

P3.5P: Pose Estimation with Unknown Focal Length

Changchang Wu
Google Inc.

Abstract

It is well known that the problem of camera pose estimation with unknown focal length has 7 degrees of freedom. Since each image point gives 2 constraints, solving this problem requires a minimum of 3.5 image points of 4 known 3D points, where 0.5 means either x or y coordinate of an image point. We refer to this minimal problem as **P3.5P**. However, the existing methods require 4 full image points to solve the camera pose and focal length [21, 1, 3, 23]. In this paper, we present a general solution to the true minimal P3.5P problem with up to 10 solutions. The remaining image coordinate is then used to filter the candidate solutions, which typically results in a single solution for good data or no solution for outliers. Experiments show the proposed method significantly improves the efficiency over the state of the art methods while maintaining a high accuracy.

1. Introduction

The Perspective- n -Point (PnP) problem is a classic problem in computer vision, and a key component of many techniques, including incremental structure from motion [18, 6, 22] and location recognition [10, 16]. Given a set of n 2D image points in a camera and their corresponding known 3D points, the problem is to solve for the unknown camera parameters, including the pose and sometimes the intrinsics.

Let m be the number of independently unknown parameters of the camera, the minimal problem requires $n = \frac{m}{2}$ image points to recover the unknowns. For example, the homogeneous 3×4 camera projection matrix has $m = 11$ degrees of freedom, so it requires $n = 5.5$ image points to compute [9]. Take P3P as another example, the calibrated pose estimation has $m = 6$ unknowns (3 for rotation and 3 for translation) and thus requires $n = 3$ image points [5, 13]. This paper addresses the $m = 7$ case, where the unknowns are the camera pose and a single focal length parameter. Obviously the corresponding minimum problem is for $n = 3.5$ image points, which we call the P3.5P problem.

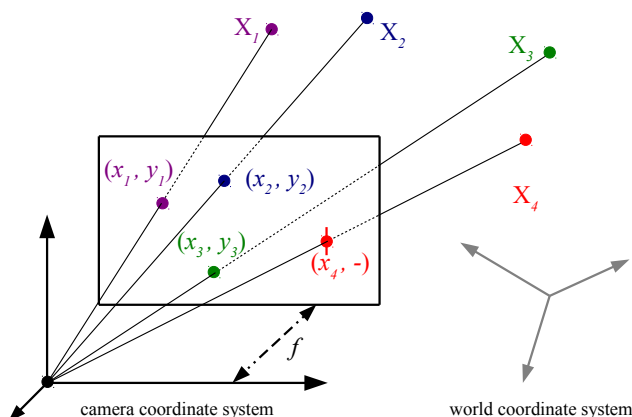


Figure 1. The P3.5P problem. The input are the 3 image points $(x_1, y_1)(x_2, y_2)(x_3, y_3)$ plus either x_4 or y_4 of the 4th image point and their corresponding 3D points $\{X_1, X_2, X_3, X_4\}$ in the world coordinate system. The 7 image coordinates give sufficient constraints for solving the unknown focal length and camera pose.

lem. See Figure 1 for an illustration.

The pose estimation problem with a single unknown focal length is a widely used approximation for the pose estimation of uncalibrated cameras, where the other intrinsic parameters are assumed known (e.g. principal point at image center, zero skew). In particular, such a minimal parametrization is often used in the 3D reconstruction of internet photos [18, 6, 22]. Despite extensive studies on various PnP problems over the past decades, surprisingly there does not exist a general minimal solution for the P3.5P problem. The state-of-the-art methods require $n = 4$ full image points instead to solve for the $m = 7$ unknowns, which leads to an over-constrained P4P (also called P4Pf) problem [21, 1, 3, 23]. This paper presents the first general solution to the P3.5P problem, which is truly minimal and additionally has simpler polynomial systems than [3, 2, 23].

The extra image coordinate is straightforwardly useful for solution verification. Among the set of solutions from the 3.5 image points, we consider the ones with large reprojection error for the remaining coordinate as invalid and discard them. This reliable reduction of solution candidates

makes the RANSAC-based pose estimation more efficient.

2. Background

Let P be the projection matrix to solve and P^k be the k -th row of P . For each known 3D point $i = 1, 2, 3, 4$, we use X_i to denote its inhomogeneous coordinates and $\widetilde{X}_i = [X_i^\top, 1]^\top$ to denote its homogeneous coordinates. The observed image point (x_i, y_i) of X_i is given by perspective projection

$$(x_i, y_i) = \left(\frac{P^1 \widetilde{X}_i}{P^3 \widetilde{X}_i}, \frac{P^2 \widetilde{X}_i}{P^3 \widetilde{X}_i} \right).$$

The image coordinates x_i and y_i provide two independent linear constraints on P as the following

$$0 = P^1 \widetilde{X}_i - x_i P^3 \widetilde{X}_i \quad (1)$$

$$0 = P^2 \widetilde{X}_i - y_i P^3 \widetilde{X}_i, \quad (2)$$

which we refer to as the **image coordinate constraints**. Any PnP problem is essentially to solve for the projection matrix from $2n$ image coordinate constraints. For example, 11 such equations gives to a linear system that allows for solving a 3×4 projection matrix. It can be seen that the image coordinate constraint naturally handles any 0.5 image point (that is, single image coordinate), because the two constraints from x_i and y_i are independent. Without loss of generality, we use the 7 image coordinate constraints given by $\{x_1, y_1, x_2, y_2, x_3, y_3, x_4\}$ to solve the P3.5P problem.

