Photogrammetry & Robotics Lab

Projective 3-Point (P3P) Algorithm / Spatial Resection

Cyrill Stachniss
5 Minute Preparation for Today

5 Minutes with Cyrill
Projective 3 Point

https://www.ipb.uni-bonn.de/5min/
Camera Localization

Given known 3D control points (X, Y, Z)

**Task:** estimate the pose of the camera
Camera Localization

Given:
- 3D coordinates of object points $X_i$

Observed:
- 2D image coordinates $x_i$ of the object points

Wanted:
- Extrinsic parameters $R$, $X_O$ of the calibrated camera
Reminder: Mapping Model

Direct linear transform (DLT) maps any object point $X$ to the image point $x$

$$
x = KR[I_3 - X_O]X
= P X
$$
Reminder: Camera Orientation

\[ x = KR[l_3] - X_O \] \[ X = P \cdot X \]

- **Intrinsics (interior orientation)**
  - Intrinsic parameters of the camera
  - Given through matrix \( K \)

- **Extrinsics (exterior orientation)**
  - Extrinsic parameters of the camera
  - Given through \( X_O \) and \( R \)
Direct Linear Transform (DLT)

Relation to DLT: Compute the 11 intrinsic and extrinsic parameters

\[ x = KR\begin{bmatrix} I_3 \mid - X_O \end{bmatrix}X \]

- **Observed image point**
- **Control point coordinates (given)**
- **3 rotations**
- **3 translations**
- **c, s, m, x_H, y_H**
Projective 3-Point Algorithm (or Spatial Resection)

Given the intrinsic parameters, compute the 6 extrinsic parameters

\[ x = KR[I_3 | - X_O]X \]

- 3 rotations
- 3 translations
- Control point coordinates (given)
- Observed image points
- \( c, s, m, x_H, y_H \) (given)
P3P/SR vs. DLT

- **P3P/SR: Calibrated camera**
  - 6 unknowns
  - We need at least 3 points

- **DLT: Uncalibrated camera**
  - 11 unknowns
  - We need at least 6 points
  - Assuming an affine camera (straight-line preserving projection)
Orienting a calibrated camera by using \( \geq 3 \) points

P3P/Spatial Resection (direct solution)
Problem Formulation

Given:
- 3D coordinates $X_i$ of $I \geq 3$ object points
- Corresponding image coordinates $x_i$ recorded using a calibrated camera

Task:
- Estimate the 6 parameters $X_O, R$
- Direct solution (no initial guess)
Different Approaches

- Different approaches: Grunert 1841, Killian 1955, Rohrberg 2009, ...
- Here: direct solution by Grunert
- **2-step process**
  1. Estimate length of projection rays
  2. Estimate the orientation
Direct Solution by Grunert

2-Step process

Estimate
1. length of projection rays
2. orientation
P3P/SR Model

- Coordinates of object points within the camera system are given by

\[ s_i^k x_i^s = R(X_i - X_O) \quad i = 1, 2, 3 \]

ray directions pointing to the object points
P3P/SR Model

- Coordinates of object points within the camera system are given by
  \[ s_i k x_i^s = R(X_i - X_O) \quad i = 1, 2, 3 \]

- From image coordinates, we obtain the directional vector of projection ray
  \[ k x_i^s = -\text{sign}(c)N(K^{-1}x_i) \]

  ensure ray directions are pointing to the object points
P3P/SR Model

- Coordinates of object points within the camera system are given by

\[ s_i^k x_i^s = R(X_i - X_O) \quad i = 1, 2, 3 \]

- From image coordinates, we obtain the directional vector of projection ray

\[ k x_i^s = -\text{sign}(c)N(K^{-1}x_i) \]

spherical normalization

\[ N(x) = \frac{x}{|x|} \]
1. Get Length of Projection Rays
1. Get Length of Projection Rays

- Start with computing the angle between rays:

\[
\cos \gamma = \frac{(X_1 - X_0) \cdot (X_2 - X_0)}{||X_1 - X_0|| \cdot ||X_2 - X_0||}
\]
1. Get Length of Projection Rays

\[
\alpha = \arccos \left( \frac{k_x}{s}, \frac{k_x}{s} \right)
\]

\[
\beta = \arccos \left( \frac{k_x}{s}, \frac{k_x}{s} \right)
\]

\[
\gamma = \arccos \left( \frac{k_x}{s}, \frac{k_x}{s} \right)
\]

Angles are directly computable by observing the image points and known intrinsics.
1. Get Length of Projection Rays

\[ a = \|X_3 - X_2\| \]
\[ b = \|X_1 - X_3\| \]
\[ c = \|X_2 - X_1\| \]

