Noname manuscript No.
(will be inserted by the editor)

Infinitesimal Plane-based Pose Estimation

Toby Collins · Adrien Bartoli

Received: date / Accepted: date

Abstract Estimating the pose of a plane given a set of point correspondences is a core problem in computer vision with many applications including Augmented Reality (AR), camera calibration and 3D scene reconstruction and interpretation. Despite much progress over recent years there is still the need for a more efficient and more accurate solution, particularly in mobile applications where the run-time budget is critical. We present a new analytic solution to the problem which is far faster than current methods based on solving Pose from n Points (PnP) and is in most cases more accurate. Our approach involves a new way to exploit redundancy in the homography coefficients. This uses the fact that when the homography is noisy it will estimate the true transform between the model plane and the image better at some regions on the plane than at others. Our method is based on locating a point where the transform is well estimated, and using only the local transformation at that point to constrain pose. This involves solving pose with a local non-redundant 1^{st}-order PDE. We call this framework Infinitesimal Plane-based Pose Estimation (IPPE), because one can think of it as solving pose using the transform about an infinitesimally small region on the plane. We show experimentally that IPPE leads to very accurate pose estimates. Because IPPE is analytic it is both extremely fast and allows us to fully characterise the method in terms of degeneracies, number of returned solutions, and the geometric relationship of these solutions. This characterisation is not possible with state-of-the-art PnP methods since they solve pose via numerical root-finding.

Keywords Plane · Pose · SfM · PnP · Homography

This research has received funding from the EU FP7 ERC research grant 307483 FLEXABLE. Code is available at http://www.tobycollins.net/research/IPPE.

Toby Collins
ALCoV-ISIT, UMR 6284 CNRS/UdA
E-mail: Toby.Collins@gmail.com

Adrien Bartoli
ALCoV-ISIT, UMR 6284 CNRS/UdA
E-mail: Adrien.Bartoli@gmail.com

1 Introduction

Plane-based Pose Estimation (PPE) is a fundamental problem in computer vision and is the basis for many important applications. At its core PPE means recovering the relative pose of a *model plane* with respect to a camera's 3D coordinate frame from a single image of that plane. Applications include estimating the pose of textured planar surfaces visible in an image or using planar markers to perform AR [22,29]. Another important application is camera calibration using views of a planar calibration target [2,40,35,13]. In the classic pipeline, first the camera's intrinsics are estimated, then PPE is performed to obtain the camera's extrinsics, which is followed by joint intrinsic/extrinsic refinement. Other important applications of PPE include camera/projector calibration [4] and Shape-from-Texture [6,19,26,25].

There exist already many methods for solving PPE. These can be broken down into two main categories. The first category solves PPE by decomposing the associated plane-to-view homography [40,35,6,30]. These methods are known as *Homography Decomposition (HD)* methods. The second category treats PPE as a special case of the general rigid pose estimation problem from point correspondences. When the camera is perspective, this is known as the *PnP problem* where n denotes the number of correspondences. We use the term Planar-PnP to be a general PnP method which can

handle the plane as a special case. HD works using the fact that the transform induced by perspective or affine projection of a plane is a homography. Once estimated the homography can be factored very efficiently to give a pose estimate. Solutions to HD exist for perspective cameras [40,35] and for weak-perspective cameras [6, 30]. We call these PHD methods and WPHD methods respectively. PnP methods work by optimising pose using a cost function related to the correspondence transfer error. This is the error in the predicted positions of point correspondences compared with their measured positions. Research on PnP has either focused on the special cases of $n = 3$ and $n = 4$ [7,10,11,15,33,14,21], or for solving the problem with arbitrary n [33,9,1,24, 23,28,30,20,34,18].

There are two main differences between PHD and Planar-PnP. Firstly state-of-the-art Planar-PnP methods significantly outperform PHD methods with respect to noise. Secondly, PHD methods return only a single solution. This means they can fail badly under certain imaging conditions. For example, when the homography is affine PPE is not solvable uniquely [34]. When in weak-perspective conditions there exists a rotation ambiguity that corresponds to an unknown reflection of the plane about the camera's z-axis [30]. This can happen when imaging small planes, or planes at a distance significantly larger than the camera's focal length. In these conditions the reprojection error of the two solutions can both be explained by noise, and so the single PHD solution can be far from the true solution about 50% of the time. By contrast most recent Planar-PnP methods can return multiple solutions which are minima of their associated cost functions. Ideally one of these corresponds to the true solution.

Approach, motivation and overview. The current approach to achieve high-accuracy PPE is to first obtain an initial estimate using a non-iterative PHD or Planar-PnP method, and then iteratively refine it by optimising the reprojection error. The refined solution gives the Maximum Likelihood (ML) estimate with a Gaussian IID noise model for the correspondences. If the initialisation method returns multiple solutions (which correspond to the minima of some cost function), then each of these are refined and the one with the lowest reprojection error is usually used as the pose estimate. There is an ongoing demand for developing a more efficient initialisation method. Ideally one that returns few solutions, and ultimately be sufficiently accurate to eliminate the need for refinement altogether. Achieving this is particularly important for mobile or embedded applications where reducing the runtime cost is imperative. Given that PHD is significantly faster than Planar-PnP methods, we aim to find a solution that performs as quickly as PHD, but with similar or better accuracy than Planar-PnP methods.

PHD uses an 8-DoF homography matrix to estimate the 6-DoF pose. Therefore the problem involves redundant constraints. PHD deals with this redundancy by solving for the best-fitting pose via an algebraic least-squares cost. This assumes that the noise of the homography is IID Gaussian, which is usually not a good approximation [5]. We propose an alternative method that uses the redundancy in the homography coefficients to provide far better pose estimates. Our method is based on the fact that when the homography has been estimated from noisy correspondences, the accuracy of this transform is *spatially-varying*. That is, the homography will predict the transformation better at some points on the model plane than others. Our method is based on identifying a point on the model plane where the transform is best predicted, and then solving pose with a non-redundant, local system using motion information only at that point. We use 1^{st}-order error propagation to find this point, which turns out to be well approximated by the centroid of the points on the model plane.

Our main theoretical contribution is to show how pose can be solved exactly via a PDE using 0^{th} and 1^{st}-order transform information at a point on the model plane. We call our approach Infinitesimal Plane-based Pose Estimation (IPPE). We use this name because it can be thought of as solving pose using transform information within an infinitesimally-small region about a single point on the model plane. To solve IPPE we use the fact that the PPE problem can be cast as a variational problem where we equate two functions. The first function is the composition of 3D rigid embedding and camera projection. The second function is the transform of the plane onto the camera's image, estimated by the homography. These two functions should be equivalent up to noise. The technique we use is to equate these functions by equating their Taylor series representations. By truncating the Taylor series at 1^{st}-order, we form a local 1^{st}-order PDE giving six constraints on pose. We show that these constraints boil down to a univariate quadratic equation whose solution is equivalent to finding the largest singular value of a 2×2 matrix. It is important to note that IPPE is not the same as solving PPE by linearising the projection equations with a Taylor approximation (as is done when the perspective camera is replaced with an affine approximation [30, 20]). That is, *IPPE does not involve any linearisation because is uses an exact representation of the projection equations via the PDE.*

There is also an important connection between IPPE and the P3P problem. Specifically, IPPE is the

solution to the P3P problem when the three points are non-colinear and their mutual separation becomes infinitesimally small. A formal study of P3P for infinitesimally separated points has not been presented in the literature before, so our analysis of IPPE contributes to the understanding of P3P.

IPPE takes as inputs the coefficients of a homography, and so it requires a minimum of four point correspondences. Unlike PHD, IPPE does not break down if the homography is affine. Empirically we show that IPPE performs very well through extensive simulation and real experiments. It consistently performs better than PHD, and in most cases outperforms competitive Planar-PnP methods, whilst being far faster because it solves pose analytically. Furthermore its analytic solution permits a full characterisation of the method. Specifically, we give answers to the following core questions:

Q1 *For what inputs does IPPE guarantee to return at least one physically valid solution?* Answer: All homography matrices with rank greater than 1. This includes affine homographies.
Q2 *How many physically valid solutions does IPPE return?* Answer: One or two.
Q3 *What is the geometric relationship between the returned solutions of IPPE?* Answer: They correspond to a reflection of the plane about a single viewing ray.
Q4 *For what inputs does IPPE estimate translation uniquely?* Answer: All homographies whose rank is greater than 1.
Q5 *For what inputs does IPPE estimate rotation uniquely?* Answer: When the plane is tangential to a 3D sphere centred at the camera's centre-of-projection.
Q6 *Does IPPE introduce any artificial degeneracies?* Answer: It does not.

Understanding whether a method introduces artificial degeneracies is important. When solving PPE with a particular method two types of degeneracies can occur. The first type are called *generic* degeneracies. These occur when the geometric configuration of the camera, plane and point correspondences are such that PPE cannot be solved uniquely. No method can estimate the plane's pose in these cases. The second type are called *artificial* degeneracies. These occur when the PPE problem is well-posed, but the method fails to return the correct solution due to the geometric configuration. For example, PHD introduces at least one artificial degeneracy which is when the homography's perspective terms are negligible. Competitive Planar-PnP methods do not solve pose analytically, and so it is virtually impossible to have complete answers to the above six core questions. They can usually give upper bounds on question 1, but questions 2-6 are left unanswered. For instance [24, 23] can give between zero and four solutions with no theoretical guarantees that the solutions will be geometrically valid.

Paper structure. In §2 we review current state-of-the-art PPE methods. In §3 we present IPPE, its solution and proofs for the six core questions above. In §4 we evaluate IPPE against state-of-the art methods using a large range of simulation experiments. In §5 we evaluate IPPE in three common applications; estimating the pose of a textured planar surface from sparse keypoint correspondences, estimating the pose of a planar checkerboard target and estimating the pose of planar AR markers from four corner correspondences. Finally in §6 we present our conclusions and directions for future work.

Background and notation. Vectors are given in lowercase bold and matrices in upper-case bold. Scalars are given in regular italic. For a 2D matrix $\mathbf{M}$, M_{ij} denote the element in $\mathbf{M}$ at row i, column j. For a vector $\mathbf{v}$, v_i denotes its i^{th} element. We use $\mathbf{M}_{ij}$ to denote the top-left $i \times j$ submatrix of $\mathbf{M}$. We use $SS_{2\times 2}$ to denote the 2×2 sub-Stiefel manifold in SO_3 (*i.e.* $\mathbf{M}$ is in $SS_{2\times 2}$ if it is a 2×2 submatrix of some 3×3 rotation matrix). $\|\mathbf{M}\|_F$ denotes the Frobenius norm of a matrix and $\|\mathbf{v}\|_2$ denotes the $\mathcal{L}_2$ norm of a vector. $\mathbf{I}_k$ denotes the $k \times k$ identity matrix. We use $\hat{\mathbf{M}}$ to denote a noisy measurement of $\mathbf{M}$. We define the model plane in world coordinates on the plane $z = 0$. We denote the rigid transform mapping a point in world coordinates to the camera's coordinate frame by the rotation $\mathbf{R} \in SO_3$ and translation $\mathbf{t} \in \mathbb{R}^3$. We use $\mathbf{s}_i^\top$ to be the i^{th} row of $\mathbf{R}$ and $\mathbf{r}_i$ to be the i^{th} column of $\mathbf{R}$. We assume that the camera is calibrated, and any distortion effects have been undone as a pre-processing step. For perspective cameras the projection of a point in the camera's coordinate frame onto the image is determined by the camera's intrinsic calibration matrix $\mathbf{K}$:

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where f_x and f_y denotes the camera's effective focal length along the x and y axes (in pixels), $\mathbf{c} = [c_x, c_y]^\top$ denote the camera's principal point and s denotes the camera's skew. We use the function $\pi([x, y, z]^\top) = z^{-1}[x, y]^\top$ to convert a point $[x, y, z]^\top$ in homogeneous

3D coordinates to inhomogeneous 2D coordinates. Perspective projection of a 3D point $\mathbf{x}$ in camera coordinates is thus given by $\pi(\mathbf{K}\mathbf{x})$. For a point $\mathbf{q}$ in the camera's image we use $\tilde{\mathbf{q}}$ to be its position in normalised coordinates:

$$\tilde{\mathbf{q}} = \mathbf{K}_{22}^{-1}(\mathbf{q} - \mathbf{c}) \tag{2}$$

We define $\{\mathbf{q}_i\}$, with $i \in \{1, 2, ...n\}$ to be the set of n correspondences where $\mathbf{u}_i \in \mathbb{R}^2$ is a point's position on the model plane and $\mathbf{q}_i \in \mathbb{R}^2$ is its position in the image. Without loss of generality we assume $\{\mathbf{u}_i\}$ is zero-centred: $\sum_{i=1}^{n} \mathbf{u}_i = \mathbf{0}$. For a homography matrix $\mathbf{H}$ we define $\Omega_{\mathbf{H}} \subset \mathbb{R}^2$ to be the subspace of $\mathbb{R}^2$ that does not map via $\mathbf{H}$ to the line at infinity: $\mathbf{u} \in \Omega_{\mathbf{H}}$ iff $[H_{31}\, H_{32}\, H_{33}] \left[\mathbf{u}^\top 1\right]^\top \neq 0$.

2 Related Work

2.1 Homography Decomposition (HD)

The first main approach to PPE involves estimating the homography associated with the model-to-image transform. This is followed by HD which gives an analytic solution to pose [40,35,6].

2.1.1 Perspective Homography Decomposition (PHD)

The transform from a point $\mathbf{u} \in \mathbb{R}^2$ on the model plane to the image of a perspective camera is described by the following homogeneous system:

$$\begin{bmatrix} \tilde{\mathbf{q}} \\ 1 \end{bmatrix} \propto [\mathbf{I}_3\, \mathbf{0}] \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ 0 \\ 1 \end{bmatrix} \propto \mathbf{H} \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} \tag{3}$$

Points $\tilde{\mathbf{q}}$ and $\mathbf{u}$ are related by a 3×3 matrix $\mathbf{H}$ known as the *model-to-view homography*. This is given by $\lambda\mathbf{H} = \left[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}\right]$ for some $\lambda \in \mathbb{R}$. We assume $\mathbf{H}$ has been estimated up to noise:

$$\hat{\mathbf{H}} \overset{\text{def}}{=} \mathbf{H} + \varepsilon_H = \lambda^{-1}[\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{t}] + \varepsilon_H \tag{4}$$

where ε_H denotes a 3×3 measurement noise matrix. In the absence of noise the columns of $\hat{\mathbf{H}}$ give $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{t}$ uniquely. Denoting $\hat{\mathbf{h}}_j$ to be the j^{th} column of $\hat{\mathbf{H}}$, λ is given trivially by $\hat{\lambda} = \|\hat{\mathbf{h}}_1\|_2^{-1} = \|\hat{\mathbf{h}}_2\|_2^{-1}$. $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{t}$ are then given by the columns of $\hat{\lambda}\hat{\mathbf{H}}$. From $\mathbf{r}_1$ and $\mathbf{r}_2$ the full rotation matrix is recovered with $\mathbf{R} = [\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{r}_1 \times \mathbf{r}_2]$. With noise, pose can be estimated in a least squares sense as proposed by Zhang [40] and Sturm [35]. Zhang's method works by first relaxing orthonormality between $\mathbf{r}_1$ and $\mathbf{r}_2$. This gives the estimates $\hat{\mathbf{r}}_j = \hat{\lambda}_j \hat{\mathbf{h}}_j$, $j \in \{1, 2\}$ with $\hat{\lambda}_j = \|\hat{\mathbf{h}}_j\|_2^{-1}$. λ and $\mathbf{t}$ are estimated with $\hat{\lambda} = (\hat{\lambda}_1 + \hat{\lambda}_2)/2$, and $\hat{\mathbf{t}} = \hat{\lambda}\hat{\mathbf{h}}_3$. $\mathbf{r}_3$ is then estimated with $\hat{\mathbf{r}}_3 = \hat{\mathbf{r}}_1 \times \hat{\mathbf{r}}_2$. The matrix $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \hat{\mathbf{r}}_3]$ is then projected onto the closest member of SO_3 (in the Frobenius sense) to give a valid rotation matrix using Singular Value Decomposition (SVD).

Sturm's method differs in that it does not first relax orthonormality. Instead ε_H is assumed to be IID Gaussian and the ML solution is found by solving the least squares problem:

$$\min_{\lambda, \mathbf{r}_1, \mathbf{r}_2, \mathbf{t}} \left\| \lambda\hat{\mathbf{H}} - \left[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}\right] \right\|_2^2 \text{ s.t. } [\mathbf{r}_1\, \mathbf{r}_2]^\top [\mathbf{r}_1\, \mathbf{r}_2] = \mathbf{I}_2 \tag{5}$$

This can be solved very efficiently by taking the SVD of the left 3×2 submatrix of $\hat{\mathbf{H}}$. Zhang and Sturm's methods have been shown empirically to perform similarly and are very fast.

2.1.2 Weak-Perspective Homography Decomposition (WPHD)

HD has also been applied to estimate pose with weak-perspective cameras [6,30]. Weak-perspective projection is a linear projection that comes by linearising perspective projection about a point on the camera's optical axis [8,20]. The transform from a point $\mathbf{u} \in \mathbb{R}^2$ on the model plane to the image of a weak-perspective camera is described by the following homogeneous system:

$$\begin{aligned} &\tilde{\mathbf{q}} = \alpha\, [\mathbf{I}_2|\mathbf{0}] \left(\mathbf{R} \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} + \mathbf{t} \right) + \varepsilon_{wp} \Leftrightarrow \\ &\begin{bmatrix} \tilde{\mathbf{q}} \\ 1 \end{bmatrix} = \mathbf{A}_{wp} \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} + \begin{bmatrix} \varepsilon_{wp} \\ 1 \end{bmatrix} \\ &\mathbf{A}_{wp} \overset{\text{def}}{=} \alpha \begin{bmatrix} \mathbf{R}_{22} & \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \\ \mathbf{0}^\top & 1 \end{bmatrix} \end{aligned} \tag{6}$$

$\varepsilon_{wp} \in \mathbb{R}^2$ denotes modelling error introduced by approximating perspective projection with weak-perspective projection. ε_{wp} becomes smaller when the variation of the model's depth is small compared to its average depth (*i.e.* the plane is small and/or its tilt angle is small), and when it projects closely to the camera's principal point [20]. α is the inverse depth of the plane along the optical axis. Given an estimate of $\mathbf{A}_{wp}$, when ε_{wp} is neglected we obtain a unique estimate for α and two estimates for $\mathbf{R}$. These correspond to a two-fold solution ambiguity that are equivalent to reflecting the model in camera coordinates about the plane $z = 0$ [30]. For either estimate of $\mathbf{R}$, $\mathbf{t}$ can be computed uniquely. If the camera's intrinsics are known up to its focal length, the two solutions to $\mathbf{R}$ can be computed, but $\mathbf{t}$ cannot be computed [6].

2.1.3 Comparing Perspective and Weak-Perspective Homography Decomposition

PHD and WPHD are different in three main respects. Firstly in WPHD the system is not redundant, because $\mathbf{A}_{wp}$ gives 6 equations for pose, whereas $\mathbf{H}$ gives 8 equations. Thus noise *must* be neglected in WPHD to have a well-posed problem (because it is exact). Secondly, in WPHD the solution to $\mathbf{R}$ is *always two-fold ambiguous*, except in the special case when the plane is fronto-parallel to the camera [6]. In PHD it is *always unique*. However PHD fails when there is a small amount of noise and $\mathbf{H}$ tends towards being affine [34]. Thirdly, WPHD tends to return worse solutions when $\mathbf{H}$ is not affine due to the modelling error induced by linearising perspective projection.

2.2 Pose Estimation from n Point Correspondences (PnP)

The second main approach to PPE is to solve $\mathbf{R}$ and $\mathbf{t}$ directly from point correspondences. These solve the PnP problem and treat planar models as a special case. PnP methods can be broadly divided into those which solve for small, fixed n, or those which handle the general case. The P3P problem has been studied extensively [7,10,11,15,33,14] and yields up to four solutions when the points are non-colinear. Thus, additional points are required in general to solve pose uniquely [10,39]. For planes, P4P has a unique solution when no 3 points are colinear [21]. Methods which solve the general PnP problem aim to exploit the redundancy of more correspondences to achieve higher accuracy. General PnP methods can be broadly divided into whether they are non-iterative [33,9,1,24,23] or iterative [28,30, 20,34]. Early non-iterative PnP methods were either computationally expensive and did not scale well for large n [33,1], or cheap but quite sensitive to noise [9].

The earliest practical solutions to PnP when n is large involved iteratively approximating perspective projection with an affine camera, using either the weak-perspective camera [30] or the para-perspective camera [20]. Both [30] and [20] solved the problem in a similar way. First pose was computed with the affine camera. Next the error induced by the affine camera approximation was estimated, and this error was fed back into the system to adjust the constraints on pose. Pose was then re-computed with this adjusted system. The process then iterated between estimating the affine approximation error, adjusting the pose constraints and estimating pose. For planar models the pose estimates at each iteration are two-fold ambiguous. To prevent the solution space exploding two-fold with each iteration [30] and [20] pruned the solutions. Two solutions were maintained in [30], with one being eliminated if its perspective reprojection error was large relative to the other. In [20] both solutions were retained in the first iteration. These initialised two search branches, and for each branch only one solution was picked at each iteration (that which had the smallest reprojection error). Finally the single solution was chosen with smallest reprojection error. The major limitation of these methods is that they are rather slow and neither convergence nor optimality can be guaranteed. It also becomes hard to distinguish the correct pose when either noise is large, or the error in the perspective approximation is large. We note here that [30] and [20] are related to our proposed framework in one sense. IPPE instantiated with the para-perspective and weak-perspective cameras give the same solution as the first iteration of [30] and [20] respectively. Where IPPE differs is in being able to properly handle the perspective camera exactly and non-iteratively.

Lu *et al.* [28] proposed an accurate iterative PnP method called RPP that does not make an affine camera approximation. The method is provably convergent and remains one of the best performing PnP methods to date. It was later extended by Schweighofer and Pinz [34] to handle ambiguous cases for planes. In [34] first a pose is estimated using [28], and then a second solution is found corresponding to a local minimum of the reprojection error with respect to a 1-DoF rotation. Thus two solutions are returned and if their reprojection errors are similar it indicates an ambiguous configuration. This method is called RPP-SP. The shortcomings of RPP-SP are that if the solution provided by [28] is poor, it is not likely to find a good second solution. Secondly, it is relatively slow as it relies on [28] to estimate the first pose. Thirdly, it is very difficult to geometrically characterise the pose ambiguity, as the second solution is found from the roots of a 4^{th} order polynomial (two of which are guaranteed to be imaginary).

More recently efficient non-iterative PnP methods have been proposed which are significantly faster than iterative ones. EPnP [23] solves the problem numerically in $O(n)$ by re-representing the 3D points using a weighted sum of four virtual control points. This means the problem size does not grow with n and so scales well for hundreds of points. A Direct Least Squares (DLS) approach was presented in [18]. Very recently RPnP has been proposed [24]. This is another non-iterative $O(n)$ solution which solves PnP by grouping the points into subsets of size three. Each subset corresponds to a P3P problems that is solvable with a 4^{th}-order polynomial. These polynomials are combined in a least-squares

manner to form a 7^{th} order polynomial whose roots each give a solution to pose. The accuracy of RPnP rivals [28], yet is far faster to compute. However, RPnP makes no guarantees on the number of returned solutions. Furthermore a geometric characterisation of its solutions is impossible.

3 Infinitesimal Plane-based Pose Estimation (IPPE)

We now present IPPE. We start by showing that given $\hat{\mathbf{H}}$, we can constrain pose using a local 1^{st}-order PDE. This PDE involves estimates of the 0^{th} and 1^{st}-order terms of the model-to-image transform function at a *single* point on the model plane. These terms are computed analytically from $\hat{\mathbf{H}}$. When $\hat{\mathbf{H}}$ contains errors, the PDE will have error-in-variables. The advantage of this PDE being local is that we are free to apply it anywhere on the model plane. Thus we can apply it at the point where we expect to have the *best* local estimate of the transform. This leads to a reduction of the error-in-variables in the PDE, and leads to a more accurate pose estimate.

