Very Fast Solution to the PnP Problem with Algebraic Outlier Rejection

Luis Ferraz[1]   Xavier Binefa[1]   Francesc Moreno-Noguer[2]

[1]Department of Information and Communication Technologies, UPF, Barcelona, Spain
[2]Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain

`luis.ferraz@upf.edu, xavier.binefa@upf.edu, fmoreno@iri.upc.edu`

Abstract

We propose a real-time, robust to outliers and accurate solution to the Perspective-n-Point (PnP) problem. The main advantages of our solution are twofold: first, it integrates the outlier rejection within the pose estimation pipeline with a negligible computational overhead; and second, its scalability to arbitrarily large number of correspondences. Given a set of 3D-to-2D matches, we formulate pose estimation problem as a low-rank homogeneous system where the solution lies on its 1D null space. Outlier correspondences are those rows of the linear system which perturb the null space and are progressively detected by projecting them on an iteratively estimated solution of the null space. Since our outlier removal process is based on an algebraic criterion which does not require computing the full-pose and reprojecting back all 3D points on the image plane at each step, we achieve speed gains of more than $100\times$ compared to RANSAC strategies. An extensive experimental evaluation will show that our solution yields accurate results in situations with up to 50% of outliers, and can process more than 1000 correspondences in less than $5ms$.

1. Introduction

The goal of the Perspective-n-Point (PnP) problem is to estimate the position and orientation of a calibrated camera from known 3D-to-2D point correspondences between a 3D model and their image projections. The PnP is at the core of many Computer Vision tasks with applications in other areas such as Robotics, Photogrammetry or Augmented Reality. Despite being a topic studied for more than a century (the first solution dates back to 1841 [11]) the last decade has witnessed a wide body of literature in which the primary goal has been to build efficient and accurate solutions scalable to a large number of correspondences.
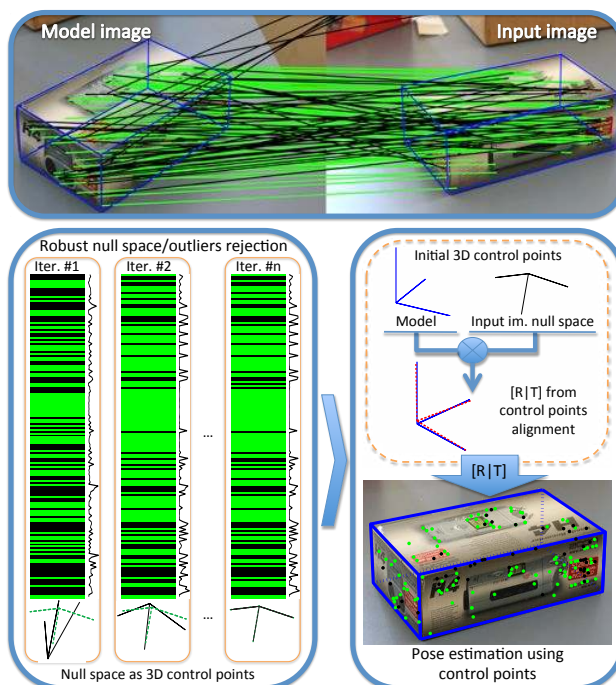
Figure 1. **Our approach to simultaneously estimate pose and reject outliers. Top:** Sample result, where green and black lines indicate the inlier and outlier matches, respectively. The 3D model is overlaid in blue in the input image, using the estimated pose. The overall process (outlier rejection and pose estimation) is done in less than 5 ms. **Bottom-left:** Our PnP solution iteratively computes the null space of a linear system. Here we plot the matrix of this system, where rows represent correspondences. The error per match of the estimated null-space is plotted on the side of each matrix, and is used to detect outliers (black rows). The convergence of the null space is plotted at the bottom of each matrix, where dotted lines depict the true null space. **Bottom-right:** The null space gives an estimate of the 3D model in camera coordinates. We finally compute the pose using rigid alignment.

Along these lines, the Efficient PnP (EPnP) [22] was the first closed-form solution to the problem with $O(n)$ complexity and little loss of accuracy with respect to the most accurate iterative methods [21]. The main contribution of

this work was to introduce a fixed number of virtual control points to represent whatever number of 3D points. The problem was then reduced to that of estimating the position of the control points, which was efficiently done using linearization techniques. Subsequent works dealing with an arbitrary number of points, have improved the accuracy of the EPnP specially for the minimal cases with $n = \{3, 4, 5\}$ correspondences. This has been essentially achieved by replacing the linearization approaches with more sophisticate polynomial solvers [19] along with new reformulations of the PnP as a least squares problem [13, 33, 32].

In any event, previous PnP solutions do not directly handle the fundamental problem of outliers, and they assume that all correspondences are at most corrupted by noise, but not by mismatches. Outliers are typically rejected at a preliminary stage using P3P algorithms [17] in combination with RANSAC-based schemes [8, 12] . Then, PnP methods are applied with all the remaining inlier matches. This initial rejection stage, though, significantly lessens the efficiency properties of the whole pose estimation process, especially for large number of outliers.

In this paper we propose a novel solution to the PnP problem that integrates an outlier-rejection mechanism and does not need to resort to an independent and separate strategy. Like in RANSAC, our approach also iterates to remove outliers. However, instead of using a geometric criterion to reject them, where at each step one needs to compute the full pose and project back all points, we use much direct criterion that results from exploring the algebraic error of a linear system inherent of our PnP formulation (See Fig. 1). In addition, this process typically converges in less than 5 iterations, even in situations where up to 50% of the correspondences are outliers. As we will show in the experimental section, this results in speed-ups of up to 100$\times$ compared to the standard P3P+RANSAC+PnP strategies, while yielding similar or even better accuracies.