Given through known control point coordinates
Use the Law of Cosines

In triangle $X_0, X_1, X_2$

\[ s_1^2 + s_2^2 - 2s_1s_2 \cos \gamma = c^2 \]

wanted \hspace{1cm} known
Use the Law of Cosines

Analogously in all three triangles

\[ a^2 = s_2^2 + s_3^2 - 2s_2 s_3 \cos \alpha \]

\[ b^2 = s_1^2 + s_3^2 - 2s_1 s_3 \cos \beta \]

\[ c^2 = s_1^2 + s_2^2 - 2s_1 s_2 \cos \gamma \]
Compute Distances

- We start from:
  \[ a^2 = s_2^2 + s_3^2 - 2s_2s_3 \cos \alpha \]

- Define:
  \[ u = \frac{s_2}{s_1}, \quad v = \frac{s_3}{s_1} \]

- Substitution leads to:
  \[ a^2 = s_1^2(u^2 + v^2 - 2uv \cos \alpha) \]

- Rearrange to:
  \[ s_1^2 = \frac{a^2}{u^2 + v^2 - 2uv \cos \alpha} \]
Compute Distances

- Use the same definition
  \[ u = \frac{s_2}{s_1} \quad v = \frac{s_3}{s_1} \]

- And perform the substitution again for:
  \[ b^2 = s_1^2 + s_3^2 - 2s_1 s_3 \cos \beta \]
  \[ c^2 = s_1^2 + s_2^2 - 2s_1 s_2 \cos \gamma \]
Compute Distances

Analogously, we obtain

$$s_1^2 = \frac{a^2}{u^2 + v^2 - 2uv \cos \alpha}$$

$$= \frac{b^2}{1 + v^2 - 2v \cos \beta}$$

$$= \frac{c^2}{1 + u^2 - 2u \cos \gamma}$$
Rearrange Again

Solve one equation for $u$ put into the other

\[ s_1^2 = \frac{a^2}{u^2 + v^2 - 2uv \cos \alpha} \]
\[ s_1^2 = \frac{b^2}{1 + v^2 - 2v \cos \beta} \]
\[ s_1^2 = \frac{c^2}{1 + u^2 - 2u \cos \gamma} \]

Results in a fourth degree polynomial

\[ A_4 v^4 + A_3 v^3 + A_2 v^2 + A_1 v + A_0 = 0 \]
Forth Degree Polynomial

\[ A_4 v^4 + A_3 v^3 + A_2 v^2 + A_1 v + A_0 = 0 \]

\[ A_4 = \left( \frac{a^2 - c^2}{b^2} - 1 \right)^2 - \frac{4c^2}{b^2} \cos^2 \alpha \]

\[ A_3 = 4 \left[ \frac{a^2 - c^2}{b^2} \left( 1 - \frac{a^2 - c^2}{b^2} \right) \cos \beta \right. \]

\[ \left. - \left( 1 - \frac{a^2 + c^2}{b^2} \right) \cos \alpha \cos \gamma + 2 \frac{c^2}{b^2} \cos^2 \alpha \cos \beta \right] \]
Forth Degree Polynomial

\[ A_4v^4 + A_3v^3 + A_2v^2 + A_1v + A_0 = 0 \]

\[
A_2 = 2 \left[ \left( \frac{a^2 - c^2}{b^2} \right)^2 - 1 + 2 \left( \frac{a^2 - c^2}{b^2} \right)^2 \cos^2 \beta \right.

\[ + 2 \left( \frac{b^2 - c^2}{b^2} \right) \cos^2 \alpha \]

\[ - 4 \left( \frac{a^2 + c^2}{b^2} \right) \cos \alpha \cos \beta \cos \gamma \]

\[ + 2 \left( \frac{b^2 - a^2}{b^2} \right) \cos^2 \gamma \right] \]
Forth Degree Polynomial

\[ A_4 v^4 + A_3 v^3 + A_2 v^2 + [A_1 v + [A_0] = 0 \]

\[
A_1 = 4 \left[ -\left( \frac{a^2 - c^2}{b^2} \right) \left( 1 + \frac{a^2 - c^2}{b^2} \right) \cos \beta \\
+ \frac{2a^2}{b^2} \cos^2 \gamma \cos \beta \\
- \left( 1 - \left( \frac{a^2 + c^2}{b^2} \right) \right) \cos \alpha \cos \gamma \right]
\]

\[
A_0 = \left( 1 + \frac{a^2 - c^2}{b^2} \right)^2 - \frac{4a^2}{b^2} \cos^2 \gamma
\]
Forth Degree Polynomial

\[ A_4 v^4 + A_3 v^3 + A_2 v^2 + A_1 v + A_0 = 0 \]