3.1 Local Constraints on Pose with a 1^{st}-order PDE

The variational system that describes the rigid embedding and perspective projection of the model plane is simple. We use $s(\mathbf{u}) = \mathbf{R}[\mathbf{u}^\top, 0]^\top + \mathbf{t} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ to denote the true (but unknown) embedding from world to camera coordinates. s is then composed with the projection function π to give the *plane-to-image transform* w:

$$w(\mathbf{u}) \stackrel{\text{def}}{=} \pi\left(\mathbf{H}\left([\mathbf{u}^\top, 1]^\top\right)\right) = (\pi \circ s)(\mathbf{u}) \tag{7}$$

$\mathbf{H}$ is the noise-free homography that transforms the model plane to normalised image coordinates. Consider a single point $\mathbf{u}_0 \in \Omega_{\mathbf{H}}$ that does not map via $\mathbf{H}$ to the line at infinity. Eq. (7) provides us with two 0^{th}-order constraints on s with:

$$w(\mathbf{u}_0) = \pi\left(\mathbf{H}\left([\mathbf{u}_0^\top, 1]^\top\right)\right) = (\pi \circ s)(\mathbf{u}_0),\ w(\mathbf{u}_0) \in \mathbb{R}^2 \tag{8}$$

Because π is smooth and s is a linear transform, w is also smooth. Thus by differentiating Eq. (7) we can obtain four 1^{st}-order constraints on s via the product rule:

$$J_w(\mathbf{u}_0) = (J_\pi \circ s)(\mathbf{u}_0) J_s(\mathbf{u}_0),\ J_w(\mathbf{u}_0) \in \mathbb{R}^{2\times 2} \tag{9}$$

where J_f denotes the function that computes the Jacobian matrix of f. Because s is a rigid transform $J_s(\mathbf{u}_0) = \mathbf{R}_{32}$ so:

$$J_s(\mathbf{u}_0)^\top J_s(\mathbf{u}_0) = \mathbf{I}_2 \tag{10}$$

Our goal is to estimate $\mathbf{t}$ and $\mathbf{R}$ by first estimating $s(\mathbf{u}_0)$ and $J_s(\mathbf{u}_0)$ by solving a 1^{st}-order PDE using Eq. (8), Eq. (9) and Eq. (10). Because Eq. (8) and Eq. (9) give us six constraints (which is the minimal number of constraints needed to estimate pose), we can solve this PDE *pointwise*. That is, for a given $\mathbf{u}_0$ we estimate $s(\mathbf{u}_0)$ and $J_s(\mathbf{u}_0)$, and from these we can recover $\mathbf{t}$ and $\mathbf{R}$.

We write this problem using the unknown vector $\mathbf{x} = s(\mathbf{u}_0) \in \mathbb{R}^3$, which is the 3D position of $\mathbf{u}_0$ in the camera's 3D coordinate frame, and the unknown matrix $\mathbf{R}_{32} = J_s(\mathbf{u}_0)$. Substituting these into Eq. (8), Eq. (9) and Eq. (10) gives what we call the *IPPE Problem*. This writes as follows:

$$\begin{array}{l}\text{find}\,\mathbf{x}, \mathbf{R}\ \text{s.t.} \\ \begin{cases} \pi(\mathbf{x}) = w(\mathbf{u}_0) & (a) \\ J_\pi(\mathbf{x})\mathbf{R}_{32} = J_w(\mathbf{u}_0) & (b) \\ \mathbf{R}_{32}^\top \mathbf{R}_{32} = \mathbf{I}_2 & (c) \\ x_3 > 0 & (d) \end{cases}\end{array} \tag{11}$$

The additional constraint (11-d) enforces that for $\mathbf{u}_0$ to be visible in the image it must lie in front of the camera. The constraints in Problem (11) only involve $\mathbf{R}_{32}$. Given a solution to $\mathbf{R}_{32}$ the third column of $\mathbf{R}$ is recovered uniquely by the cross-product of the two columns in $\mathbf{R}_{32}$. To recover $\mathbf{t}$ from $\mathbf{x}$ and $\mathbf{R}$ we use the definition of $\mathbf{x}$: $\mathbf{x} = s(\mathbf{u}_0) = \mathbf{R}\left[\mathbf{u}_0\, 0\right]^\top + \mathbf{t}$. Thus given a solution to Problem (11) pose is given by:

$$\begin{array}{l}\mathbf{R} = \left[\mathbf{R}_{32} \mid \mathbf{R}_{32}\begin{bmatrix}1\\0\end{bmatrix} \times \mathbf{R}_{32}\begin{bmatrix}0\\1\end{bmatrix}\right] \\ \mathbf{t} = \mathbf{x} - \mathbf{R}\begin{bmatrix}\mathbf{u}_0\\0\end{bmatrix}\end{array} \tag{12}$$

In practice we do not have access to the noise-free homography $\mathbf{H}$. Instead we have access to a noisy estimate $\hat{\mathbf{H}}$ computed from the point correspondences $\{\mathbf{u}_i\}$ and $\{\tilde{\mathbf{q}}_i\}$. We therefore must work with noisy estimates of $w(\mathbf{u}_0)$ and $J_w(\mathbf{u}_0)$, which we denote by $\mathbf{v} \in \mathbb{R}^2$ and $\mathbf{J} \in \mathbb{R}^{2\times 2}$ respectively. We assume that $\hat{H}_{33} = 1$ (which can be ensured by rescaling $\hat{\mathbf{H}}$), and so $\mathbf{v}$ and $\mathbf{J}$ are given by:

$$\mathbf{v} \stackrel{\text{def}}{=} \pi(\hat{\mathbf{H}}[\mathbf{u}_0{}^\top 1]^\top) \approx w(\mathbf{u}_0) \tag{13}$$

$$
\begin{aligned}
&\mathbf{J} \stackrel{\text{def}}{=} (1 + u_x\hat{H}_{31} + u_y\hat{H}_{32})^{-2} \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \approx J_w(\mathbf{u}_0) \\
&[u_x, u_y]^\top \stackrel{\text{def}}{=} \mathbf{u}_0 \\
&J_{11} \stackrel{\text{def}}{=} \hat{H}_{11} - \hat{H}_{31}\hat{H}_{13} + u_x(\hat{H}_{11}\hat{H}_{32} - \hat{H}_{31}\hat{H}_{12}) \\
&J_{12} \stackrel{\text{def}}{=} \hat{H}_{12} - \hat{H}_{32}\hat{H}_{13} + u_y(\hat{H}_{12}\hat{H}_{31} - \hat{H}_{32}\hat{H}_{11}) \\
&J_{21} \stackrel{\text{def}}{=} \hat{H}_{21} - \hat{H}_{31}\hat{H}_{23} + u_x(\hat{H}_{21}\hat{H}_{32} - \hat{H}_{31}\hat{H}_{22}) \\
&J_{22} \stackrel{\text{def}}{=} \hat{H}_{22} - \hat{H}_{32}\hat{H}_{23} + u_y(\hat{H}_{22}\hat{H}_{31} - \hat{H}_{32}\hat{H}_{21})
\end{aligned}
\tag{14}
$$

$\mathbf{v}$ and $\mathbf{J}$ can be defined for any $\mathbf{u}_0 \in \Omega_{\hat{\mathbf{H}}}$.

Eq. (11-a) gives estimates for x_1 and x_2 in terms of x_3 via $[x_1, x_2]^\top = x_3\mathbf{v}$. Substituting this into $J_\pi(\mathbf{x})$ gives:

$$
J_\pi(\mathbf{x}) \approx J_\pi(x_3[\mathbf{v}^\top 1]^\top) = x_3^{-1}\left[\mathbf{I}_2 \mid -\mathbf{v}\right] \tag{15}
$$

Thus Problem (11) is reduced to one in x_3 and $\mathbf{R}$. To simplify further we make a change of variables $\gamma \stackrel{\text{def}}{=} x_3^{-1}$ to give Problem (11) in terms of γ and $\mathbf{R}$:

$$
\boxed{\begin{array}{l} \text{find}\,\gamma, \mathbf{R} \quad \text{s.t.} \\ \begin{cases} \gamma\left[\mathbf{I}_2 \mid -\mathbf{v}\right]\mathbf{R}_{32} = \mathbf{J} & (a) \\ \mathbf{R}_{32}^\top\mathbf{R}_{32} = \mathbf{I}_2 & (b) \\ \gamma > 0 & (c) \end{cases} \end{array}} \tag{16}
$$

Given a solution to Problem (16) the plane's pose is given from Eq. (12) using $\mathbf{x} = \gamma^{-1}\left[\mathbf{v}^\top 1\right]^\top$.

One can also construct the PPE problem using alternative camera projection models. We show in Appendix A that when we use weak-perspective or para-perspective models we obtain a problem with exactly the same form as Problem (16). The difference is that the affine approximation made by these cameras lead to different values for $\mathbf{v}$ and $\mathbf{J}$. Therefore a solution to Problem (16) is general because it handles perspective, para-perspective and weak-perspective cameras as special cases.

3.2 Statistical Motivation for IPPE and Choosing $\mathbf{u}_0$

We defer our solution to Problem (16) until the next section. We first consider two important questions:

Q1 *When there is noise in the correspondences (and hence noise in* $\hat{\mathbf{H}}$*,* $\mathbf{v}$ *and* $\mathbf{J}$*), how does changing* $\mathbf{u}_0$ *affect Problem (16)*?

Q2 *How can we choose* $\mathbf{u}_0$ *such that Problem (16) is* least *affected by noise in the correspondences?*

We have studied these questions based on a statistical analysis of how errors in the correspondences propagate through $\hat{\mathbf{H}}$ to $\mathbf{v}$ and $\mathbf{J}$. We then show how this propagated error varies as a function of $\mathbf{u}_0$. The answers we find to the above two questions provide the statistical motivation for why IPPE is a very sensible approach to PPE in the first place. *This is because the error in both* $\mathbf{v}$ *and* $\mathbf{J}$ *varies as a function of* $\mathbf{u}_0$*, and the error is approximately minimal at the centroid of* $\{\mathbf{u}_i\}$. By choosing $\mathbf{u}_0$ to be at the point where the error in $\mathbf{v}$ and $\mathbf{J}$ is least, then IPPE solves pose using a system of equations with the lowest error-in-variables. Note that when there is no noise in $\hat{\mathbf{H}}$ we have error-free estimates of $\mathbf{v}$ and $\mathbf{J}$ for all $\mathbf{u}_0$. It would therefore make no difference where we positioned $\mathbf{u}_0$, because for any $\mathbf{u}_0$ γ and $\mathbf{R}$ would be estimated without error.

Recall that IPPE can be thought of as solving PPE using constraints from the motion of an infinitesimally small region on the model plane (centred at $\mathbf{u}_0$). IPPE might seem counter intuitive because when we think about pose estimation we might imagine that using an infinitesimally small region would lead to instability. *This is in fact the opposite.* Note that if $\{\mathbf{u}_i\}$ were to be themselves infinitesimally separated then with a small amount of noise the PPE problem itself would be totally unstable. In IPPE however the motion at an infinitesimal region about $\mathbf{u}_0$ is predicted from $\hat{\mathbf{H}}$ via points that *are* spatially separated.

We assume the correspondences $\{\tilde{\mathbf{q}}_i\}$ in the image are perturbed from their true positions by zero-mean Gaussian IID noise. This model has been shown many times to be a good approximation in practice [17]. We denote $\hat{\mathbf{q}} \in \mathbb{R}^{2n}$ to be the vector that holds $\{\tilde{\mathbf{q}}_i\}$ as a single column vector. We use $\mathbf{\Sigma}_{\hat{\mathbf{q}}} = \sigma^2\mathbf{I}_{2n}$ to denote the uncertainty covariance matrix of $\hat{\mathbf{q}}$, where σ^2 is the correspondence noise variance.

3.2.1 Uncertainty in $\mathbf{v}$

We start by considering the uncertainty in $\mathbf{v}$ given noisy correspondences and show how this varies as a function of $\mathbf{u}_0$. We do this by modelling the 1^{st}-order effects of propagating errors in $\hat{\mathbf{q}}$ through $\hat{\mathbf{H}}$ to $\mathbf{v}$. Recall that $\{\mathbf{u}_i\}$ is zero-centred so that its centroid is at the origin. We write the 2×2 uncertainty covariance matrix of $\mathbf{v}$ as a function of $\mathbf{u}_0$ by $\mathbf{\Sigma}_{\mathbf{v}}(\mathbf{u}_0) : \mathbb{R}^2 \to \mathcal{S}(2)$, where $\mathcal{S}(2)$ is the space of 2×2 covariance matrices. Because $\mathbf{\Sigma}_{\mathbf{v}}(\mathbf{u}_0) \succeq \mathbf{0}$ we can minimise the uncertainty in $\mathbf{v}$ by finding $\mathbf{u}_0$ that minimises the trace of $\mathbf{\Sigma}_{\mathbf{v}}(\mathbf{u}_0)$.

Theorem 1 *The point that minimises* $\text{trace}\left(\mathbf{\Sigma}_{\mathbf{v}}(\mathbf{u}_0)\right)$ *(the uncertainty in* $\mathbf{v}$*) is given up to* 1^{st}*-order by the centroid of* $\{\mathbf{u}_i\}$.

Proof

The optimal 1^{st}-order approximation of $\hat{\mathbf{H}}$ is given by the ML affine transform $\hat{\mathbf{H}} \approx \begin{bmatrix} \hat{\mathbf{A}}_{ML} & \hat{\mathbf{t}}_{ML} \\ \mathbf{0}^\top & 1 \end{bmatrix}$, which is the least squares affine transform that maps $\{\mathbf{u}_i\}$ to $\{\tilde{\mathbf{q}}_i\}$. When we use the 1^{st}-order approximation $\mathbf{v} \approx \hat{\mathbf{A}}_{ML}\mathbf{u} + \hat{\mathbf{t}}_{ML}$, $\mathbf{\Sigma}_{\mathbf{v}}(\mathbf{u}_0)$ is given by:

$$[\mathbf{\Sigma}_{\mathbf{v}}(\mathbf{u}_0)]_{ij} \approx \begin{cases} \frac{\sigma}{n} + (\mathbf{u}_0 - \mathbf{0})^\top (\bar{\mathbf{U}}^\top \bar{\mathbf{U}})^{-1} (\mathbf{u}_0 - \mathbf{0}) & i = j \\ 0 & i \neq j \end{cases} \tag{17}$$

$\bar{\mathbf{U}}$ is the $2 \times n$ matrix that holds $\{\mathbf{u}_i\}$. $\bar{\mathbf{U}}^\top \bar{\mathbf{U}} \succ \mathbf{0}$ is the covariance matrix of $\{\mathbf{u}_i\}$. Eq. (17) is straightforward to prove using 1^{st}-order uncertainty propagation, and we include a short derivation in Appendix B. Eq. (17) tells us that to 1^{st}-order the uncertainty in $\mathbf{v}$ induced by noise in $\{\tilde{\mathbf{q}}_i\}$ follows a Gaussian distribution with isotropic variance and centred at the origin. Thus the variance of $\mathbf{v}$ increases quadratically with respect to the distance $\mathbf{u}_0$ is from the origin. The value $\hat{\mathbf{u}}_0 \in \mathbb{R}^2$ that minimises the uncertainty in $\mathbf{v}$ is that which minimises $\text{trace}(\mathbf{\Sigma}_{\mathbf{v}}(\mathbf{u}_0))$. This is unique and given by $\hat{\mathbf{u}}_0 = \mathbf{0}$ (*i.e.* the centroid of $\{\mathbf{u}_i\}$). □

Consequently a good strategy to reduce the uncertainty in $\mathbf{v}$ is to position $\mathbf{u}_0$ at the centroid of $\{\mathbf{u}_i\}$.

3.2.2 Uncertainty in $\mathbf{J}$

We also want $\mathbf{u}_0$ to reduce the uncertainty in $\mathbf{J}$. This is less simple than the uncertainty in $\mathbf{v}$ because it involves studying the second-order properties of $\hat{\mathbf{H}}$ (*i.e.* the variation of its Jacobian with respect to $\mathbf{u}_0$). Recall that $\mathbf{J}$ is a function of both $\mathbf{u}_0$ and $\hat{\mathbf{H}}$, and $\hat{\mathbf{H}}$ is a function of $\hat{\mathbf{q}}$. Consider first $\hat{\mathbf{q}}$. The Taylor expansion of $\text{vec}(\mathbf{J})$ about $\hat{\mathbf{q}}$ is:

$$\text{vec}(\mathbf{J}) = \text{vec}(\mathbf{J}(\hat{\mathbf{q}})) + \frac{\partial}{\partial \hat{\mathbf{q}}} \text{vec}(\mathbf{J}) \triangle \hat{\mathbf{q}} + \mathcal{O}^2 \tag{18}$$

We use $\mathbf{\Sigma}_{\mathbf{J}}$ to denote the 4×4 covariance matrix of $\text{vec}(\mathbf{J})$. Because $\mathbf{\Sigma}_{\hat{\mathbf{q}}} = \sigma^2 \mathbf{I}_{2n}$, this is given to 1^{st}-order by:

$$\mathbf{\Sigma}_{\mathbf{J}} \approx \sigma^2 \frac{\partial}{\partial \hat{\mathbf{q}}} \text{vec}(\mathbf{J}) \frac{\partial}{\partial \hat{\mathbf{q}}} \text{vec}(\mathbf{J})^\top \tag{19}$$

We use $\mathbf{\Sigma}_{\mathbf{J}}(\mathbf{u}_0)$ to denote explicitly the dependence of $\mathbf{\Sigma}_{\mathbf{J}}$ on $\mathbf{u}_0$. Our goal is to find the $\mathbf{u}_0$ that minimises $\text{trace}\left(\mathbf{\Sigma}_{\mathbf{J}}(\mathbf{u}_0)\right)$ (*i.e.* the uncertainty in $\mathbf{J}$). This involves an analysis of $\frac{\partial}{\partial \hat{\mathbf{q}}} \text{vec}(\mathbf{J})$, which depends on the algorithm used to compute $\hat{\mathbf{H}}$ (and hence $\mathbf{J}$) from $\hat{\mathbf{q}}$. We analyse the most well-established algorithm, which is the normalised Direct Linear Transform (DLT) algorithm [17]. Recall that normalisation means modifying $\{\mathbf{u}_i\}$ and $\{\tilde{\mathbf{q}}_i\}$ so that the point sets are zero-centred and the average distance of each point set to the origin is $\sqrt{2}$. We use $\{\mathbf{u}'_i\}$ and $\{\tilde{\mathbf{q}}'_i\}$ to denote the normalised point sets and $\hat{\mathbf{H}}'$ to be the homography that maps $\{\mathbf{u}'_i\}$ to $\{\tilde{\mathbf{q}}'_i\}$ using the DLT algorithm.

Theorem 2 *When the perspective terms in* $\hat{\mathbf{H}}'$ *(i.e.* $\hat{H}'_{31}$ *and* $\hat{H}'_{32}$*) are negligible a point that minimises* $\text{trace}\left(\mathbf{\Sigma}_{\mathbf{J}}(\mathbf{u}_0)\right)$ *(the uncertainty in* $\mathbf{J}$*) is the centroid of* $\{\mathbf{u}_i\}$.

Theorem 2 depends on the following lemmas:

Lemma 1 *A point that minimises* $\text{trace}\left(\mathbf{\Sigma}_{\mathbf{J}}(\mathbf{u}_0)\right)$ *is given to* 1^{st}*-order by a point where a change in* $\{\tilde{\mathbf{q}}_i\}$ *induces the smallest change in* $\mathbf{J}$.

Lemma 2 *When the perspective terms of* $\hat{\mathbf{H}}'$ *(i.e.* $\hat{H}'_{31}$ *and* $\hat{H}'_{32}$*) are small the minimisation of* $\text{trace}\left(\mathbf{\Sigma}_{\mathbf{J}}(\mathbf{u}_0)\right)$ *is a convex quadratic problem.*

The proofs of Theorem 2 and these lemmas are based on [5] which shows how the error in $\hat{\mathbf{q}}$ propagates to $\hat{\mathbf{H}}'$. We give the proofs in Appendix C.

When the perspective terms in $\hat{\mathbf{H}}'$ are small but non-negligible, an optimal solution to $\arg\min_{\mathbf{u}_0} \left[\text{trace}\left(\mathbf{\Sigma}_{\mathbf{J}}(\mathbf{u}_0)\right)\right]$ can be found easily, since Lemma 2 tells us it is a convex quadratic problem. When the perspective terms in $\hat{\mathbf{H}}'$ are non-negligible an optimal solution is not guaranteed to be precisely at the centroid of $\{\mathbf{u}_i\}$. However Theorem 1 and Lemma 2 tell us that as the perspective terms in $\hat{\mathbf{H}}'$ become smaller then an optimal solution tends towards the centroid of $\{\mathbf{u}_i\}$. In real imaging conditions usually the perspective terms in $\hat{\mathbf{H}}'$ are small and we have found that the centroid of $\{\mathbf{u}_i\}$ is very close to the optimal solution. In practice it can therefore be used as an approximate minimiser of $\text{trace}\left(\mathbf{\Sigma}_{\mathbf{J}}(\mathbf{u}_0)\right)$.

Summary. We have provided answers to the two questions at the beginning of this section with a statistical analysis of how the uncertainty in the point correspondences propagates through $\hat{\mathbf{H}}$ to $\mathbf{v}$ and $\mathbf{J}$. The uncertainty is a function of $\mathbf{u}_0$. The uncertainty in $\mathbf{v}$ is minimised to 1^{st} order by setting $\mathbf{u}_0$ to be the centroid of $\{\mathbf{u}_i\}$. Assuming that the perspective terms in the normalised homography $\hat{\mathbf{H}}'$ are small (which is usually the case in common imaging conditions), the uncertainty in $\mathbf{J}$ is minimised by solving a convex quadratic problem. This is also approximately minimised by setting $\mathbf{u}_0$ to be the centroid of $\{\mathbf{u}_i\}$. Recall that $\mathbf{u}_0$ must be in $\Omega_{\hat{\mathbf{H}}}$. This is always satisfied by the centroid of $\{\mathbf{u}_i\}$ because it is at the origin, and this never maps to the line at infinity: $\begin{bmatrix} \hat{H}_{31} & \hat{H}_{32} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{0}^\top & 1 \end{bmatrix}^\top = 1 \neq 0$.

3.3 Solving IPPE

Our solution to Problem (16) does not require $\mathbf{u}_0$ to be positioned at the centroid of $\{\mathbf{u}_i\}$. In practice this is where we position it to reduce error-in-variables. The main results in this section is the analytic solution to Problem (16) and proofs of the following theorems:

Theorem 3 ***(Solution existence and uniqueness in*** γ***).*** *When* $\mathbf{u}_0 \in \Omega_{\hat{\mathbf{H}}}$, $\mathbf{J} \neq \mathbf{0}$ *the solution to* γ *in Problem (16) always exists and is unique.*

Theorem 4 ***(Two-fold ambiguity in*** $\mathbf{R}$***).*** *When* $\mathbf{u}_0 \in \Omega_{\hat{\mathbf{H}}}$, $\mathbf{J} \neq \mathbf{0}$ *a solution to* $\mathbf{R}$ *in Problem (16) always exists and there are at most two solutions to* $\mathbf{R}$*. These correspond to reflecting the model plane in camera coordinates about a plane whose normal points along the line-of-sight* $[\mathbf{v}^\top 1]^\top$*.* $\mathbf{R}$ *has a unique solution iff the model plane in camera coordinates is tangential to a sphere centred at the optical axis.*

3.3.1 Input Bounds

Problem (16) can be setup using any $\mathbf{u}_0 \in \Omega_{\hat{\mathbf{H}}}$ (since if $\mathbf{u}_0 \notin \Omega_{\hat{\mathbf{H}}}$ then $\mathbf{J}$ is undefined).

Theorem 5 ***(Generic Degeneracy).*** *If there exists* $\mathbf{u}_0 \in \Omega_{\hat{\mathbf{H}}}$ *such that* $\mathbf{J} = 0$ *then* $\hat{\mathbf{H}}$ *is rank-1 and no algorithm can solve pose from* $\hat{\mathbf{H}}$*.*

Proof

It is simple to show from Eq. (14) that $\mathbf{J} = 0 \Leftrightarrow \hat{\mathbf{H}} = [\hat{H}_{13}\, \hat{H}_{23}\, 1]^\top [\hat{H}_{13}\, \hat{H}_{23}\, 1]$. Therefore $\mathbf{J} = 0 \Rightarrow \text{rank}(\hat{\mathbf{H}}) = 1$ and all points on the model plane map in the image to a single point (which is at $[\hat{H}_{13}\, \hat{H}_{23}]^\top$). This is a degenerate configuration that occurs when the model plane is infinitely far from the camera. In this case no algorithm can recover its pose. □

We therefore restrict solving Eq. (16) to when $\mathbf{u}_0 \in \Omega_{\hat{\mathbf{H}}}$ and $\mathbf{J} \neq \mathbf{0}$. The solution we now present gives a physically valid solution for all these inputs. This means our solution does not introduce any artificial degeneracies.