2. Related work

The PnP problem has been primarily addressed for a fixed and small number of correspondences. Several closed-form solutions have been proposed to solve the P3P [11, 6, 17, 9], the P4P [8], and the problem with $n = \{4, 5\}$ points [30]. Yet, while these minimal problems can be very efficiently solved, they have the drawback of being very sensitive to noise. This is attenuated by exploiting data redundancy of larger point sets. The most straight-forward algorithm for doing so, is the $O(n)$ Direct Linear Transformation (DLT) [12]. However, DLT does not make use of the fact that in the PnP problem the calibration parameters are assumed to be known in advance, and generally gives poorer results than "specialized" PnP approaches. These, can be roughly split into iterative and non-iterative techniques.

Iterative PnP approaches optimize an objective function involving all correspondences. Standard objective functions are based on geometric errors (e.g. 2D reprojection) [24]. This method, though, is computationally expensive as it performs a demanding exploration of the solution space to avoid local minima solutions. In the Procrustes PnP (PPnP) [10], the error between the 3D points and their estimated locations is iteratively minimized in a least squares sense. Yet, since this approach is not combined with a global search strategy, it is prone to fall into local minima. An alternative to geometric errors, is to optimize algebraic errors, which are faster to compute [2]. This is explored in [21], that minimizes an algebraic error resulting from deviations in the line of sight of the 3D-to-2D correspondences. Again, this approach is sensitive to local minima.

Most of the limitations undergone by iterative methods (i.e, local minima solutions and/or high computational cost) may be overcome by non-iterative approaches. Paradoxically, early solutions were not computationally tractable, as they considered all n points as unknowns of the problem. Retrieving their locations, after imposing inter-point distances, lead to $O(n^5)$ [26] and $O(n^8)$ [1] solutions. More recently, there have been several closed form and $O(n)$ formulations that can afford arbitrarily large point sets. The first of these techniques was the EPnP [22, 18], that reduced the PnP to retrieving the position of four control points spanning any number n of 3D points. The inter-point distance constraints were only considered between these control points, and the resulting quadratic polynomials were solved with simple linearization techniques.

Subsequent works have improved the accuracy of the EPnP, still in $O(n)$, by replacing the linearization with polynomial solvers. For instance, the Robust PnP (RPnP) [19] explicitly retrieves the roots of a seventh degree polynomial that results from the least square minimization of multiple P3P problems. The Direct-Least-Squares (DLS) [13] formulates a nonlinear cost function, that produces a fourth order polynomial system and is solved using the Macauley matrix method. The main drawback in DLS is related with the Cayley representation used for the rotations, which is degenerate at 180 degrees. A new unpublished version of the DLS that avoids this problem is provided in the author's webpage. Finally, [33, 32], propose two direct minimization methods using a quaternion parameterization solved by means of a Gröbner basis solver, resulting in the so-called Accurate and Scalable PnP (ASPnP) [33] and the Optimal PnP (OPnP) [32], two of the most competitive techniques of the state-of-the-art.

Yet, as we have pointed out above, the essential problem of dealing with outliers has not been directly handled by previous PnP solutions. To the best of our knowledge this task is set aside, as an independent preliminary step based on RANSAC-like strategies [19, 33, 32], where a series of P3P problems are solved and their solutions evaluated on all

points. When a large consensus is found it is finally evaluated on a PnP method. The main drawback of this two stage strategy is that the efficiency of the PnP algorithm is not fully exploited, as most of the time is spent in the multiple evaluations of the P3P.

There are related geometric problems proposing alternative outlier-rejection schemes that are intrinsically integrated within the problem that is being solved. For instance, to robustify PCA [5, 14] use influence functions and [3, 34] use L_1-norm. In [16, 15] the L_1-norm is used within the optimization scheme to handle certain amount of outliers in matrix factorization and multiview geometry problems. Another way to handle the outlier rejection problem, is to focus on maximizing the number of inliers that satisfy the model that is being optimized. Dual problem [23] or the minimax formulations [31] are used for this purpose, although they are limited to relatively a small number of outliers ($< 35\%$). Larger amounts have been recently handled using a truncated L_2-norm [7]. However, since this approach has a $O(n^d)$ complexity, where d is the dimensionality of the model, it is constrained to problems with relative small dimensionalities, like estimating planar motions or stereo triangulations.

In this paper we draw inspiration in these outlier rejection schemes and integrate them within the linear formulation of the PnP solution proposed by the EPnP algorithm. The combination of both ingredients lets us to progressively discard outliers with a simple step-like loss function and estimate the null space of the linear system in just a few iterations, each executed in $O(n)$ time.

3. Robustified EPnP

We next describe the main ingredients of our robust version of the PnP problem. We first review and improve the linear formulation of the problem that results from the EPnP algorithm [22]. Then we show that based on this linear formulation, we can easily formulate a robust outlier rejection procedure based on the minimization of the algebraic error of the linear system. Once an initial outlier-free solution is obtained, the pose can be estimated in an straight-forward manner applying our Procrustes-based alignment.