Most existing P4P and PnP methods cannot be extended to solve the P3.5P problem. Instead of using the image coordinate constraints directly, these methods derive new constraints based on the equality of 3D point distances in the world and camera coordinate systems. For 4 coplanar 3D points, Abidi and Chandra [1] derived a set of simple distance ratio constraints from the tetrahedron volume, which was later generalized by Bujnak et al. to handle 4 arbitrary 3D points [3, 2]. Recently, Zheng et al. developed a new general P4P algorithm by exploiting the equality of the angles defined by 3D point triplets [23]. In order to parametrize all the 3D points in the camera coordinate system, such methods require 4 complete image points along with their depth parameters, which are incapable of handling a single image coordinate. Our proposed method directly uses the image coordinate constraints, thus does not have such limitations.

Triggs's P4P algorithm [21] in fact can be extended to handle the P3.5P problem, but it is inherently degenerate for planar structures. This method first builds a linear system for the projection matrix from the image coordinate constraints, and then parametrizes the projection matrix using the null space, which is $4 = (12 - 4 * 2)$ dimensional for 4 image points and would be $5 = (12 - 7)$ dimensional for

3.5 image points. For a single unknown focal length, there exist 4 constraints on the dual image of the absolute conic (DIAC) [21], which is over-constrained for the 4 dimensional null space but can lead to a finite number of solutions for the 5 dimensional null space. However, similar to P4P, such a derived P3.5P algorithm would have degeneracy issues with planar structures, because planar scenes provide fewer independent constraints on the calibration [20].

By modeling the P3.5P problem in Macaulay 2 [8], we find it having 10 non-trivial solutions (focal length $f \neq 0$) for non-coplanar points, and 8 non-trivial solutions for coplanar points. Therefore, an optimal general minimum solver for P3.5P should have 10 solutions (trivial or non-trivial, real or imaginary). When applying Trigg's method [21] to the P3.5P problem, we find it producing 16 solutions, where 6 out of 16 are trivial with zero focal lengths. Since P4P is over-constrained, the existing general solutions [3, 23] discard some available constraints in order to create a polynomial system with a finite number of solutions. In particular, the P4P algorithm by Bujnak et al. [3] can produce the expected number of 10 solutions, while the method by Zheng et al. [23] has 16 solutions. In this paper, we introduce a method for P3.5P with the desired minimal number of 10 solutions.

The goal of this paper is to develop a general and compact solution to the P3.5P problem that can work for both coplanar points and non-coplanar points. To tackle this problem, we find it necessary to introduce a new parametrization for cameras with unknown focal length.

3. Minimal parametrization for P3.5P

Let us first examine the naive and standard parametrization for a camera with unknown focal length. We will show its limitations in camera pose estimation and then propose a novel compact parametrization for P3.5P.

3.1. Limitations of the standard parametrization

Except for the unknown focal length f , the camera intrinsic is assumed to have square pixels, no skew, and $(0, 0)$ principal point. Let R be the camera rotation and C be the camera center, the projection matrix can be written as

$$P = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} R [I \mid -C]. \quad (3)$$

Such a naive parametrization is not ideal for minimal absolute pose problems, due to the following two reasons.

First, it has a two-fold redundancy. There always exists a different set of camera parameters for the same projection

matrix as follows

$$P = \begin{bmatrix} -f & & \\ & -f & \\ & & 1 \end{bmatrix} \left(\begin{bmatrix} -1 & & \\ & -1 & \\ & & 1 \end{bmatrix} R \right) [I \mid -C]. \quad (4)$$

This alternative interpretation has a negated focal length and an additional 180° rotation around the z -axis. When using both focal length and camera rotation, the number of solutions to minimal pose estimation problems would be unfortunately doubled. Two examples of such solution doubling effect can be found in [11, 12], where the camera parameters similarly include $1/f$ and rotation. Most other absolute pose solvers avoid such a straightforward camera parametrization, and instead either represent the projection matrix using linear combination of null space [21, 15, 4] or solve for the point depths [1, 3, 23].

Second, there exists an infinite number of **trivial solutions** for coplanar points. The 7 or 8 image coordinate constraints can be trivially satisfied by the following:

$$P^1 = P^2 = 0_{1 \times 4} \text{ and} \quad (5)$$

$$P^3 \widetilde{X}_i = 0, \quad i = 1 \dots 4 \quad (6)$$

From the parametrization in Equation 3, we see that Equation 5 can be simply satisfied by $f = 0$. Equation 6 requires the 4 points to be on the camera plane. Because the full 3 degrees of freedom of the rotation always allow to align the normal of the camera plane with the 3D point plane, the remaining one degree freedom of in-plane rotation leads to an infinite number of trivial solutions. This degeneracy can be avoided by parametrizing $1/f$ instead of f as in [11, 12], but the number of solutions is still doubled.

3.2. A new camera parametrization

We seek for new camera parametrization that can avoid the solution doubling as well as the planar degeneracy. Noticing that the alternative camera interpretation has an additional rotation around the z -axis on the left side of the original rotation matrix, we decompose the rotation matrix R into two independent components:

$$R = R_\theta R_\rho = R(z, \theta) R(\Phi, \rho) \quad (7)$$

where R_θ is a rotation by θ around the z -axis and R_ρ is a rotation by ρ around a unit axis Φ in the xy plane ($\Phi \perp z$). See Table 1 for the decomposition we derive using quaternions. In the literature of affine structure from motion, a similar rotation decomposition of $R_\rho R_\theta$ was proposed for two-view reconstruction by Koenderink and van Doorn [14, 17], which only differs by the order of R_θ and R_ρ . For affine structure from motion, R_θ first aligns the epipolar lines of two images and R_ρ corresponds to the bas-relief ambiguity of the 3D reconstruction.