3.3.2 Simplification

We rewrite the left side of Eq. (16-a) as follows:

$$\gamma \left[\mathbf{I}_2 | -\mathbf{v}\right] \mathbf{R}_{32} = \gamma \left[\mathbf{I}_2 | -\mathbf{v}\right] \mathbf{R}_v \mathbf{R}_v^\top \mathbf{R}_{32} \tag{20}$$

We define $\mathbf{R}_v \in SO_3$ as a rotation that rotates $[\mathbf{I}_2 | -\mathbf{v}]$ such that for some $\mathbf{B} \in \mathbb{R}^{2\times 2}$, $[\mathbf{I}_2 | -\mathbf{v}]\, \mathbf{R}_v = [\mathbf{B}|\mathbf{0}]$. $\mathbf{B}$ is rank-2 because $[\mathbf{I}_2 | -\mathbf{v}]$ and $\mathbf{R}_v$ are rank-2 and rank-3 respectively. We solve Problem (16) in terms of the rotation matrix $\tilde{\mathbf{R}} \stackrel{\text{def}}{=} \mathbf{R}_v^\top \mathbf{R}$, and then recover $\mathbf{R}$ with $\mathbf{R} = \mathbf{R}_v \tilde{\mathbf{R}}$. Eq. (16-a) becomes:

$$\begin{array}{ll} \gamma \left[\mathbf{I}_2 | -\mathbf{v}\right] \mathbf{R}_{32} = \mathbf{J} & \Leftrightarrow \\ \gamma \left[\mathbf{B}|\mathbf{0}\right] \tilde{\mathbf{R}}_{23} = \mathbf{J} & \Leftrightarrow \\ \gamma \tilde{\mathbf{R}}_{22} = \mathbf{A}, \;\; \mathbf{A} \stackrel{\text{def}}{=} \mathbf{B}^{-1}\mathbf{J} & \end{array} \tag{21}$$

Because $\mathbf{J}$ is at least rank-1 $\mathbf{A}$ is at least rank-1. Therefore we have reduced Problem (16) to the decomposition of a 2×2 matrix $\mathbf{A}$ (which is at least rank-1) into a positive scale term (γ) and a 2×2 sub-Stiefel matrix ($\tilde{\mathbf{R}}_{22}$). Once decomposed we then reconstruct the original rotation matrix $\mathbf{R}$ from $\tilde{\mathbf{R}}_{22}$.

3.3.3 Analytic Solution to Problem (16)

The solution to γ is:

$$\begin{array}{l} \gamma = \sigma_1^A = \frac{1}{2}\left(a_u + a_w + \sqrt{(a_u - a_w)^2 + 4a_v^2}\right) \\ \begin{bmatrix} a_u & a_v \\ a_v & a_w \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{A}\mathbf{A}^\top \end{array} \tag{22}$$

where σ_1^A is the largest singular value of $\mathbf{A}$. We denote the third column of $\mathbf{R}_v$ by $\mathbf{r}_{v3}$. Because $[\mathbf{I}_2 | -\mathbf{v}]\, \mathbf{R}_v = [\mathbf{B}|\mathbf{0}]$, $[\mathbf{I}_2 | -\mathbf{v}]\, \mathbf{r}_{v3} = \mathbf{0}$ and so by rearrangement $\mathbf{r}_{v3} \propto [\mathbf{v}^\top 1]^\top$. Thus $\mathbf{R}_v$ is any rotation which aligns the z-axis to $[\mathbf{v}^\top 1]^\top$. We define $\mathbf{R}_v$ uniquely by using the smallest rotation that aligns the z-axis to $[\mathbf{v}^\top 1]^\top$. This is given by Rodrigues' formula:

$$\begin{array}{l} \mathbf{R}_v = \mathbf{I}_3 + \sin\theta [\mathbf{k}]_\times + (1 - \cos\theta)[\mathbf{k}]_\times^2 \\ t \stackrel{\text{def}}{=} \|\mathbf{v}\|_2 \\ s \stackrel{\text{def}}{=} \left\| [\mathbf{v}^\top \mathbf{1}]^\top \right\|_2 \\ \cos\theta \stackrel{\text{def}}{=} 1/s \\ \sin\theta \stackrel{\text{def}}{=} \sqrt{1 - 1/s^2} \\ [\mathbf{k}]_\times \stackrel{\text{def}}{=} 1/t \begin{bmatrix} \mathbf{0} & \mathbf{v} \\ -\mathbf{v}^\top & 0 \end{bmatrix} \end{array} \tag{23}$$

$\mathbf{R}$ has two solutions which we denote by $\mathbf{R}_1, \mathbf{R}_2 \in SO_3$. These are:

$$\begin{array}{l} \mathbf{R}_1 = \mathbf{R}_v \tilde{\mathbf{R}}_1, \;\; \mathbf{R}_2 = \mathbf{R}_v \tilde{\mathbf{R}}_2 \\ \tilde{\mathbf{R}}_1 \stackrel{\text{def}}{=} \begin{bmatrix} \tilde{\mathbf{R}}_{22} & +\mathbf{c} \\ +\mathbf{b}^\top & a \end{bmatrix} \\ \tilde{\mathbf{R}}_2 \stackrel{\text{def}}{=} \begin{bmatrix} \tilde{\mathbf{R}}_{22} & -\mathbf{c} \\ -\mathbf{b}^\top & a \end{bmatrix} \\ \tilde{\mathbf{R}}_{22} = \gamma^{-1}\mathbf{A} \\ \mathbf{b} = \text{rank}_1\left(\mathbf{I}_2 - \tilde{\mathbf{R}}_{22}^\top \tilde{\mathbf{R}}_{22}\right) = \left[\sqrt{r_u} \;\; \text{sign}(r_v)\sqrt{r_w}\right]^\top \\ \begin{bmatrix} r_u & r_v \\ r_v & r_w \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{I}_2 - \tilde{\mathbf{R}}_{22}^\top \tilde{\mathbf{R}}_{22} \\ \begin{bmatrix} \mathbf{c} \\ a \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_{22} \\ \mathbf{b}^\top \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} \tilde{\mathbf{R}}_{22} \\ \mathbf{b}^\top \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{array} \tag{24}$$

3.3.4 Proof of Theorem 3

The decomposition $\gamma\tilde{\mathbf{R}}_{22} = \mathbf{A}$ has a simple solution in γ because the largest singular value of a matrix in $SS_{2\times 2}$ is 1:

$$\begin{array}{l}\tilde{\mathbf{R}}_{22} \in SS_{2\times 2} \Leftrightarrow \exists \mathbf{U}, \mathbf{V}, \sigma \text{ s.t.} \\ \begin{cases} \mathbf{U}\begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix}\mathbf{V}^\top = \tilde{\mathbf{R}}_{22} \\ \mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_2,\ 0 \leq \sigma \leq 1 \end{cases}\end{array} \tag{25}$$

We denote an SVD of $\mathbf{A}$ by $\mathbf{A} = \mathbf{U}_A\left[\text{diag}(\sigma_1^A, \sigma_2^A)\right]\mathbf{V}_A^\top$, with $\sigma_1^A > 0$, $\sigma_1^A \geq \sigma_2^A$ and $\mathbf{U}_A^\top\mathbf{U}_A = \mathbf{V}_A^\top\mathbf{V}_A = \mathbf{I}_2$. Because a singular value matrix is unique when the singular values are sorted by magnitude, the solution to γ is unique:

$$\begin{array}{ll} \gamma\tilde{\mathbf{R}}_{22} = \mathbf{A} & \Leftrightarrow \\ \gamma\mathbf{U}\begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix}\mathbf{V}^\top = \mathbf{U}_A\begin{bmatrix} \sigma_1^A & 0 \\ 0 & \sigma_2^A \end{bmatrix}\mathbf{V}_A^\top & \Rightarrow \\ \gamma = \sigma_1^A & \end{array} \tag{26}$$

Because $\sigma_1^A > 0$ Eq. (16-c) is satisfied by $\gamma = \sigma_1^A$. Therefore when $\mathbf{J} \neq \mathbf{0}$ the solution to γ in Problem (16) always exists and is unique. □

3.3.5 Proof of Theorem 4

Because γ has a unique solution when $\mathbf{J} \neq \mathbf{0}$ then $\tilde{\mathbf{R}}_{22} = \gamma^{-1}\mathbf{A}$ is a unique solution to $\tilde{\mathbf{R}}_{22}$. We then complete the Stiefel matrix $\tilde{\mathbf{R}}_{32}$ using orthonormality constraints. Let $\mathbf{b}^\top$ denote the third row of $\tilde{\mathbf{R}}_{32}$. We have $\tilde{\mathbf{R}}_{32}^\top\tilde{\mathbf{R}}_{32} = \mathbf{I}_2 \Leftrightarrow \mathbf{I}_2 - \tilde{\mathbf{R}}_{22}^\top\tilde{\mathbf{R}}_{22} = \mathbf{b}^\top\mathbf{b}$, so $\mathbf{b}$ is given by the rank-1 decomposition of $(\mathbf{I}_2 - \tilde{\mathbf{R}}_{22}^\top\tilde{\mathbf{R}}_{22})$. Let $\sigma_d > 0$ be the non-zero singular value of $(\mathbf{I}_2 - \tilde{\mathbf{R}}_{22}^\top\tilde{\mathbf{R}}_{22})$ and $\mathbf{d}$ be a singular vector for σ_d. There exist two solutions to $\mathbf{b}$ which are $\pm\sqrt{\sigma_d}\mathbf{d}$. Thus there exist two solutions to $\tilde{\mathbf{R}}_{32}$ using either solution to $\mathbf{b}$ as its third row. We then complete $\tilde{\mathbf{R}}$ uniquely from either solution to $\tilde{\mathbf{R}}_{32}$ by forming its third column with the cross-product of the two columns in $\tilde{\mathbf{R}}_{32}$. Therefore there exist two solutions to $\tilde{\mathbf{R}}$, and because $\mathbf{R} = \mathbf{R}_v\tilde{\mathbf{R}}$ there exist two solutions to $\mathbf{R}$.

Recall that $\mathbf{v}$ is the 2D point where $\mathbf{u}_0$ is located in the image (in normalised coordinates). Therefore $[\mathbf{v}^\top 1]^\top$ is a line-of-sight starting at the camera's optic centre and passing through $\mathbf{v}$. Eq. (24) factorises the two solutions to $\mathbf{R}$ into two rotations. First the rotation $\tilde{\mathbf{R}}$ is applied (using either $\tilde{\mathbf{R}}_1$ or $\tilde{\mathbf{R}}_2$ and then the rotation $\mathbf{R}_v$ is applied. From Eq. (24) the rotation of a 3D point $[\mathbf{u}^\top 0]^\top$ on the model plane according to $\tilde{\mathbf{R}}_1$ or $\tilde{\mathbf{R}}_2$ is related by:

$$\tilde{\mathbf{R}}_2[\mathbf{u}^\top 0]^\top = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}\tilde{\mathbf{R}}_1[\mathbf{u}^\top 0]^\top \tag{27}$$

Therefore the difference between rotating the point by either $\tilde{\mathbf{R}}_1$ or $\tilde{\mathbf{R}}_2$ corresponds to reflecting it about the model plane's z axis.

The two solutions to $\mathbf{R}$ are formed by first rotating the model plane by either $\tilde{\mathbf{R}}_1$ or $\tilde{\mathbf{R}}_2$. These rotations are equivalent up to a reflection in the model's z-axis. This is followed by a second rotation $\mathbf{R}_v$ which aligns the model plane's z-axis with the line-of-sight $[\mathbf{v}^\top 1]^\top$. The combined effect is a two-fold solution corresponding to a reflection of the model plane about a plane whose normal (in camera coordinates) points along $[\mathbf{v}^\top 1]^\top$.

$\mathbf{R}$ has a single solution iff $\tilde{\mathbf{R}}_1 = \tilde{\mathbf{R}}_2$. From Eq. (24) $\tilde{\mathbf{R}}_1 = \tilde{\mathbf{R}}_2 \Leftrightarrow \mathbf{c} = -\mathbf{c} \Leftrightarrow \mathbf{c} = \mathbf{0} \Leftrightarrow \mathbf{b} = \mathbf{0} \Leftrightarrow a = 1$. Therefore $\tilde{\mathbf{R}}_1$ (and hence $\tilde{\mathbf{R}}_2$) is a within-plane rotation that does not change the model plane's normal. The plane's normal is therefore only changed by $\mathbf{R}_v$ which rotates it to point along the line-of-sight $[\mathbf{v}^\top 1]^\top$. This is equivalent to saying that $\mathbf{R}$ has a unique solution iff the model plane in camera coordinates is tangential to a sphere centred at the optical axis. □

3.4 Disambiguation

Using Eq. (12) the two solutions to the plane's pose are:

$$\begin{array}{l}\left(\mathbf{R}_1, \mathbf{t}_1 = \gamma^{-1}\begin{bmatrix}\mathbf{v}\\1\end{bmatrix} - \mathbf{R}_1\begin{bmatrix}\mathbf{u}_0\\0\end{bmatrix}\right) \\ \left(\mathbf{R}_2, \mathbf{t}_2 = \gamma^{-1}\begin{bmatrix}\mathbf{v}\\1\end{bmatrix} - \mathbf{R}_2\begin{bmatrix}\mathbf{u}_0\\0\end{bmatrix}\right)\end{array} \tag{28}$$

It is possible to resolve which of these solutions is correct by inspecting their reprojection errors. We use the fact that within the correspondences there must exist three correspondences that are not colinear (otherwise a homography could not have been computed uniquely [17]). Without loss of generality let these be $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$.

Lemma 3 ***(Disambiguation)****. Given three non-colinear points $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \in \mathbb{R}^2$ on the model plane, for any $\mathbf{u}_0 \in \Omega_{\hat{\mathbf{H}}}$ the two pose solutions in Eq. (28) will, if different, project either $\mathbf{u}_1$, $\mathbf{u}_2$ or $\mathbf{u}_3$ to two different image points.*

Lemma 3 is proved easily by contradiction in Appendix D.

Lemma 3 tells us that in the absence of noise if $\mathbf{R}_1 \neq \mathbf{R}_2$ then the reprojection errors of $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ will all be zero only for the correct pose. With noise we can robustly disambiguate pose by inspecting the reprojection errors using all point correspondences. The reprojection error for each pose is:

$$e(\mathbf{R}_j, \mathbf{t}_j) = \sum_{i=1}^{n}\left\|\pi\left(\mathbf{R}_j\begin{bmatrix}\mathbf{u}_i\\0\end{bmatrix} + \mathbf{t}_j\right) - \tilde{\mathbf{q}}_i\right\|_2^2,\ j \in \{1, 2\}$$

(29)

We use $(\mathbf{R}^*, \mathbf{t}^*)$ to denote the pose solution with lowest e. We are then faced with accepting or rejecting the second pose as an alternative hypothesis. Pose is ambiguous if e_1 and e_2 are similar; specifically if the reprojection error of either pose is indistinguishable to noise. A decision can be made using a likelihood ratio test however this involves selecting a confidence bound, which is application specific. Instead we return both solutions with their reprojection errors, and leave it up to the end application to choose whether to reject the alternative hypothesis.

3.5 The Front-facing Constraint

Problem (16) enforces the physical assumption that the surface must lie in front of the camera for it to be imaged (*i.e.* $\gamma > 0$). However it does not enforce which *side* the plane's surface can be viewed from. When the model is translucent, correspondences could come from either side of the plane. When the model is opaque we have an additional constraint on $\mathbf{R}$ because correspondences can only come from the plane's front-facing side. Without loss of generality let the model plane's normal point away from its z axis. The *front-facing constraint* is $[\mathbf{v}^\top 1]\mathbf{r}_3 \geq 0$ (*i.e.* the cosine of the angle between the surface normal in camera coordinates and the line-of-sight $\left[\mathbf{v}^\top 1\right]^\top$ must be non-negative). The IPPE problem with the front-facing constraint is:

$$\begin{aligned} &\text{find}\,\gamma, \mathbf{R} \quad \text{s.t.} \\ &\begin{cases} \gamma\left[\mathbf{I}_2 | -\mathbf{v}\right]\mathbf{R}_{32} = \mathbf{J} & (a) \\ \mathbf{R}_{32}^\top \mathbf{R}_{32} = \mathbf{I}_2 & (b) \\ \gamma > 0 & (c) \\ [\mathbf{v}^\top 1]\left(\mathbf{R}_{32}[1\,0]^\top \times \mathbf{R}_{32}[0\,1]^\top\right) \geq 0 & (d) \end{cases} \end{aligned} \tag{30}$$

This includes the front-facing constraint (Eq. (30-d)) written in terms of $\mathbf{R}_{32}$.

Lemma 4 *If* $\det(\mathbf{J}) < 0$ *then Problem (30) has no solution.*

Proof

It is simple to show by rearrangement:

$$[\mathbf{v}^\top 1]\left(\mathbf{R}_{32}[1\,0]^\top \times \mathbf{R}_{32}[0\,1]^\top\right) = \det\left(\left[\mathbf{I}_2 | -\mathbf{v}\right]\mathbf{R}_{32}\right) \tag{31}$$

From Eq. (30-a) $\det\left(\left[\mathbf{I}_2 | -\mathbf{v}\right]\mathbf{R}_{32}\right) = \det(\gamma^{-1}\mathbf{J})$, so Eq. (30-d) $\Leftrightarrow \det(\gamma^{-1}\mathbf{J}) \geq 0 \Leftrightarrow \gamma^{-2}\det(\mathbf{J}) \geq 0$. Because $\gamma > 0$, Eq. (30-d) $\Leftrightarrow \det(\mathbf{J}) \geq 0$ which contradicts $\det(\mathbf{J}) < 0$. Therefore when $\det(\mathbf{J}) < 0$ Problem (30) has no solution. □

Conversely, if $\det(\mathbf{J}) > 0$ then Eq. (30-d) is redundant, because Eq. (30-c) and $\det(\mathbf{J}) > 0 \Rightarrow$ Eq. (30-d). Therefore when $\det(\mathbf{J}) > 0$ the front-facing constraint adds nothing to the problem. To summarise, when $\det(\mathbf{J}) < 0$ there is no front-facing solution to the plane's pose (from Lemma 4), but when $\det(\mathbf{J}) \geq 0$ both solutions to its pose will be front-facing. Therefore the front-facing constraint cannot be used to disambiguate the correct pose.

3.6 The Connection Between IPPE and P3P

To complete our analysis of IPPE we now give the connection between IPPE and P3P. This connection comes from the fact that $\mathbf{J}$ can be represented in two equivalent ways. The first is to compute it by differentiating $\hat{\mathbf{H}}$, as we have done in IPPE. The second is to compute it from the motion of three non-colinear *virtual* points that transform according to $\hat{\mathbf{H}}$, but which are separated by an infinitesimal distance. By linearising the P3P equations with respect to the points' positions on the model plane, in the limit as they tend to the same point we arrive at the IPPE equations in Eq. (11). This connection is important because Theorems 3 and 4 give a full characterisation of what happens in P3P as the points' separation tends to zero.

In P3P there are three non-colinear model points $\{\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2\}$, $\mathbf{u}_i \in \mathbb{R}^2$ and we have estimates $\{\tilde{\mathbf{q}}_0, \tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2\}$, $\tilde{\mathbf{q}}_i \in \mathbb{R}^2$ of their position in the image in normalised coordinates. Without loss of generality let $\mathbf{u}_0 = \mathbf{0}$. The six P3P equations write as:

$$\begin{aligned} &\tfrac{1}{t_3}\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \tilde{\mathbf{q}}_0 && (a) \\ &\tfrac{1}{t_3 + [R_{31}\ R_{32}]\mathbf{u}_1}\begin{bmatrix} t_1 + [R_{11}\ R_{12}]\,\mathbf{u}_1 \\ t_2 + [R_{21}\ R_{22}]\,\mathbf{u}_1 \end{bmatrix} = \tilde{\mathbf{q}}_1 && (b) \\ &\tfrac{1}{t_3 + [R_{31}\ R_{32}]\mathbf{u}_2}\begin{bmatrix} t_1 + [R_{11}\ R_{12}]\,\mathbf{u}_2 \\ t_2 + [R_{21}\ R_{22}]\,\mathbf{u}_2 \end{bmatrix} = \tilde{\mathbf{q}}_2 && (c) \end{aligned} \tag{32}$$

Theorem 6 ***(Relationship between IPPE and P3P).*** *In the limit when the separation of the three points in P3P tends to zero, the P3P problem becomes the IPPE problem.*

Proof of Theorem 6

When the separation of $\{\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2\}$ is small Eq. (32-b,c) can be approximated to 1^{st}-order with a Taylor expansion of their left sides with respect to $\mathbf{u}_1$ and $\mathbf{u}_2$ about the model plane's origin. After some simplification the six equations become:

$$\begin{aligned} &\tfrac{1}{t3}\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \tilde{\mathbf{q}}_0 && (a) \\ &\tfrac{1}{t_3}\left[\mathbf{I}_2 \ -\tfrac{1}{t_3}\begin{bmatrix} t_1 \\ t_2 \end{bmatrix}\right]\mathbf{R}_{32}\mathbf{U} + \mathcal{O}^2 = \mathbf{Q} && (b) \end{aligned} \tag{33}$$

with:

$$\begin{array}{l}\mathbf{Q} \overset{\text{def}}{=} [\tilde{\mathbf{q}}_1 \;\; \tilde{\mathbf{q}}_2] \;\; (a) \\ \mathbf{U} \overset{\text{def}}{=} [\mathbf{u}_1 \;\; \mathbf{u}_2] \;\; (b)\end{array} \tag{34}$$

and $\mathcal{O}^2$ denoting terms beyond 1^{st}-order. When $\mathcal{O}^2$ is neglected Eq. (33) approximates the P3P equations and this approximation becomes better as $\mathbf{u}_1$ and $\mathbf{u}_2$ approach the origin. We combine Eq. (33-a) and Eq. (33-b) to give what we call the *Infinitesimal P3P Problem*:

$$\begin{array}{l}\text{find}\, t_3, \mathbf{R} \quad \text{s.t.} \\ \begin{cases} \frac{1}{t_3}\left[\mathbf{I}_2 \; -\tilde{\mathbf{q}}_0\right]\mathbf{R}_{32} = \mathbf{Q}\mathbf{U}^{-1} + \mathcal{O}^2 & (a) \\ \mathbf{R}_{32}^\top \mathbf{R}_{32} = \mathbf{I}_2 & (b) \\ t_3 > 0 & (c) \end{cases}\end{array} \tag{35}$$

Note that because $\{\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2\}$ are non-colinear then $\mathbf{U}$ is rank-2 and so is invertible. Problem (16) and Problem (35) clearly have identical structure.

The left sides of Eq. (16-a) and Eq. (35-a) are the same by equating variable names. Because $1/t_3$ is the inverse-depth of the point $\mathbf{u}_0$ in camera coordinates, it is equal to γ. because $\tilde{\mathbf{q}}_0$ is the position of $\mathbf{u}_0$ in the camera's image, it is equal to $\mathbf{v}$. This implies the right sides of Eq. (16-a) and Eq. (35-a) are the same, which implies $\mathbf{J} = \mathbf{Q}\mathbf{U}^{-1} + \mathcal{O}^2$. Therefore in the limit when $\{\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2\}$ are infinitesimally separated, $\mathcal{O}^2 = \mathbf{0}$ and $\mathbf{J} = \mathbf{Q}\mathbf{U}^{-1}$, and so the P3P problem becomes the IPPE problem. □

Theorems 3 and 4 therefore give a full characterisation of what happens in P3P as the points' separation tends to zero. The characteristics are:

- As the separation of the 3 points tends to zero in the limit the solution to translation becomes unique.
- As the separation of the 3 points tends to zero in the limit the solution to rotation becomes two-fold ambiguous.
- This rotation ambiguity corresponds to a reflection of the points about a plane whose normal points along a line-of-sight that passes through the points.
- There is no rotation ambiguity if the 3 points become tangential to a sphere passing through the line-of-sight.

IPPE can therefore be thought of as solving pose by generating three infinitesimally separated virtual points centered at $\mathbf{u}_0$ and recovering pose using their positions in the image from $\hat{\mathbf{H}}$. Given this relationship between IPPE and P3P, one might ask why do this when we could generate three virtual points anywhere on the model plane and solve pose using P3P. The answer is that because $\hat{\mathbf{H}}$ is noisy positioning the virtual correspondences at different locations will cause P3P to return different results, and different numbers of results (between zero and four). The question would then be where is it best to position the points to ensure we obtain an accurate and physically valid solution. This question is interesting, but has not been studied previously in the literature. IPPE provides an answer to this question. That is, they should be infinitesimally separated and positioned at the centroid of the real correspondences. This stems from the statistical analysis in §3.2. We will show in §4.6 that IPPE performs significantly better than P3P using virtual correspondences positioned at other locations.

