

# **Photogrammetry & Robotics Lab**

## **Projective 3-Point (P3P) Algorithm / Spatial Resection**

**Cyrill Stachniss**

---

# 5 Minute Preparation for Today



<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  $\mathbf{X}_i$

## Observed:

- 2D image coordinates  $\mathbf{x}_i$  of the object points

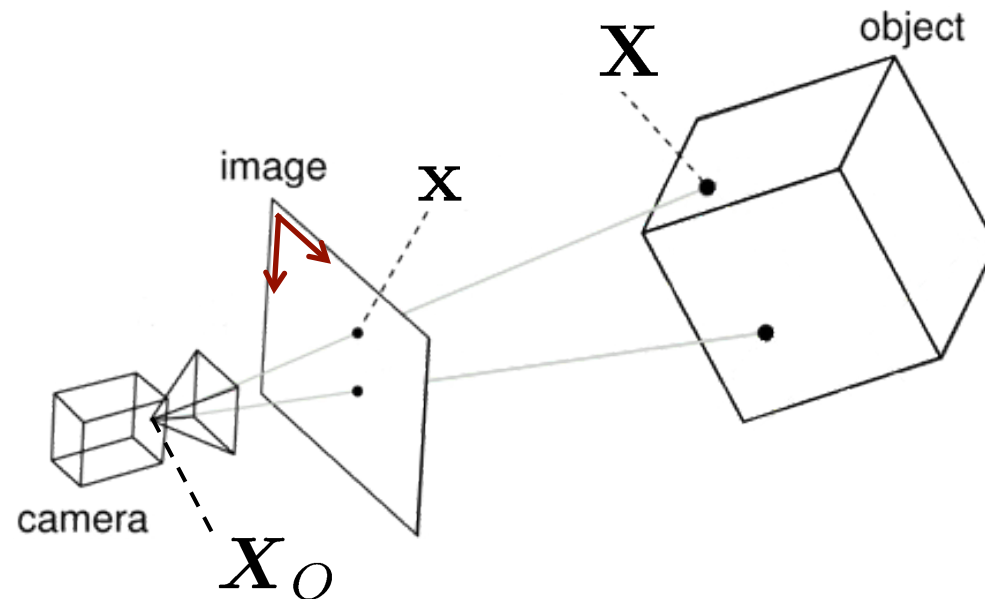
## Wanted:

- Extrinsic parameters  $R, \mathbf{X}_O$  of the calibrated camera

# Reminder: Mapping Model

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

$$\begin{aligned}\mathbf{x} &= KR[I_3 | -\mathbf{X}_O]\mathbf{X} \\ &= \mathbf{P} \mathbf{X}\end{aligned}$$



# Reminder: Camera Orientation

$$\mathbf{x} = \mathbf{K} \mathbf{R} [I_3 | -\mathbf{X}_O] \mathbf{X} = \mathbf{P} \mathbf{X}$$

- **Intrinsics (interior orientation)**
  - Intrinsic parameters of the camera
  - Given through matrix  $\mathbf{K}$
- **Extrinsics (exterior orientation)**
  - Extrinsic parameters of the camera
  - Given through  $\mathbf{X}_O$  and  $\mathbf{R}$

# Direct Linear Transform (DLT)

**Relation to DLT :** Compute the **11 intrinsic and extrinsic parameters**

$$\mathbf{x} = \mathbf{K} \mathbf{R} [I_3 | -\mathbf{X}_O] \mathbf{X}$$

The diagram illustrates the Direct Linear Transform (DLT) equation  $\mathbf{x} = \mathbf{K} \mathbf{R} [I_3 | -\mathbf{X}_O] \mathbf{X}$  with the following annotations:

- observed image point**: Points to  $\mathbf{x}$  with a black arrow.
- $\mathbf{c}, \mathbf{s}, \mathbf{m}, \mathbf{x}_H, \mathbf{y}_H$** : Points to  $\mathbf{K}$  with a red arrow.
- 3 rotations**: Points to  $\mathbf{R}$  with a red arrow.
- 3 translations**: Points to the  $-\mathbf{X}_O$  term with a red arrow.
- control point coordinates (given)**: Points to  $\mathbf{X}$  with a black arrow.

# Projective 3-Point Algorithm (or Spatial Resection)

Given the intrinsic parameters, compute the **6 extrinsic parameters**

$$\mathbf{x} = \mathbf{K} \mathbf{R} [I_3 | -\mathbf{X}_O] \mathbf{X}$$

The diagram illustrates the projective 3-point algorithm equation  $\mathbf{x} = \mathbf{K} \mathbf{R} [I_3 | -\mathbf{X}_O] \mathbf{X}$ . Annotations with arrows point to each term:

- observed image points**: points to  $\mathbf{x}$
- $c, s, m, x_H, y_H$  (given)**: points to  $\mathbf{K}$
- 3 rotations**: points to  $\mathbf{R}$
- 3 translations**: points to  $-\mathbf{X}_O$
- control point coordinates (given)**: points to  $\mathbf{X}$



# 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:

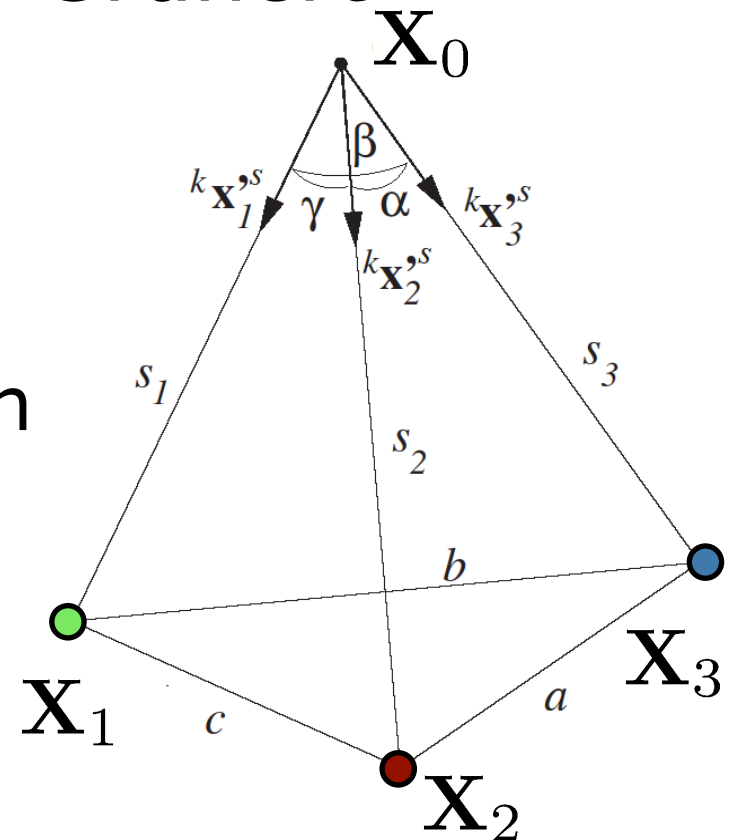
- 3D coordinates  $X_i$  of  $I \geq 3$  object points
- Corresponding image coordinates  $x_i$  recorded using a calibrated camera