3.1. Revisiting the EPnP Linear Formulation

Let us assume we are given a set of 3D-to-2D correspondences between n 3D reference points $\mathbf{p}_i^w = [x_i^w, y_i^w, z_i^w]^\top$ expressed in a world coordinate system w and their 2D projections $\mathbf{u}_i = [u_i, v_i]^\top$, for $i = 1, \ldots, n$. Let $\mathbf{A}$ be the camera internal calibration matrix, also assumed to be known in advance. Given these assumptions, the goal of the PnP is to estimate the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ that align the camera and world coordinate frames. As it is standard, this is addressed through the minimization of a cost function based on the accumulated reprojection errors. For

each point i, we have the following perspective constraint:

$$d_i \begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix} = \mathbf{A} \left[\mathbf{R} | \mathbf{t} \right] \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} , \tag{1}$$

where d_i is the depth of the point. Following [22], $\mathbf{p}_i^w$ can be rewritten in terms of the barycentric coordinates of four control points $\mathbf{c}_j^w$, $j = 1, \ldots, 4$, chosen so as to define an orthonormal basis centered at the origin of the world coordinate system. Every reference point, can therefore be expressed as $\mathbf{p}_i^w = \sum_{j=1}^4 \alpha_{ij} \mathbf{c}_j^w$.

Note that the barycentric coordinates α_{ij} are independent on the coordinate system, and specifically they remain the same when writing the reference points in the camera coordinate system c. That is, $\mathbf{p}_i^c = \sum_{j=1}^4 \alpha_{ij} \mathbf{c}_j^c$.

If we replace the term $\mathbf{R}\mathbf{p}_i^w + \mathbf{t}$ by $\mathbf{p}_i^c$, multiply both sides of Eq. 1 by $\mathbf{A}^{-1}$ and substitute the last row of the resulting equation into the first two rows, the perspective projection constraint for one single correspondence can be rewritten as the following linear system of two equations:

$$\begin{bmatrix} 1 & 0 & -u_i^c \\ 0 & 1 & -v_i^c \end{bmatrix} \mathbf{B}_i \mathbf{x} = \mathbf{0} \tag{2}$$

where $[u_i^c, v_i^c, 1]^\top = \mathbf{A}^{-1}[u_i, v_i, 1]^\top$ are the normalized 2D coordinates, $\mathbf{B}_i$ is a sparse 3×12 matrix built from the barycentric coordinates α_{ij}, and $\mathbf{x} = [\mathbf{c}_1^{c\top}, \ldots, \mathbf{c}_4^{c\top}]^\top$ is our unique unknown, a 12-dimensional vector made of the control point coordinates in the camera reference system. At this point we can exploit the particular sparsity pattern of $\mathbf{B}_i$ to write the Eq. 2 as a Kronecker product:

$$\begin{bmatrix} \alpha_{i1} & \alpha_{i2} & \alpha_{i3} & \alpha_{i4} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & -u_i^c \\ 0 & 1 & -v_i^c \end{bmatrix} \mathbf{x} = \mathbf{0} \tag{3}$$

Finally, concatenating these equations for all n correspondences can be expressed as a linear system $\mathbf{M}\mathbf{x} = \mathbf{0}$ where $\mathbf{M}$ is a $2n \times 12$ matrix.

It is worth to point that formulating the problem from u_i^c and v_i^c coordinates and with the Kronecker product is not used in the original EPnP, resulting in significant speedups when building the correspondence matrix $\mathbf{M}$.

3.2. Robust Null Space Estimation

As discussed in the original EPnP formulation [22] in the noise-free case the rank of the null-space of $\mathbf{M}$ is one, but due to noise this rank is assumed that can grow up to four. EPnP then proceeds by separately solving each of the cases and retaining the one that minimizes the reprojection error of all points. However, as it will be shown in the results section, this strategy is sensitive to the presence of outliers.

In this paper, instead, we assume the rank of the null-space of $\mathbf{M}$ to be always one, and propose a strategy to compute this null space while removing outliers and gross error correspondences. Indeed, the noise and outliers force

Algorithm 1 Robust Null Space Estimation

Input: $\mathbf{M}$: $2n \times 12$ corresp. matrix; $\delta_{\max}$: max. algebraic error
Output: $\mathbf{W}, \mathbf{x}$
Initialize: $\mathbf{W} = \mathbf{I}_{2n}, \xi = \text{Inf}$

1: **loop**
2:     $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathbf{M}^{\top}\mathbf{W}\mathbf{M});$
        where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_{12}];$ $\mathbf{S} = \text{diag}(s_1, \ldots, s_{12})$
3:     $\mathbf{x} \leftarrow \mathbf{v}_k;$ where $k : s_k = \min(s_1, \ldots, s_{12})$
4:     $\boldsymbol{\epsilon} = \mathbf{M}\mathbf{x}$
5:     $\varepsilon_i = \|[\epsilon_{2i-1}, \epsilon_{2i}]\|$
6:     $\varepsilon_{\max} = \mathbf{Q}_{25\%}(\varepsilon_1, \ldots, \varepsilon_n)$
7:     **if** $\varepsilon_{\max} > \xi$ **then**
8:         **return** $\mathbf{W}, \mathbf{x}$
9:     **else**
10:         $\xi = \varepsilon_{\max}$
11:     **end if**
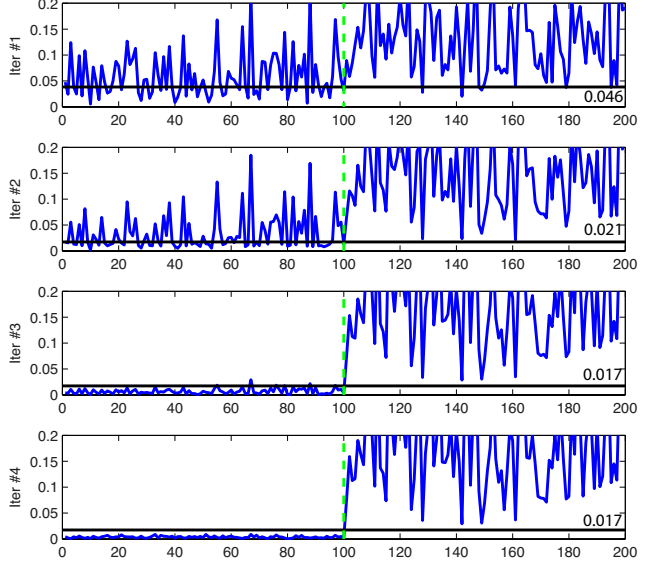12:   Update w_i using Eq. 6
13: **end loop**