3.7 IPPE Algorithm and Summary

We now summarise IPPE in pseudocode. We break this down into two components. The first component is the solution to Problem (16). This takes as inputs $\mathbf{v}$ and $\mathbf{J}$, and returns γ, $\mathbf{R}_1$ and $\mathbf{R}_2$. We give this in Algorithm 1. Note that all steps involve only simple floating point operations. It is therefore extremely fast to perform, fully analytic and does not require any additional numerical libraries (*e.g.* computing eigen decompositions or root finding, as is required in most PnP approaches [24,23, 11,33,10,7,15,37,1]). We have proved that Algorithm 1 does not introduce any artificial degeneracies. That is, it guarantees that a positive scale factor γ and two rotation matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ will be returned for all $\mathbf{v} \in \mathbb{R}^2$ and $\mathbf{J} \neq \mathbf{0}$. This means that it may handle cases such as when the plane is viewed obliquely (*i.e.* when its normal is orthogonal to the line-of-sight, meaning $\mathbf{J}$ is rank-1. Although this is not likely to occur in practice (because in such situation we would likely not be able to compute correspondences) it does say that Algorithm 1 will not induce instability as a result of the way it estimates pose. Unlike PHD, Algorithm 1 can be used when the homography is an affine transform (since all that is required is $\mathbf{J} \neq \mathbf{0}$). Therefore, unlike PHD it will not encounter instability when the amount of perspective distortion in the homography is small.

The second component involves taking as input a set of point correspondences and the camera intrinsics, constructing $\mathbf{v}$ and $\mathbf{J}$, calling Algorithm 1, and returning two pose estimates. This is given in Algorithm 2. In the absence of noise we can estimate translation without error from Eq. (28). With noise we have found that $\mathbf{t}$ can be estimated slightly more accurately using the solution to $\mathbf{R}$ and estimating it in a Linear Least Squares sense. The cost function we use is as follows:

$$\begin{array}{l} c(\mathbf{t}; [\mathbf{r}_1 \, \mathbf{r}_2 \, \mathbf{r}_3], \{\mathbf{u}_i\}, \{\tilde{\mathbf{q}}_i\}) \overset{\text{def}}{=} \\ \sum_{i=1}^n \left\| \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} [\mathbf{u}_i^\top 0]^\top + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} - (\mathbf{r}_3 [\mathbf{u}_i^\top 0]^\top + t_3)\tilde{\mathbf{q}}_i \right\|_2^2 \end{array} \tag{36}$$

Eq. (36) is derived from the Maximum Likelihood cost but is convex because we minimise the error in 3D camera space rather than in 2D image space. Solving Eq. (36) is very efficient, and is the solution to a Linear Least Squares system of the form: $\|\mathbf{W}_j\mathbf{t}_j - \mathbf{b}_j\|_2^2$. $\mathbf{W}_j$ is a $2n \times 3$ matrix and $\mathbf{b}_j$ is a $2n \times 1$ vector. Eq. (36) is solved by $\mathbf{t}_j = (\mathbf{W}_j^\top \mathbf{W}_j)^{-1}\mathbf{W}_j\mathbf{b}_j$. It is straightforward to show that $\mathbf{W}$ must be rank-3, so the solution to $\mathbf{t}_j$ is a unique global minimum. The computational overhead for computing $\mathbf{t}_j$ in this way is very small because $\mathbf{W}_j^\top \mathbf{W}_j$ is a 3×3 matrix (and so its inverse is very fast to compute).

Algorithm 1 IPPE: The solution to Problem (16)

Require: $\mathbf{v} \in \mathbb{R}^2$ and $\mathbf{J} \in \mathbb{R}^{2\times 2}, \mathbf{J} \neq \mathbf{0}$

function IPPE($\mathbf{v}, \mathbf{J}$)
  Compute $\mathbf{R}_v$ from $\mathbf{v}$ ▷ (Eq. (23))
  $[\mathbf{B}|\mathbf{0}] \leftarrow [\mathbf{I}_2| - \mathbf{v}]\,\mathbf{R}_v$
  $\mathbf{A} \leftarrow \mathbf{B}^{-1}\mathbf{J}$
  $\gamma \leftarrow \sigma_1^A$ ▷ the largest singular value of $\mathbf{A}$ (Eq. (22))
  $\tilde{\mathbf{R}}_{22} \leftarrow \gamma^{-1}\mathbf{A}$
  $\mathbf{b} \leftarrow \text{rank}_1\left(\mathbf{I}_2 - \tilde{\mathbf{R}}_{22}^\top \tilde{\mathbf{R}}_{22}\right)$ ▷ (Eq. (24))
  $\begin{bmatrix}\mathbf{c}\\ a\end{bmatrix} \leftarrow \begin{bmatrix}\tilde{\mathbf{R}}_{22}\\ \mathbf{b}^\top\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} \times \begin{bmatrix}\tilde{\mathbf{R}}_{22}\\ \mathbf{b}^\top\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}$
  $\mathbf{R}_1 \leftarrow \mathbf{R}_v \begin{bmatrix}\tilde{\mathbf{R}}_{22} & +\mathbf{c}\\ +\mathbf{b}^\top & a\end{bmatrix}$, $\mathbf{R}_2 \leftarrow \mathbf{R}_v \begin{bmatrix}\tilde{\mathbf{R}}_{22} & -\mathbf{c}\\ -\mathbf{b}^\top & a\end{bmatrix}$
  return $\gamma, \mathbf{R}_1, \mathbf{R}_2$

Algorithm 2 Correspondence-based IPPE for Perspective Cameras

Require:
  $\{\mathbf{u}_1, \mathbf{u}_2, .., \mathbf{u}_n\},\ \mathbf{u}_i \in \mathbb{R}^2$ ▷ A set of n points on the model plane. These are zero centred: $\sum_i \mathbf{u}_i = \mathbf{0}$
  $\{\mathbf{q}_1, \mathbf{q}_2, .., \mathbf{q}_n\},\ \mathbf{q}_i \in \mathbb{R}^2$ ▷ The correspondences of each point in the camera's image
  $\mathbf{K}$ ▷ The camera intrinsics matrix

function IPPE($\{\mathbf{u}_i\}, \{\mathbf{q}_i\}, \mathbf{K}$)
  $[\tilde{\mathbf{q}}_i^\top 1]^\top \leftarrow \mathbf{K}^{-1}[\mathbf{q}_i^\top, 1]^\top$ ▷ $\tilde{\mathbf{q}}_i$ is $\mathbf{q}_i$ in normalised coordinates
  $\mathbf{H} \leftarrow \text{homog}\,(\{\mathbf{u}_i\}, \{\tilde{\mathbf{q}}_i\})$ ▷ Best fitting homography between $\{\mathbf{u}_i\}$ and $\{\tilde{\mathbf{q}}_i\}$, $H_{33} = 1$
  $\mathbf{J} \leftarrow \begin{bmatrix}J_{11} & J_{12}\\ J_{21} & J_{22}\end{bmatrix}$ ▷ $\mathbf{J}$ is the Jacobian of $\pi(\mathbf{H}[\mathbf{u}_0\ 1]^\top)$ at $\mathbf{u}_0 = \mathbf{0}$
  $J_{11} \leftarrow H_{11} - H_{31}H_{13}$
  $J_{12} \leftarrow H_{12} - H_{32}H_{13}$
  $J_{21} \leftarrow H_{21} - H_{31}H_{23}$
  $J_{22} \leftarrow H_{22} - H_{32}H_{23}$
  $\mathbf{v} \leftarrow [H_{13}, H_{23}]^\top$ ▷ $\mathbf{v}$ is $\pi(\mathbf{H}[\mathbf{u}_0\ 1]^\top)$ at $\mathbf{u}_0 = \mathbf{0}$
  $(\gamma, \mathbf{R}_1, \mathbf{R}_2) \leftarrow \text{IPPE}(\mathbf{v}, \mathbf{J})$
  $\mathbf{t}_1 \leftarrow (\mathbf{W}_1^\top \mathbf{W}_1)^{-1}\mathbf{W}_1\mathbf{b}_1$
  $\mathbf{t}_2 \leftarrow (\mathbf{W}_2^\top \mathbf{W}_2)^{-1}\mathbf{W}_2\mathbf{b}_2$ ▷ Solution to (36)
  return $\{\mathbf{R}_1, \mathbf{t}_1\}, \{\mathbf{R}_2, \mathbf{t}_2\}$

4 Experimental Evaluation with Simulated Data

In this section we give a detailed comparison of the performance of IPPE using simulation experiments. We break this section into three parts. The first part compares IPPE against PHD using five different methods to estimate the homography. We have found IPPE combined with Harker and O'Leary's method [16] to perform the best. This performs marginally better than when using the DLT and approximately the same when using the ML estimate. We call this combination IPPE+HO.

In the second section we compare IPPE+HO against competitive state-of-the-art PnP methods. We give a detailed breakdown of this comparison along two axes. The first is the number of correspondences n, which we break down into small n (*i.e.* between 4 and 10) and medium-to-large n (*i.e.* between 8 and 50). The second axis is broken down into simulations where the PPE problem is unambiguous, and simulations where it is ambiguous. When a simulation is ambiguous, it means that there are multiple pose solutions that can reasonably explain the image data. In these cases we do not force the methods to return a single solution, but instead they can return multiple solutions. The best of these solutions with respect to ground-truth is used to measure the methods' accuracy. By contrast in unambiguous cases, the methods are forced to return a single solution as the one with smallest reprojection error, and it is only this solution which is evaluated.

In the third section we compare IPPE against solving pose via P3P, using three virtual correspondences estimated from the homography. The purpose of this evaluation is to test whether IPPE performs better than using some other strategies for positioning the virtual correspondences.

4.1 Simulation Setup

We use a testing framework similar to [23,24]. A perspective camera is setup and a planar model is embedded and projected into the camera's image. The model is a zero-centred square region on the plane $z = 0$ with variable width w. The camera has width 640 and height 480 pixels and the intrinsic matrix is:

$$\mathbf{K} = \begin{bmatrix} f & 0 & 320 \\ 0 & f & 240 \\ 0 & 0 & 1 \end{bmatrix} \tag{37}$$

with f being the focal length with a default $f = 800$ pixels. We then randomly sample from the space of rigid

embeddings as follows. We uniformly sample a point in the image $\hat{\mathbf{p}}$ and create the ray $\mathbf{w} = [\hat{\mathbf{p}}^\top 1]^\top$. We then project this ray out to a random depth d. d is uniformly drawn from the interval $d \sim \mathrm{U}(f/2, 2f)$. We then compute the translation component as $\mathbf{t} = d\mathbf{w}$. The rotation $\mathbf{R}$ is determined as follows. We first create an in-plane rotation $\mathbf{R}(\theta)$, $\hat{\theta} \sim \mathrm{U}(0, 2\pi)$. This is followed by an out-of-plane rotation $\mathbf{R}(\psi, q_x, q_y)$ with axis $\mathbf{r} = 1/k[q_x, q_y, 0]^\top$, $k = \|[q_x, q_y]\|_2$ and $q_x, q_y \sim \mathrm{U}(-1, +1)$. The angle is $\psi \sim \mathrm{U}(0, \psi_{max})$. ψ_{max} denotes the maximum angle in radians such that the plane's tilt angle with respect to the viewing ray is less than 80 degrees. The rotation is given by $\mathbf{R} = \mathbf{R}(\psi, q_x, q_y)\mathbf{R}(\theta)$. We then synthesise n point correspondences. Their positions in the model plane are $\{[u_i, v_i]^\top\}$ with $u_i, v_i \sim \mathrm{U}(-w/2, +w/2)$. These points are then projected in the image via $\{\mathbf{K}, \mathbf{R}, \mathbf{t}\}$ to give their corresponding image positions $\{[x_i, y_i]^\top\}$. To measure an algorithm's sensitivity to noise in the image we perturb each point (x_i, y_i) with additive zero-mean Gaussian noise with standard deviation σ_I. We also test sensitivity to noise in the model view by perturbing each point (u_i, v_i) with Gaussian noise with standard deviation σ_M. We keep only those embeddings where all point correspondences lie in front of the camera and project within the image. We denote the tuple $(\{u_i, v_i\}_k, \{x_i, y_i\}_k, \mathbf{R}_k, \mathbf{t}_k)$ to be the data for the k^{th} test sample.

4.2 Well-Posed and Ill-Posed Conditions

In the special case when $\sigma_I = \sigma_M = 0$ planar pose is recoverable uniquely. When $\sigma_I > 0$ and/or $\sigma_M > 0$ there may be instances when pose estimation is ambiguous. That is, an alternative rigid hypothesis P_2 exists which projects the point set $\{u_i, v_i\}$ close to $\{x_i, y_i\}$. It is important to separate ambiguous from unambiguous cases. In an ambiguous case a method returning a single solution may pick an incorrect pose similar to P_2. In this case it is not the algorithm which is to blame for these errors but the posedness of the problem. We therefore measure performance for each algorithm in two modes.

Mode 1 is where each algorithm returns *one* solution. HD methods always return one solution. IPPE, and most PnP methods can return multiple solutions. In Mode 1 we force these algorithms to return the solution with lowest reprojection error. In order to obtain meaningful statistics we must ensure that test samples in Mode 1 are sufficiently unambiguous. In §3.4 we have shown that pose is ambiguous iff an affine homography can model the transformation between correspondences. To judge whether a test sample $(\{u_i, v_i\}_k, \{x_i, y_i\}_k, \mathbf{R}_k, \mathbf{t}_k)$ is ambiguous we measure how many times more likely the data is predicted by a perspective homography $\mathbf{H}_p$ than an affine homography $\mathbf{H}_a$. We compute $\mathbf{H}_p$ with the ground truth transform $(\mathbf{R}, \mathbf{t})$ and refine with Gauss-Newton iterations. We compute $\mathbf{H}_a$ with a least squares fit of the correspondences (which is also the ML estimate for affine projection). We then measure the log-likelihood ratio:

$$D = l\left(\{x_i, y_i\}_k; \{u_i, v_i\}_k, \sigma_I, \mathbf{H}_p\right) - l\left(\{x_i, y_i\}_k; \{u_i, v_i\}_k, \sigma_I, \mathbf{H}_a\right) \quad (38)$$

$l(\cdot;\cdot,\cdot\mathbf{H})$ denotes the data log-likelihood given the transform $\mathbf{H}$. We judge a sample to be ambiguous if $D < \tau_a$. Only unambiguous samples are selected for testing algorithms in Mode 1. A small τ_a means that more samples are rejected as being ambiguous whereas a larger τ_a means fewer. It is not critical for us to finely tune τ_a, we merely wish to select a value which eliminates cases which are clearly ambiguous to ensure that algorithms tested in Mode 1 are tested in well-conditioned cases. In mode 1 we use $\tau_a = 5$.

Mode 2 is when we keep *all* samples, and allow algorithms to return multiple solutions.

4.3 Summary of Experimental Parameters and Error Metrics

In Table 1 we give a summary of the experimental free parameters for the synthetic experiments. We denote $\{\hat{\mathbf{R}}_k, \hat{\mathbf{t}}_k\}$ to be the rotation and translation estimated by a given algorithm given $(\{u_i, v_i\}_k, \{x_i, y_i\}_k, \mathbf{R}_k, \mathbf{t}_k)$. Similarly to previous works [23,24] we measure error with two metrics:

1. $RE(\hat{\mathbf{R}})$. The Rotational Error (in degrees). This is the angle of the minimal rotation needed to align $\hat{\mathbf{R}}$ to $\mathbf{R}$. This is given by taking the angle of the axis/angle representation of $\hat{\mathbf{R}}^\top\mathbf{R}$.
2. $TE(\hat{\mathbf{t}})$. The Translational Error (%). This is the relative error in translation, given by $TE(\hat{\mathbf{t}}, \mathbf{t}) = \|\hat{\mathbf{t}} - \mathbf{t}\|_2 / \|\mathbf{t}\|_2$.

Parameter	Meaning
f	Focal length
w	Model plane width
n	Number of correspondences
σ_I	Correspondence noise (image)
σ_M	Correspondence noise (model)
Mode	Either in Mode 1 or Mode 2 (§4.2)

Table 1 Free parameters in synthetic experiments.

For each error metric we measure three statistics; the standard deviation, the mean and median error.

4.4 IPPE versus Perspective Homography Decomposition

We start by comparing IPPE against the two existing PHD methods, which we denote by HDSt [35] HDZh [40]. Because these return only a single solution we perform the tests in Mode 1 (*i.e.* unambiguous cases). We compare across 5 different Homography Estimation (HE) methods. This is to (*i*) assess the sensitivity of an algorithm with respect to the choice of HE method, and (*ii*) to determine which HE method leads to best pose estimates. The HE methods we test are as follows:

1. **DLT** (non iterative). The Direct Linear Transform [17].
2. **TAUB** (non iterative). The Taubin estimate [36].
3. **HO** (non iterative). Harker and O'Leary [16] based on Total Least Squares (TLS) with equilibration.
4. **MLGN** (iterative). ML minimiser using Gauss-Newton iterations. MLGN is initialised with the best non-iterative solution from 1,2 or 3.
5. **STGN** (iterative). Symmetric transfer error minimiser using Gauss-Newton iterations. STGN is initialised with the best non-iterative solution from 1,2 or 3. STGN is used in place of MLGN when $\sigma_M > 0$.

We have run a series of 5 experiments (E1 to E5) by varying the parameters in Table 1 to cover a range of imaging conditions. Note that there is redundancy in scaling both f and w, therefore we keep f constant and only vary w. The parameter instantiations for each experiment are shown in Table 2. We present summary statistics over 5,000 simulated poses in Tables 3-7. For each HE method, we have highlighted in blue the pose estimation method which gives the lowest average error. TAUB consistently performs the worst for HDZh, HDSt and IPPE. We see that using the DLT gives lowest errors for HDZh and HDSt. The best performing HE method for IPPE is HO, which is very closely followed by DLT. HO is also the fastest method; between 5-6 times faster to compute than the DLT [16]. We also see that IPPE consistently outperforms HDZh and HDSt for all HE methods. A visual comparison of methods is shown in the graphs in Figure 1. The five rows correspond to the five experiments, and the columns show mean and median errors in rotation and translation. To reduce clutter we plot results only with the best performing HE method for HDZh and HDSt (the DLT). We can see a clear improvement in performance for IPPE in all error statistics, across all experiments. Also it shows that IPPE is rather insensitive to the choice of HD method.

	E1	**E2**	**E3**	**E4**	**E5**
f	800	800	800	800	800
w	200	300	200→400	250	350
n	10	5→40	12	15	8
σ_I	0→6	2	3	3.5	3.5
σ_M	0	0	0	0→7	0→5
Mode	1	1	1	1	1

Table 2 Varying imaging conditions in synthetic experiments E1-E5.

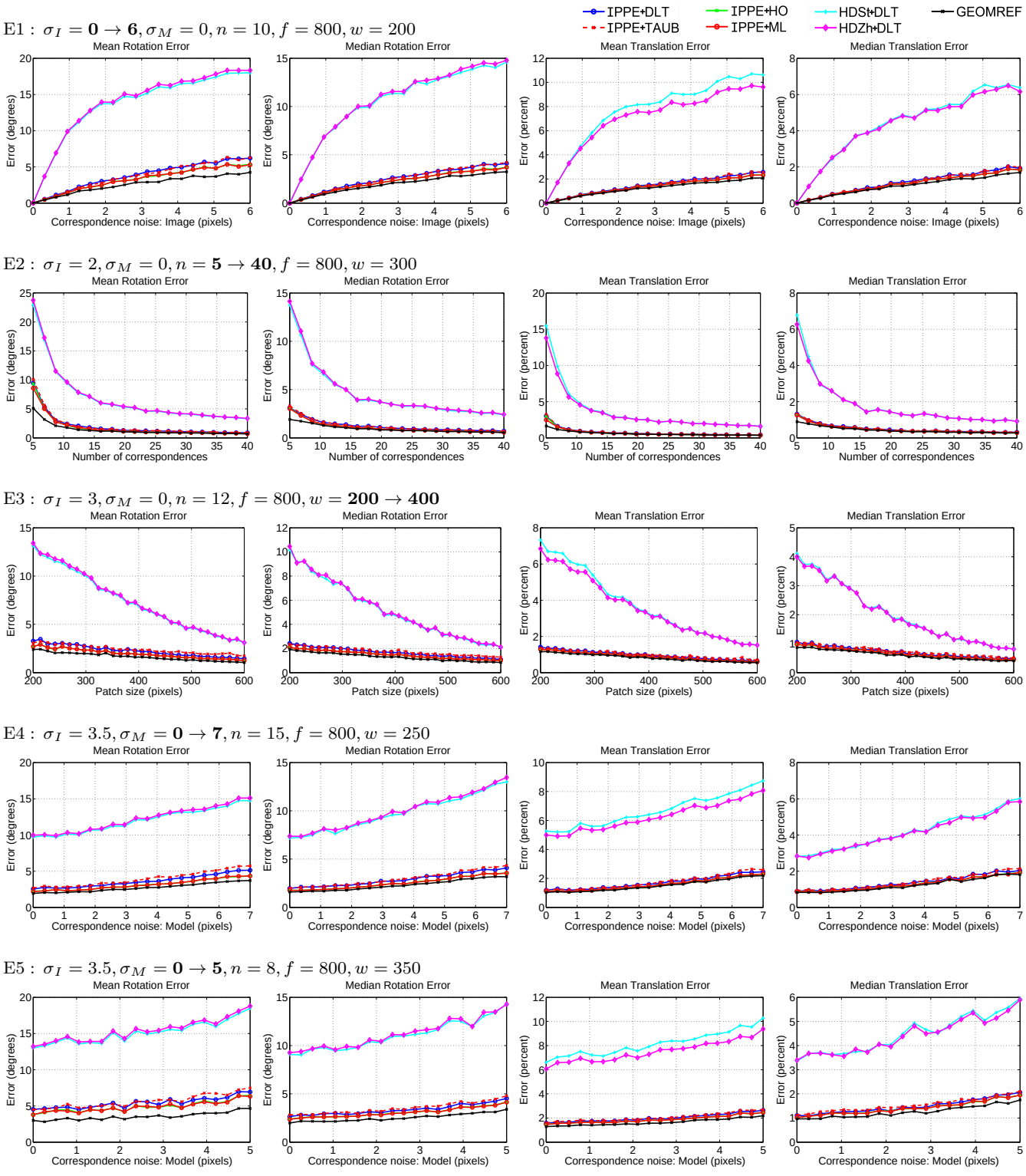

Fig. 1 Synthetic experimental results: Comparing the pose estimation accuracy of IPPE with PHD (E1-E5).