## 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

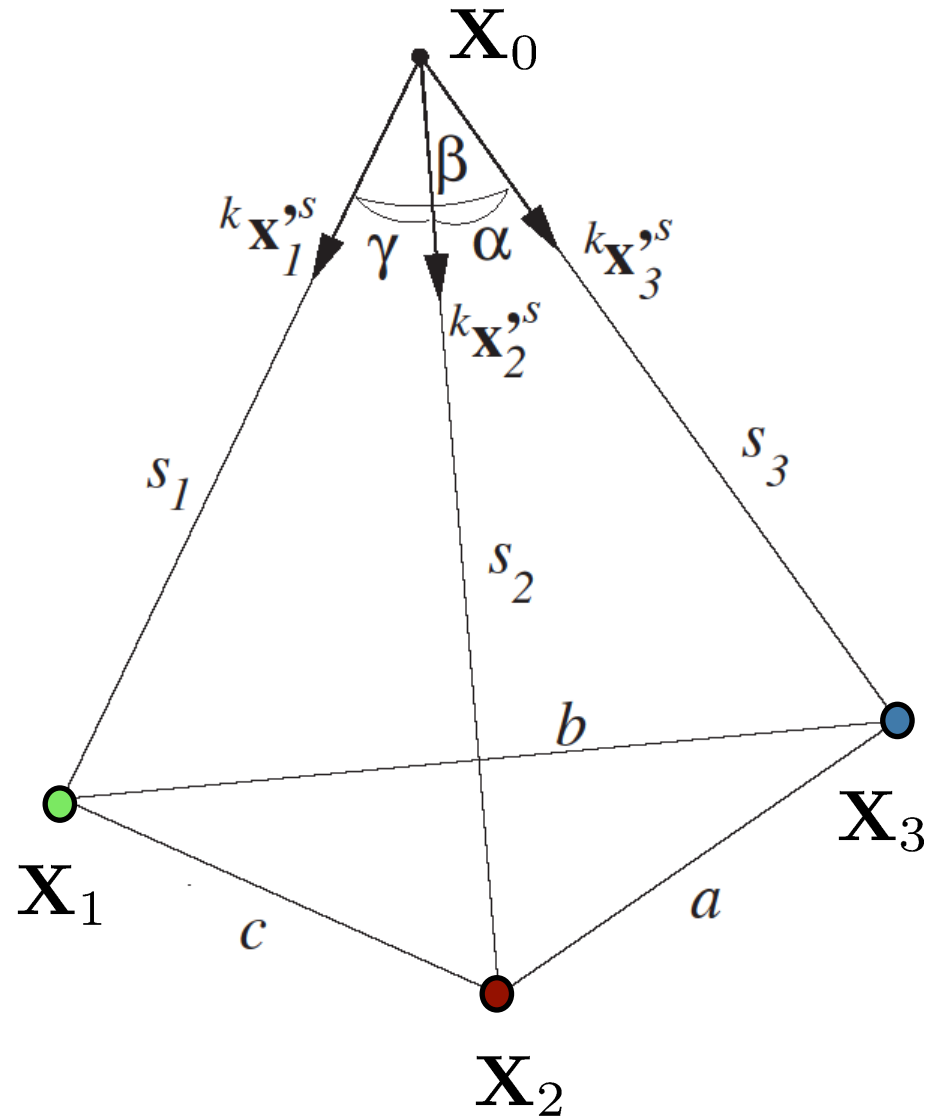


# Direct Solution by Grunert

## 2-Step process

Estimate

1. length of projection rays
2. orientation

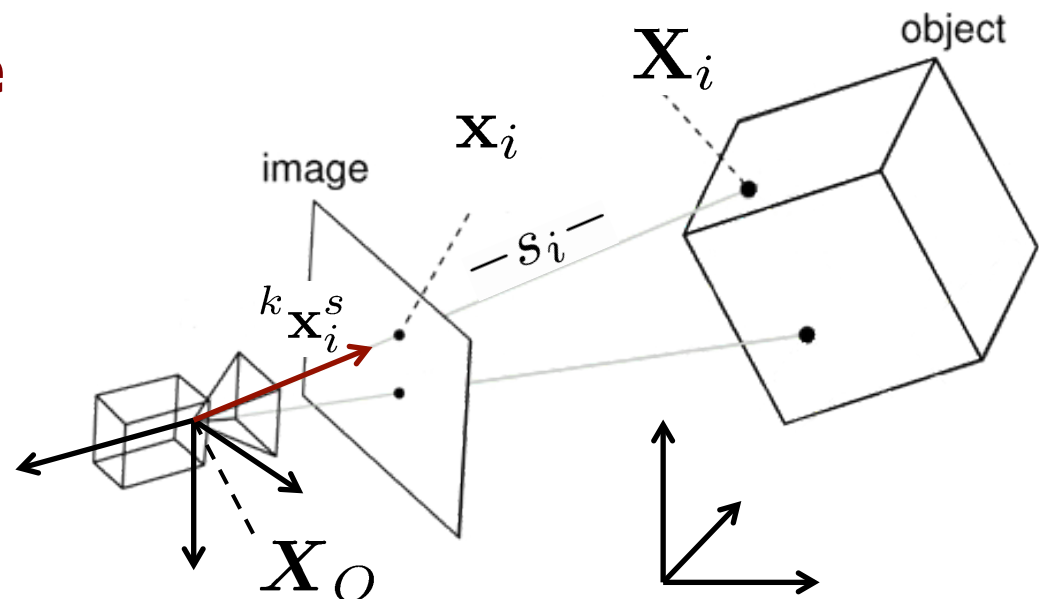


# P3P/SR Model

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

$$s_i \mathbf{x}_i^s = R(\mathbf{X}_i - \mathbf{X}_O) \quad i = 1, 2, 3$$

ray directions  
pointing to the  
object points



# P3P/SR Model

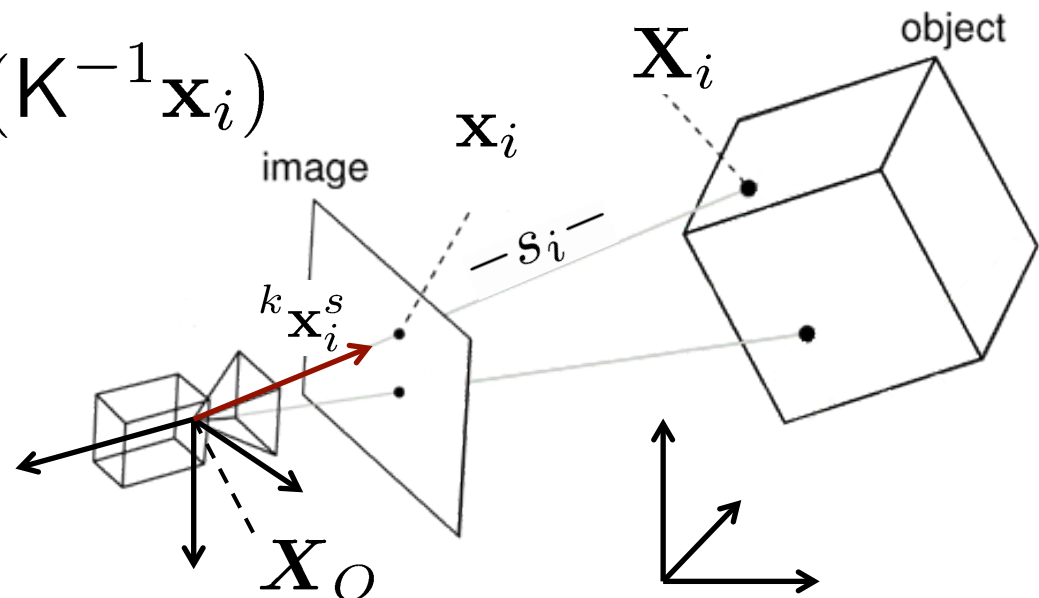
- Coordinates of object points within the camera system are given by

$$s_i {}^k \mathbf{x}_i^s = R(\mathbf{X}_i - \mathbf{X}_O) \quad i = 1, 2, 3$$

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

$${}^k \mathbf{x}_i^s = -\text{sign}(c) N(K^{-1} \mathbf{x}_i)$$

**ensure ray  
directions are  
pointing to the  
object points**



# P3P/SR Model

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

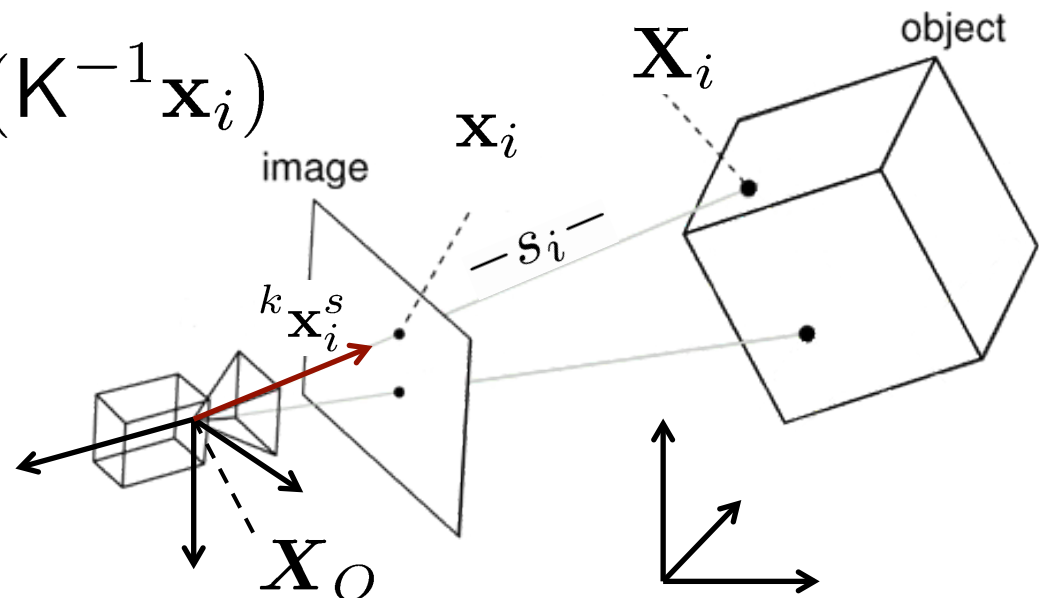
$$s_i {}^k \mathbf{x}_i^s = R(\mathbf{X}_i - \mathbf{X}_O) \quad i = 1, 2, 3$$