Figure 2. Example of outlier rejection with 100 inliers (shown on the left-most part of the graphs) and 100 outliers. The blue line represents the algebraic error of each correspondence, and the horizontal black line is the threshold $\max(\varepsilon_{\max}, \delta_{\max})$ used in Eq. 6 to classify each match as inlier or outlier. δ_{max} is the only threshold we need to provide, and we set it to 0.017, which approximately represents an image noise of $\tau = 10$ pixels for a focal length $f = 800$.

the numerical rank of the null-space of $\mathbf{M}$ to be zero, i.e, $\mathbf{M}$ has no null singular values, although a few of them are very close to zero. In order to remove outliers, and compute a noise-free version of $\mathbf{M}$, which we denote $\mathbf{L}$, we pose the following minimization

$$\arg\min_{\mathbf{L}, \mathbf{W}} \quad \|\mathbf{W}(\mathbf{M} - \mathbf{L})\|^2 \tag{4}$$

$$\text{subject to} \quad \text{rank}(\mathbf{L}) = \text{rank}(\mathbf{M}) - 1$$

where $\mathbf{W} = \text{diag}(w_1, w_1, \ldots, w_n, w_n)$ is a $2n \times 2n$ diagonal matrix with binary entries indicating if each correspondence is considered as inlier ($w_i = 1$) or outlier ($w_i = 0$).

Since we seek to estimate the null space of $\mathbf{M}$, (i.e. the vector $\mathbf{x}$), we rewrite the problem by multiplying the previous minimization by $\mathbf{x}$. Given that $\mathbf{x}$ and the rows in $\mathbf{L}$ lie into complementary subspaces, $\mathbf{L}\mathbf{x} = \mathbf{0}$, the constrained minimization of Eq. 4 can be turned into a non-linear minimization of an algebraic error with unknowns $\mathbf{x}$ and $\mathbf{W}$,

$$\arg\min_{\mathbf{x}, \mathbf{W}} \|\mathbf{W}\mathbf{M}\mathbf{x}\|^2 . \tag{5}$$

To solve this optimization we sequentially estimate $\mathbf{x}$ and $\mathbf{W}$, as detailed in Algorithm 1. Initially, all correspondences are assumed to be inliers, and thus $\mathbf{W}$ is initialized to the identity matrix. We then compute the singular value decomposition of $\mathbf{M}^{\top}\mathbf{W}\mathbf{M}$, and take $\mathbf{x}$ to be the eigenvector associated to the smallest singular value . Let $\boldsymbol{\epsilon} = \mathbf{M}\mathbf{x}$ be the residual vector, and $\varepsilon_i = \|[\epsilon_{2i-1}, \epsilon_{2i}]\|$ the algebraic error associated to the i-th correspondence. All of the entries of the matrix $\mathbf{W}$ are updated according to a simple step function:

$$w_i = \begin{cases} 1 & \text{if } \varepsilon_i \leq \max(\varepsilon_{\max}, \delta_{\max}) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\varepsilon_{\max} = \mathbf{Q}_{25\%}(\varepsilon_1, \ldots, \varepsilon_n)$ returns the algebraic error of the correspondence that is at the boundary of the lowest 25% quartile. This function is used as a robust estimator to reject those correspondences with largest algebraic errors. For this purpose, we also evaluated using the median operator. Yet, the median has a breakdown point (maximum number of supported outliers) of 50%, while, as we will show in the results section, the breakdown point of the $\mathbf{Q}_{25\%}$ reaches the 60%.

Furthermore, note that if we solely used this criterion, in an outlier-free case we would be unnecessarily rejecting inlier correspondences. In order to avoid this situation and achieve faster convergence rates we introduce an algebraic error threshold $\delta_{\max}$ that needs to be reached in order to consider a specific correspondence as outlier, similar to the maximum reprojection error τ used in standard RANSAC to classify matches as outliers. We observed that the algebraic error grows with τ and it lessens when increasing the camera focal length f. We modeled this observation by $\delta_{\max} = k\tau/f$ (where both τ and f are expressed in pixels) and established $k = 1.4$ from a large number of synthetic experiments without outliers, in which we computed the maximum algebraic error for different amounts of noise and different focal lengths.

This process is repeated until the convergence of the null-space $\mathbf{x}$ which usually happens in less than 5 iterations.

