, and a fourth correspondence can be used to choose the correct one
    - PnP requires 3+1 correspondences.
- Multiple algorithms exist and are implemented in OpenCV
- Add RANSAC to make it robust.

Pose vs. position

The pose of a camera is given by $\boldsymbol{R}$ and $\boldsymbol{t}$

This is not the orientation and position of the camera

$$\boldsymbol{T}_{\text{world}\rightarrow\text{cam}} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

$$\boldsymbol{T}_{\text{cam}\rightarrow\text{world}} = \boldsymbol{T}^{-1}_{\text{world}\rightarrow\text{cam}} = \begin{bmatrix} \boldsymbol{R}^\mathsf{T} & -\boldsymbol{R}^\mathsf{T}\boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

Pose vs. position

The orientation of a camera is given by $\boldsymbol{R}^\top$

The position of a camera is given by $-\boldsymbol{R}^\top t$

This is important in order to plot the camera

Putting it all together

Outline

1. Use the essential or fundamental matrix to estimate the pose of the second camera
2. Triangulate points using the known camera poses
3. Use PnP or camera resectioning to estimate the pose of the next camera
4. Repeat steps 2. and 3.

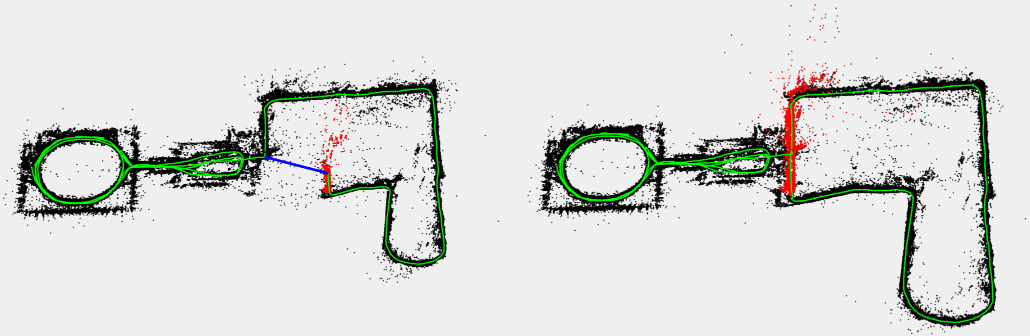
Outline

1. Use the essential or fundamental matrix to estimate the pose of the second camera
2. Triangulate points using the known camera poses
3. Use PnP or camera resectioning to estimate the pose of the next camera
4. Repeat steps 2. and 3.
5. Use (windowed) bundle adjustment

Feature tracking

- Some points can be tracked through many frames
- Keep track of them to make your model drift less

Loop closure



The exercise

- Bigger exercise (two weeks)
- Estimate essential matrix
- Estimate 3D points
- SolvePnP

Learning objectives

After this lecture you should be able to:

- choose the correct decomposition of the essential matrix
- explain why the scale of the translation is unknown
- explain the Perspective-n-Point problem
- implement a simple visual odometry algorithm

Next week

Next week: Today's exercise is bigger than usual, so you have two weeks to complete it.

You can also spend next week catching up on previous exercises.

Exercise time!