	R	R_θ	R_ρ
$\begin{bmatrix} \eta_w \\ \eta_x \\ \eta_y \\ \eta_z \end{bmatrix}$	$\eta_w^2 + \eta_z^2 \neq 0$	$\begin{bmatrix} \eta_w \\ 0 \\ 0 \\ \eta_z \end{bmatrix}$	$\begin{bmatrix} 1 \\ \frac{\eta_w \eta_x + \eta_z \eta_y}{\eta_w^2 + \eta_z^2} \\ \frac{\eta_w \eta_y - \eta_z \eta_x}{\eta_w^2 + \eta_z^2} \\ 0 \end{bmatrix}$
	$\eta_w = \eta_z = 0$	$\begin{bmatrix} \cos \frac{\theta}{2} \\ 0 \\ 0 \\ \sin \frac{\theta}{2} \end{bmatrix}$	$\begin{bmatrix} 0 \\ \eta_x \cos \frac{\theta}{2} + \eta_y \sin \frac{\theta}{2} \\ \eta_y \cos \frac{\theta}{2} - \eta_x \sin \frac{\theta}{2} \\ 0 \end{bmatrix}$

Table 1. The decomposition of rotation matrix $R = R_\theta R_\rho$. Let $\eta = [\eta_w, \eta_x, \eta_y, \eta_z]^\top$ be the quaternion representation of R , it is straightforward to derive the above decomposition using the Hamilton product of quaternions. The decomposition is unique if $\eta_w^2 + \eta_z^2 \neq 0$. Otherwise, when $\eta_w = \eta_z = 0$, there is a valid decomposition for any angle θ , which is not unique. Note when $\eta_w = \eta_z = 0$, R and R_ρ are 180 degree rotations around some axis in the xy plane, which is rare in practice.

Substituting R with the two decomposed rotations in Equation 3, the projection matrix is transformed into

$$P = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} R(z, \theta) R(\Phi, \rho) [I \mid -C] \\ = \begin{bmatrix} f \cos \theta & -f \sin \theta & \\ f \sin \theta & f \cos \theta & \\ & & 1 \end{bmatrix} R(\Phi, \rho) [I \mid -C]. \quad (8)$$

Let $f_c = f \cos \theta$ and $f_s = f \sin \theta$, we define a rotated calibration matrix K_θ for convenience

$$K_\theta = K_\theta(f_c, f_s) = \begin{bmatrix} f_c & -f_s & \\ f_s & f_c & \\ & & 1 \end{bmatrix}. \quad (9)$$

Finally, we arrive at a new parametrization for our camera that has a single unknown focal length:

$$P = K_\theta R_\rho [I \mid -C] = [K_\theta R_\rho \mid T], \quad (10)$$

where the translation vector $T = -K_\theta R_\rho C$ is basically the 4-th column of the projection matrix. For solving the minimal pose problem, we parametrize the translation vector T instead of the camera position C , which reduces the complexity and redundancy in the multiplications.

The calibration matrix and the rotation R_θ around the z -axis are now combined into the rotated calibration matrix $K_\theta(f_c, f_s)$. The new parametrization successfully eliminates the two-fold redundancy of negated focal lengths,

because the two alternative interpretations now have the same parameters: $f_c = f \cos \theta = (-f) \cos(\theta + \pi)$ and $f_s = f \sin \theta = (-f) \sin(\theta + \pi)$.

3.3. The P3.5P problem with 10 solutions

In order to build a problem with a minimal finite number of solutions, the decomposition of $R = R_\theta R_\rho$ needs to be unique. As shown in Table 1, there is an infinite number of possible decomposition if the quaternion of R is in the form of $[0, *, *, 0]^\top$. The problem however can be easily avoided. When there exists a finite number of possible cameras, we can always rotate the world coordinate system to avoid having such solutions, which we will discuss the details in Section 4.4. From a moment, we assume the camera rotation R has no solution in the quaternion form of $[0, *, *, 0]^\top$.

Since the quaternion of R is not in the form of $[0, *, *, 0]^\top$, the w -component of the quaternion of R_ρ is non-zero. We can use a homogeneous quaternion vector $[1, q_x, q_y, 0]^\top$ to represent the rotation R_ρ , which gives a degree 2 polynomial matrix as follows

$$R_\rho = \begin{bmatrix} 1 + q_x^2 - q_y^2 & 2q_x q_y & 2q_y \\ 2q_x q_y & q_y^2 - q_x^2 + 1 & -2q_x \\ -2q_y & 2q_x & 1 - q_x^2 - q_y^2 \end{bmatrix}. \quad (11)$$

Including the rotated calibration matrix $K_\theta(f_c, f_s)$, the rotation R_ρ parametrized by q_x and q_y , and the translation vector $T = [t_x, t_y, t_z]^\top$, our P3.5P problem has the following 7 unknowns to solve

$$\{f_c, f_s, q_x, q_y, t_x, t_y, t_z\} \quad (12)$$

The 7 image coordinate constraints are now polynomials of degree 3 in the 7 parameters.