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

$${}^k \mathbf{x}_i^s = -\text{sign}(c) N(K^{-1} \mathbf{x}_i)$$

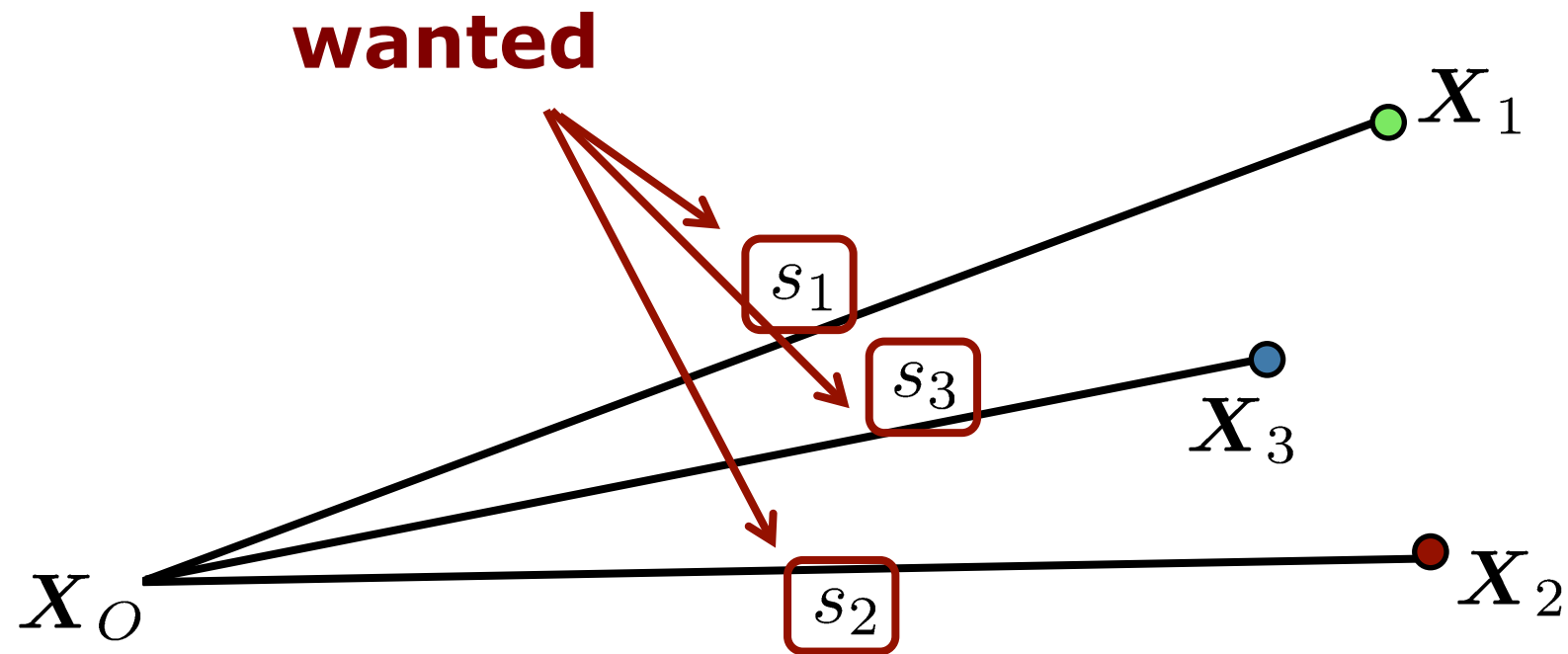
**spherical  
normalization**

$$N(\mathbf{x}) = \frac{\mathbf{x}}{|\mathbf{x}|}$$





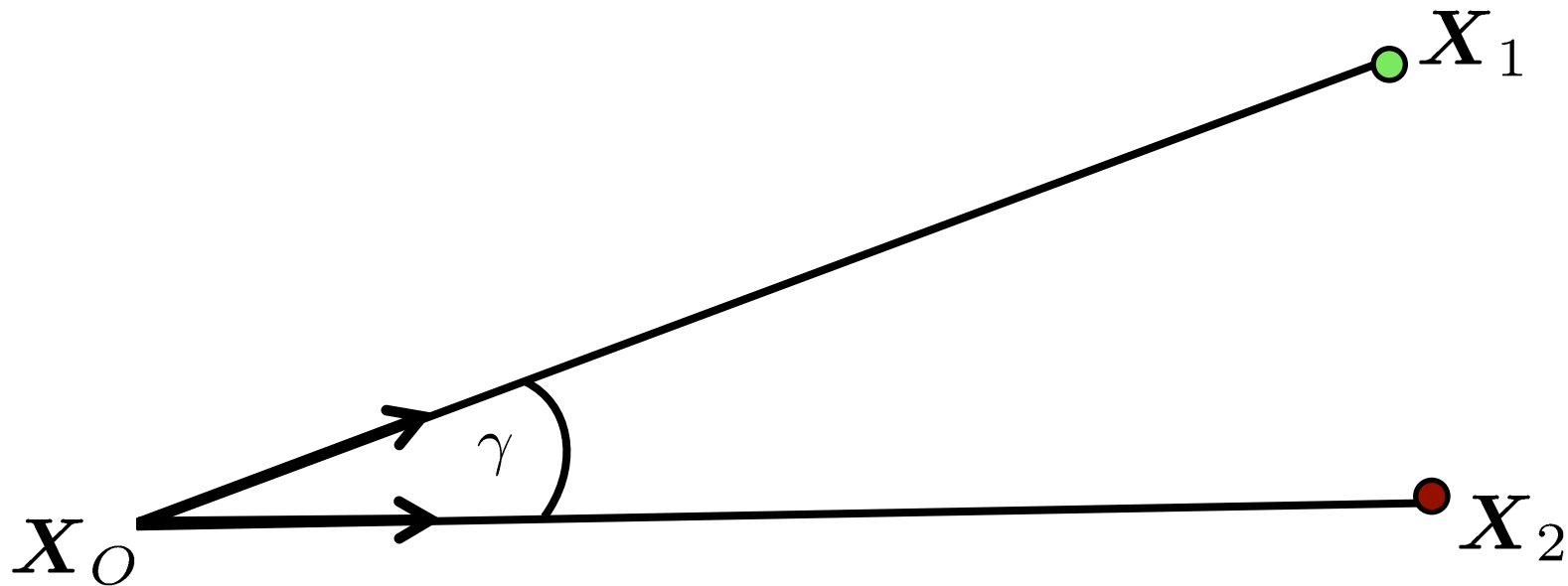
# 1. Get Length of Projection Rays



# 1. Get Length of Projection Rays

- Start with computing the angle between rays:

$$\cos \gamma = \frac{(\mathbf{X}_1 - \mathbf{X}_0) \cdot (\mathbf{X}_2 - \mathbf{X}_0)}{\|\mathbf{X}_1 - \mathbf{X}_0\| \|\mathbf{X}_2 - \mathbf{X}_0\|}$$



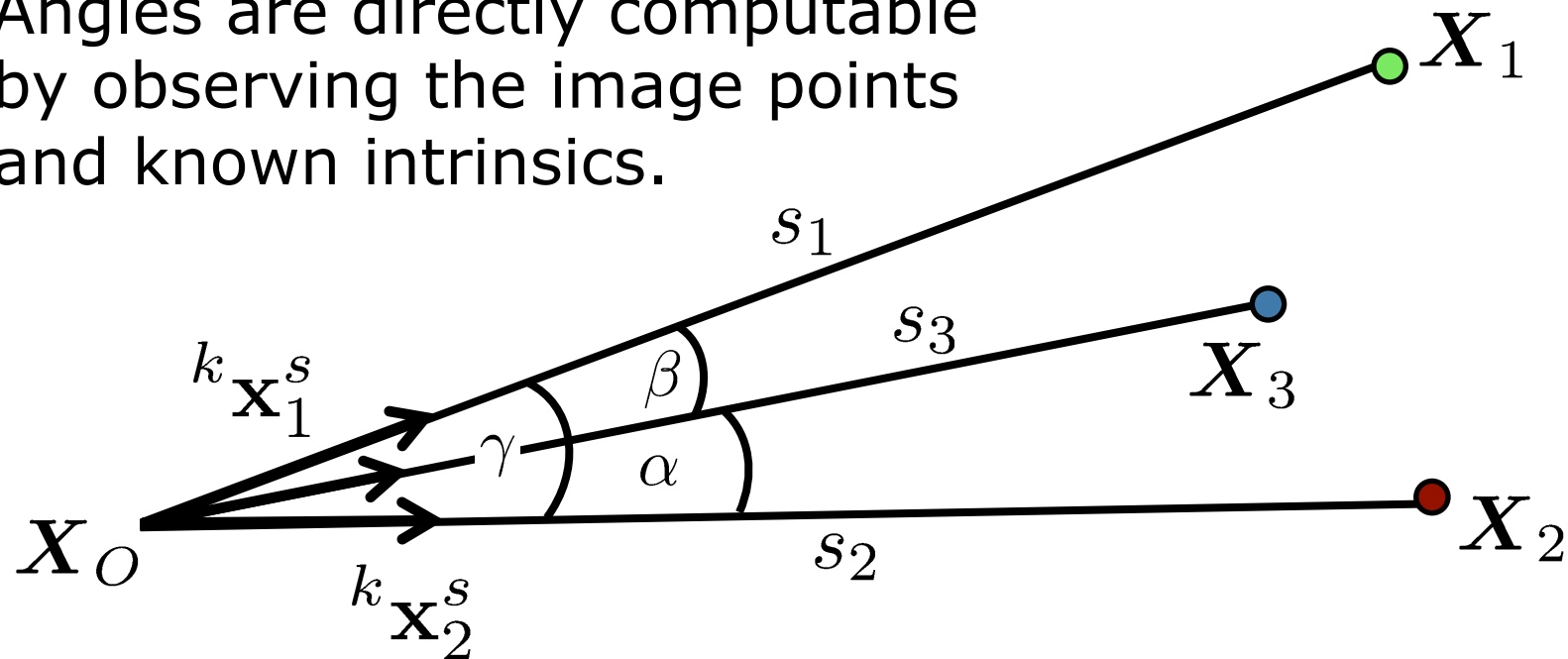
# 1. Get Length of Projection Rays

$$\alpha = \arccos \left( {}^k\mathbf{x}_2^s, {}^k\mathbf{x}_3^s \right)$$

$$\beta = \arccos \left( {}^k\mathbf{x}_3^s, {}^k\mathbf{x}_1^s \right)$$

$$\gamma = \arccos \left( {}^k\mathbf{x}_1^s, {}^k\mathbf{x}_2^s \right)$$

Angles are directly computable by observing the image points and known intrinsics.