E1: RE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0
0.632	1.06±2.44	6.76±6.68	6.87±6.8	1.11±2.89	6.83±6.92	6.93±7.07	0.949±2.09	6.43±6.57	6.53±6.68	0.948±2.09	6.43±6.55	6.53±6.67
1.58	2.52±5.69	12.5±10.8	12.6±11	2.63±6.8	12.7±11.6	12.8±11.8	2.23±5.2	11.9±10.6	12±10.7	2.23±5.2	11.9±10.5	12±10.7
2.21	3.28±8.03	14.2±12.6	14.4±12.9	3.22±7.79	13.7±12.4	13.9±12.7	2.99±7.82	13±11.9	13.2±12.2	2.93±7.48	13±11.8	13.2±12.1
3.16	4.05±7.6	15.8±13.2	16.1±13.7	4.33±9.18	15.2±13.4	15.6±14.1	3.66±7.66	14.3±12.4	14.7±13.1	3.79±8.39	14.1±12.1	14.5±12.7
3.79	4.7±8.85	16.3±12.7	16.6±12.9	4.84±10.3	15.9±13.3	16.2±13.7	4.07±8.73	14.9±12.2	15.2±12.4	4.07±8.73	14.9±12.1	15.1±12.3
E1: TE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0
0.632	0.42±0.401	3.34±4.48	3.26±4.21	0.437±0.41	3.37±4.48	3.29±4.24	0.403±0.377	3.19±4.37	3.12±4.13	0.403±0.377	3.18±4.36	3.11±4.13
1.58	0.95±0.873	6.87±8.75	6.43±7.72	0.965±0.884	6.85±8.9	6.41±7.81	0.91±0.83	6.48±8.41	6.1±7.43	0.908±0.834	6.45±8.38	6.08±7.41
2.21	1.2±1.12	7.98±9.87	7.32±8.53	1.2±1.17	7.98±10.2	7.29±8.77	1.13±1.07	7.45±9.61	6.86±8.33	1.12±1.06	7.41±9.55	6.83±8.25
3.16	1.6±1.46	8.84±10.3	8.12±8.89	1.59±1.37	8.37±10.1	7.71±8.72	1.49±1.32	8.01±9.56	7.4±8.28	1.48±1.32	7.91±9.34	7.33±8.09
3.79	1.83±1.6	9.31±10.1	8.48±8.64	1.84±1.61	9.01±10.2	8.15±8.66	1.7±1.45	8.51±9.58	7.75±8.2	1.7±1.46	8.46±9.47	7.71±8.13

Table 3 IPPE versus Perspective Homography Decomposition: Experiment E1

E2: RE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
5	10.1±22.1	23.4±25.7	24.2±26.8	9.46±21	22.9±25	23.8±26.3	9.33±21.3	22.3±24.6	23.1±25.7	8.59±19.2	22±24.1	22.7±25.2
8.68	3.14±7.28	11.4±11.3	11.6±11.3	2.98±6.59	11.4±11.6	11.5±11.6	2.79±6.95	10.7±10.8	10.7±10.7	2.75±6.78	10.6±10.7	10.7±10.7
14.2	1.75±2.01	7.55±6.79	7.67±6.88	1.79±3.41	7.05±6.74	7.14±6.79	1.51±2.84	6.7±6.32	6.78±6.38	1.55±3.19	6.66±6.34	6.74±6.4
17.9	1.52±1.31	6.11±5.37	6.23±5.61	1.51±1.2	5.71±5.23	5.8±5.33	1.29±0.985	5.28±4.81	5.37±4.93	1.29±0.986	5.28±4.8	5.36±4.93
23.4	1.25±1.11	4.95±4.11	5.05±4.2	1.21±0.944	4.57±4.03	4.64±4.09	1.01±0.779	4.32±3.81	4.39±3.88	1.01±0.78	4.31±3.8	4.38±3.87
27.1	1.2±1.17	4.73±4.07	4.8±4.14	1.15±0.921	4.32±3.86	4.39±3.91	0.96±0.732	4.08±3.62	4.14±3.68	0.959±0.733	4.08±3.61	4.13±3.67
E2: TE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
5	3.23±10.4	16.1±21.1	14.4±19.1	3.01±10.1	15.5±20.5	13.8±18.5	2.91±9.18	15.2±20.2	13.6±18.2	2.49±6.62	14.6±19.2	13±17.1
8.68	1.14±1.25	7.85±9.83	7.22±9.01	1.15±1.22	6.07±8.63	5.65±7.49	1.07±1.13	5.67±8.23	5.26±7.14	1.07±1.14	5.62±8.07	5.23±7
14.2	0.785±0.688	3.64±4.54	3.56±4.26	0.765±0.674	3.56±4.61	3.47±4.3	0.72±0.61	3.34±4.27	3.26±4.03	0.719±0.611	3.31±4.24	3.24±4.01
17.9	0.669±0.581	3.05±3.73	2.99±3.51	0.663±0.601	2.83±3.48	2.77±3.28	0.61±0.541	2.66±3.3	2.62±3.13	0.61±0.54	2.64±3.3	2.6±3.12
23.4	0.533±0.458	2.45±2.81	2.42±2.71	0.518±0.466	2.19±2.67	2.18±2.6	0.488±0.43	2.11±2.57	2.1±2.5	0.487±0.43	2.1±2.55	2.08±2.48
27.1	0.515±0.438	2.28±2.69	2.26±2.62	0.512±0.437	2.19±2.57	2.18±2.53	0.469±0.397	2.06±2.43	2.06±2.39	0.468±0.397	2.05±2.43	2.04±2.39

Table 4 IPPE versus Perspective Homography Decomposition: Experiment E2

E3: RE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
200	3.07±4.36	13.5±10.5	13.7±10.8	3.27±5.9	13.1±10.5	13.4±11	2.7±4.19	12.3±9.92	12.5±10.2	2.7±4.18	12.2±9.84	12.5±10.2
228	3.12±5.19	12.5±10.5	12.8±10.9	2.96±4.06	11.9±10.1	12.2±10.5	2.61±4.12	11.2±9.83	11.5±10.2	2.6±4.11	11.2±9.67	11.5±10.1
269	2.94±4.11	11.2±9.66	11.5±10	2.91±5.7	10.8±10	11±10.2	2.51±4.05	10.1±9.2	10.3±9.49	2.51±4.05	10±9.02	10.2±9.3
297	2.77±4.72	10.6±9.33	10.8±9.44	2.75±5.33	10.1±9.28	10.2±9.4	2.33±4.59	9.5±8.71	9.63±8.79	2.32±4.59	9.49±8.81	9.64±8.9
338	2.73±4.01	9.22±8.51	9.36±8.67	2.61±4.34	8.52±8.39	8.6±8.47	2.28±3.62	8.1±7.82	8.2±7.94	2.28±3.62	8.13±7.94	8.22±8.06
366	2.39±3.08	8.38±7.54	8.52±7.57	2.26±3.24	7.88±7.47	8.01±7.59	1.95±2.6	7.3±6.77	7.42±6.84	1.95±2.61	7.29±6.8	7.4±6.85
E3: TE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
200	1.35±1.2	7.66±9.23	7.09±8.03	1.41±1.29	7.33±8.9	6.84±7.77	1.28±1.17	7.02±8.59	6.54±7.51	1.28±1.16	6.97±8.53	6.49±7.45
228	1.33±1.12	6.88±8.45	6.42±7.46	1.34±1.21	6.65±8.29	6.2±7.29	1.23±1.05	6.23±7.93	5.81±7.05	1.23±1.05	6.16±7.86	5.76±7
269	1.22±1.05	6.21±7.92	5.76±6.91	1.2±1.04	5.98±7.7	5.56±6.73	1.1±0.946	5.61±7.31	5.24±6.42	1.1±0.946	5.56±7.18	5.2±6.29
297	1.12±1.03	5.85±7.12	5.48±6.27	1.12±1.02	5.39±6.75	5.08±5.99	1.04±0.945	5.15±6.43	4.87±5.72	1.04±0.943	5.08±6.34	4.81±5.67
338	1.13±1.02	4.58±5.77	4.41±5.3	1.08±1	4.17±5.28	4.02±4.85	1.02±0.933	4.03±5.09	3.9±4.72	1.02±0.935	4±5.02	3.87±4.68
366	0.993±0.869	4.04±4.73	3.9±4.37	0.985±0.868	3.91±4.9	3.82±4.59	0.906±0.803	3.67±4.47	3.59±4.22	0.905±0.8	3.65±4.45	3.57±4.21

Table 5 IPPE versus Perspective Homography Decomposition: Experiment E3

E4: RE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
0	2.67±3.53	10.3±7.91	10.6±8.19	2.53±3.94	9.71±8.19	9.98±8.48	2.23±3.29	9.11±7.43	9.38±7.71	2.18±2.77	9.06±7.4	9.33±7.68
0.737	2.82±3.81	10.7±8.23	10.9±8.52	2.66±3.27	9.7±8.21	9.91±8.44	2.44±5.09	9.23±7.55	9.4±7.76	2.44±5.1	9.16±7.43	9.33±7.62
1.84	3±3.09	11.8±9.3	12±9.48	2.95±4.5	10.7±8.95	10.8±9.04	2.45±2.59	10.2±8.39	10.3±8.5	2.46±2.66	10.1±8.36	10.3±8.46
2.58	3.37±4.15	12.3±9.42	12.6±9.67	3.24±3.92	11.2±8.99	11.5±9.22	2.8±3.38	10.6±8.36	10.8±8.55	2.79±3.38	10.5±8.29	10.8±8.47
3.68	3.97±6.09	13.4±9.42	13.7±9.52	3.57±5.17	12.1±9.38	12.2±9.49	3.13±5.06	11.5±8.66	11.7±8.75	3.06±4.5	11.4±8.52	11.6±8.62
4.42	4.16±4.08	14.7±9.75	15±9.98	3.9±4.36	13±8.92	13.1±9.06	3.28±2.67	12.3±8.38	12.5±8.52	3.28±2.66	12.2±8.37	12.4±8.52
E4: TE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
0	1.22±1.06	5.85±6.91	5.54±6.16	1.19±1.06	5.26±6.47	5±5.85	1.11±0.954	5.01±6.19	4.79±5.63	1.11±0.948	5±6.14	4.78±5.57
0.737	1.21±1.09	5.69±7.06	5.39±6.36	1.19±1.03	5.22±6.38	4.94±5.71	1.1±0.942	4.99±6.21	4.75±5.6	1.1±0.943	4.96±6.12	4.73±5.52
1.84	1.38±1.12	6.28±7.32	5.94±6.56	1.38±1.17	5.63±6.51	5.37±5.82	1.26±1.05	5.35±6.24	5.09±5.61	1.26±1.05	5.32±6.14	5.05±5.54
2.58	1.47±1.19	6.81±7.65	6.38±6.75	1.47±1.21	6.22±7.33	5.85±6.43	1.37±1.08	5.88±6.9	5.54±6.1	1.37±1.08	5.83±6.83	5.5±6.05
3.68	1.75±1.71	7.52±8.22	7.03±7.29	1.66±1.33	6.56±7.29	6.2±6.46	1.55±1.23	6.4±7.14	6.05±6.38	1.55±1.25	6.31±7.02	5.98±6.28
4.42	1.9±1.52	8.07±8.66	7.46±7.59	1.82±1.49	7.24±8.13	6.73±7.1	1.69±1.32	6.88±7.76	6.41±6.8	1.69±1.32	6.74±7.63	6.3±6.7

Table 6 IPPE versus Perspective Homography Decomposition: Experiment E4

E5: RE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
0	4.49±9.32	13.8±13.4	14.1±13.8	4.54±10.4	13±13.4	13.2±13.7	3.83±8.71	12.4±12.6	12.6±13	3.79±8.54	12.3±12.5	12.5±12.8
0.526	4.73±9	14.5±13.7	14.7±14	4.78±9.8	13.8±13.2	14±13.5	4.41±9.69	13.3±13	13.4±13.2	4.36±9.57	13.1±13	13.3±13.2
1.32	4.86±9.02	14.2±13.3	14.4±13.5	4.81±10.2	13.7±13.4	13.9±13.6	4.47±9.79	13±12.8	13.1±13	4.47±9.81	12.9±12.5	13.1±12.7
1.84	5.26±10.1	15.7±14.6	15.9±14.8	5.4±11.5	15.1±15.6	15.3±15.9	4.68±9.86	14.3±14.4	14.4±14.6	4.72±10.3	14.2±14.4	14.4±14.6
2.63	5.59±10.9	15.6±13.2	15.9±13.6	5.57±11.6	14.9±12.8	15.2±13.2	4.78±9.91	14.3±12.7	14.6±13.2	4.93±10.6	14.1±12.4	14.4±13
3.16	5.85±10.8	16.3±13.8	16.7±14.3	5.89±12.1	15.5±14.2	15.9±14.8	5.08±10.3	14.8±13.4	15.2±14.1	5.26±11.1	14.8±13.4	15.2±14
E5: TE	TAUB			DLT			HO			ML		
σ_I	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh	IPPE	HDSt	HDZh
0	1.6±1.55	7.23±9.51	6.61±8.21	1.58±1.54	6.61±8.74	6.07±7.55	1.45±1.39	6.41±8.61	5.91±7.46	1.46±1.39	6.35±8.46	5.87±7.34
0.526	1.7±1.6	7.49±9.76	6.98±8.51	1.66±1.58	7.14±9.47	6.62±8.25	1.58±1.49	6.84±9.15	6.39±8	1.57±1.48	6.72±8.95	6.29±7.79
1.32	1.81±1.74	7.49±9.43	6.93±8.13	1.75±1.67	7.14±9.36	6.67±8.15	1.62±1.5	6.86±9.25	6.43±8.02	1.63±1.49	6.82±9.06	6.38±7.87
1.84	1.91±1.78	8.36±10.9	7.68±9.36	1.87±1.76	7.82±10.3	7.23±8.88	1.74±1.6	7.45±9.93	6.94±8.6	1.75±1.6	7.31±9.63	6.82±8.35
2.63	1.95±1.82	8.83±10.6	8.11±9.19	1.9±1.83	8.28±9.99	7.65±8.74	1.8±1.59	7.98±9.89	7.39±8.57	1.8±1.56	7.84±9.62	7.26±8.3
3.16	2.15±1.96	8.83±10.4	8.15±9.05	2.06±1.87	8.35±10.3	7.75±8.98	1.92±1.7	8±9.86	7.46±8.62	1.91±1.69	7.85±9.59	7.35±8.43

Table 7 IPPE versus Perspective Homography Decomposition: Experiment E5

4.5 IPPE versus PnP methods

We now compare IPPE against state-of-the-art PnP methods. We use HO for estimating the homography between correspondences. The following names are used for the compared methods:

- **IPPE+HO** (non iterative): Proposed method using homography estimated with [16].
- **RPP-SP** (non iterative): Schweighofer and Pinz [34]. This is the extension of Lu *et al.* [28] to handle ambiguities.
- **EPnP** (non iterative): Moreno-Noguer *et al.* [34].
- **RPnP** (non iterative): Li *et al.* [24].
- **HDZh+DLT** (non iterative): The best performing HD method.
- **GEOMREF** (iterative): Iterative refinement using a geometric criteria (see below).

GEOMREF is used as the gold standard. This is initialised using the compared method which gives the solution with the lowest residual error, and refined with Gauss-Newton iterations. When $\sigma_M = 0$ we use the ML error as the geometric cost. When $\sigma_M > 0$ we use the symmetric transfer error.

We start with a series of Mode 1 experiments. We divided these into two parts. The first part measures performance when the number of correspondences is medium-to-large ($n = 8 \rightarrow 50$). The second part measures performance when the number of correspondences is small ($n = 4 \rightarrow 10$). We make this division to assist visualising results as methods perform far better with larger n. The division also helps study two properties; the accuracy of an algorithm with low numbers of correspondences and how well an algorithm exploits correspondence redundancy. In total we perform 12 experiments (E6-E17). There are 6 for $n = 4 \rightarrow 10$ (E6-E11) and 6 for $n = 8 \rightarrow 50$ (E12-E17). The experimental parameters are shown in Table 8.

	E6	**E7**	**E8**	**E9**	**E10**	**E11**
f	800	800	800	800	800	800
w	300	300	300	300	300	300
n	8→50	8→50	8→50	8→50	8→50	8→50
σ_I	0.5	3	8	2	2	2
σ_M	0	0	0	0.5	3	8
Mode	1	1	1	1	1	1

	E12	**E13**	**E14**	**E15**	**E16**	**E17**
f	800	800	800	800	800	800
w	300	300	300	300	300	300
n	4→10	4→10	4→10	4→10	4→10	4→10
σ_I	0.5	3	8	2	2	2
σ_M	0	0	0	0.5	3	8
Mode	1	1	1	1	1	1

Table 8 Varying imaging conditions in synthetic experiments E6-E17.

4.5.1 Medium to Large n

The results for experiments E6-E11 are shown in Figure 2. With respect to rotation we see that across all conditions IPPE+HO is consistently the best performing method (excluding refinement with GEOMREF). There is a clear improvement in performance with respect to the next best non-iterative method (RPnP). The performance of RPP-SP with respect to mean error remains larger than IPPE+HO. With respect to median error, RPP-SP approaches but never exceeds IPPE+HO for larger n. When n goes beyond 15 the performance of IPPE+HO is very close to GEOMREF. Turning to translation error we see a similar ranking of methods. The difference between IPPE+HO and RPP-SP is smaller than for rotation error. There is negligible difference between IPPE+HO and RPP-SP in translation performance in experiments E9-E11 (when noise increases in the model). The next best non-iterative method (RPnP) performs behind IPPE+HO and RPP-SP with respect to translation error for all experiments.

We can see that IPPE+HO is the best performing non-iterative method in the range $n = 8 \rightarrow 50$. We also see that beyond $n = 15$ the performance gains in refining the IPPE+HO solution with GEOMREF are very small in all experiments. This is true when there is correspondence noise in the image, model, or both. The same cannot be said in all experiments for the other methods. This has important practical implications as it suggests that when speed is an important priority, one can do away with iterative refinement and use the IPPE+HO solution. A rule of thumb would be when $n > 15$.

4.5.2 Small n

We now turn to the performance evaluation with $n = 4 \rightarrow 10$. The results are shown in Figure 3. Here we see that for $n \geq 6$ IPPE+HO is the best performing method (excluding GEOMREF) with respect to rotation across all conditions. For $n \geq 6$ IPPE+HO performs as well as or better than the next best method (RPP-SP) with respect to translation. For $n = 4$ IPPE+HO is outperformed by RPnP and RPP-SP. RPnP does well for $n = 4$, although there is a clear performance gap between RPnP and GEOMREF. This gap is larger for larger σ_M, indicating RPnP has difficulty with noise in the model. The performance of IPPE+HO is significantly worse at $n = 4$ than $n = 5$. The reason is two-fold. Firstly the homography is computed from

4 point correspondences, and because of the lack of redundancy the homography overfits. For $n > 4$ there is redundancy and this leads to considerably lower error. The second reason is that the configuration of correspondences in the model affects the sensitivity of homography estimation to noise. Because the correspondences are uniformly sampled on the model plane some configurations can lead to a poorer conditioning of the homography estimation problem. We refer the reader to [5] where a detailed analysis is given on the stability of homography estimation by 1^{st}-order perturbation theory.

Experiments E12-E17 suggest that IPPE+HO should not be used when $n < 6$, as better results would be obtained with RPnP. However in practical applications this is not always true. We now study the case when the model's points are not drawn randomly on the plane, but rather four are located on corners of the square region: $(u, v)_1 = 1/2(w, w)$, $(u, v)_2 = 1/2(w, -w)$, $(u, v)_3 = 1/2(-w, -w)$, $(u, v)_4 = 1/2(-w, w)$. This is typically the case in AR-based planar pose estimation. The remaining $n - 4$ points are positioned with uniform probability within the region. We then studied the algorithms' performances in these configurations. We ran six experiments (E18-E23) using this new sampling scheme. The experimental parameters are listed in Table 9. These are the same as experiments E6-E11, but we have reduced the plane size from 300 to 100. The reason for this is that the new sampling scheme means the correspondences span a larger region on the model, and thus reduces the influence of noise.

The results for these experiments are shown in Figure 4. We see that now IPPE+HO significantly outperforms RPnP with respect to rotation and translation for all n. This is in contrast to when the points are located randomly on the model (Figure 3). The next best performing method is RPP-SP. With respect to rotation, RPP-SP is consistently outperformed by IPPE+HO. With respect to translation IPPE+HO performs at least as good as or better than RPP-SP.

	E18	**E19**	**E20**	**E21**	**E22**	**E23**
f	800	800	800	800	800	800
w	100	100	100	100	100	100
n	4→10	4→10	4→10	4→10	4→10	4→10
σ_I	0.5	2	5	2	2	2
σ_M	0	0	0	0.5	1	3
Mode	1	1	1	1	1	1

Table 9 Varying imaging conditions in synthetic experiments E18-E23. Correspondences are selected four of them positioned on the corners of the plane.

4.5.3 Ambiguous Cases

In the final set of synthetic experiments we investigate algorithm performance in Mode 2 (without excluding ambiguous cases). Here algorithms are permitted to return multiple solutions, and we compute error with respect to the closest solution to the ground truth. Ambiguous cases occur when the amount of perspective distortion is small, which can be controlled by reducing the plane's size. We give the experimental parameters in in Table 10 using the same selection method as E18-E23 with at least four correspondences positioned on the corners of the plane. Here we have reduced the plane size to 50, which meant many ambiguous cases were included. The performance graphs are shown in Figure 5. Here we see a similar performance trend to E18-E23. IPPE+HO consistently does very well. It is the best performing method with respect to rotation (excluding GEOMREF) in all conditions, with a very small gap between IPPE+HO and GEOMREF. The performance gap for smaller n becomes smaller, and for $n = 4$ it is virtually indistinguishable. IPPE+HO performs as well as or better than RPP-SP in translation. HDZh and EPnP performs rather worse than IPPE+HO, RPnP and RPP-SP, and their errors are beyond the axis range.

	E24	**E25**	**E26**	**E27**	**E28**	**E29**
f	800	800	800	800	800	800
w	50	50	50	50	50	50
n	4→10	4→10	8→40	8→10	4→10	4→10
σ_I	0.5	1	2	1	1	1
σ_M	0	0	0	0.5	1	2
Mode	2	2	2	2	2	2

Table 10 Varying imaging conditions in synthetic experiments E24-E29. These experiments tested algorithm performance in Mode 2.

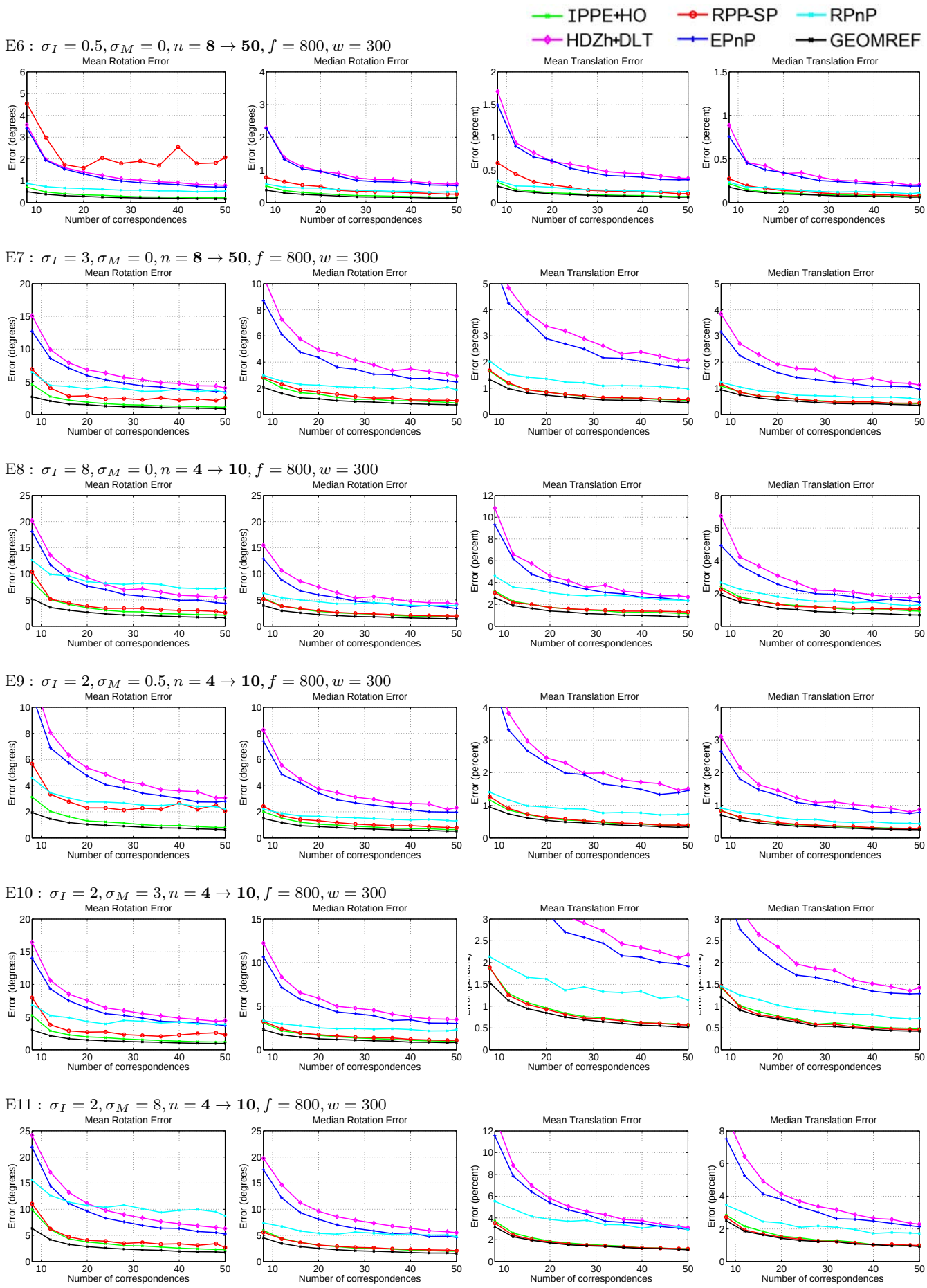

Fig. 2 Synthetic experiments: Comparing pose accuracy of IPPE+HO with previous state-of-the-art methods (E6-E11)