By modeling the 7 polynomials in Macaulay 2, we find it producing 10 solutions for both general points and coplanar points. When the 4 points are coplanar, 2 out of the 10 solutions are trivial solutions. Equation 5 is now trivially satisfied by $f_c = f_s = t_x = t_y = 0$. Equation 6 generally gives 3 independent constraints on P^3 , while P^3 also has 3 degrees of freedom (2 for R_ρ and 1 for t_z). This turns out to give exactly the 2 trivial solutions. We are able to avoid the planar degeneracy because the unknown rotation R_ρ in P^3 has only 2 degrees of freedom.

4. Solving the P3.5P problem

In this section, we first eliminate the translation vector T from the image coordinate constraints, then further eliminate f_c and f_s from the problem, and finally build a polynomial system of degree 6 for solving q_x and q_y . An outline of our algorithm can be found in Table 2.

1. Detect and fix possible degeneracy	Sec. 4.4
2. Eliminate the translation vector T	Sec. 4.1
3. Eliminate the focal length f_c and f_s	Sec. 4.2
4. Solve for R_ρ using Gröbner basis	Sec. 4.3
5. Solve for f_c, f_s and T using SVD	Sec. 4.3
6. Add degeneracy-fix rotation to solutions	Sec. 4.4
7. Filter solutions using the extra coordinate	Sec. 4.5

Table 2. The outline of our new P3.5P algorithm.

4.1. Eliminating the translation vector T

Let R_ρ^1, R_ρ^2 and R_ρ^3 be the three rows of R_ρ , the perspective projection of point X_i can be written as

$$P \widetilde{X}_i = K_\theta R_\rho X_i + T = \begin{bmatrix} (f_c R_\rho^1 - f_s R_\rho^2) X_i + t_x \\ (f_s R_\rho^1 + f_c R_\rho^2) X_i + t_y \\ R_\rho^3 X_i + t_z \end{bmatrix}$$

Let λ_i be the depth of the 3D point $X_i \in \{X_1, X_2, X_3\}$, for which we have a complete image point (x_i, y_i) :

$$\lambda_i [x_i, y_i, 1]^\top = P \widetilde{X}_i. \quad (13)$$

We can obtain the translation T as a function of λ_i

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \lambda_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} - P \widetilde{X}_i = \begin{bmatrix} \lambda_i x_i - (f_c R_\rho^1 - f_s R_\rho^2) X_i \\ \lambda_i y_i - (f_s R_\rho^1 + f_c R_\rho^2) X_i \\ \lambda_i - R_\rho^3 X_i \end{bmatrix} \quad (14)$$

By substituting the translation vector T into the perspective projection, we obtain for any 3D point X_j the following

$$P \widetilde{X}_j = \begin{bmatrix} (f_c R_\rho^1 - f_s R_\rho^2) (X_j - X_i) + \lambda_i x_i \\ (f_s R_\rho^1 + f_c R_\rho^2) (X_j - X_i) + \lambda_i y_i \\ R_\rho^3 (X_j - X_i) + \lambda_i \end{bmatrix}.$$

For any image coordinate x_j and y_k such that $j \neq i$ and $k \neq i$, the image coordinate constraints now become polynomials in $\{f_c, f_s, q_x, q_y, \lambda_i\}$ as follows:

$$(f_c R_\rho^1 - f_s R_\rho^2 - x_j R_\rho^3)(X_j - X_i) + \lambda_i (x_i - x_j) = 0 \quad (15)$$

$$(f_s R_\rho^1 + f_c R_\rho^2 - y_k R_\rho^3)(X_k - X_i) + \lambda_i (y_i - y_k) = 0 \quad (16)$$

There are a total of 5 such polynomials defined by the remaining 5 image coordinates (other than x_i and y_i).

Because λ_i appears linearly in the 5 new image coordinate constraints from point X_i (Equation 15 and 16), we can eliminate λ_i using any 2 of the 5 equations. For example, by multiplying Equation 15 by $(y_i - y_k)$ and Equation 16 by $(x_i - x_j)$ and then subtracting them, we can obtain the following polynomial constraint:

$$0 = (y_i - y_k)(f_c R_\rho^1 - f_s R_\rho^2 - x_j R_\rho^3)(X_j - X_i) - (x_i - x_j)(f_s R_\rho^1 + f_c R_\rho^2 - y_k R_\rho^3)(X_k - X_i), \quad (17)$$

which is of degree 3 in $\{f_c, f_s, q_x, q_y\}$. Since the polynomial equation encodes the geometric relationship between 4 image coordinates, we simply call it the **quadruple constraint** by $\{(x_i, y_i), x_j, y_k\}$.

Given the 5 image coordinate constraints from a 3D point X_i , there are 10 possible equation pairs, which give 10 quadruple constraints. We find that there are only 4 linearly independent equations from the 10. Considering the 3×10 equations from $\{X_1, X_2, X_3\}$, there turns out to be still only 4 linearly independent equations. For simplicity, we manually pick 4 linearly independent quadruple constraints out of the 3×10 equations:

$$\begin{aligned} &\{(x_1, y_1), x_2, y_3\}, \\ &\{(x_1, y_1), x_3, y_2\}, \\ &\{(x_2, y_2), x_4, y_3\}, \\ &\{(x_3, y_3), x_4, y_2\}. \end{aligned}$$

These 4 equations use exactly 7 image coordinates without y_4 , which clearly differ with the existing P4P methods.