Solve for \( v \) to get \( s_1, s_2, s_3 \) through:

\[
\begin{align*}
s_1^2 &= \frac{b^2}{1+v^2-2v \cos \beta} \\
s_3 &= v \ s_1
\end{align*}
\]

\[
a^2 = s_2^2 + s_3^2 - 2s_2 s_3 \cos \alpha \Rightarrow s_2 = \ldots
\]

**Problem:**
up to 4 possible solutions!

\[ \{s_1, s_2, s_3\}_{1\ldots4} \]
Example for Multiple Solutions

- Assume $a = b = c$ and $\alpha = \beta = \gamma$
- Tilting the triangle $(X_1, X_2, X_3)$ has no effect on $(a, b, c)$ and $(\alpha, \beta, \gamma)$
Four Solutions
How to Eliminate This Ambiguity?

- Known approximate solution (e.g., from GPS) or
- Use 4th points to confirm identify the correct solution

Unique solution for

\[ s_1, s_2, s_3 \]
2. Orientation of the Camera

Given:
- Distances and direction vectors to the control points

Task:
- Estimate 6 extrinsic parameters
2. Orientation of the Camera

1. Compute 3D coordinates of the control points in the camera system

\[ k \mathbf{X}_i = s_i \ k \mathbf{x}_i^s \quad i = 1, 2, 3 \]

That’s what we just discussed!
2. Orientation of the Camera

1. Compute 3D coordinates of the control points in the camera system

\[ k \mathbf{X}_i = s_i k \mathbf{x}_i^s \quad i = 1, 2, 3 \]

2. Compute coordinate transformation for

\[ k \mathbf{X}_i = R(\mathbf{X}_i - \mathbf{X}_O) \]
P3P/SR in a Nutshell

\[ X_i, x_i, K \quad i = 1, 2, 3, 4 \]

Compute angles

\[ \alpha, \beta, \gamma \]

Computes distances

\[ \{s_1, s_2, s_3\}_{1\ldots4} \]

Identify correct solution through 4\textsuperscript{th} point

\[ s_1, s_2, s_3 \]

Compute coordinate transformation

\[ R, X_0 \]
Critical Surfaces

“Critical cylinder”

- If the projection center lies on a cylinder defined by the control points
- Small changes in angles lead to large changes in coordinates
- Instable solution
Outlier Handling with RANSAC

Use direct solution to find correct solution among set of corrupted points

- Assume I ≥ 3 points
  
  1. Select 3 points randomly
  2. Estimate parameters of SR/P3P
  3. Count the number of other points that support current hypotheses
  4. Select best solution

- Can deal with large numbers of outliers in data
More Recent Solutions

- Further solutions gave been proposed after Grunert’s solutions of 1841
- New methods still have ambiguities when using 3 control points only
- $4^{th}$ point needed for disambiguation
- Faster to compute
- Numerically more stable
- Partially less complex
Recent Approaches

Complete Solution Classification for the Perspective-Three-Point Problem

Xiao-Chuan Gao, Blinder,  ECE, Xiao-Hong Hou, Jiannong Tang, and Hang Fei Cheng

Abstract

The Perspective-Three-Point (PTP) problem arises in many applications, including computer vision, robotics, and computer graphics. It involves determining the position and orientation of a camera from three known points in the scene. This problem is known to have up to four solutions, each of which corresponds to a different configuration of the camera and the points.

A novel parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation

Laurence Kneip

Abstract

The Perspective-Three-Point (PTP) problem arises in many applications, including computer vision, robotics, and computer graphics. It involves determining the position and orientation of a camera from three known points in the scene. This problem is known to have up to four solutions, each of which corresponds to a different configuration of the camera and the points.

A novel parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation

Laurence Kneip
Recent Approaches

Lambda Twist: An Accurate Fast Robust Perspective Three Point (P3P) Solver.
Mikael Persson 2019 | 0000-0003-1593-3754 | and Klas Nordberg 1

Abstract: We present Lambda Twist, a novel P3P solver which is accurate, fast and robust. Current state-of-the-art P3P solvers that allow non-linear solutions are also very accurate, at a cost of being slow and unstable. Lambda Twist is an algorithm that solves the Perspective Three-Point Problem (P3P) with a runtime of $O(N)$, where $N$ is the number of points. The key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.

Keywords: P3P, Visual Odometry, Camera Geometry

1 Introduction

We are concerned with the accuracy and consistency of odometry estimation on low-power hardware and systems. Since both odometry and localization errors independently have a large impact on the accuracy, we have a consistent solution to the P3P problem (Figure 1).

For odometry, the key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.