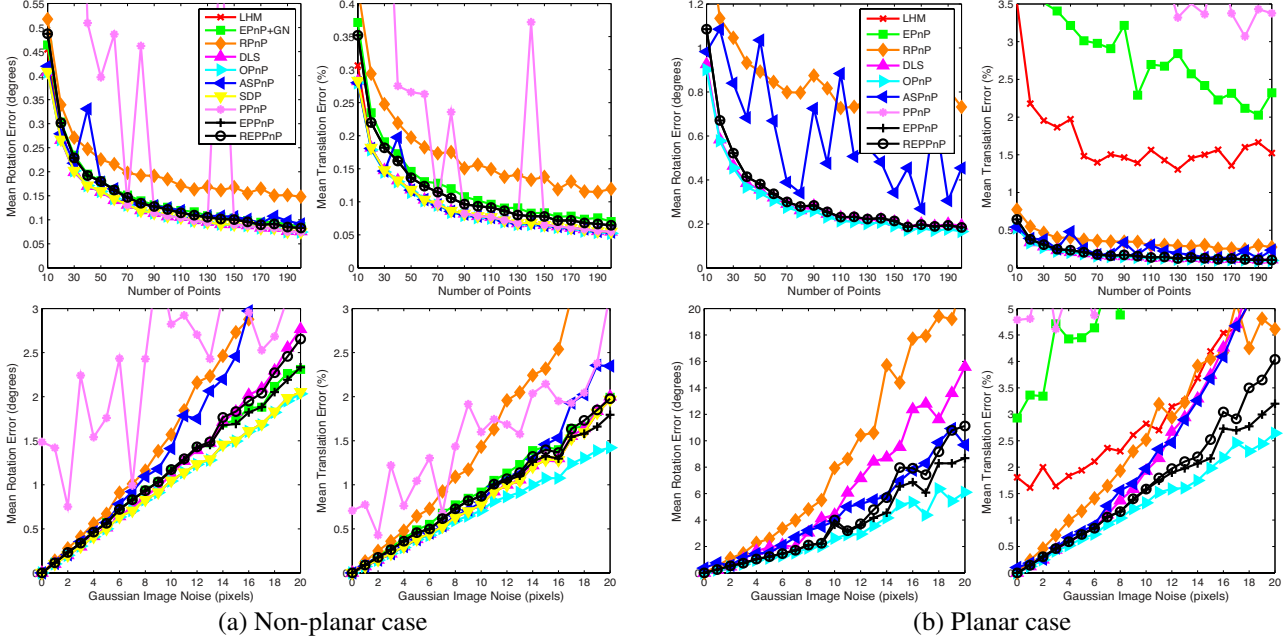|	(a) Non-planar case	(b) Planar case

Figure 3. Synthetic experiments, varying the number of correspondences (first row) and the amount of Gaussian noise (second row).

Fig. 2 shows an example of the execution of our algorithm with 50% of outliers, which converged after 4 iterations.

3.3. Estimating Scale and Absolute Pose

So far we have estimated a solution $\mathbf{x}$ of the system $\mathbf{Mx} = \mathbf{0}$ up to scale, i.e, any scaled version $\gamma\mathbf{x}$ would also be a solution of that linear system. In addition, $\mathbf{x}$ is an estimation of the position of the control points $\mathbf{c}_j^c$ in camera referencial, but our ultimate goal is to estimate the $\mathbf{R}$ and $\mathbf{t}$ that yield to absolute camera pose in the world coordinate system. This is linearly calculated using the generalization of the Orthogonal Procrustes problem [28], which solves, in closed-form, the following minimization problem:

$$\arg\min_{\gamma,\mathbf{R},\mathbf{t}} \sum_{j=1}^{4} \|\mathbf{R}\mathbf{c}_j^w + \mathbf{t} - \gamma\mathbf{c}_j^c\|^2 \qquad (7)$$

$$\text{subject to } \mathbf{R}^\top\mathbf{R} = \mathbf{I}_3$$

Additionally, and in a similar manner as [18] extends the original EPnP with a final Gauss-Newton optimization, we also propose an iterative refinement of $\mathbf{x}$. For this purpose we consider the estimation of the control points positions from Eq. 7, $\hat{\mathbf{c}}_j^c = \mathbf{R}\mathbf{c}_j^w + \mathbf{t}$, and build an estimation of the null-space vector $\hat{\mathbf{x}} = [(\hat{\mathbf{c}}_1^c)^\top, \dots, (\hat{\mathbf{c}}_4^c)^\top]^\top$. This vector is then projected on an extended null space of $\mathbf{M}$, with four components, and recompute $\mathbf{x}$ as:

$$\mathbf{x} \leftarrow \mathbf{N}\boldsymbol{\beta} : \arg\min_{\boldsymbol{\beta}} \|\mathbf{N}\boldsymbol{\beta} - \hat{\mathbf{x}}\|^2 \qquad (8)$$

where $\mathbf{N}$ is a 12×4 matrix representing the 4 eigenvectors in the assumed null space of $\mathbf{M}$ and $\boldsymbol{\beta}$ is a 4-dimensional

vector of weights. This minimization is solved in closed form, and yields updated values $\mathbf{x}$. These new estimates are in turn fed again to Eq. 7 to recompute γ, $\mathbf{R}$ and $\mathbf{t}$ until the convergence of both equations.

Since this optimization is computed just over the four control points, its computational overhead is negligible.

3.4. The Planar Case

Just as with the EPnP [22], the planar case requires a slight modification of the method. Since in this case only three control points are necessary to span the reference points onto the plane, the dimensionality of our vector of unknowns $\mathbf{x}$ drops to 9, and $\mathbf{M}$ becomes a $2n \times 9$ matrix of correspondences. Besides these changes the rest of the algorithm remains completely unchanged.

4. Experimental results

We compare the accuracy and scalability of our method, with and without the outlier rejection mechanism, against state-of-the-art on synthetic and real data. Our method is implemented in MATLAB and the source code will be made publicly available in the authors webpage.