4.2. Eliminating the focal length f_c and f_s

Because f_c and f_s again appear linearly in the polynomials for the quadruple constraints, we can rewrite the 4 polynomials as the multiplication of a 4×3 degree 2 polynomial matrix $F(q_x, q_y)$ with vector $[f_c, f_s, 1]^T$.

$$\underbrace{F(q_x, q_y)}_{4 \times 3} \begin{bmatrix} f_c \\ f_s \\ 1 \end{bmatrix} = 0 \quad (18)$$

Since these equations have non-trivial solutions, we must have $\text{rank}(F(q_x, q_y)) \leq 2$. Therefore, each 3×3 sub-matrix of $F(q_x, q_y)$ is rank deficient and has 0 determinant. By computing the determinants of the 4 possible 3×3 sub-matrices of $F(q_x, q_y)$, we are able to eliminate f_c and f_s and derive 4 polynomials in just two unknowns $\{q_x, q_y\}$.

It can be verified that the 4 determinant polynomials give the expected number of 10 solutions. Because there are only 2 variables, any subset of 2 or 3 of the 4 determinant polynomials still give a finite number solutions, but there may be false solutions. In fact, any 2 polynomials of the 4 give 18 solutions, and 3 of the 4 give 10 or 12 solutions.

Since each element of $F(q_x, q_y)$ is of degree 2, the obtained polynomials of the 3×3 determinants are of degree 6. In comparison, the polynomials used in recent general P4P methods [3] and [23] are of degree 8 and 7 respectively. By solving the true minimal P3.5P problem instead of the over-constrained P4P problem, we are able to obtain simpler polynomials with lower degrees, which undoubtedly would lead to an improved speed.

4.3. Solving for R_ρ and then other parameters

By employing the standard Gröbner basis eigenvalue method, we solve the 4 degree 6 polynomials for the two

unknowns q_x and q_y . If we multiply the 4 polynomials with $\{q_x^2, q_x q_y, q_x, q_y, 1\}$, there are 20 polynomials with 43 monomials of degree 8 and lower. It can be verified that these 20 polynomials are linearly independent in general and contain all the elements need for the Gröbner basis with respect to the GRevLex order. The 20 polynomials can be stacked into a 20×43 coefficient matrix. After Gauss-Jordan elimination of the matrix, it is then straightforward to construct the 10×10 action matrix for solving the polynomials. In fact, 13 out of the 43 monomials are not used by the action matrix construction, so we actually only need to construct a 20×30 coefficient matrix for the G-J elimination. This coefficient matrix is often called elimination template. It is worth noting that our elimination template is much smaller than those of previous general methods, which are 154×180 in [3], later reduced to 53×63 in [2] by an exhaustive search, and most recently 36×52 for the polynomials of [23].

Instead of using Gröbner basis to solve the polynomial system, [23] opted for two alternative methods. The polynomial eigenvalue (polyeig) factorization is shown to have higher accuracy, while the characteristics polynomial (CP) technique can lead to higher speed with a sacrifice to the accuracy. Both strategies should be applicable to our polynomial system, if we use only 2 of the 4 bivariate determinant polynomials. In this work, we have only explored the Gröbner basis eigenvalue method, which is suitable for pure C++ implementation and has high accuracy. We find our GB-based P3.5P solver has a comparable accuracy as the polynomial eigenvalue solver of [23], while the P3.5P solver has a much higher speed thanks to the reduced size of the elimination template.

For each solution of q_x and q_y , the remaining unknowns can be solved using singular value decomposition. To be specific, $[f_c, f_s, 1]^T$ is the null space vector of $F(q_x, q_y)$. With the recovered f_c and f_s , another linear system can be built using the image coordinate constraints to recover the translation vector T .

4.4. Detecting and avoiding possible degeneracies

As discussed earlier, our proposed parametrization is degenerate if there exists a solution with R in the quaternion form of $[0, *, *, 0]^T$. We may detect possible degeneracies by explicitly solving for such cameras, which turns out to have the following form

$$P = \begin{bmatrix} a & b & 0 & t_x \\ b & -a & 0 & t_y \\ 0 & 0 & 1 & t_z \end{bmatrix}. \quad (19)$$

Similar to the elimination of translation in Section 4.1, the quadruple constraints now give 4 linear constraints for a and b , which can be written as

$$D_{4 \times 3} [a, b, 1]^T = 0.$$

If $\text{rank}(D) = 3$, the linear system has no solution and the normal P3.5P algorithm can proceed. If $\text{rank}(D) \leq 1$, the linear system has an infinite number of solutions, which means the P3.5P or P4P problems cannot be solved.

If $\text{rank}(D) = 2$, the linear system has a single solution, then our parametrization of rotation is degenerate for the input. To avoid such solutions, we can simply add an auxiliary rotation to the world coordinate system (e.g. a random rotation). Since there exists up to 10 solutions, such an auxiliary rotation can be easily found. After finding the solutions for the transformed non-degenerate P3.5P problem, the solutions to the original problem can be obtained by adding the auxiliary rotation.

If not adding the auxiliary rotation for degenerate data, we find that the rank of the elimination template be less than 20, which would fail to solve the polynomial system. This degeneracy has an extremely low probability to occur in real problems, so we recommend skipping the degeneracy avoidance for simplicity in RANSAC-based applications, where some failure iterations are acceptable. For synthetic problems, we do find it degenerate if all 3D points have the same z value, for which the degeneracy can be fixed by rotating 90 degrees around the x -axis.