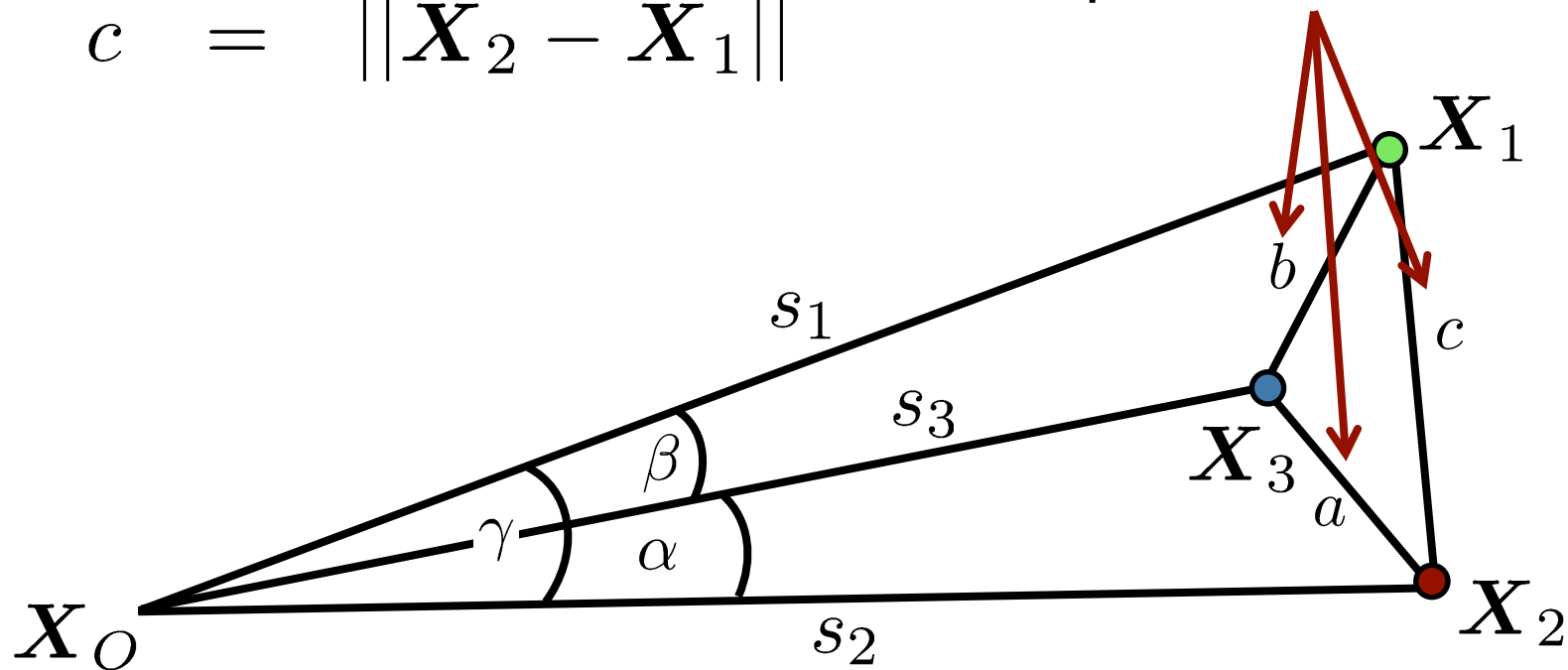
# 1. Get Length of Projection Rays

$$a = ||\mathbf{X}_3 - \mathbf{X}_2||$$

$$b = ||\mathbf{X}_1 - \mathbf{X}_3||$$

$$c = ||\mathbf{X}_2 - \mathbf{X}_1||$$

Given through  
known control  
point coordinates

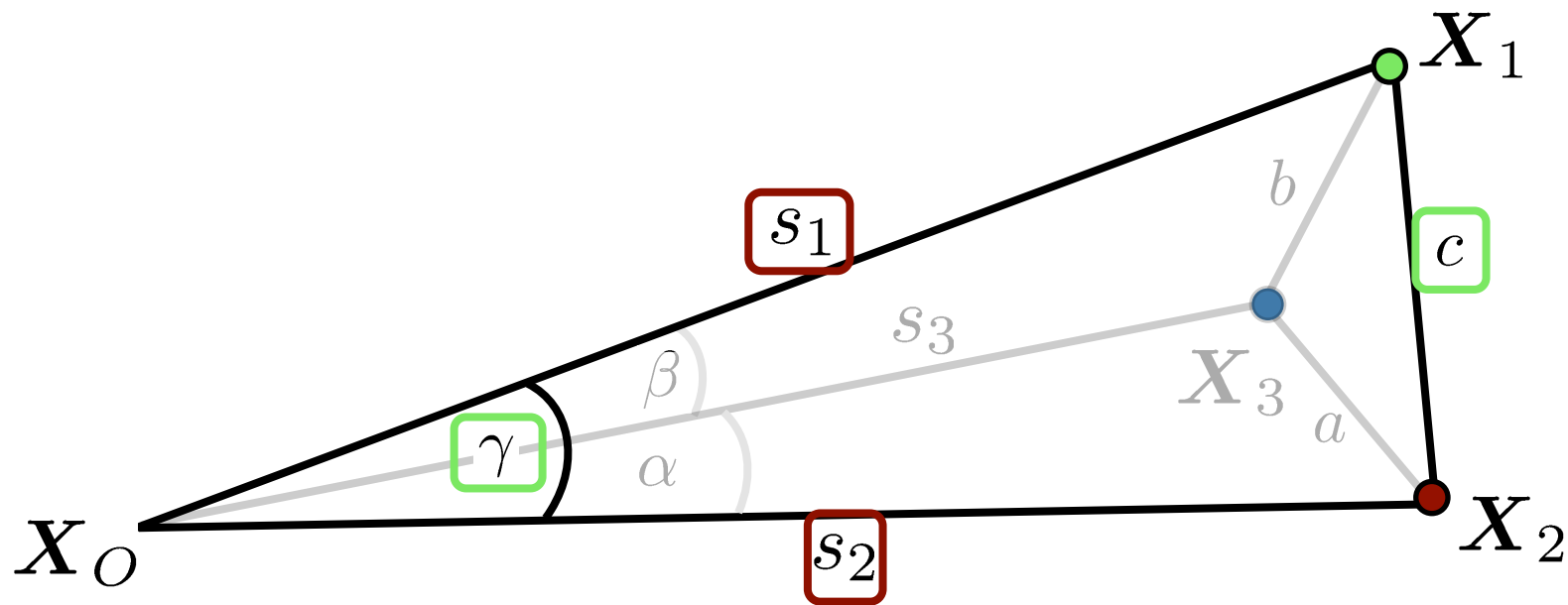


# Use the Law of Cosines

In triangle  $X_0, X_1, X_2$

$$s_1^2 + s_2^2 - 2 \boxed{s_1} \boxed{s_2} \cos \boxed{\gamma} = \boxed{c^2}$$

wanted                      known



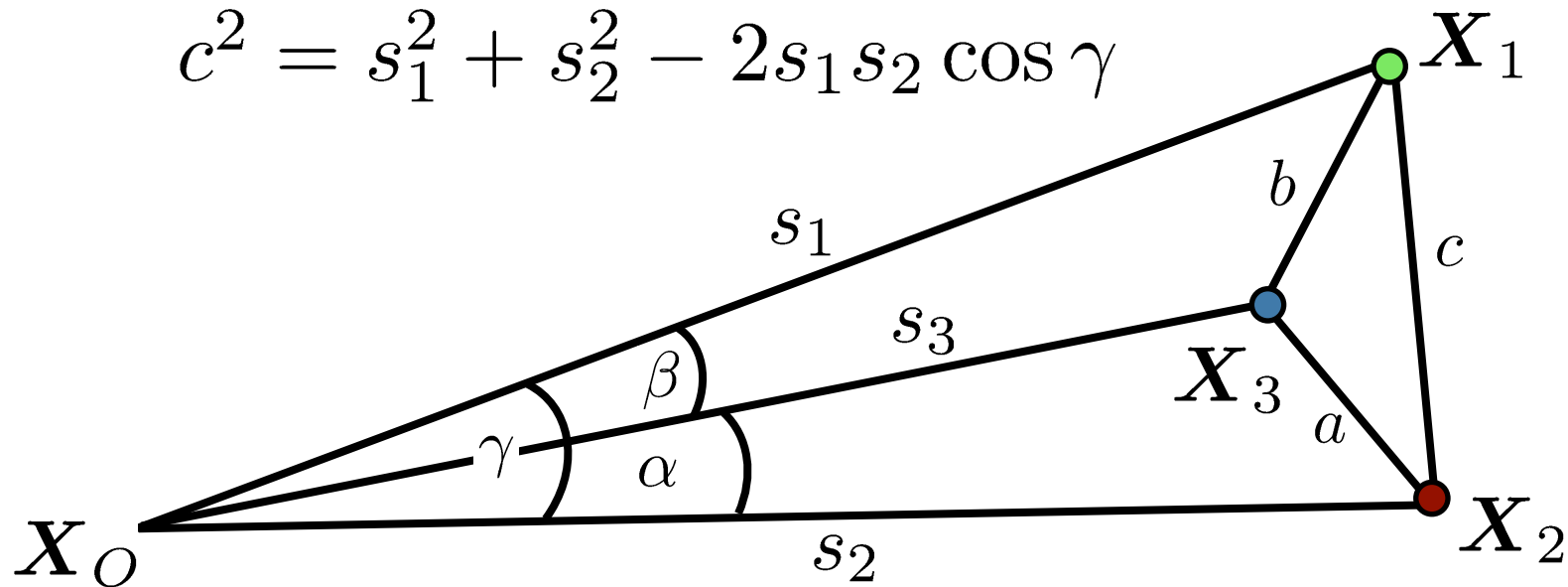
# Use the Law of Cosines

Analogously in all three triangles

$$a^2 = s_2^2 + s_3^2 - 2s_2s_3 \cos \alpha$$

$$b^2 = s_1^2 + s_3^2 - 2s_1s_3 \cos \beta$$

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



# Compute Distances

- We start from:

$$a^2 = s_2^2 + s_3^2 - 2s_2s_3 \cos \alpha$$

- Define:  $u = \frac{s_2}{s_1}$        $v = \frac{s_3}{s_1}$

- 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:

$$b^2 = s_1^2 + s_3^2 - 2s_1s_3 \cos \beta$$

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



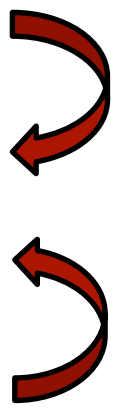
# Compute Distances

Analogously, we obtain

$$\begin{aligned}s_1^2 &= \frac{a^2}{u^2 + v^2 - 2uv \cos \alpha} \\ &= \frac{b^2}{1 + v^2 - 2v \cos \beta} \\ &= \frac{c^2}{1 + u^2 - 2u \cos \gamma}\end{aligned}$$