4.1. Synthetic experiments

We assume a virtual calibrated camera with image size of 640×480 pixels, focal length of 800 and principal point in the image center. We randomly generated 3D-to-2D correspondences, where 3D reference points were distributed into the x, y, z interval $[-2, 2] \times [-2, 2] \times [4, 8]$.
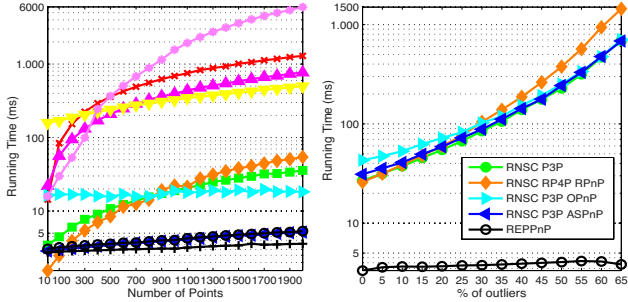
Figure 4. Running time. Varying the number of outlier-free correspondences (left) and varying the % of outliers (right). In the left, the color codes and line styles are the same as those used in Fig 3.

Figure 5. Synthetic experiments for different levels of outliers. Top: Relative error of the null-space of $\mathbf{Mx} = \mathbf{0}$ computed without removing outliers (left); and algebraic error for the inlier and outliers correspondences projected onto this null space (right). The deviation levels of the initially estimated null space let to clearly identify inliers and outliers, based on their algebraic error. Bottom: Inlier geometric reprojection errors in pixels (left), and mean translation error (right) for the pose estimated with state-of-the-art PnP solutions when outlier correspondences are not removed. Rotation error has similar levels of inaccuracy. Note that even for small percentage of outliers the obtained solutions are completely wrong.

We also added Gaussian noise to the 2D image coordinates and different percentages of outliers, produced by randomizing the 2D position of the projections. Finally, we chose the ground-truth translation $\mathbf{t}_{\text{true}}$ as the centroid of the 3D reference points and we randomly generated a ground truth rotation matrix $\mathbf{R}_{\text{true}}$. As a metric errors we used the same as in [19, 32]. The absolute error is measured in degrees between the $\mathbf{R}_{\text{true}}$ and the estimated $\mathbf{R}$ as $e_{\text{rot}}(\text{deg}) = \max_{k=1}^{3}\{\text{acos}(\mathbf{r}_{k,\text{true}}^{\top} \cdot \mathbf{r}_k) \times 180/\pi\}$ where $\mathbf{r}_{k,\text{true}}$ and $\mathbf{r}_k$ are the k-th column of $\mathbf{R}_{\text{true}}$ and $\mathbf{R}$. The translation error is computed as $e_{\text{trans}}(\%) = \|\mathbf{t}_{\text{true}} - \mathbf{t}\|/\|\mathbf{t}\| \times 100$.

All the plots discussed in this section were created by running 500 independent MATLAB simulations and report the average and median rotation and translation errors.

4.1.1 Number of Correspondences and Noise

In these experiments we compared the accuracy and running time of our method assuming there are no outlier correspondences. Therefore, our method could be applied without the outlier rejection process. However, since we want to show that this process does not affect the final results, we consider two versions of our method referred to as Robust Efficient Procrustes PnP (REPPnP) for the outlier rejection case, and Efficient Procrustes PnP (EPPnP) for the one without the outlier-rejection scheme.

We have compared our formulations against the most recent PnP approaches: the robust version of DLS [13], ASPnP [33], OPnP [32], RPnP [19], PPnP [10], EPnP + GN [18], SDP [29], and the LHM [21].

The first row of Fig. 3 plots the accuracy for increasing number of correspondences, from $n = 10$ to 200, with constant Gaussian noise of $\sigma = 2$ pixels. The second row depicts the errors for increasing amounts of noise, from $\sigma = 0$ to 20, and a constant number of correspondences $n = 30$. Both experiments for planar and non-planar configurations, show that the proposed approach yields accurate solutions, comparable to the best state-of-the-art solutions. Just to give significance of the difference between the errors
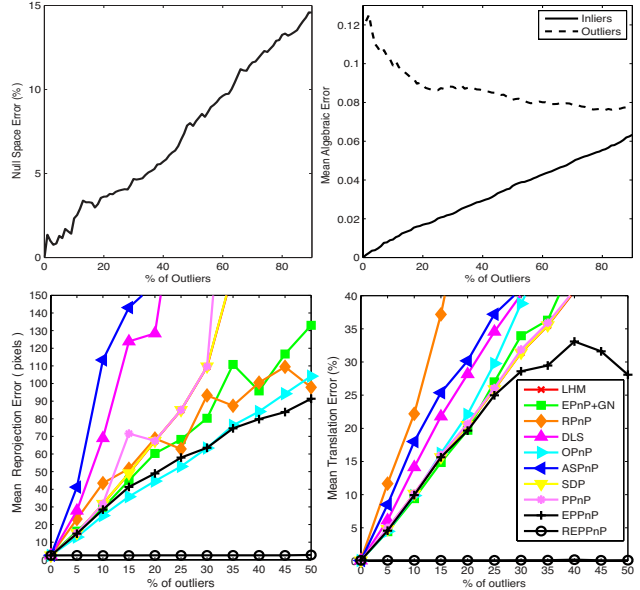
in our method and the most accurate approach (OPnP), in the worst case it is equivalent to a mean deviation of about 0.5 mm in estimating the position of points randomly distributed within a cube of 40 cm side located at a 60 cm of the camera. For the planar case, it is interesting to note how our approach significantly improves the performance of the EPnP, mostly due to the use of the Procrustes refinement stage.