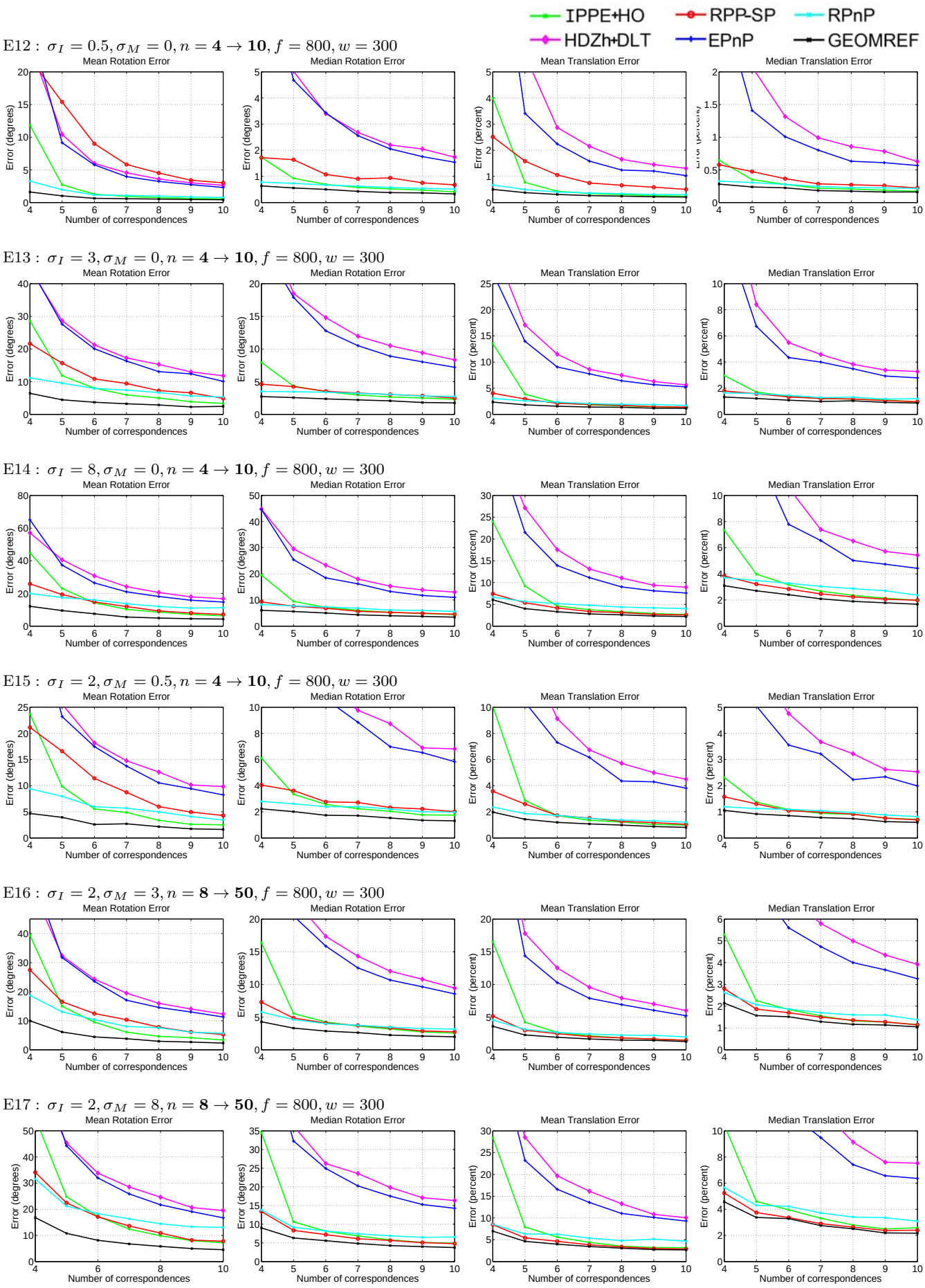

Fig. 3 Synthetic experiments: Comparing pose accuracy of IPPE+HO with previous state-of-the-art methods (E12-E17)

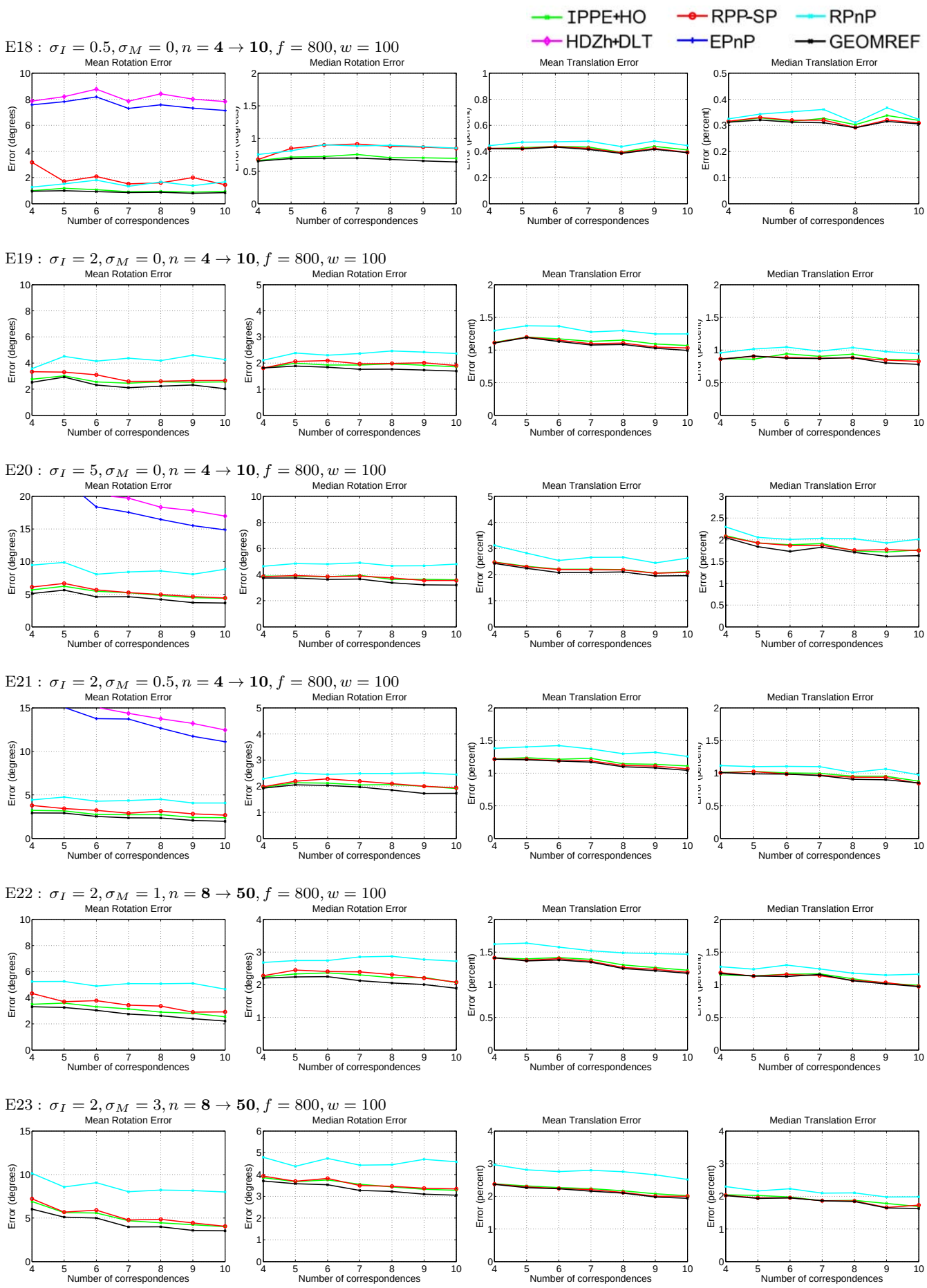

Fig. 4 Synthetic experiments: Comparing pose accuracy of IPPE+HO with previous state-of-the-art methods (E18-E23) The corners of the planar region are used as four correspondences.

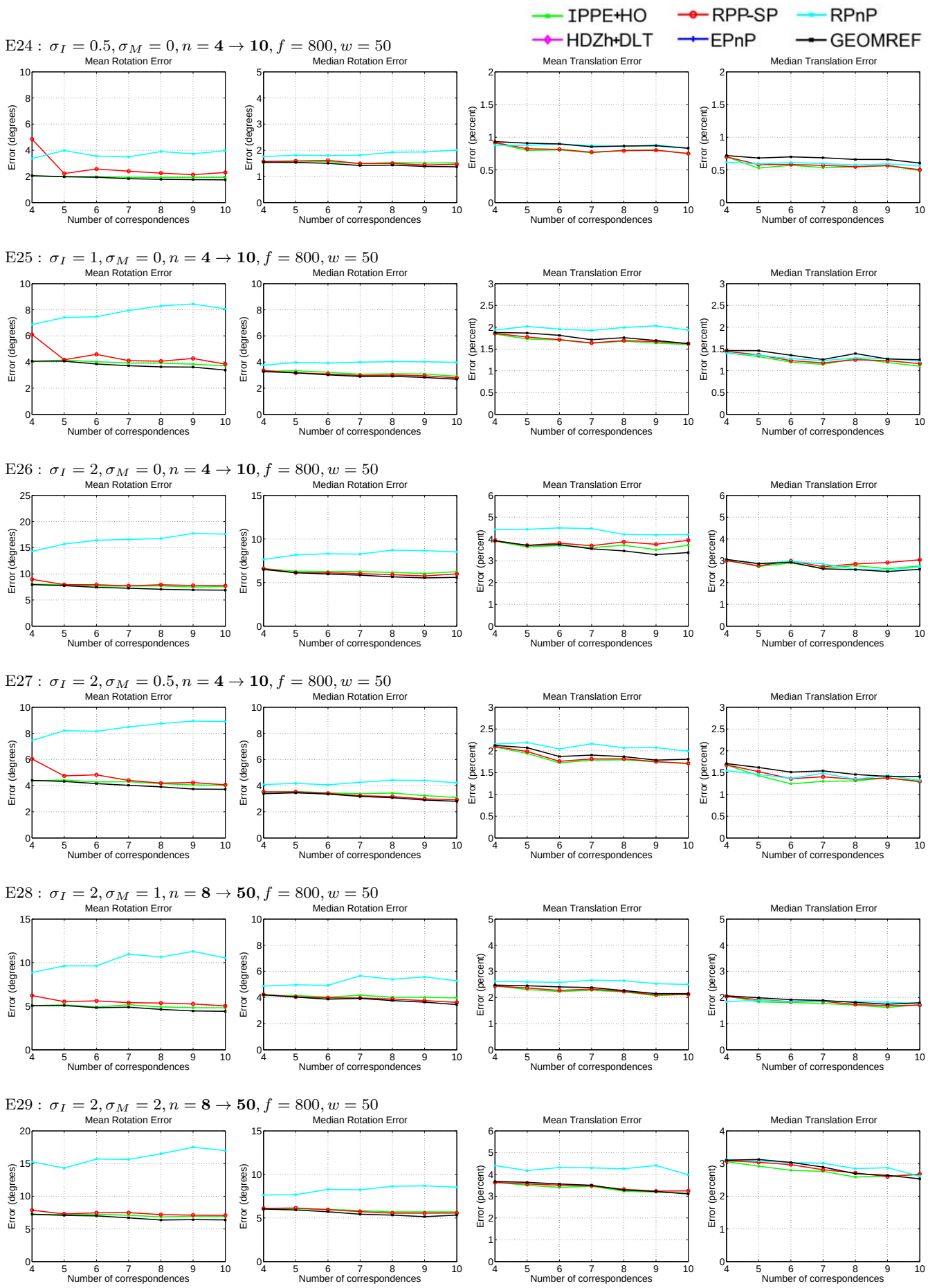

Fig. 5 Synthetic experiments: Comparing pose accuracy of IPPE+HO with previous state-of-the-art methods in Mode (E24-E29). The corners of the planar region are used as four correspondences.

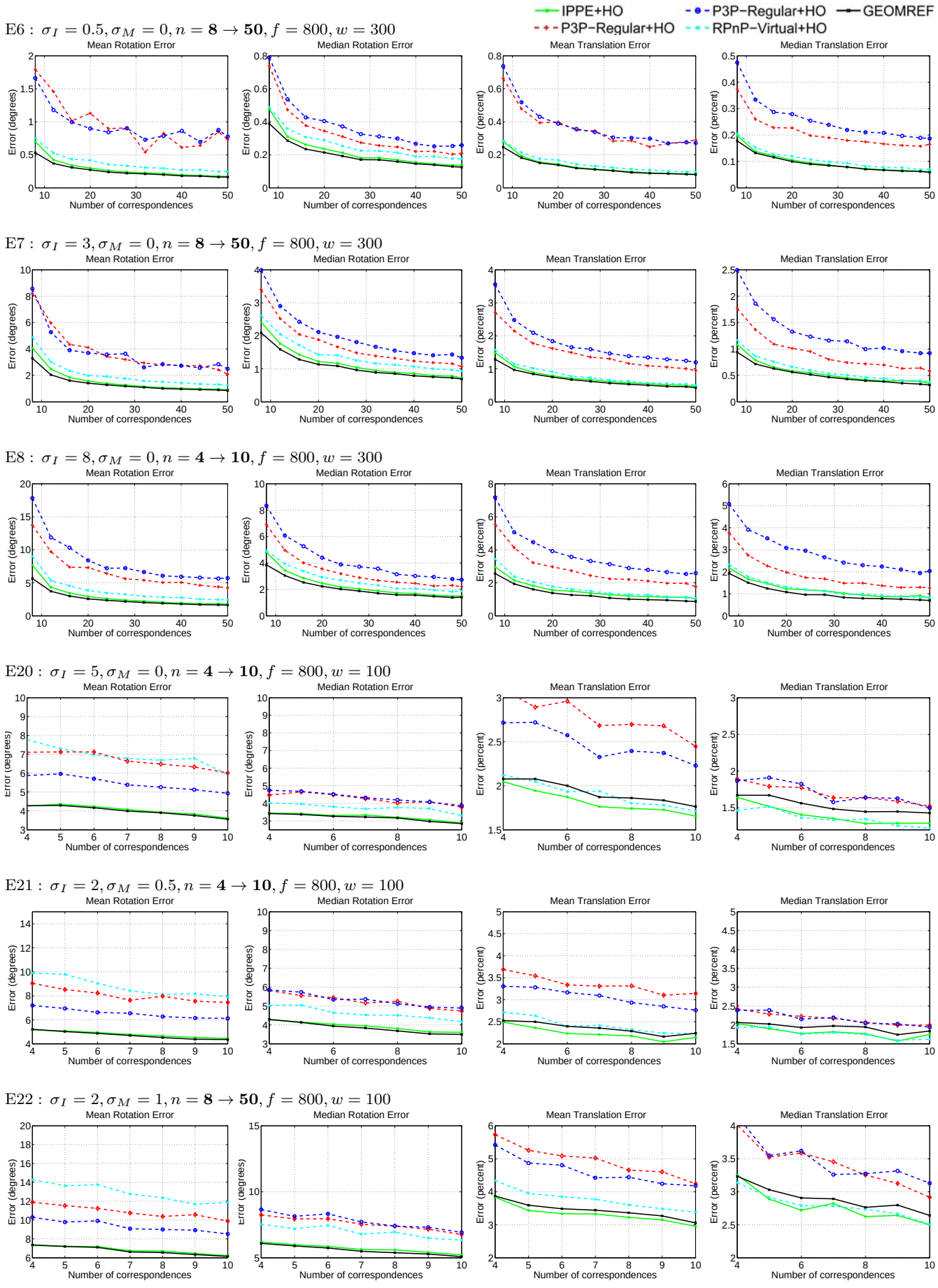

Fig. 6 Synthetic experiments: Comparing pose accuracy of IPPE+HO with P3P performed on virtual point correspondences computed from the homography.

4.6 IPPE Versus P3P with Virtual Correspondences

In the final part of our simulation experiments we compare IPPE against P3P using virtual point correspondences. Specifically given an estimate of $\mathbf{H}$, and a virtual point $\mathbf{u}'_j$ positioned on the model plane, we compute its correspondence in the image with $\mathbf{q}'_j = h(\mathbf{H}[\mathbf{u}'^{\top}_j, 1]^{\top})$. We have tested three different choices for positioning the $\mathbf{u}'_j$. These are as follows:

- **P3P-Random**: We compute the bounding box of $\{\mathbf{u}_i\}$ and position three points randomly within this box.
- **P3P-Regular**: We compute the bounding box of $\{\mathbf{u}_i\}$ and position three points on the bottom-left, top-left and top-right boundaries of this this box.
- **RPnP-Virtual**: We use the original set of points.

Pose is solved for P3P-Random and P3P-Regular using the method in [12]. Pose for RPnP-Virtual is solved using RPnP (which splits the points into multiple P3P problems), but using their positions in the image predicted by $\mathbf{H}$, rather than the measured correspondences. P3P-Random suffers from the problem that it may return zero solutions. We have found that this occurs in practice between 3-4% of the time depending on noise. To make the comparison simple we compute performance statistics for P3P-Random using only instances where it returned at least one solution. By contrast because the points in P3P-Regular are at right-angles, it is guaranteed to return at least one solution, and at most two [12].

To maintain a fair comparison we compared P3P-Random, P3P-Regular and RPnP-Virtual using the homography estimated using HO. We have found that IPPE+HO consistently performs better than P3P-Random, P3P-Regular and RPnP-Virtual across the experiments presented earlier in this section. For brevity we present the results for just for experiments E6-E8 and E18-E20 for these methods. This is given in Figure 6.

5 Experimental Evaluation with Real Data

In this section we evaluate the algorithms on three applications involving real images. The first is to estimate the pose of a planar target from keypoint matches. The second is to estimate the pose of a planar checker-board target. The third is to estimate the pose of small planar AR markers.

5.1 Planar Pose Estimation from Keypoint Matches

In this experiment a series of images of a 120×90mm planar test surface was photographed in normal indoor light conditions. The series comprises 28 images, three of which are shown in Figure 7. The camera used is a Nikon D3100 DSLR with image resolution 2304×1536 pixels. The camera was calibrated with Bouguet's calibration toolbox [3] with focal length $f_x = 3204$ pixels, $f_y = 3220$ pixels. A fronto-parallel model image was constructed by undistorting and rectifying the first of these images. We computed correspondences between the model view and all input images using standard automatic methods. Specifically we used VLFeat's SIFT implementation [38] with putative matches computed using Lowe's ratio test [27] and performed RANSAC to find inlier correspondences (an inlier threshold of 5 pixels was used). This resulted in between 250-400 correspondences found in each image. We then computed gold standard pose estimates for each image using all inlier correspondences by minimising the symmetric transfer error with Gauss-Newton iterations. Given the large number of correspondences all tested methods perform quite well. We compute error statistics over all 28 images. In Table 11 we list accuracy with respect to the gold standard. Here IPPE+HO is the most accurate method with RPP-SP following in second.

	R Error (degrees)	**t** Error (%)
IPPE+HO	0.1249	0.0375
HDZh+DLT	2.8650	0.2691
RPP-SP	1.4951	1.0877
EPnP	1.9347	1.0496
RPnP	0.1850	0.2132

Table 11 Accuracy of algorithms on the 'Game cover' dataset using all correspondences. Accuracy is computed with respect to the gold standard pose combined by GEOMREF.

We used this dataset to study the accuracy of the algorithms as conditions become more challenging. Specifically, when using smaller numbers of correspondences drawn from sub-regions of the model. Conditions become harder as the region becomes smaller because *(i)* there are fewer correspondences and *(ii)* the problem becomes ambiguous because the homography becomes affine. Each image is processed as follows. For each correspondence, we collect all correspondences that lie within a circular window of radius $r$mm. If there are less than 3 neighbouring correspondences we discard the window. Otherwise we compute pose and measure the error with respect to the gold standard pose (computed using the entire surface). We varied r within the range $[5.22...50]$ mm. The results are shown in Figure

8. The first and second rows show the error in rotation and translation respectively as r varies from 5.22mm to 20.9mm. We also give $\bar{n}$; the average number of correspondences within each sub-window. These vary from $\bar{n} = 4.64$ to $\bar{n} = 23.8$. The third and fourth rows show errors in rotation and translation for larger windows; r varying from 26.1mm to 41.1mm. In each graph we plot cumulative error distributions. The distribution of a method performing well will push towards the top left of the axes. For the smallest window size $r = 5.22$mm all methods perform poorly, including GEOMREF. This is because at this small scale the problem is severely ill-conditioned. As r increases all methods perform better. IPPE+HO performs marginally worse than RPP-SP for $r = 5.22$mm ($\bar{n} = 4.64$). However beyond $r = 10.4$mm IPPE+HO consistently performs very close to GEOMREF, and consistently performs as well as or better than the next-best method (RPP-SP). This agrees with the synthetic experiments, where, for randomly positioned correspondences IPPE+HO starts to outperform other methods for $n \geq 8$. Beyond $r = 20.9$mm one can see that IPPE+HO and GEOMREF can find the correct solution nearly all the time; 99.6% of samples have a rotation error less than 10%.

5.2 Pose Estimation of a Planar Checker Pattern

The second set of real experiments involves estimating the pose of a planar checkerboard pattern. We have experimented with two datasets. The first is a series of 20 images captured by a standard 720p smartphone camera in normal indoor lighting conditions. We used a checker surface comprising 21×30 squares each of size 9.22mm. Figure 9 shows three example images in this dataset. The second dataset is a publicly-available one from the Matlab Calibration Toolbox. This comprises 20 images of a 12×12 checkerboard with square size 30mm.

We compute model-to-image correspondences using the Matlab Calibration Toolbox. This involves manually clicking the four corners of the model in an image, and the corresponding homography is used to initialise all checker corners. These are then refined to sub-pixel accuracy with gradient descent. For the first dataset we have 628 correspondences per image. All methods perform well using this amount of data. To differentiate the methods we perform a similar experiment to §5.1 to see how well they perform on smaller checkerboard. For each image, we draw all $m \times m$ checker sub-patterns, where m was varied from 2 to the width of the checkerboard. We then compute pose for each sub-region and compare to the gold standard. Figure 10 shows the results for the first dataset. Here we see that IPPE+HO and RPP-SP are virtually indistinguishable from GEOMREF. However, as we will show from Table 12 IPPE+HO is between 50 and 70 times faster than RPP-SP. HDZh and EPnP perform significantly worse. We see a similar trend for the dataset from the Matlab Calibration Toolbox in Figure 11.

5.3 Pose Estimation of Augmented Reality Markers

In the last set of experiments we evaluate performance for estimating the pose of AR markers. This task typically involves the following processing pipeline: *(i)* to detect the position of the marker approximately in the input image. This involves finding image regions which match a marker's characteristic pattern. *(ii)* to refine the four corner positions to sub-pixel accuracy. *(iii)* to use the corners to estimate 3D pose. *(iv)* (optional) pose refinement. Here we compare the accuracy of IPPE to previous methods for solving *(iii)*. Because $n = 4$ the homography is computed exactly from the point correspondences (*i.e.* there is no redundancy since the correspondences provided 8 constraints on the homography). When $n = 4$ the homography can be solved very efficiently with an analytic solution.

We use the following experimental setup. The open source library ArUcO [29] is used to generate 300 uniquely-identifiable markers each of width 7.90mm. The markers were rotated by a random angle and distributed evenly over 9 A4 sheets of paper. These sheets were printed using a high-precision laser printer, corrected for anisotropic printer scaling. The papers were then fixed to a large planar background surface, by tiling them in a 3×3 grid. We ran plane-based bundle-adjustment to accurately determine the relative positions of each sheet of paper on the background. This allowed us to have a composite planar model of all 300 AR markers.

We then captured two video sequences with a 720p smartphone camera. The first one viewed the markers at close range, with the average distance between sensor and plane to be 52.1cm. The second was at mid-range with the average distance of 102.2cm. We ran ArUcO's marker detector and rejected any video frames where fewer than 10 markers were detected (typically occurring when high motion blur is present). From the remaining frames we randomly selected 30 from both videos to comprise two test sets. Example frames from the close and medium range sets are shown in the top and bottom rows of Figure 12 respectively. We then tested the performance of the algorithms for these datasets. For each image in a dataset, a gold-standard pose was computed using gradient-based refinement using the positions of *all* detected AR markers. The per-

formance of an algorithm was measured by how close its pose estimate using a *single* AR marker was to the gold standard. We plot the results in Figure 13. We compute rotation error, and also the error of the estimated depth of the centre of the AR marker (in mm). Here we see that IPPE, RPnP and RPP-SP are the best performing methods and perform very close to GEOMREF. RPnP performs very slightly worse for rotation than IPPE and RPP-SP. There is a noticeable tail in rotation error for GEOMREF; approximately 5% of markers have errors greater than 10 degrees. The reason for these outliers is because of tracking errors; very occasionally the corner predictions are far from their true positions (*e.g.* greater than 5 pixels) when the gradient-based refinement gets trapped in an incorrect local minimum. HDZh and EPnP are significantly worse at solving this problem, and show a significant performance drop for the mid-range dataset. Even though the accuracy of IPPE, RPnP and RPP-SP is quite similar, IPPE is significantly faster to compute. Because the homography is computed analytically, IPPE computes pose *entirely analytically* using only simple floating point operations for this problem. This is in contrast to RPP-SP, which is iterative, and RPnP, which involves numerical root finding for a 7^{th} order polynomial.