## Rearrange Again

Solve one equation for  $u$  put into the other

$$\begin{aligned}s_1^2 &= \frac{a^2}{u^2 + v^2 - 2uv \cos \alpha} \\s_1^2 &= \frac{b^2}{1 + v^2 - 2v \cos \beta} \\s_1^2 &= \frac{c^2}{1 + u^2 - 2u \cos \gamma}\end{aligned}$$


**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

$$\boxed{A_4}v^4 + \boxed{A_3}v^3 + A_2v^2 + A_1v + A_0 = 0$$

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

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

# Forth Degree Polynomial

$$A_4 v^4 + A_3 v^3 + A_2 v^2 + A_1 v + A_0 = 0$$

$$\begin{aligned} A_2 = & 2 \left[ \left( \frac{a^2 - c^2}{b^2} \right)^2 - 1 + 2 \left( \frac{a^2 - c^2}{b^2} \right)^2 \cos^2 \beta \right. \\ & + 2 \left( \frac{b^2 - c^2}{b^2} \right) \cos^2 \alpha \\ & - 4 \left( \frac{a^2 + c^2}{b^2} \right) \cos \alpha \cos \beta \cos \gamma \\ & \left. + 2 \left( \frac{b^2 - a^2}{b^2} \right) \cos^2 \gamma \right] \end{aligned}$$

# Forth Degree Polynomial

$$A_4 v^4 + A_3 v^3 + A_2 v^2 + \boxed{A_1} v + \boxed{A_0} = 0$$

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

$$\boxed{A_0} = \left( 1 + \frac{a^2 - c^2}{b^2} \right)^2 - \frac{4a^2}{b^2} \cos^2 \gamma$$

## 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:

$$s_1^2 = \frac{b^2}{1+v^2-2v \cos \beta}$$

$$s_3 = v s_1$$

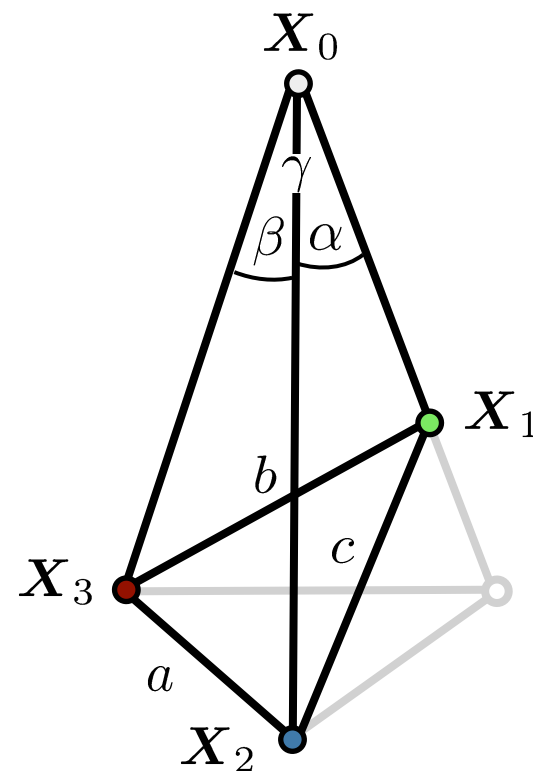
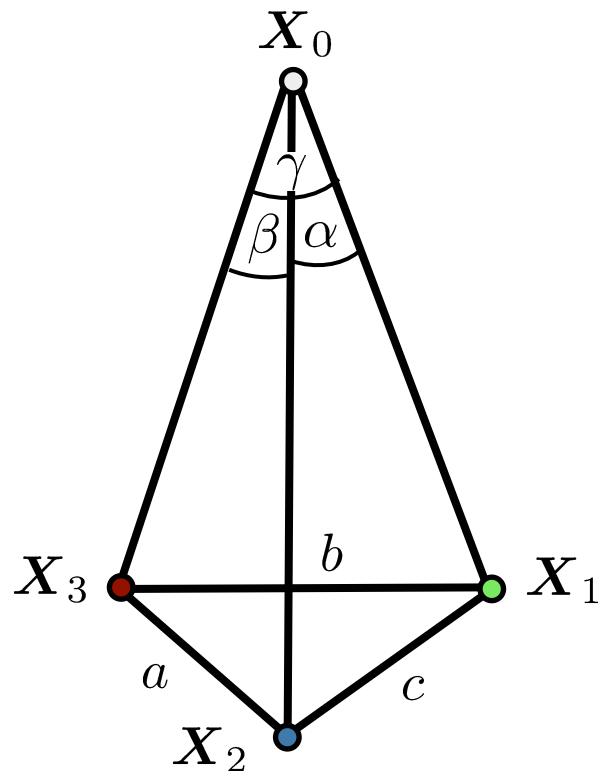
$$a^2 = s_2^2 + s_3^2 - 2s_2 s_3 \cos \alpha \Rightarrow s_2 = \dots$$

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

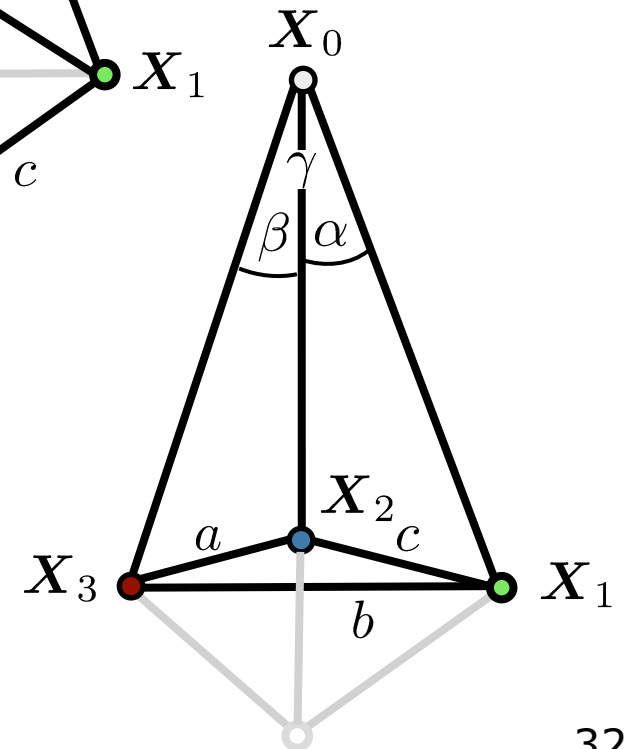
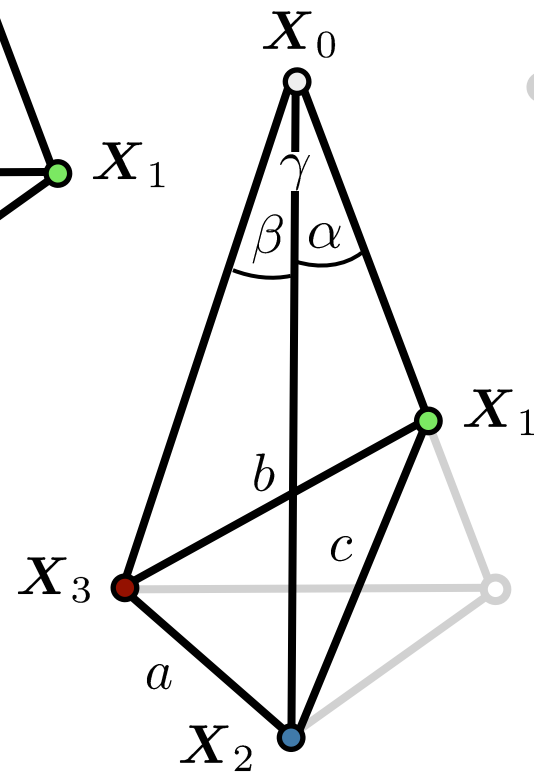
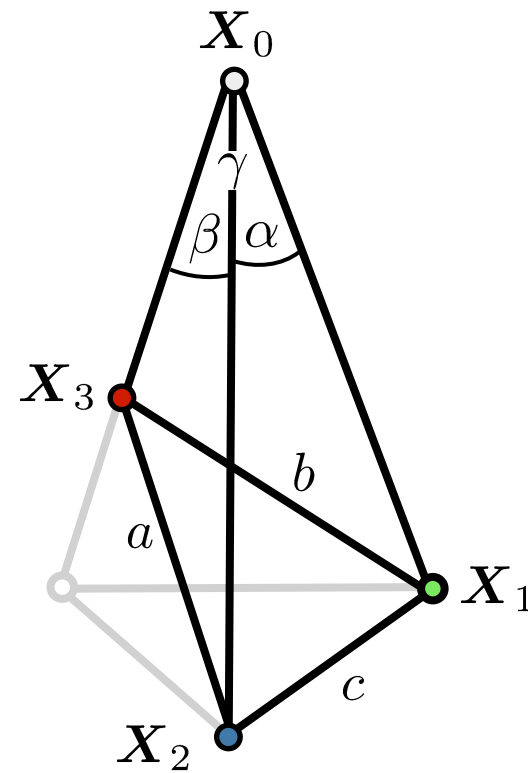
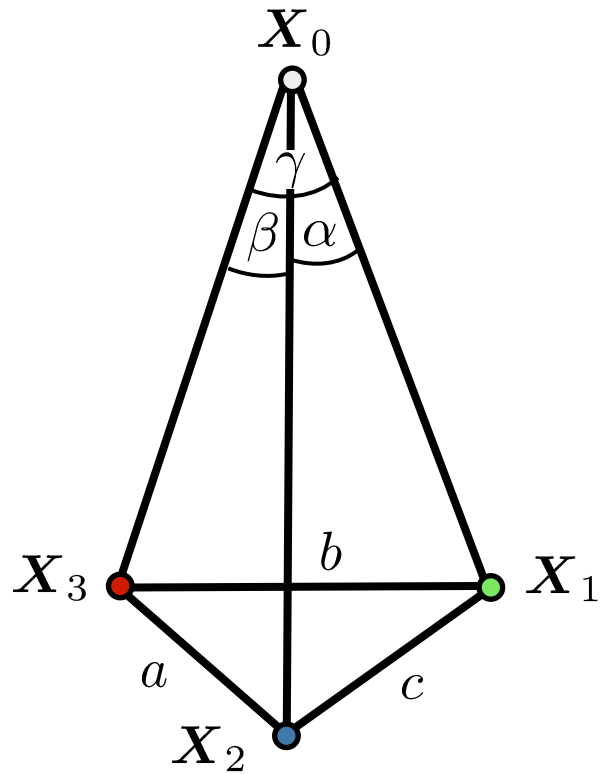
$$\{s_1, s_2, s_3\}_{1\dots 4}$$