In addition, as we have discussed before, we want to make clear that our approach is not intended to work in minimal cases. In fact, similar to EPnP, the performance of our approach drops for $n \leq 6$. Instead, we seek to exploit the consistency of large number of correspondences. This is indeed a realistic situation, as current keypoint detectors are able to extract hundreds of reliable feature points in standard images. And what is most important, we can process all this amount of points very efficiently, and as we will show below, even under the presence of outliers.

Fig. 4(left) shows the computation time of all methods in the outlier free case, for an increasing number of correspondences, from $n = 10$ to 2000, and with a fixed $\sigma = 2$. This experiment was done on an Intel Core i7 CPU to 2.7Ghz
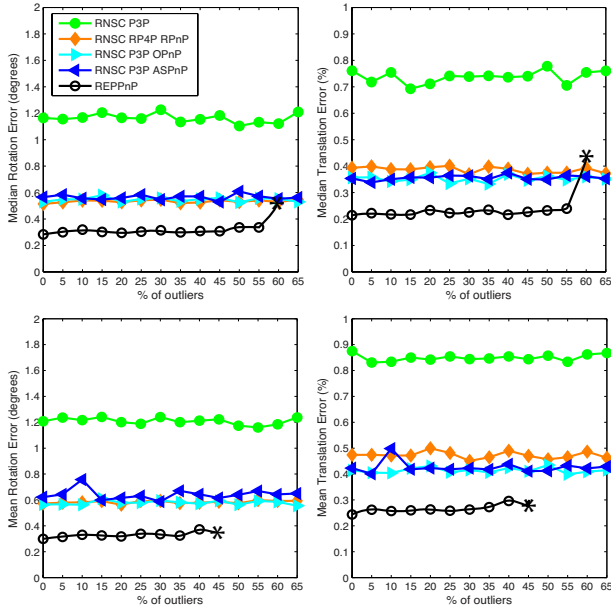
Figure 6. Synthetic experiments, varying the % of outlier correspondences. The real inliers are fixed to 100 with $\sigma = 5$. The symbol $*$ represents the last value before the breakdown point.

and all methods are implemented in MATLAB. Note that our EPPnP method is the fastest one with almost constant computational cost fixed to ≈ 3 ms whatever number of correspondences. Closely followed by ASPnP and our robust implementation REPPnP, which despite being computed in a situation without outliers, it executes several iterations.

4.1.2 Robustness to Outliers

The main contribution of our approach is to embed the outlier rejection scheme within the pose estimation pipeline. The success of our approach holds in part on the fact that the null-space $\mathbf{x}_{\text{ini}}$ computed initially (i.e., the solution of $\mathbf{Mx} = \mathbf{0}$ before removing any outlier) "only" deteriorates linearly with the presence of outliers. We show this in Fig. 5(top-left) where we plot $\|\mathbf{x}_{\text{true}} - \mathbf{x}_{\text{ini}}\|/\|\mathbf{x}_{\text{ini}}\| \times 100$ for increasing amounts of outliers. Note that the difference between the true and the initially estimated null spaces is kept within reasonable bounds for large percentages of outliers. This lets us to easily detect the outlier correspondences, as they generally have larger algebraic errors than inlier correspondences when projected onto this initial null space. We show this in Fig. 5(top-right), where we plot the algebraic error $\mathbf{Mx}_{\text{ini}}$.

This robustness to the presence of outliers, contrasts with the sensitivity undergone by the geometric error. If we use the same experimental setup, and compute the 2D reprojection using the state-of-the-art PnP methods described above, we see that the obtained solutions are so corrupted that inliers and outliers cannot be distinguished from geometric
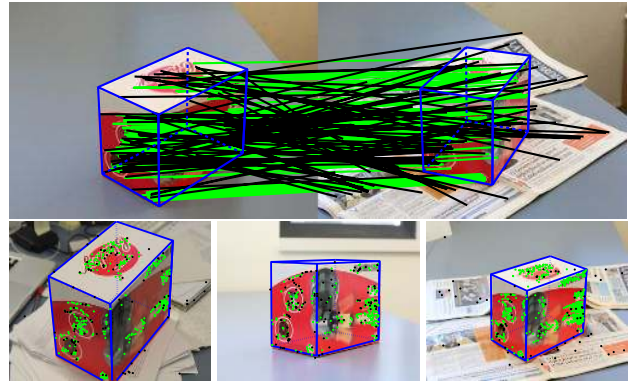

Figure 7. Real image examples. Top: Inlier and outlier correspondences found by our method between a model (left) and a test image (rigth). Bottom: Examples of fitting using our method.(Green and black dots represent the inliers and outliers respectively)

error alone (see Fig. 5(bottom-left)). Obviously this results in total failures of these methods, if outliers are not initially removed (Fig.. 5(bottom-right)).

In Fig. 6 we therefore compare the performance of our robust approach, and other PnP methods when used in a RANSAC strategy, concretely [12], with the following combinations of minimal and general approaches: (RANSAC+P3P [17]); (RANSAC + RP4P + RPnP [19]); (RANSAC + P3P [17] + ASPnP [33]); and (RANSAC + P3P [17] + OPnP [32]). Note that for percentages of outliers below 45% our approach yields slightly better results than other approches, we suspect it comes from the fact of using a different criterion to choose the inliers correspondences (Eq. 6). As mentioned above, the breakdown point of our approach is between 50% and 60% of outliers, situation which a RANSAC based strategy can still handle. However as seen in Fig. 4(right) this is at the expense of a significant increase in the computation time, which is more than 100x larger than the time required by our approach[1].