5.3.1 Timing Information

We have computed the time required to perform each of the compared methods as a function of n. This has been done on a standard Intel i7-3820 desktop PC running 64-bit Matlab 2012a. For all compared algorithms we use the code provided by the authors. We use our own Matlab implementation of IPPE. Note that these are not the fastest implementations, and speedups would be gained with for example C implementations. However benchmarking all methods with Matlab gives a fair comparison and reveals how computation time scales with n. For a given n we simulated 500 randomised configurations using the simulation setup in §4.1. Figure 14 and Table 12 shows processing time as n varies from 4 to 650. For IPPE and HDZh with $n = 4$, we use an analytic formula to estimate the homography. This requires approximately 50 floating-point operations and is faster than solving with DLT and HO, yet yields the same result. RPP-SP is by far the slowest method. IPPE+HO is the fastest method. It is marginally faster than HDZh+DLT, but considerably faster than EPnP, RPnP and RPP-SP. In Table 12 at $n = 4$ we see that IPPE is approximately 6.7 times faster than EPnP and 6.2 times faster than RPnP. IPPE is approximately 75 times faster than RPP-SP. EPnP, RPnP and IPPE are all $O(n)$ methods. We can see from Figure 14 that the graph's slope is considerably lower for IPPE than for EPnP and RPnP. This is because IPPE is time-bounded by the cost of computing the homography, which itself is very fast even for large n. At $n = 500$, IPPE is only about 1.5 times slower than at $n = 6$. By contrast EPnP and RPnP are approximately 4.3 and 9.6 times slower at $n = 500$ than $n = 6$.

n	IPPE+HO	HDZh+DLT	RPP-SP	EPnP	RPnP
4	0.150	0.261	11.101	1.012	0.940
6	0.387	0.497	14.211	0.883	0.965
10	0.398	0.517	22.444	0.929	1.011
60	0.420	0.527	51.260	1.024	1.475
160	0.494	0.605	138.508	1.705	2.908
340	0.555	0.669	258.100	2.853	5.659
500	0.602	0.715	408.362	3.760	9.205
700	0.657	0.771	483.992	4.849	13.905

Table 12 Table showing computation time (in ms) for solving PPE with compared methods.

Fig. 7 Images taken from the 'Game cover' dataset. Images were captured with a Nikon D3100 DSLR with image resolution 2304×1536 pixels. We used SIFT [27] to compute putative feature matches with each image containing between 200-500 features.

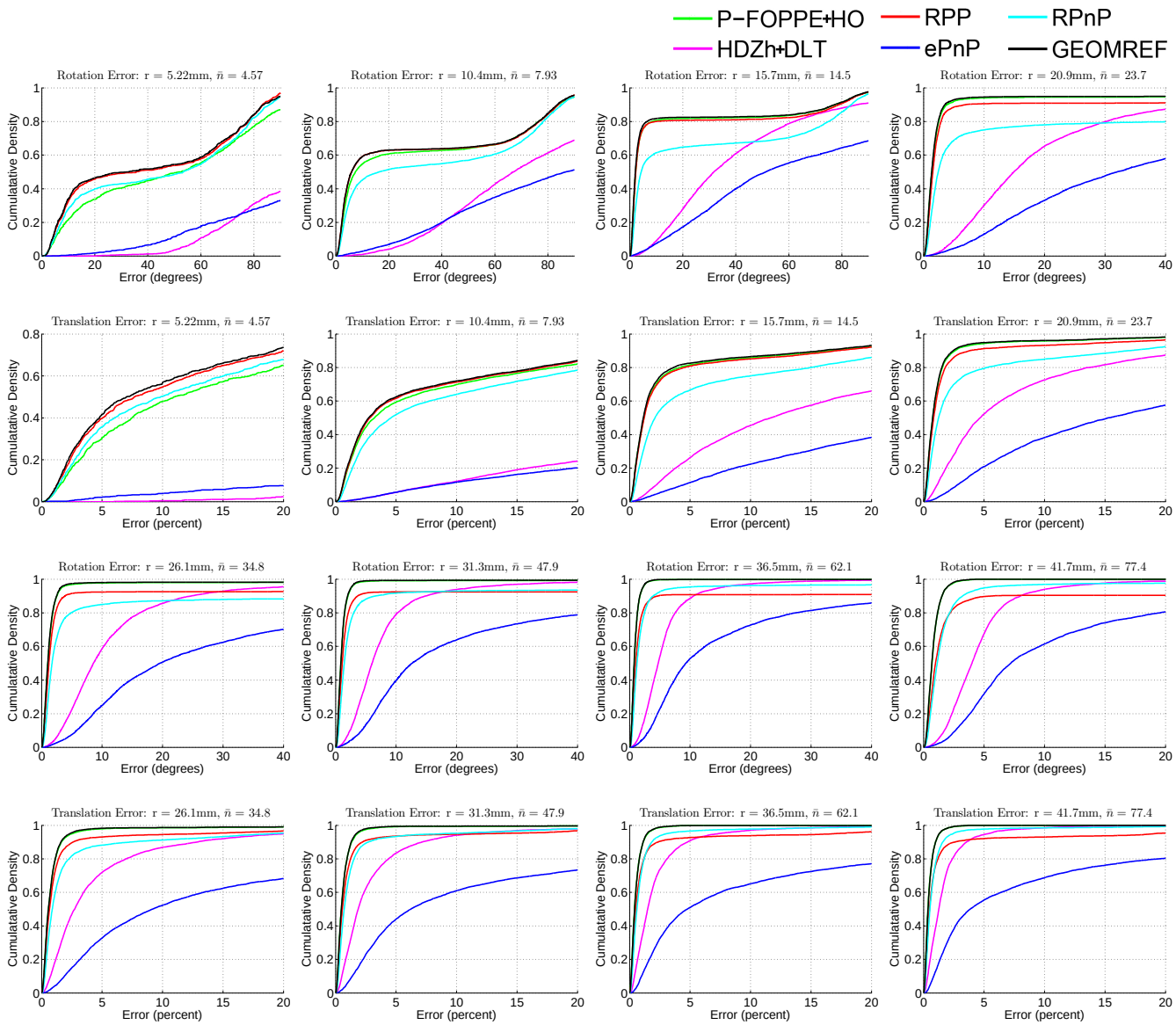

Fig. 8 Real experimental results (pose estimation using the 'Game cover' dataset): Comparing pose accuracy with varying window sizes.

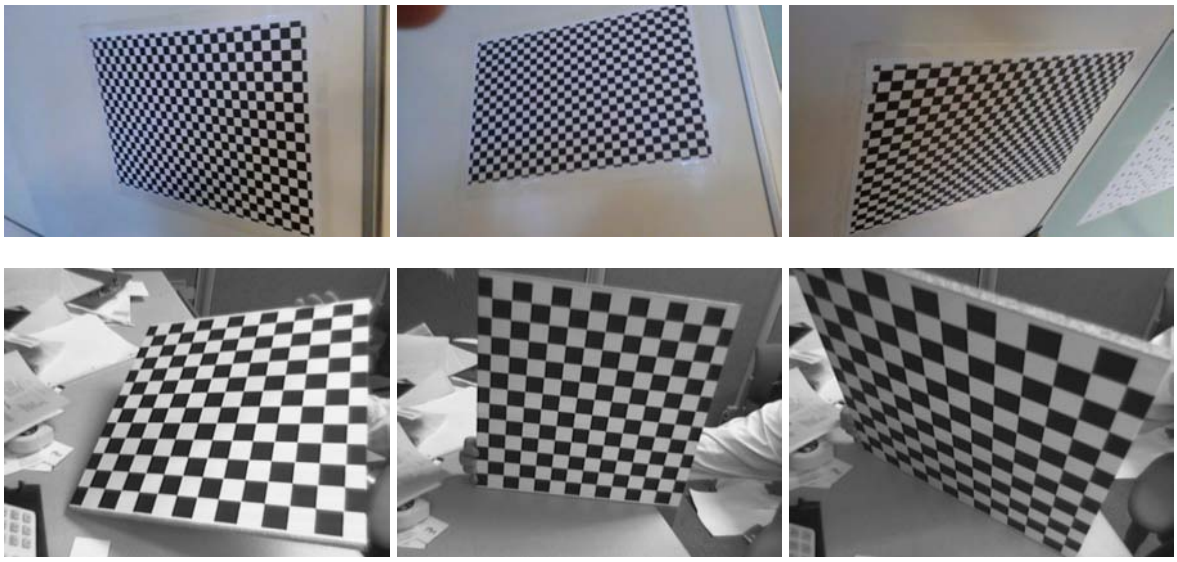

Fig. 9 Example views of two checkerboard test surfaces. Top row: views of a 193 × 276mm target captured by a 720p smartphone. Bottom row: views of a 360 × 360mm target from the public dataset supplied with the Matlab Calibration Toolbox. The performance of IPPE+HO and RPP-SP is virtually indistinguishable to GEOMREF.

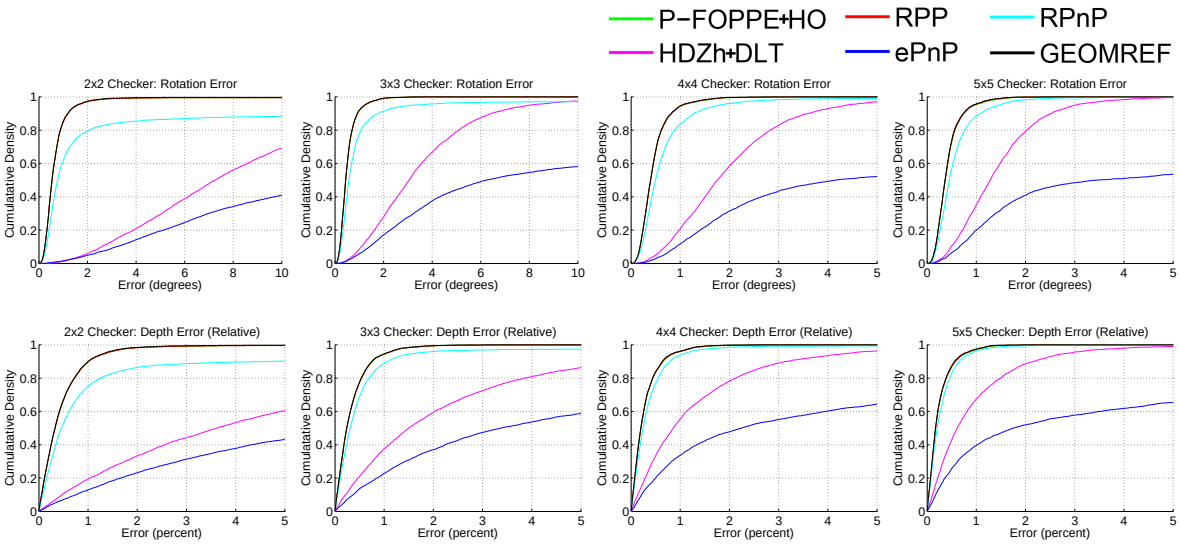

Fig. 10 Real experiments (checkerboard pose estimation captured with 720p smartphone): Comparing pose accuracy with varying checker sizes. The performance of IPPE+HO and RPP-SP is virtually indistinguishable to GEOMREF.

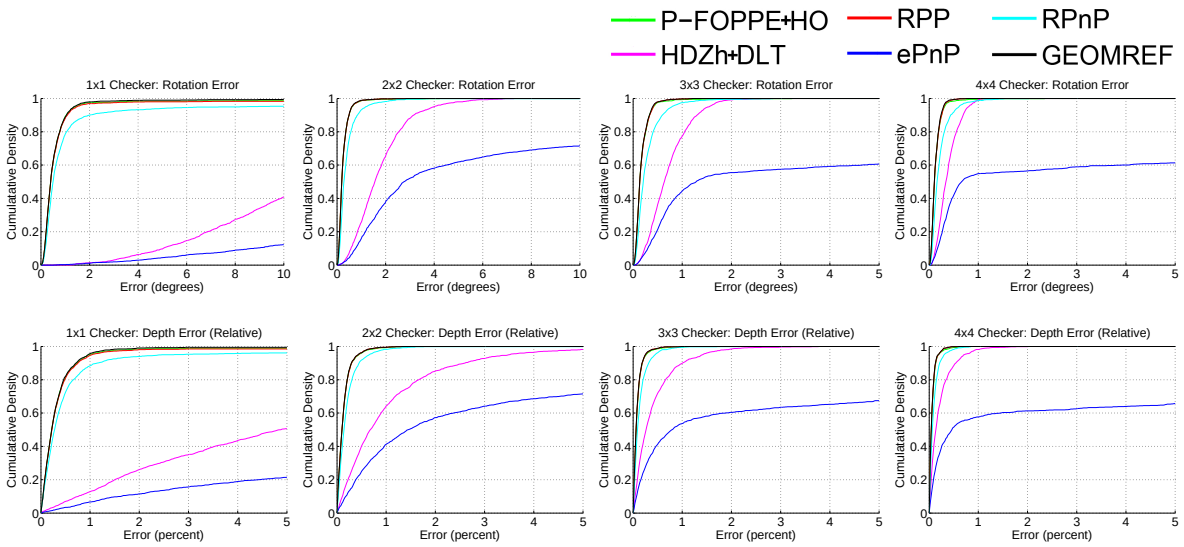

Fig. 11 Real experiments (checkerboard pose estimation with data from the Matlab Calibration Toolbox): Comparing pose accuracy with varying checker sizes.

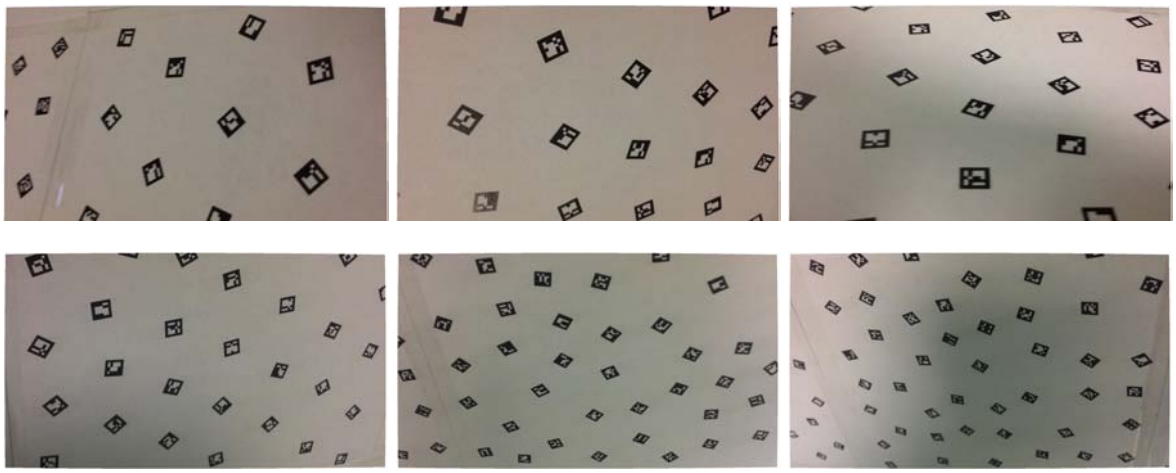

Fig. 12 Example views of AR markers captured by a 720p smartphone. Top row: close-range views. Bottom row: medium-range views.

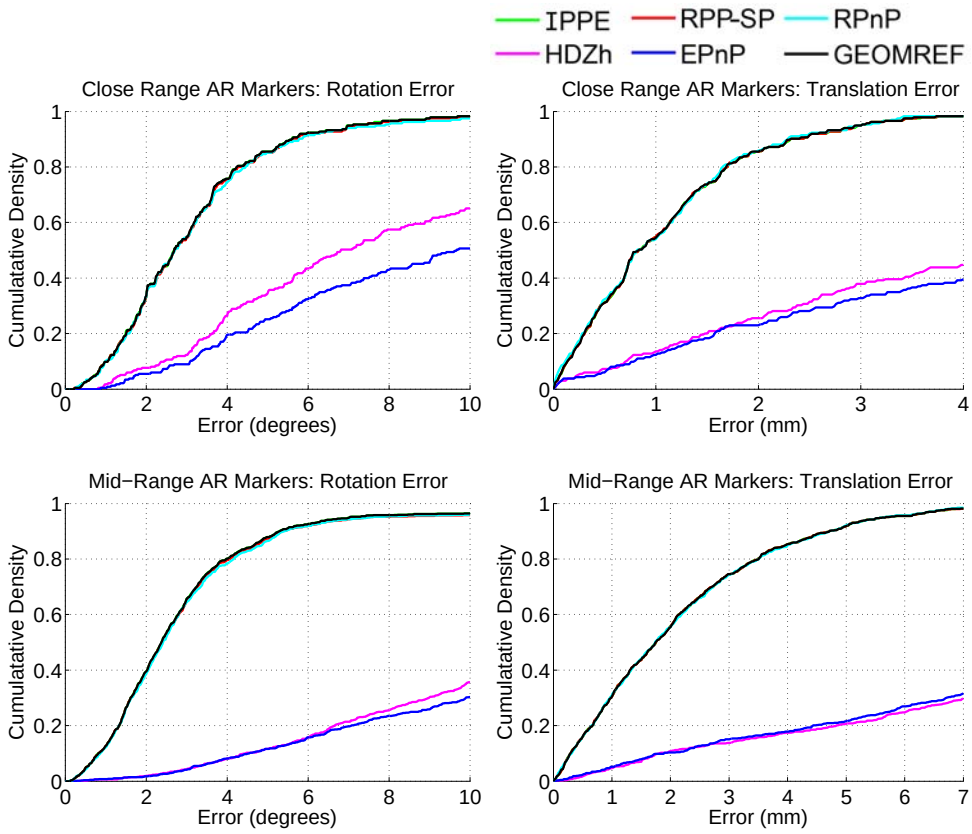

Fig. 13 Real experiments (AR marker pose estimation). Results are divided into close-range (left column) and mid-range (right column) conditions. Because each marker has four point correspondences at its four courners, the homography is computed exactly without requiring HO or DLT methods. There is very little to distinguish IPPE, RPP-SP and RPnP in terms of accuracy, and all perform very similarly to GEOMREF. This indicates that for this application there is no real benefit in refining their pose estimates with maximum likelihood refinement. However IPPE is by far the fastest and simplest of these three methods (see Table 12 with $n = 4$).

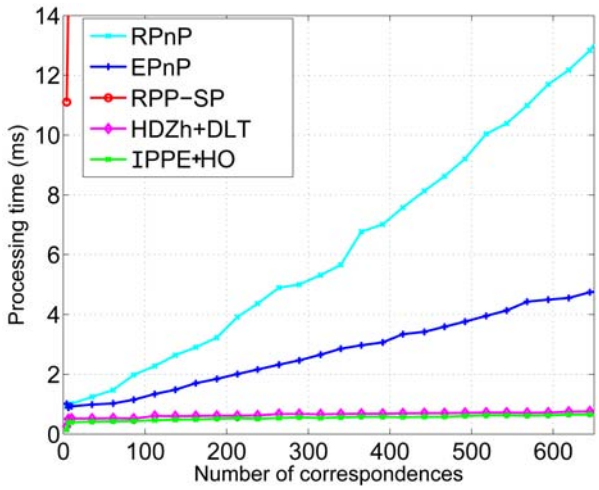

Fig. 14 Graph showing computation time (in ms) for solving PPE with compared methods. Benchmarking was performed on a standard Intel i7-3820 desktop PC. We use Matlab implementations provided by the authors for HDZh+DLT, RPP-SP, EPnP and RPnP. We use our own Matlab implementation for IPPE. The implementation of HO is provided by the authors.

6 Conclusion

We have presented the Infinitesimal Plane-based Pose Estimation (IPPE) algorithm. The core idea behind IPPE is to use the fact that a noisy homography will predict the transform w between the model plane and the image better at some locations than others. Our premise is that a good way to exploit the redundancy in the homography is to locate the point where the transform w is best estimated, and then solve pose exactly using local 1^{st}-order information of w at that point. We have presented the statistical justification for this approach. When the homography is estimated by noisy point correspondences we have shown using error propagation that estimates of w *and* J_w is made with highest certainty at the centroid of the model points. An equivalent way to say this is that the centroid is the point where a small perturbation in the correspondences will induce the smallest change in w and J_w.

We have then shown that given an estimate of w and J_w at a particular point $\mathbf{u}_0$, we can solve pose with a non-redundant 1^{st}-order PDE. This PDE is exact and does not make any 1^{st}-order approximations of the projection process. The solution to IPPE has some attractive properties. These include guarantees on the number of physical solutions (this is at most two, but never fewer than one), the fact that it never introduces artificial degeneracies, and allows a clear understanding of how these solutions relate geometrically. Unlike perspective homography decomposition, IPPE handles perspective and affine homographies transparently and does not break down when the amount of perspective distortion is small. Unlike affine homography decomposition, IPPE does not introduce any modelling error by approximating perspective projection with a linear transform.

We have performed a thorough empirical evaluation of IPPE and have shown that it performs very well in practice. It substantially outperforms homography decomposition and in most cases outperforms modern PnP methods (whilst being substantially faster). When the point correspondences come from AR markers, camera calibration targets or a large number of 2D keypoints such as SIFT, there really is no good reason to use another method over IPPE.

There is also a deep connection between IPPE and the P3P problem. This is that the solutions to P3P will tend to the solutions to IPPE if we create three virtual correspondences with infinitesimal separation centred at $\mathbf{u}_0$, and use the homography to estimate their positions in the image. One might then ask is there a better strategy than IPPE for positioning these virtual correspondences? Using error propagation analysis the answer appears to be no, because as the three points tend away from the centroid the uncertainty in their positions predicted by the homography increases quadratically. This has been confirmed empirically in our experiments.

In the future we aim to apply IPPE to related problems that are currently solved with classic homography decomposition, including plane-based pose estimation with intrinsic calibration and plane-based Structure-from-Motion. In terms of the broader picture, IPPE is a solution to a problem that involves estimating a transform using a redundant set of constraints that have error-in-variables. The redundancy is exploited by finding the point in the transform's domain with the least error-in-variables via uncertainty propagation, and then solving the transform using an exact (*i.e.* non-redundant) local system at that point. We hope that this strategy may be of use in other vision problems for estimating transforms when there exists smooth variation in the error-in-variables.

Appendices

A IPPE using the Para-perspective and Weak-perspective Cameras

Para-perspective projection approximates perspective projection by linearising π about some 3D point $\mathbf{x}_c =$

$[x_c, y_c, z_c]^\top$ in camera coordinates. We denote this by $\pi_{pp}(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^2$. To reduce approximation error $\mathbf{x}_c$ is chosen to be the centroid of the model's points [31,32]. $\mathbf{x}_c$ can be parameterised by a 2D point $\tilde{\mathbf{q}}_c$ in normalised coordinates, scaled by a depth z_c: $\mathbf{x}_c = z_c[\tilde{\mathbf{q}}_c^\top, 1]^\top$. π_{pp} is then given by:

$$\pi_{pp}(\mathbf{x}) = \tilde{\mathbf{q}}_c + z_c^{-1} \left[\mathbf{I}_2 \mid -\tilde{\mathbf{q}}_c \right] \mathbf{x} \tag{39}$$

Because para-perspective projection is an affine transform, $\hat{\mathbf{H}}$ is also an affine transform, and computed by the best fitting affine transform that maps $\{\mathbf{u}_i\}$ to $\{\tilde{\mathbf{q}}_i\}$. The Jacobian of the model-to-image transform w is therefore constant, which we denote by $\mathbf{J}_a \in \mathbb{R}^{2\times 2}$. We can then estimate z_c (*i.e.* the depth of the centroid of the correspondences in camera coordinates) and estimate the plane's rotation using IPPE by replacing π with π_{pp}. This leads to an instance of Problem (16) with substitutions $\mathbf{J} \leftarrow \mathbf{J}_a$, $\mathbf{v} \leftarrow \tilde{\mathbf{q}}_c$ and $\gamma \leftarrow z_c^{-1}$.

The weak-perspective camera can be treated similarly to the para-perspective camera. The difference is that in weak-perspective projection the linearisation is done at a 3D point passing through the camera's optical axis. The weak-perspective projection function $\pi_{wp}(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^2$ is given by:

$$\pi_{wp}(\mathbf{x}) = \tilde{\mathbf{q}}_c + z_0^{-1} \left[\mathbf{I}_2 \mid \mathbf{0} \right] \mathbf{x} \tag{40}$$

where z_0 approximates the depth of the plane along the camera's optical axis. We can estimate z_0 and estimate the plane's rotation using IPPE by replacing π with π_{wp}. This leads to an instance of Problem (16) with substitutions $\mathbf{J} \leftarrow \mathbf{J}_a$, $\mathbf{v} \leftarrow \mathbf{0}$ and $\gamma \leftarrow z_0^{-1}$.