4.5. Filtering the candidate solutions

Now the P3.5P algorithm has used 7 image coordinates to find up to 10 possible solutions, and there is still one remaining image coordinate to utilize. Intuitively, the remaining image coordinate can be used to verify the recovered solutions and filter out the invalid ones. We compute the reprojection error for the last image coordinate, and reject a solution if the error is large. In addition, we can also discard the solutions that result in negative point depths. Note similar solution filtering is applicable to other existing P4P solvers (e.g. [3, 23]), and these solvers are supposed to produce approximately the same number of filtered solutions.

5. Experiments

In this section, we evaluate the performance of the proposed P3.5P algorithm and compare to existing general P4P solvers [2, 23]. Unless filtering is explicitly mentioned, all the real solutions from P3.5P are used in the evaluation. Out of the various solvers by [2], we pick the most accurate ratio-based GB solver (139×153 elimination template) and call it **Bujnak-ratio-GB**. The method by [23] has two variants that solve the polynomials by polynomial eigenvalue (polyeig) and characteristics polynomial (CP) respectively, which we refer to as **Zheng-polyeig** and **Zheng-CP**.

5.1. Solver speed

Table 3 shows the speed comparison for our P3.5P solver, three example GB-based P4P solvers from [2] and

the non-GB-based P4P solvers from [23]. Among the GB-based solvers, our P3.5P solver has the smallest elimination template and is clearly the fastest one. Thanks to the reduced size of the elimination template, the speed of our method is close to the *Zheng-CP* solver, which however has slightly worse accuracy. Note that the characteristic polynomial method is also applicable to our polynomial system if higher speed is desired.

Solver	Polynomial solving method	Time
Our P3.5P	GB, 20×30 G-J elimination	0.109ms
Ratio [2]	GB, 53×63 G-J elimination	0.336ms
	GB, 86×96 G-J elimination	0.929ms
	GB, 139×153 G-J elimination	3.320ms
Zheng [23]	GB, 36×52 G-J elimination	0.257ms
	polyeig	1.648ms
	Characteristic polynomial	0.067ms

Table 3. Comparison of the solver speed (the polynomial solving step). The timings for the GB-based solvers are benchmarked using G-J elimination and Eigen decomposition of random matrices.

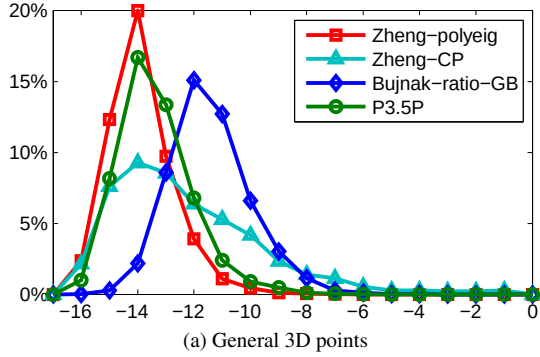
5.2. Stability and accuracy

Similar to the experiments by [23], we first generate the 3D points in the camera coordinate system and their corresponding observations, and then produce the 3D points in the world coordinate by applying a random 3D similarity transformation. The focal length is randomly chosen from 200 to 2000 pixels. For general 3D points, the points in the camera coordinate are randomly distributed in the box of $[-2, 2] \times [-2, 2] \times [4, 8]$. For coplanar points, the points are generated similarly with a common z value and then rotated randomly around the centroid.

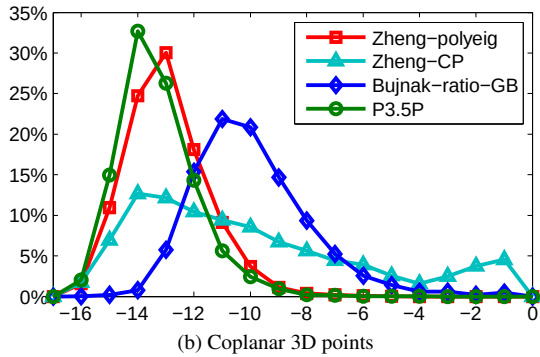
We compare the recovered focal lengths $\{f\}$ with the ground truth f_g , and examine the relative focal length error $\delta = \left| \frac{f - f_g}{f_g} \right|$. If there are multiple solutions, the focal length with smallest error is taken. If a solver returns no solution, the error is considered infinite. Since the focal length error is a reasonable indicator for the errors in other parameters, this paper evaluates the accuracy of only the focal length solutions due to the limit of space.

Figure 2 shows the distribution of $\log_{10} \delta$ for noise free data accumulated from 4000 runs. For both general and coplanar 3D points, it can be seen that our P3.5P has an overall similar numeric stability as *Zheng-polyeig* and outperforms *Zheng-CP* and *Bujnak-ratio-GB*. We believe a further improvement is possible by using polyeig.

Now we add zero-mean Gaussian noise to the image points, and study the accuracy of the solvers under varying level of noise. Taking into account of image normalization, we use a camera with fixed focal length of 200 pixels. Figure 3 reports the median relative focal length error δ from



(a) General 3D points



(b) Coplanar 3D points

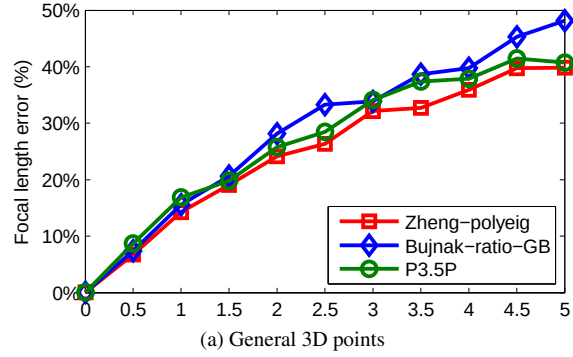
Figure 2. Numeric stability on noise-free data. The horizontal axis is \log_{10} of the relative focal length error. The proposed P3.5P algorithm has comparable stability as *Zheng-polyeig* for both general 3D points and coplanar 3D points.