# 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 4<sup>th</sup> points to confirm identify the correct solution



**Unique solution for**

$s_1, s_2, s_3$

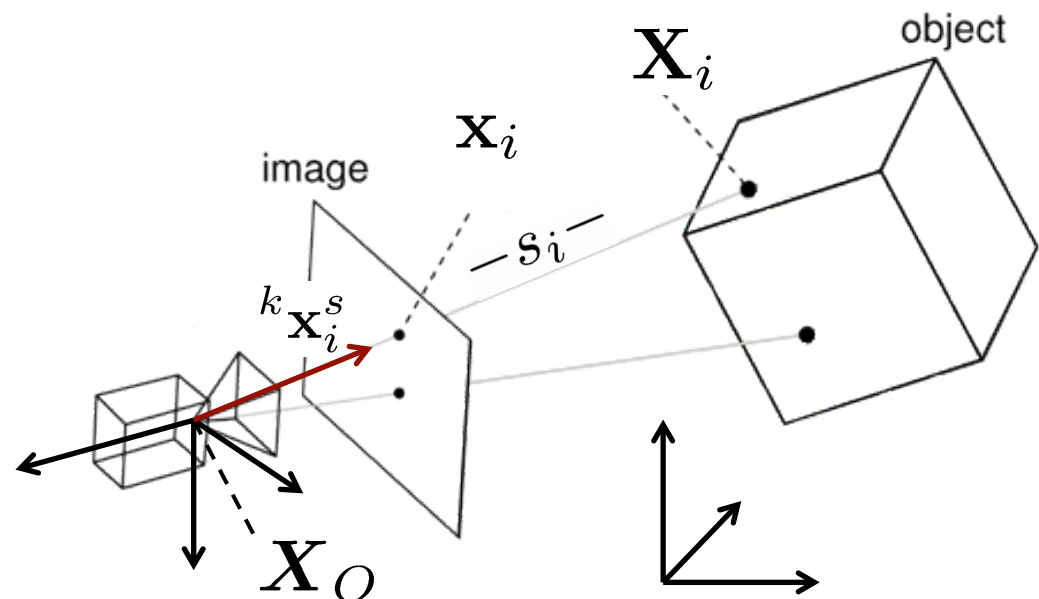
## 2. Orientation of the Camera

### Given:

- Distances and direction vectors to the control points

### Task:

- Estimate 6 extrinsic parameters

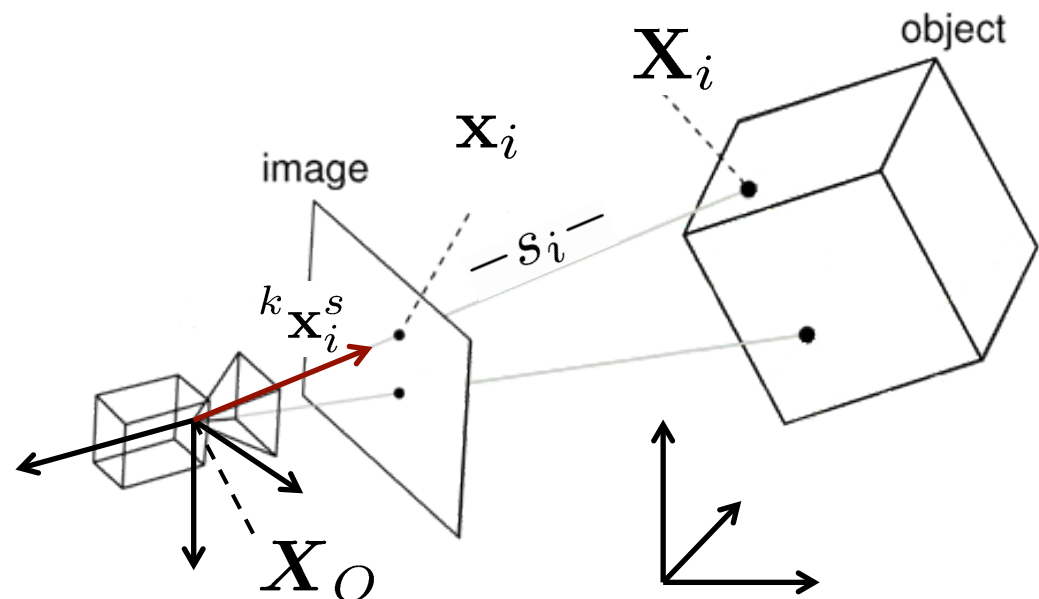


## 2. Orientation of the Camera

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!**



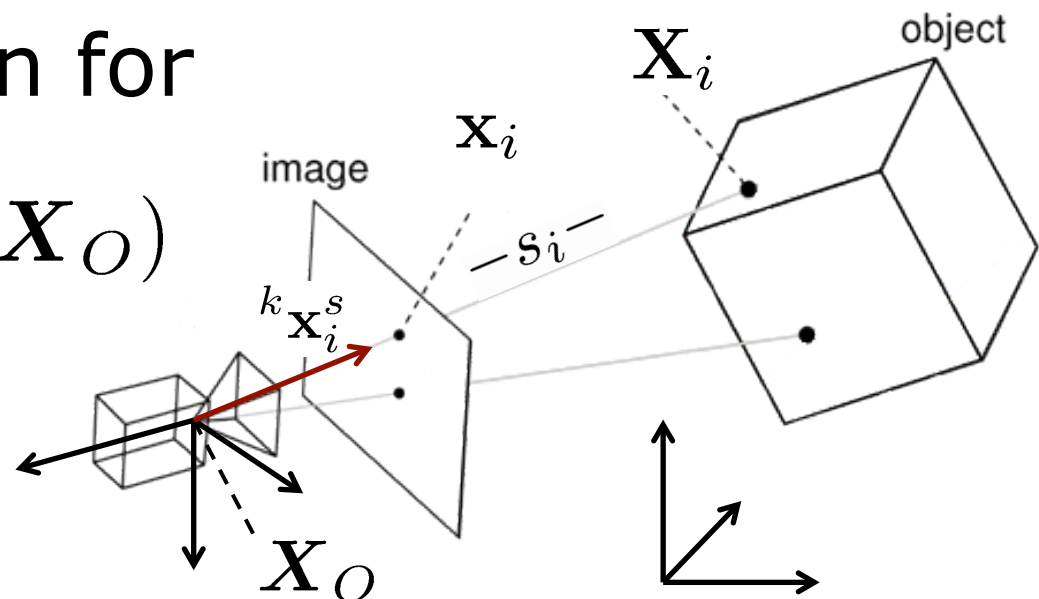
## 2. Orientation of the Camera

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

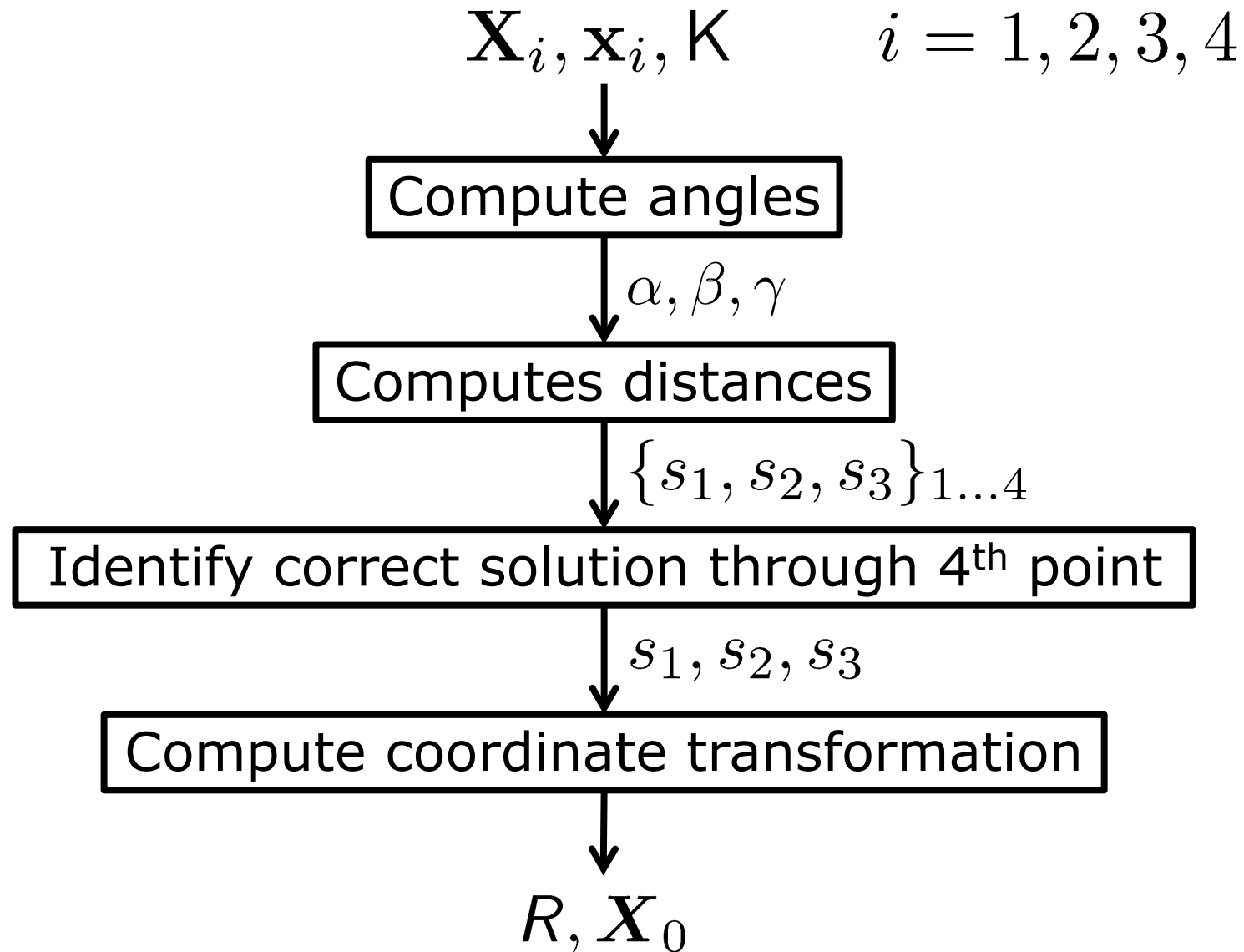
$${}^k\mathbf{X}_i = s_i {}^k\mathbf{x}_i^s \quad i = 1, 2, 3$$

2. Compute coordinate transformation for

$${}^k\mathbf{X}_i = R(\mathbf{X}_i - \mathbf{X}_O)$$



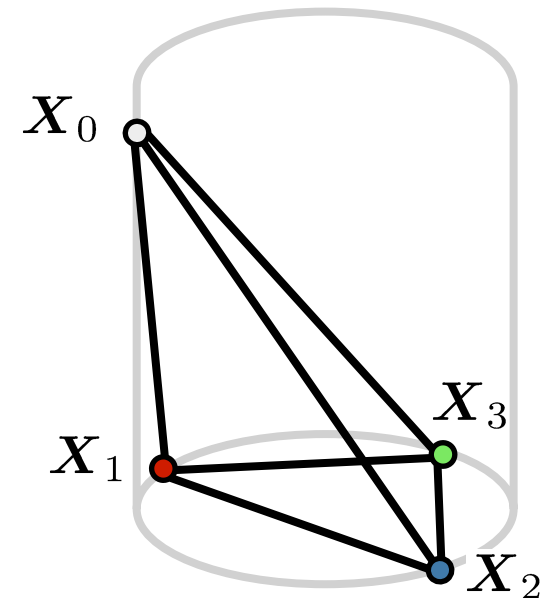
# P3P/SR in a Nutshell



# Critical Surfaces

## “Critical cylinder”

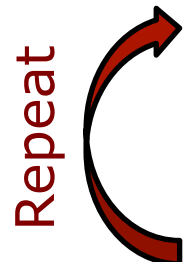
- If the projection center lies on a cylinder defined by the control points
- Small changes in angles lead to large changes in coordinates
- Unstable solution



# Outlier Handling with RANSAC

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

- Assume  $I \geq 3$  points

- Repeat 
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 have been proposed after Grunert's solutions of 1841
- New methods still have ambiguities when using 3 control points only
- 4<sup>th</sup> point needed for disambiguation
- Faster to compute
- Numerically more stable
- Partially less complex



# Recent Approaches

930

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 8, AUGUST 2003

## Complete Solution Classification for the Perspective-Three-Point Problem

Xiao-Shan Gao, Member, IEEE, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng

**Abstract**—In this paper, we use two approaches to solve the Perspective-Three-Point (P3P) problem: the algebraic approach and the geometric approach. In the algebraic approach, we use Wu-Ritt's zero decomposition algorithm to give a complete triangular decomposition for the P3P equation system. This decomposition provides the first complete analytical solution to the P3P problem. We also give a complete solution classification for the P3P equation system, i.e., we give explicit criteria for the P3P problem to have one, two, three, and four solutions. Combining the analytical solutions with the criteria, we provide an algorithm, CASSC, which may be used to find complete and robust numerical solutions to the P3P problem. In the geometric approach, we give some pure geometric criteria for the number of real physical solutions.

**Index Terms**—Perspective-Three-Point problem, pose determination, analytical solutions, solution classification, geometric criteria, Wu-Ritt's zero decomposition method.

### 1 INTRODUCTION

The Perspective-n-Point (PnP) problem is originated from camera calibration [1], [2], [3], [4]. Also known as pose estimation, it is to determine the position and orientation of the camera with respect to a scene object from  $n$  corresponding points. It concerns many important fields such as computer animation [5], computer vision [3], automation, image analysis, and automated cartography [2], photogrammetry [6], robotics [1], and model-based machine vision system [7], etc. Fischler and Bolles [2] summarized the problem as follows:

"Given the relative spatial locations of  $n$  control points, and given the angle to every pair of control points from an additional point called the Center of Perspective ( $C_p$ ), find the lengths of the line segments joining  $C_p$  to each of the control points."

The study of the PnP problem mainly consists of two aspects:

1. Design fast and stable algorithms that can be used to find all or some of the solutions of the PnP problem.
2. Give a classification for the solutions of the PnP problem, i.e., give the conditions under which the problem has one, two, three or four solutions.

There are many results for the first problem and the second problem is still open. The aim of this paper is to give a complete and effective solution to the above two problems for the P3P problem.

The P3P problem is the smallest subset of control points that yields a finite number of solutions. In 1981, Fischler and