B Proof of Eq. (17)

We prove Eq. (17) using a general form with point correspondences in d-dimensional space. $\mathbf{U} \in \mathbb{R}^{d\times n}$ denotes the set of points in the domain space, where n is the number of points. $\mathbf{Q} \in \mathbb{R}^{d\times n}$ denotes the corresponding set of points in the target space (of the same dimensionality d). We use $\bar{\mathbf{U}}$ to denote $\mathbf{U}$ but zero-meaned (so that the sum of the rows of $\bar{\mathbf{U}}$ are zero). Let $\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{A}} & \hat{\mathbf{t}} \\ \mathbf{0}^\top & 1 \end{bmatrix}$ denote the maximum likelihood homogeneous affine transform that maps $\bar{\mathbf{U}}$ to $\mathbf{Q}$, with $\hat{\mathbf{A}} \in \mathbb{R}^{d\times d}, \hat{\mathbf{t}} \in \mathbb{R}^d$. $\hat{\mathbf{M}}$ is given by:

$$\begin{aligned} &\hat{\mathbf{t}} = \mathbf{Q}\mathbf{1} \\ &\hat{\mathbf{A}} = (\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{q}, \ \mathbf{B} \stackrel{\text{def}}{=} \mathbf{I}_d \otimes \bar{\mathbf{U}}^\top \end{aligned} \tag{41}$$

where $\mathbf{1}$ is the all-ones $n \times 1$ vector and $\mathbf{q} \in \mathbb{R}^{dn\times 1}$ denotes $\mathbf{Q}$ stacked into a column vector. The transformation of a point $\mathbf{u} \in \mathbb{R}^d$ in the domain according to $\hat{\mathbf{M}}$ is given by: $f(\mathbf{u}) = \mathbf{V}\hat{\mathbf{M}}$, where $\mathbf{V} \stackrel{\text{def}}{=} \mathbf{I}_d \otimes \mathbf{u}^\top$. Suppose $\mathbf{Q}$ is corrupted by IID zero-mean Gaussian noise with variance σ. The uncertainty covariance matrix in $\mathbf{q}$ is $\mathbf{\Sigma}_\mathbf{q} = \sigma\mathbf{I}_2$ and using propagation of uncertainty, the uncertainty in the position of $\mathbf{u}$ transformed according to $\hat{\mathbf{M}}$ is given by the $n \times n$ covariance matrix $\mathbf{\Sigma}_{f(\mathbf{u})}$:

$$\begin{aligned} &\mathbf{\Sigma}_{f(\mathbf{u})} = \mathbf{\Sigma}_{\hat{\mathbf{t}}} + \mathbf{\Sigma}_{\hat{\mathbf{A}}} && (a) \\ &\mathbf{\Sigma}_{\hat{\mathbf{t}}} = \tfrac{\sigma}{n}\mathbf{I}_n && (b) \\ &\mathbf{\Sigma}_{\hat{\mathbf{A}}} = \sigma\mathbf{V}^\top\hat{\mathbf{M}}\hat{\mathbf{M}}^\top\mathbf{V} = \mathbf{V}^\top(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{V} && (c) \\ &\Leftrightarrow \left[\mathbf{\Sigma}_{\hat{\mathbf{A}}}\right]_{ij} = \begin{cases} \mathbf{u}^\top(\bar{\mathbf{U}}^\top\bar{\mathbf{U}})^{-1}\mathbf{u} & i = j \\ 0 & i \neq j \end{cases} && (d) \end{aligned} \tag{42}$$

The step from Eq. (42-c) to Eq. (42-d) is made because of the block-diagonal structure of $(\mathbf{B}^\top\mathbf{B})^{-1}$.

C Proof of Theorem 2

Proof of Lemma 1. Lemma 1 comes directly from Eq. (19). To first order we have:

$$\arg\min_{\mathbf{u}_0} \operatorname{trace}\left(\mathbf{\Sigma}_\mathbf{J}(\mathbf{u}_0)\right) = \arg\min_{\mathbf{u}_0} \left\| \frac{\partial}{\partial\hat{\mathbf{q}}} \operatorname{vec}(\mathbf{J}) \right\|_F^2 \tag{43}$$

Eq. (43) tells us that to minimise the uncertainty in $\mathbf{J}$ we should find $\mathbf{u}_0$ where a small change in the correspondences in the image changes $\mathbf{J}$ the least. □

Proof of Lemma 2. Let $\mathbf{J}'$ denote the Jacobian of $\mathbf{H}'$, and $\tilde{\mathbf{q}}'_i = s_q\tilde{\mathbf{q}}_i + \mathbf{t}_q$ for some $s_q \in \mathbb{R}^+$ and $\mathbf{t}_q \in \mathbb{R}^2$. Recall the centroid of $\{\mathbf{u}_i\}$ is already at the origin, and so $\mathbf{u}'_i = s_u\mathbf{u}_i$ for some $s_u \in \mathbb{R}^+$. We use $\hat{\mathbf{q}}'$ to be the vector of length $2n$ that holds $\{\tilde{\mathbf{q}}'_i\}$. Using the product rule we have $\frac{\partial \operatorname{vec}(\mathbf{J})}{\partial \hat{\mathbf{q}}'} = s_q \frac{\partial \operatorname{vec}(\mathbf{J})}{\partial \hat{\mathbf{q}}}$. Because $s_q \in \mathbb{R}^+$ we have:

$$\begin{aligned} \arg\min_{\mathbf{u}_0} \operatorname{trace}\left(\mathbf{\Sigma}_\mathbf{J}(\mathbf{u}_0)\right) &= \arg\min_{\mathbf{u}_0} \left\| \tfrac{\partial}{\partial\hat{\mathbf{q}}} \operatorname{vec}(\mathbf{J}) \right\|_F^2 \\ &= \arg\min_{\mathbf{u}_0} \left\| \tfrac{\partial}{\partial\hat{\mathbf{q}}'} \operatorname{vec}(\mathbf{J}) \right\|_F^2 \end{aligned} \tag{44}$$

Normalising $\{\tilde{\mathbf{q}}_i\}$ therefore does not affect the solution. We then make the coordinate transform $\mathbf{u} \leftarrow s_u\mathbf{u}$, and solve Problem (44) using $\{\mathbf{u}'_i\}$ in place of $\{\mathbf{u}_i\}$ and $\mathbf{J}'$ in place of $\mathbf{J}$. Suppose a solution to this is given by $\hat{\mathbf{u}}'_0$. By undoing the coordinate transform, a solution to the original problem is given by $s_u^{-1}\hat{\mathbf{u}}'_0$.

When the perspective terms of $\mathbf{H}'$ (H'_{31} and H'_{32}) are small a good approximation to $\mathbf{J}'$ can be made by linearising with respect to H'_{31} and H'_{32} about $H'_{31} = H'_{32} = 0$. This linearisation gives:

$$w(\mathbf{u}_0) \approx \begin{bmatrix} -H'_{31}H'_{11}u_x^2 + (-H'_{13}H'_{12} - H'_{32}H'_{11})u_xu_y \\ -H'_{31}H'_{21}u_x^2 + (-H'_{13}H_{22} - H'_{32}H'_{21})u_xu_y \\ H'_{11}u_x - H'_{32}H'_{12}u_y^2 + H'_{12}u_y \\ H'_{21}u_x - H'_{32}H'_{22}u_y^2 + H'_{22}u_y \end{bmatrix} + \tag{a}$$

$$\operatorname{vec}(\mathbf{J}') = \operatorname{vec}\left(\tfrac{\partial w}{\partial \mathbf{u}}(\mathbf{u}_0)\right) \approx \begin{bmatrix} H'_{11} - H'_{31}(2H'_{11}u_x + H'_{12}u_y) - H'_{32}H'_{11}u_x \\ H'_{21} - H'_{31}(2H'_{21}u_x + H'_{22}u_y) - H'_{32}H'_{21}u_x \\ H'_{12} - H'_{32}(H'_{11}u_x + 2H'_{12}u_y) - H'_{31}H'_{12}u_x \\ H'_{22} - H'_{32}(H'_{21}u_x + 2H'_{22}u_y) - H'_{31}H'_{22}u_x \end{bmatrix} \tag{b}$$

(45)

The approximation of $\text{vec}(\mathbf{J}')$ in Eq. (45-b) is linear in $\mathbf{u}_0$, and so $\frac{\partial}{\partial \hat{\mathbf{q}}'}\text{vec}(\mathbf{J}')$ is also linear in $\mathbf{u}_0$. This means $\left\|\frac{\partial}{\partial \hat{\mathbf{q}}'}\text{vec}(\mathbf{J}')\right\|_F^2$ is of the form:

$$\left\|\frac{\partial}{\partial \hat{\mathbf{q}}'}\text{vec}(\mathbf{J}')\right\|_F^2 \approx \mathbf{u}_0^\top \mathbf{Q}\mathbf{u}_0 + \mathbf{b}^\top \mathbf{u}_0 + c \tag{46}$$

for some 2×2 matrix $\mathbf{Q}$ (which is either positive definite or positive semi-definite), a 2×1 vector $\mathbf{b}$ and a constant scalar c. □

Using the product rule we have:

$$\begin{aligned}&\left\|\tfrac{\partial}{\partial \hat{\mathbf{q}}'}\text{vec}(\mathbf{J}')\right\|_F^2 = \left\|\tfrac{\partial}{\partial \mathbf{h}'}\text{vec}(\mathbf{J}')\tfrac{\partial \mathbf{h}'}{\partial \hat{\mathbf{q}}'}\right\|_F^2 \\ &= \text{trace}\left(\tfrac{\partial}{\partial \mathbf{h}'}\text{vec}(\mathbf{J}')\mathbf{C}\tfrac{\partial}{\partial \mathbf{h}'}\text{vec}(\mathbf{J}')^\top\right) \\ &\mathbf{h}' \stackrel{\text{def}}{=} \text{vec}(\mathbf{H}'), \quad \mathbf{C} \stackrel{\text{def}}{=} \tfrac{\partial}{\partial \hat{\mathbf{q}}'}\mathbf{h}'\tfrac{\partial}{\partial \hat{\mathbf{q}}'}\mathbf{h}'^\top,\ \mathbf{C} \succ \mathbf{0},\end{aligned} \tag{47}$$

$\mathbf{C}$ is a 8×8 positive definite matrix that has been studied in [5]. When $\mathbf{H}'$ is approximately affine the perspective terms H'_{31} and H'_{32} and the translational terms H'_{13} and H'_{23} are negligible. When $H'_{31} = H'_{32} = H'_{13} = H'_{23} = 0$, $\frac{\partial}{\partial \mathbf{h}'}\text{vec}(\mathbf{J}')$ is given by:

$$\begin{bmatrix} 1&0&0&0&0&0 & -2H'_{11}u_x - H'_{12}u_y & -H'_{11}u_y & 0 \\ 0&0&0&1&0&0 & -2H'_{21}u_x - H'_{22}u_y & -H'_{21}u_y & 0 \\ 0&1&0&0&0&0 & -H'_{12}u_x & -2H'_{12}u_y - H'_{11}u_x & 0 \\ 0&0&0&0&1&0 & -H'_{22}u_x & -2H'_{22}u_y - H'_{21}u_x & 0 \end{bmatrix} \tag{48}$$

It was shown that the normalisation step orthogonalises $\frac{\partial}{\partial \hat{\mathbf{q}}'}\mathbf{h}'$ [5]. This implies $\mathbf{C}$ is approximately a diagonal matrix and so:

$$\begin{aligned}&\left\|\tfrac{\partial}{\partial \hat{\mathbf{q}}'}\text{vec}(\mathbf{J}')\right\|_F^2 = \text{trace}\left(\tfrac{\partial}{\partial \hat{\mathbf{q}}}\text{vec}(\mathbf{J}')\tfrac{\partial}{\partial \hat{\mathbf{q}}}\text{vec}(\mathbf{J}')^\top\right) \\ &= \text{trace}\left(\tfrac{\partial}{\partial \mathbf{h}'}\text{vec}(\mathbf{J}')\mathbf{C}\tfrac{\partial}{\partial \mathbf{h}'}\text{vec}(\mathbf{J}')^\top\right) \\ &\approx \textstyle\sum_{ij}\left[\tfrac{\partial}{\partial \mathbf{h}'}\text{vec}(\mathbf{J}')\right]_{ij}^2 \mathbf{C}_{jj}\end{aligned} \tag{49}$$

This is a weighted sum of the (squared) elements of $\frac{\partial}{\partial \mathbf{h}'}\text{vec}(\mathbf{J}')$. The weights are $\mathbf{C}_{jj}$ which are non-negative because $\mathbf{C}$ is positive definite. Therefore when the perspective terms of $\mathbf{H}'$ are negligible $\text{trace}\left(\boldsymbol{\Sigma}_{\mathbf{J}'}(\mathbf{u}_0)\right)$ is minimised by $\mathbf{u}_0 = \mathbf{0}$, and so $\text{trace}\left(\boldsymbol{\Sigma}_{\mathbf{J}}(\mathbf{u}_0)\right)$ minimised by $\mathbf{u}_0 = s_u^{-1}\mathbf{0} = \mathbf{0}$ (*i.e.* the centroid of $\{\mathbf{u}_i\}$). □

D Proof of Lemma 3

For simplicity we centre the model's coordinate frame at $\mathbf{u}_0$, so $\mathbf{u}_i \leftarrow (\mathbf{u}_i - \mathbf{u}_0)$ and $\mathbf{u}_0 \leftarrow \mathbf{0}$. Because $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ are non-colinear at least two members of $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ cannot be $\mathbf{0}$. Without loss of generality let these be $\mathbf{u}_1$ and $\mathbf{u}_2$.

Let $\mathbf{v}_i \stackrel{\text{def}}{=} w(\mathbf{u}_i)$, $i \in \{1,2,3\}$ be the position of the three points in the image (in normalised coordinates). From Eq. (28) the two embeddings of $\mathbf{u}_i$ into camera coordinates are:

$$\begin{aligned}s_1(\mathbf{u}_i) &= \mathbf{R}_1\begin{bmatrix}\mathbf{u}_i \\ 0\end{bmatrix} + \gamma^{-1}\begin{bmatrix}\mathbf{v}_0 \\ 1\end{bmatrix} \\ s_2(\mathbf{u}_i) &= \mathbf{R}_2\begin{bmatrix}\mathbf{u}_i \\ 0\end{bmatrix} + \gamma^{-1}\begin{bmatrix}\mathbf{v}_0 \\ 1\end{bmatrix}\end{aligned} \tag{50}$$

If $s_1(\mathbf{u}_i)$ and $s_2(\mathbf{u}_i)$ project $\mathbf{u}_i$ to the same image point (*i.e.* they exist along the same line-of-sight) then pose cannot be disambiguated using the reprojection error of $\mathbf{u}_i$. This is true for $\mathbf{u}_0$ because $\mathbf{u}_0 = \mathbf{0} \Rightarrow s_1(\mathbf{u}_0) = s_2(\mathbf{u}_0) = \gamma^{-1}[\mathbf{v}_0^\top 1]^\top$. For $\mathbf{u}_i, i \neq 0$, we cannot disambiguate pose using reprojection error iff:

$$\begin{aligned}&\forall i \in \{1,2,3\}\ \exists s_i \in \mathbb{R}^+ \text{ s.t.} \\ &\mathbf{R}_1\begin{bmatrix}\mathbf{u}_i \\ 0\end{bmatrix} + \gamma^{-1}\begin{bmatrix}\mathbf{v}_0 \\ 1\end{bmatrix} = s_i\left(\mathbf{R}_2\begin{bmatrix}\mathbf{u}_i \\ 0\end{bmatrix} + \gamma^{-1}\begin{bmatrix}\mathbf{v}_0 \\ 1\end{bmatrix}\right)\end{aligned} \tag{51}$$

Using the decompositions of $\mathbf{R}_1$ and $\mathbf{R}_2$ from Eq. (24) we pre-multiply both sides of Eq. (51) by $\mathbf{R}_v^\top$ to give:

$$\begin{aligned}&\forall i \in \{1,2,3\}\ \exists s_i \in \mathbb{R}^+ \text{ s.t.} \\ &\begin{bmatrix}\gamma^{-1}\mathbf{A} \\ +\mathbf{b}^\top\end{bmatrix}\mathbf{u}_i + \tilde{\mathbf{t}} = s_i\left(\begin{bmatrix}\gamma^{-1}\mathbf{A} \\ -\mathbf{b}^\top\end{bmatrix}\mathbf{u}_i + \tilde{\mathbf{t}}\right) \\ &\tilde{\mathbf{t}} \stackrel{\text{def}}{=} \gamma^{-1}\mathbf{R}_v^\top\begin{bmatrix}\mathbf{v}_0 \\ 1\end{bmatrix}\end{aligned} \tag{52}$$

We split Eq. (52) into three cases. The first case is when $\mathbf{b} = \mathbf{0}$. In this case there is no ambiguity because from Eq.(24) $\mathbf{b} = \mathbf{0} \Leftrightarrow \tilde{\mathbf{R}}_1 = \tilde{\mathbf{R}}_2 \Leftrightarrow \mathbf{R}_1 = \mathbf{R}_2$. The second case is when $\mathbf{b} \neq \mathbf{0}$ and the top two rows of the left side of Eq. (52) are non-zero: $\gamma^{-1}\mathbf{A}\mathbf{u}_i + \tilde{\mathbf{t}}_{12} \neq \mathbf{0}$. This implies $\sigma_i = 1$. The third row of Eq. (52) then implies $\mathbf{b}^\top\mathbf{u}_i = -\mathbf{b}^\top\mathbf{u}_i$. Because $\mathbf{b} \neq \mathbf{0}$ and $\mathbf{u}_i \neq \mathbf{0}$ for $i \in \{1,2\}$, $\mathbf{b}$ must be orthogonal to $\mathbf{u}_1$ and $\mathbf{u}_2$. This implies $\mathbf{u}_1$ and $\mathbf{u}_2$ are colinear, which is a contradiction.

The third case is when $\mathbf{b} \neq \mathbf{0}$ and the top two rows of the left side of Eq. (52) are zero: $\gamma^{-1}\mathbf{A}\mathbf{u}_i + \tilde{\mathbf{t}}_{12} = \mathbf{0}$. By eliminating $\tilde{\mathbf{t}}_{12}$ and cancelling γ this implies $\mathbf{A}(\mathbf{u}_2 - \mathbf{u}_1) = \mathbf{0}$ and $\mathbf{A}(\mathbf{u}_3 - \mathbf{u}_1) = \mathbf{0}$. Because $\mathbf{u}_2 \neq \mathbf{u}_1$, this implies $\mathbf{A}$ has a nullspace. Because $\text{rank}(\mathbf{A}) \geq 1$, this implies $\text{rank}(\mathbf{A}) = 1$, and so $(\mathbf{u}_2 - \mathbf{u}_1) = \lambda(\mathbf{u}_3 - \mathbf{u}_1)$ for some $\lambda \neq 0$. This implies $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ are colinear, which is a contradiction.

To summarise, when $\mathbf{b} = \mathbf{0}$ there is no ambiguity because both solutions to pose are the same, and when $\mathbf{b} \neq \mathbf{0}$ Eq. (52) is false, and hence Eq. (51) is false. Therefore when $\mathbf{b} \neq \mathbf{0}$ Eq. (28) will project either $\mathbf{u}_1$, $\mathbf{u}_2$ or $\mathbf{u}_3$ to two different image points. □

References

1. Ansar, A., Daniilidis, K.: Linear pose estimation from points or lines. Pattern Analysis and Machine Intelligence (PAMI) **25**, 282–296 (2003)
2. Barreto, J., Roquette, J., Sturm, P., Fonseca, F.: Automatic Camera Calibration Applied to Medical Endoscopy. In: British Machine Vision Conference (BMVC) (2009)
3. Bouguet, J.Y.: A camera calibration toolbox for matlab. URL http://www.vision.caltech.edu/bouguetj/calib_doc/
4. Brown, M., Majumder, A., Yang, R.: Camera-based calibration techniques for seamless multiprojector displays. Visualization and Computer Graphics pp. 193–206 (2005)
5. Chen, P., Suter, D.: Error analysis in homography estimation by first order approximation tools: A general technique. Journal of Mathematical Imaging and Vision **33**, 281–295 (2009)
6. Collins, T., Durou, J.D., Gurdjos, P., Bartoli, A.: Single-view perspective shape-from-texture with focal length estimation: A piecewise affine approach. In: 3D Data Processing Visualization and Transmission (3DPVT10 (2010)
7. Dhome, M., Richetin, M., Lapreste, J.T.: Determination of the attitude of 3D objects from a single perspective view. Pattern Analysis and Machine Intelligence (PAMI) **11**, 1265–1278 (1989)
8. Faugeras, O., Luong, Q.T., Papadopoulou, T.: The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications. MIT Press, Cambridge, MA, USA (2001)
9. Fiore, P.D.: Efficient linear solution of exterior orientation. Pattern Analysis and Machine Intelligence (PAMI) **23**, 140–148 (2001)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**, 381–395 (1981)
11. Gao, X.S., Hou, X., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. Pattern Analysis and Machine Intelligence (PAMI) **25**, 930–943 (2003)
12. Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. Pattern Analysis and Machine Intelligence (PAMI) **25**(8), 930–943 (2003)
13. Geiger, A., Moosmann, F., Car, m., Schuster, B.: A toolbox for automatic calibration of range and camera sensors using a single shot. In: International Conference on Robotics and Automation (ICRA) (2012)
14. Haralick, R.M., Lee, C.N., Ottenberg, K., Nölle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. International Journal of Computer Vision (IJCV) **13**, 331–356 (1994)
15. Haralick, R.M., Lee, D., Ottenburg, K., Nölle, M.: Analysis and solutions of the three point perspective pose estimation problem. In: Computer Vision and Pattern Recognition (CVPR) (1991)
16. Harker, M., O'Leary, P.: Computation of homographies. In: British Computer Vision Conference (BMVC) (2005)
17. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)
18. Hesch, J.A., Roumeliotis, S.I.: A direct least-squares (DLS) method for PnP. In: International Conference on Computer Vision (ICCV) (2011)
19. Hilsmann, A., Schneider, D., Eisert, P.: Template-free shape from texture with perspective cameras. In: British Machine Vision Conference (BMVC) (2011)
20. Horaud, R., Dornaika, F., Lamiroy, B., Christy, S.: Object Pose: The Link between Weak Perspective, Paraperspective and Full Perspective. International Journal of Computer Vision (IJCV) **22**, 173–189 (1997)
21. Hung, Y., Harwood, D., Yeh, P.e.n..S.h.u.: Passive ranging to known planar point sets. Tech. rep., University of Maryland (College Park, MD US) (1984)
22. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: International Workshop on Augmented Reality (IWAR) (1999)
23. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An accurate O(n) solution to the pnp problem. International Journal of Computer Vision (IJCV) **81**, 155–166 (2009)
24. Li, S., Xu, C., Xie, M.: A robust O(n) solution to the perspective-n-point problem. Pattern Analysis and Machine Intelligence (PAMI) (2012)
25. Lobay, A., Forsyth, D.A.: Recovering shape and irradiance maps from rich dense texton fields. In: Computer Vision and Pattern Recognition (CVPR) (2004)
26. Lobay, A., Forsyth, D.A.: Shape from texture without boundaries. International Journal of Computer Vision (IJCV) **67**, 71–91 (2006)
27. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) **60**, 91–110 (2004)
28. Lu, C.P., Hager, G.D., Mjolsness, E.: Fast and globally convergent pose estimation from video images. Pattern Analysis and Machine Intelligence (PAMI) **22**, 610622 (2000)
29. Munoz-Salinas, R.: ArUco: Augmented reality library from the university of cordoba. URL http://www.uco.es/investiga/grupos/ava/node/26
30. Oberkampf, D., DeMenthon, D., Davis, L.S.: Iterative pose estimation using coplanar feature points. Computer Vision and Image Understanding (CVIU) **63**, 495–511 (1996)
31. ichi Ohta, Y., Maenobu, K., Sakai, T.: Obtaining surface orientation from texels under perspective projection. In: International Joint Conferences on Artificial Intelligence (IJCAI) (1981)
32. Poelman, C., Kanade, T.: A paraperspective factorization method for shape and motion recovery. Tech. rep. (1993)
33. Quan, L., Lan, Z.: Linear n-point camera pose determination. Pattern Analysis and Machine Intelligence (PAMI) (1999)
34. Schweighofer, G., Pinz, A.: Robust pose estimation from a planar target. Pattern Analysis and Machine Intelligence (PAMI) **28**, 2024–2030 (2006)
35. Sturm, P.: Algorithms for plane-based pose estimation. In: Computer Vision and Pattern Recognition (CVPR) (2000)
36. Taubin, G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. Pattern Analysis and Machine Intelligence (PAMI) **13**, 1115–1138 (1991)
37. Triggs, B.: Camera pose and calibration from 4 or 5 known 3D points. In: International Conference on Computer Vision (ICCV) (1999)
38. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. URL http://www.vlfeat.org/

39. Zhang, C.X., Hu, Z.Y.: A general sufficient condition of four positive solutions of the p3p problem. Journal of Computer Science and Technology **20**, 836–842 (2005)
40. Zhang, Z.: A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence (PAMI) **22**, 1330–1334 (2000)