500 runs per noise level. Overall, the accuracy of our P3.5P is sufficiently good for real applications. Nevertheless, we do find that our P3.5P solver has slightly larger errors as the noise level grows, which can be explained by the over-fitting of the 7 image coordinates. Please note the solutions from the P3.5P solver always have near 0 reprojection errors for 7 image coordinates. One possible future work is to improve the accuracy by incorporating the remaining image coordinate into the new polynomial system.

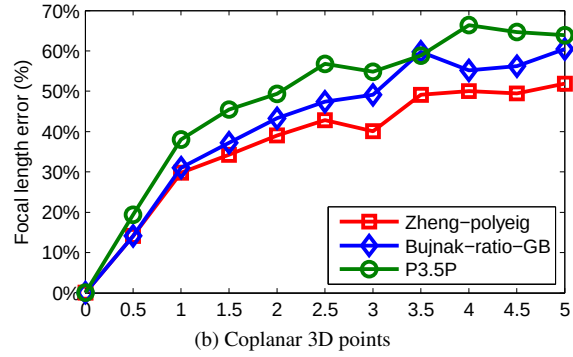
5.3. Number of solutions

We find that the P3.5P solver normally produces 2, 4 or 6 real solutions (without filtering), of which the distribution can be found in Figure 4. For coplanar points, we notice the pattern of real solutions pairs that have equal focal length and mirrored poses. Since there are up to 8 non-trivial solutions for coplanar points, this pairing matches that fact that P3P has a maximum number of 4 solutions.

For the solution filtering step, we use an adaptive threshold $\epsilon * f$, where $f = \sqrt{f_c^2 + f_s^2}$ is the focal length of the solution. To be specific, we use $\epsilon = 0.01$, which corresponds to about a 0.57 degree angular error. After filtering the solutions according to the reprojection errors and the sign of the point depth, we find both P3.5P and P4P solvers



(a) General 3D points



(b) Coplanar 3D points

Figure 3. Median relative focal length error δ under varying noise levels, where the horizontal axis is the standard deviation of the Gaussian noise in pixels. Since the ground truth focal length is 200 pixels, the experimented noise levels are relatively high.

typically producing a single solution for good data, which is reasonable for an over-determined problem. See Table 4 for the detailed numbers. Similarly when the point data contains outliers, it is unlikely that all the reprojection errors are small for the 4 points. We contaminate the image points by adding 100 pixel offset to a random coordinate (out of 8), and experiments show all the solvers usually producing 0 solution after the filtering.

Since P4P is over-determined, we should discard these obviously invalid solutions. The simple filtering is particularly good for efficient RANSAC applications, because a much smaller set of candidate solutions need to be tested.

5.4. Real images

We have integrated the proposed P3.5P solver into an incremental reconstruction system, and found it working well for 3D reconstruction of real images. After recovering the camera poses, we apply the PMVS algorithm [7] to compute the dense models. Figure 5 shows the dense reconstruction examples for both planar scenes and non-planar scenes. The well recovered camera pose leads to the satisfactory dense reconstruction.

	No noise, no outliers				With outlier	
	Mean # of real solutions		Runs with $\delta < 10^{-8}$		Mean # of real solutions	
	General	Coplanar	General	Coplanar	General	Coplanar
Zheng-polyeig	4.270	3.938	99.90%	99.32%	3.441	2.840
Zheng-CP	2.424	2.458	91.30%	68.53%	1.575	1.252
Bujnak-Ratio-GB	3.685	3.374	96.88%	79.45%	2.945	2.368
P3.5P	4.563	3.557	99.65%	99.55%	3.981	2.698
P3.5P w/ filtering	1.129	1.209	99.65%	99.15%	0.095	0.147

Table 4. Comparison of the the number of solutions. Although we only show the P3.5P stats for the filtering, the other solvers produce approximately the same number of solutions if we filter the solutions by point depth sign and the summed reprojection errors.

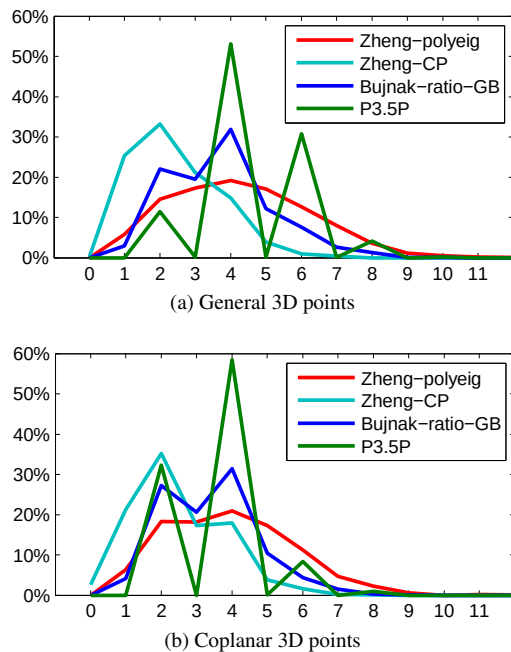


Figure 4. The number of real solutions. Here we count all the real solutions from P3.5P, while the code of other solvers have already done some filtering. Note the even number of solutions are due to the fact imaginary eigen values show up in pairs.