Bolles [2] presented the RANSAC algorithm. They have noticed that there are at most four possible solutions to the P3P equation system. Hung et al. [8] presented an algorithm for computing the 3D coordinates of the perspective center relative to the camera frame. In 1991, Haralick et al. [9] reviewed the major direct solutions up to 1991, including six algorithms given by Grunert (1841), Finsterwalder (1903), Merritt (1949), Fischler and Bolles (1981), Linnainmaa et al. (1988), and Grafarend et al. (1989), respectively. They also give the analytical solution for the P3P problem with resultant computation. DeMenthon and Davis [10], [11] showed that by using approximations to the perspective, simpler computational solutions can be obtained. Qian and Lan [4] reduced the problem to a new quartic equation with Sylvester resultant and proposed a linear algebra algorithm to solve the PnP problem.

One of the important research directions on the P3P problem is its multisolution phenomenon. Fischler and Bolles [2] presented some examples of multisolutions of the P3P problem. In 1986, Wolfe [7] pointed out that the six permutations of the three control points combined with four-solution possibility can produce 24 possible camera-triangle configurations consistent with a single perspective view [6], [7]. Yuan [6] gave a necessary condition for the existence of the solution for first time. In 1991, Wolfe and Jones [12] gave a geometric explanation to this multisolution phenomenon in the image plane under the assumption of "canonical view."

In 1997, Su et al. [5] applied Wu-Ritt's zero decomposition method to find the main solution branch and some nondegenerate branches for the P3P problem. But a

## A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation

Laurent Kneip

laurent.kneip@mvz.ethz.ch

Davide Scaramuzza

davide.scaramuzza@mvz.ethz.ch

Roland Siegwart

rsiegwart@ethz.ch

Autonomous Systems Lab, ETH Zurich

### Abstract

The Perspective-Three-Point (P3P) problem aims at determining the position and orientation of the camera in the world reference frame from three 2D-3D point correspondences. This problem is known to provide up to four solutions that can then be disambiguated using a fourth point. All existing solutions attempt to first solve for the position of the points in the camera reference frame, and then compute the position and orientation of the camera in the world frame, which aligns the two point sets. In contrast, in this paper we propose a novel closed-form solution to the P3P problem, which computes the aligning transformation directly in a single stage, without the intermediate derivation of the points in the camera frame. This is made possible by introducing intermediate camera and world reference frames, and expressing their relative position and orientation using only two parameters. The projection of a world point into the parametrized camera pose then leads to two conditions and finally a quartic equation for finding up to four solutions for the parameter pairs. A subsequent backsubstitution directly leads to the corresponding camera poses with respect to the world reference frame. We show that the proposed algorithm offers accuracy and precision comparable to a popular, standard, state-of-the-art approach but at much lower computational cost (3.5 times faster). Furthermore, it provides improved numerical stability and is less affected by degenerate configurations of the selected world points. The superior computational efficiency is particularly suitable for any RANSAC-outlier rejection step, which is always recommended before applying PnP or non-linear optimization of the final solution.

### 1. Introduction

The Perspective-n-Point (PnP) problem is originated from camera calibration [1], [10], [17], [28]. Also known as pose estimation, it aims at retrieving the position and orientation of the camera with respect to a scene object from

$n$  corresponding 3D points. This problem has found many applications in computer animation [30], computer vision [16], augmented reality, automation, image analysis, automated cartography [16], photogrammetry [1], [24], robotics [15], and model-based machine vision systems [13]. In 1981, Fischler and Bolles [10] summarized the problem as follows: Given the relative spatial locations of  $n$  control points, and given the angle to every pair of control points  $P_i$  from an additional point called the center of perspective  $C$ , find the lengths of the line segments joining  $C$  to each of the control points. The next step then consists of retrieving the orientation and translation of the camera with respect to the object reference frame.