1.1 Introduction

Lambda Twist: An Accurate Fast Robust Perspective Three Point (P3P) Solver.
Mikael Persson 2019 | 0000-0003-1593-3754 | and Klas Nordberg 1

Abstract: We present Lambda Twist, a novel P3P solver which is accurate, fast and robust. Current state-of-the-art P3P solvers that allow non-linear solutions are also very accurate, at a cost of being slow and unstable. Lambda Twist is an algorithm that solves the Perspective Three-Point Problem (P3P) with a runtime of $O(N)$, where $N$ is the number of points. The key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.

Keywords: P3P, Visual Odometry, Camera Geometry

1 Introduction

We are concerned with the accuracy and consistency of odometry estimation on low-power hardware and systems. Since both odometry and localization errors independently have a large impact on the accuracy, we have a consistent solution to the P3P problem (Figure 1).

For odometry, the key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.

1.1 Introduction

Lambda Twist: An Accurate Fast Robust Perspective Three Point (P3P) Solver.
Mikael Persson 2019 | 0000-0003-1593-3754 | and Klas Nordberg 1

Abstract: We present Lambda Twist, a novel P3P solver which is accurate, fast and robust. Current state-of-the-art P3P solvers that allow non-linear solutions are also very accurate, at a cost of being slow and unstable. Lambda Twist is an algorithm that solves the Perspective Three-Point Problem (P3P) with a runtime of $O(N)$, where $N$ is the number of points. The key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.

Keywords: P3P, Visual Odometry, Camera Geometry

1 Introduction

We are concerned with the accuracy and consistency of odometry estimation on low-power hardware and systems. Since both odometry and localization errors independently have a large impact on the accuracy, we have a consistent solution to the P3P problem (Figure 1).

For odometry, the key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.

1.1 Introduction

Lambda Twist: An Accurate Fast Robust Perspective Three Point (P3P) Solver.
Mikael Persson 2019 | 0000-0003-1593-3754 | and Klas Nordberg 1

Abstract: We present Lambda Twist, a novel P3P solver which is accurate, fast and robust. Current state-of-the-art P3P solvers that allow non-linear solutions are also very accurate, at a cost of being slow and unstable. Lambda Twist is an algorithm that solves the Perspective Three-Point Problem (P3P) with a runtime of $O(N)$, where $N$ is the number of points. The key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.

Keywords: P3P, Visual Odometry, Camera Geometry

1 Introduction

We are concerned with the accuracy and consistency of odometry estimation on low-power hardware and systems. Since both odometry and localization errors independently have a large impact on the accuracy, we have a consistent solution to the P3P problem (Figure 1).

For odometry, the key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.

1.1 Introduction

Lambda Twist: An Accurate Fast Robust Perspective Three Point (P3P) Solver.
Mikael Persson 2019 | 0000-0003-1593-3754 | and Klas Nordberg 1

Abstract: We present Lambda Twist, a novel P3P solver which is accurate, fast and robust. Current state-of-the-art P3P solvers that allow non-linear solutions are also very accurate, at a cost of being slow and unstable. Lambda Twist is an algorithm that solves the Perspective Three-Point Problem (P3P) with a runtime of $O(N)$, where $N$ is the number of points. The key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.

Keywords: P3P, Visual Odometry, Camera Geometry

1 Introduction

We are concerned with the accuracy and consistency of odometry estimation on low-power hardware and systems. Since both odometry and localization errors independently have a large impact on the accuracy, we have a consistent solution to the P3P problem (Figure 1).

For odometry, the key feature of Lambda Twist is a novel and efficient parametrization of the P3P problem. The current state-of-the-art P3P solver is based on non-linear optimization, which makes it slow and unstable. The Lambda Twist parametrization is based on a linear combination of three homogeneous coordinates and is inspired by homography-based P3P solvers. The accuracy of Lambda Twist is similar to other non-linear P3P solvers, but it is significantly faster and more robust.
Orienting a calibrated camera by using > 3 control points

Spatial Resection
Iterative Solution
Overview: Iterative Solution

- Over determined system with \( I > 3 \)
- No direct solution but iterative LS
- Main steps
  - Build the system of observation equations
  - Measure image points \( \mathbf{x}_i, \ i = 1, \ldots, I \)
  - Estimate initial solution \( R, X_o \rightarrow \mathbf{x}^{(0)} \)
- Adjustment
  - Linearizing
  - Estimate extrinsic parameter \( \hat{\mathbf{x}} \)
  - Iterate until convergence
Summary

- P3P estimates the position and heading of a **calibrated camera** given control points
- Required $\geq 3$ control points
- Direct solution
  - Fast
  - Suited for outlier detection with RANSAC
- Statistically optimal solution using iterative least squares
  - Uses all available points
  - Assumes no outliers
  - Allows for accuracy assessments