6. Conclusions

This paper presents the first true minimal solver for the pose estimation problem with a single unknown focal length, which has 7 degrees of freedom. The new minimal solver can recover up to 10 solutions from 7 image coordinate (3.5 points). Our P3.5P solver has comparable accuracy as the state-of-the-art P4P solvers, and has the best speed among the accurate solvers thanks to the smallest GB elimination template. The extra image coordinate can be used to verify and filter the candidate solutions, and a single solution is usually produced after the filtering.

In order to solve the P3.5P problem, we have introduced a new parametrization for cameras with unknown focal length, which we believe having great potential in many

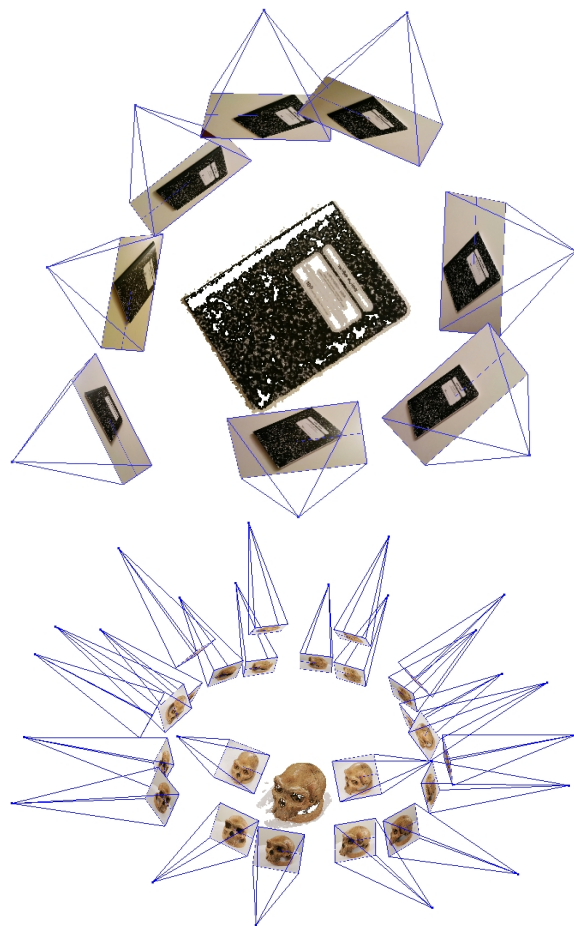


Figure 5. Two example reconstructions, for which the incremental reconstruction uses the P3.5P based camera resection. The first scene is a planar notebook and the second is a skull model. The dense point clouds are generated by the PMVS software [7].

other uncalibrated camera problems, for example, [11, 12].

Acknowledgements The author thanks Enliang Zheng for many fruitful discussions on minimal solvers. The author would also like to thank Henrik Stewénius for sharing his M2 code [19], and Yinqiang Zheng and Martin Bujnak for providing their P4P code.

References

- [1] M. A. Abidi and T. Chandra. A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation. In *PAMI*, 1995. 1, 2, 3
- [2] M. Bujnák. Algebraic solutions to absolute pose problems, 2012. 1, 2, 5, 6
- [3] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the P4P problem for camera with unknown focal length. In *CVPR*, 2008. 1, 2, 3, 5, 6
- [4] M. Bujnak, Z. Kukelova, and T. Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In *ACCV*, 2010. 3
- [5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1
- [6] J. Frahm, P. Fite Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *ECCV*, pages IV: 368–381, 2010. 1
- [7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 2010. 7, 8
- [8] D. R. Grayson and M. E. Stillman. Macaulay 2, a software system for research in algebraic geometry, 2002. 2
- [9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. 2004. 1
- [10] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009. 1
- [11] K. Josephson and M. Byrod. Pose estimation with radial distortion and unknown focal length. In *CVPR*, 2009. 3, 8
- [12] K. Josephson, M. Byrod, F. Kahl, and K. Astrom. Image-based localization using hybrid feature correspondences. In *CVPR*, 2007. 3, 8
- [13] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR*, 2011. 1
- [14] J. J. Koenderink and A. J. Van Doorn. Affine structure from motion. *Journal of the Optical Society of America*, 1991. 3
- [15] Y. Kuang and K. Åström. Pose estimation with unknown focal length using points, directions and lines. In *ICCV*, 2013. 3
- [16] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *ECCV*, 2012. 1
- [17] L. S. Shapiro, A. Zisserman, and M. Brady. 3d motion recovery via affine epipolar geometry. *IJCV*, 1995. 3
- [18] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH*, 2006. 1
- [19] H. Stewénus, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *Workshop on Omnidirectional Vision*, 2005. 8
- [20] B. Triggs. Autocalibration from planar scenes. In *ECCV*, 1998. 2
- [21] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *CVPR*, 1999. 1, 2, 3
- [22] C. Wu. Towards linear-time incremental structure from motion. In *3DV*, 2013. 1
- [23] Y. Zheng, S. Sugimoto, I. Sato, and M. Okutomi. A general and simple method for camera pose and focal length determination. In *CVPR*, 2014. 1, 2, 3, 5, 6