The Direct Linear Transformation was first developed by photogrammetrists [31] as a solution to the PnP problem—when the 3D points are in a general configuration—and then introduced in the computer vision community [7], [16]. When the points are coplanar, the homography transformation can be exploited [16] instead.

In this paper, we address the particular case of PnP for  $n = 3$ . This problem is also known as Perspective-Three-Point (P3P) problem. The P3P is the smallest subset of control points that yields a finite number of solutions. When the intrinsic camera parameters are known and we have  $n \geq 4$ , the solution is generally unique.

The P3P problem was first investigated in 1841 by Grunert [13] and in 1903 by Finsterwalder [4], who noticed that for a calibrated camera there can be up to four solutions, which can then be disambiguated using a fourth point. In the literature, there exist many solutions to this problem, which can be classified into iterative, non-iterative, linear, and non-linear ones. In 1991, Haralick et al. [13] reviewed the major direct solutions up to 1991, including the six algorithms given by Grunert (1841) [13], Finsterwalder (1903)—as summarized by Finsterwalder and Scheufele in [9]—, Merritt (1949) [25], Fischler and Bolles (1981) [10], Hung et al. (1985) [20], Linnainmaa et al. (1988) [23], and Grafarend et al. (1989) [13], respectively. They also gave the analytical solution for the P3P problem with re-

1

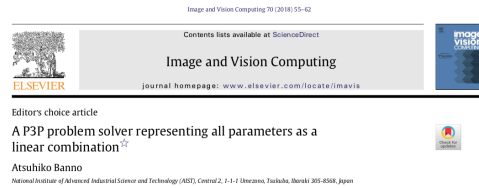
Gao  
2003

Complete Solution Classification  
for the Perspective-Three-Point  
Problem

Kneip  
2011

A novel parametrization of the  
perspective-three-point problem for a  
direct computation of absolute  
camera position and orientation

# Recent Approaches



ARTICLE INFO	ABSTRACT
<p><b>Article history:</b> Received 7 December 2016 Received in revised form 5 August 2017 Accepted 8 January 2018 Available online 15 January 2018</p> <p><b>Keywords:</b> P3P problem Localization Camera parameter Global basis</p>	<p>We propose a novel strategy for the Perspective-Three-Point (P3P) problem that determines the position and orientation of a calibrated camera from three known point pairs of 2D-3D correspondences. Starting from three similarity transformation equations that relate the global and the camera-oriented coordinates, our method treats all of the intrinsic camera parameters as a linear combination of known vectors with unknown coefficients. By reducing the number of unknowns and using a Gelb basis, the problem is converted into a fourth-order polynomial equation with a single unknown parameter. Experimental results show that our method is highly practical and precise. Moreover, the performance of our method and its robustness to image noise are similar to those of a state-of-the-art method. In addition, our method exhibits greater computational efficiency than other methods.</p> <p>© 2018 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<a href="http://creativecommons.org/licenses/by/4.0/">http://creativecommons.org/licenses/by/4.0/</a>).</p>

**1. Introduction**

Estimation of the position and orientation of a calibrated camera using the  $n$  correspondences of control points between a 2D image and the 3D world is referred to as the Perspective- $n$ -Point (PnP) problem. PnP is a fundamental problem for calculating the self-position and orientation of a camera in known environments [10,14,34]. It has been the subject of many studies and is relevant to diverse fields, such as photogrammetry [7], camera calibration [32], pose estimation [4,17], and robotics [5,21]. Various solutions to the PnP problem have been reported, such as closed methods [23], iterative methods [10,27], algebraic methods [1,20], and geometric methods [3,8,19]. These methods have been developed for not only perspective cameras but also general cameras [25,29].

Among the family of PnP problems, the Perspective-Three-Point (P3P) problem, which uses three correspondences, is a fundamental problem [6,11–12,25,26]. Three corresponding pairs constitute the minimum set for estimating the position and orientation of a camera. Therefore, this approach is useful in cases where it is difficult to extract a sufficient number of inlier correspondences from an image and the 3D scene. In addition, methods that use the minimum number of correspondences are advantageous in the case of noisy data, in the Random Sample Consensus (RANSAC) framework [7], the probability of selecting the inlier pairs in P3P is higher than that

<sup>†</sup> This paper has been recommended for acceptance by Enrique Dato.

E-mail address: [atsuhiko.banno@nec.com](mailto:atsuhiko.banno@nec.com).

<https://doi.org/10.1016/j.imvcom.2018.01.001>  
0262-8856/© 2018 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## Lambda Twist: An Accurate Fast Robust Perspective Three Point (P3P) Solver.

Mikael Persson<sup>1</sup> [0000–0002–5931–9396] and Klas Nordberg<sup>1</sup> \*

Computer Vision Laboratory, Linköping University, Sweden

**Abstract.** We present Lambda Twist: a novel P3P solver which is accurate, fast and robust. Current state-of-the-art P3P solvers find all roots to a quartic and discard geometrically invalid and duplicate solutions in a post-processing step. Instead of solving a quartic, the proposed P3P solver exploits the underlying elliptic equations which can be solved by a fast and numerically accurate diagonalization. This diagonalization requires a single real root of a cubic which is then used to find the, up to four, P3P solutions. Unlike the direct quartic solvers our method never computes geometrically invalid or duplicate solutions. Extensive evaluation on synthetic data shows that the new solver has better numerical accuracy and is faster compared to the state-of-the-art P3P implementations. Implementation and benchmark are available on github.

**Keywords:** P3P · PnP · Visual Odometry · Camera Geometry

## 1 Introduction

Pose estimation from projective observations of known model points, also known as the Perspective  $n$ -point Problem (PnP), is extensively used in geometric computer vision systems. In particular, finding the camera pose (orientation and position) from observations of  $n$  3D points in relation to a world coordinate system is often the first step in visual odometry and augmented reality systems [12,7]. It is also an important part in structure from motion and reconstruction of unordered images [1]. The minimal PnP case with a finite number of solutions requires three ( $n = 3$ ) observations in a nondegenerate configuration and is known as the P3P problem (Figure 1).

We are concerned with the latency and accuracy critical scenarios of odometry on low power hardware and AR/VR. Since both latency and localization errors independently not only break immersion, but also cause nausea, accurate solutions and minimal latency are crucial. As an example application, vision based localization for AR/VR places a few markers/beacons on a target, which are then found using a high speed camera. Ideally we would then solve the pose directly on chip without sending the full image stream elsewhere, mandating minimal cost. Further, because the markers are placed on a small area and the camera is of relatively low resolution, the markers are close to each other, meaning numerical issues due to near degenerate cases are common and the algorithm must be robust. The experiments will show that we have made substantial progress on both speed and accuracy compared to state-of-the-art.

\* name.surname@liu.se

NAKANO: SIMPLE DIRECT SOLUTION TO P3P PROBLEM

1

## A Simple Direct Solution to the Perspective-Three-Point Problem

Gaku Nakano  
[g-nakano@cq.jp.nec.com](mailto:g-nakano@cq.jp.nec.com)

Central Research Labs  
NEC Corporation  
Kawasaki, Japan

## Abstract

This paper proposes a new direct solution to the perspective-three-point (P3P) problem based on an algebraic approach. The proposed method represents the rotation matrix as a function of distances from the camera center to three 3D points, then, finds the distances by utilizing the orthogonal constraints of the rotation matrix. The formulation can be simply written because it relies only on some simple concepts of linear algebra. According to synthetic data evaluations, the proposed method gives the second-best performance against the state-of-the-art methods on both numerical accuracy and computational efficiency. In particular, the proposed method is the fastest among the quartic-equation based solvers. Moreover, the experimental results imply that the P3P problem still has an arguable issue on numerical stability regarding a point distribution and a camera pose.

## 1 Introduction

The perspective-three-point (P3P) problem, also known as the absolute camera pose estimation problem, is one of the most classical and fundamental problems in computer vision that determines the pose of a calibrated camera, *i.e.* the rotation and the translation, from three pairs of 3D point and its projection on the image plane. Since Grunert [8] gave the first solution in 1841, the P3P problem has been widely investigated [8, 9, 10] and extended to more complex camera pose estimation problems, *e.g.* for least squares case with  $n$  points (the PnP problem [8, 9, 10, 11, 12]), for uncalibrated cameras with unknown internal parameters such as focal length or lens distortion (the P3.5P [13], P4P [8, 14], P5P [8], PnP [8, 15], and PnPPr [16] problems).

Classical methods for the P3P problem [8, 9] consist of two steps: first, find the distances between the camera center and the given three 3D points; then, estimate the camera pose by solving an alignment problem of two triangles. The first step formulates a quartic equation with respect to one of the three distances by eliminating the other two based on the law of cosines. After finding the roots of the quartic equation, the second step solves the alignment problem, which is a rigid transformation between two triangles, by using a  $4 \times 4$  eigenvalue decomposition or  $3 \times 3$  singular value decomposition. Due to operations of the matrix decomposition, the numerical accuracy of the final solution becomes low despite its time-consuming processing.

© 2019. The copyright of this document resides with its authors.  
It may be distributed unchanged freely in print or electronic forms.

Banno  
2018

Persson  
2019

Nakano  
2019

A P3P problem solver  
representing all  
parameters as a linear  
combination

Lambda Twist: An Accurate  
Fast Robust Perspective  
Three Point (P3P) Solver.

A Simple Direct  
Solution to the  
Perspective-Three-  
Point Problem

**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  $x_i, i = 1, \dots, I$
  - Estimate initial solution  $R, X_o \rightarrow x^{(0)}$
  - Adjustment
    - Linearizing
    - Estimate extrinsic parameter  $\hat{x}$
    - 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