Finally, we would like to comment that we are aware that exist more efficient versions of RANSAC [27]. For instance PROSAC [4] exploits priors on the confidence of the detector . Note, however, that we could also incorporate this kind of knowledge in our approach, e.g. by weighting each correspondence based on its confidence. However, for the clarity of the paper we did not further explore these situations.

4.2. Real images

We also tested our approach in real image sequences corrupted by large amounts of outliers. Feature points and initial correspondences are obtained using SIFT [20], which yields around $200 - 400$ correspondences per image. Fig. 7(top) shows one sample input image and its set of 248

---
[1]The stopping criterion for the RANSAC is based on the number of iterations that guarantee, with a probability 0.99 that at least one minimal solution is outlier free (See. [12], page 119).

correspondences, from which 96 are outliers. Our REPPnP can detect these outliers and compute the pose in less than 4ms, $25\times$ faster than any other RANSAC-based method. The results of other detections are plotted on the bottom of Fig. 7. Note that an additional benefit of our approach, is that the outlier removal process is completely transparent to the user, simplifying thus its practical utilization.

5. Conclusions

We have proposed a very fast and accurate solution to the PnP problem which is inherently robust and does not require RANSAC-based preprocessing steps for outlier removal. The outlier rejection scheme is integrated within the pose estimation pipeline with a negligible overhead, resulting in an approach completely scalable and with almost constant computational cost for an arbitrary large number of correspondences. As a future work we plan to integrate additional priors (e.g. detector confidence levels) and explore the use of kernel functions to weight the correspondences, to expand our approach to larger percentage of outliers and even faster speeds. Following [25] we also plan to extend this approach to non-calibrated cameras.

References

[1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *PAMI*, 25(5):578–589, 2003.

[2] F. L. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Im. Proc.*, 9(1):56–71, 1979.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Jour. of ACM*, 58(3):11, 2011.

[4] O. Chum and J. Matas. Matching with prosac-progressive sample consensus. In *CVPR*, pages 220–226, 2005.

[5] F. De La Torre and M. J. Black. A framework for robust subspace learning. *IJCV*, 54(1-3):117–142, 2003.

[6] D. DeMenthon and L. Davis. Exact and approximate solutions of the perspective-three-point problem. *PAMI*, 14(11):1100–1105, 1992.

[7] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In *ECCV*, pages 738–751, 2012.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. of the ACM*, 24(6):381–395, 1981.

[9] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *PAMI*, 25(8):930–943, 2003.

[10] V. Garro, F. Crosilla, and A. Fusiello. Solving the pnp problem with anisotropic orthogonal procrustes analysis. In *3DIMPVT*, pages 262–269, 2012.

[11] J. A. Grunert. Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in geodäsie. *In Grunerts Archiv für Mathematik und Physik*, 1841.

[12] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000.

[13] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) method for pnp. In *ICCV*, pages 383–390, 2011.

[14] P. J. Huber. *Robust Statistics*. Wiley, 1981.

[15] F. Kahl, S. Agarwal, M. K. Chandraker, D. Kriegman, and S. Belongie. Practical global optimization for multiview geometry. *IJCV*, 79(3):271–284, 2008.

[16] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *PAMI*, 29(10):1834–1847, 2007.

[17] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR*, pages 2969–2976, 2011.

[18] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *IJCV*, 81(2):155–166, 2009.

[19] S. Li, C. Xu, and M. Xie. A robust o(n) solution to the perspective-n-point problem. *PAMI*, 34(7):1444–1450, 2012.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[21] C.-P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *PAMI*, 22(6):610–622, 2000.

[22] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o(n) solution to the pnp problem. In *ICCV*, pages 1–8, 2007.

[23] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. In *CVPR*, pages 1450–1457, 2010.

[24] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *PAMI*, 31(5):783–794, 2009.

[25] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *PAMI*, 35(10):2387–2400, 2013.

[26] L. Quan and Z. Lan. Linear n-point camera pose determination. *PAMI*, 21(8):774–780, 1999.

[27] R. Raguram, J.-M. Frahm, and M. Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *ECCV*, pages 500–513, 2008.

[28] P. H. Schönemann and R. M. Carroll. Fitting one matrix to another under choice of a central dilation and a rigid motion. *Psychometrika*, 35(2):245–255, 1970.

[29] G. Schweighofer and A. Pinz. Globally optimal o (n) solution to the pnp problem for general camera models. In *BMVC*, pages 1–10, 2008.

[30] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *ICCV*, volume 1, pages 278–284, 1999.

[31] J. Yu, A. Eriksson, T.-J. Chin, and D. Suter. An adversarial optimization approach to efficient outlier removal. In *ICCV*, pages 399–406, 2011.

[32] Y. Zheng, Y. Kuang, S. Sugimoto, K. Aström, and M. Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *ICCV*, pages 4321–4328, 2013.

[33] Y. Zheng, S. Sugimoto, and M. Okutomi. Aspnp: An accurate and scalable solution to the perspective-n-point problem. *Trans. on Information and Systems*, 96(7):1525–1535, 2013.

[34] X. Zhou, C. Yang, and W. Yu. Automatic mitral leaflet tracking in echocardiography by outlier detection in the low-rank representation. In *CVPR*, pages 972–979, 2012.