Exhaustive Linearization for Robust Camera Pose and Focal Length Estimation

Adrian Penate-Sanchez, Juan Andrade-Cetto, *Member, IEEE* and Francesc Moreno-Noguer

Abstract—We propose a novel approach for the estimation of the pose and focal length of a camera from a set of 3D-to-2D point correspondences. Our method compares favorably to competing approaches in that it is both more accurate than existing closed form solutions, as well as faster and also more accurate than iterative ones.

Our approach is inspired on the EPnP algorithm, a recent O(n) solution for the calibrated case. Yet, we show that considering the focal length as an additional unknown renders the linearization and relinearization techniques of the original approach no longer valid, especially with large amounts of noise. We present new methodologies to circumvent this limitation termed exhaustive linearization and exhaustive relinearization which perform a systematic exploration of the solution space in closed form. The method is evaluated on both real and synthetic data, and our results show that besides producing precise focal length estimation, the retrieved camera pose is almost as accurate as the one computed using the EPnP, which assumes a calibrated camera.

Index Terms—Camera calibration, Perspective-n-Point problem.

1 Introduction

Estimating the camera pose from n 3D-to-2D point correspondences is a fundamental and well-understood problem in computer vision. Its solution is relevant to almost every application of computer vision in the era of smart phones. The most general version of the problem requires estimating the six degrees of freedom of the pose and five calibration parameters: focal length, principal point, aspect ratio and skew. This can be established with a minimum of 6 correspondences, using the well known Direct Linear Transform (DLT) algorithm [11].

There are, though, several simplifications to the problem which turn into an extensive list of different algorithms that improve the accuracy of the DLT. The most common simplification is to assume known calibration parameters. This is the so-called Perspective-n-Point problem, for which three point correspondences suffice in its minimal version [10]. There exist also iterative solutions to the over-constrained problem with $n > 3$ point correspondences [7], [12], [21] and non-iterative solutions that vary in computational complexity and accuracy from $O(n^8)$ [1] to $O(n^2)$ [8] down to $O(n)$ [20].

For the uncalibrated case, given that modern digital cameras come with square pixel size and principal point close to the image center [4], [11], the problem simplifies to the estimation of only the focal length. Solutions exist for the minimal problem with unknown focal length [2], [18], [25], [27], and for the case with unknown focal length plus unknown radial distortion [4], [5], [14], [27].

Unfortunately, in the presence of noise and mismatches, these solutions to the minimal problem become unstable and may produce unreliable pose estimates. This is commonly addressed including an extra RANSAC [9] iterative step for outlier removal, either taking minimal or non-minimal subsets [26], but at the expense of high computational load. Recent approaches have reformulated the problem as a quasi-convex optimization problem, allowing for the estimation of global minima [6], [15], [16]. Yet, while this is a very attractive idea, the iterative nature of these approaches makes them unpractical for real-time applications, unless a very small number of correspondences is considered.

In this work we advocate for an efficient solution that can handle an arbitrarily large point sample, thus increasing its robustness to noise. Using a large point set may be especially useful for current applications such as 3D camera tracking [19] or structure-from-motion [28], which require dealing with hundreds of noisy correspondences in real time.

The method we propose fulfills these requirements: it allows estimating pose and focal length in bounded time, and since it is a non-minimal solution, it is robust to situations with large amounts of noise in the input data. Drawing inspiration on the EPnP algorithm [20], [22], we show that the solution of our problem belongs to the kernel of a matrix derived from the 3D-to-2D correspondences, and thus can be expressed as a linear combination of its eigenvectors. The weights of this linear combination become the unknowns of the problem, which we solve applying additional distance constraints.

However, solving also for the focal length has the effect that the linearization and relinearization techniques used in [20], [22] to estimate these weights are no longer valid. Several factors contribute to this: (1) the new polynomials that need to be considered are of degree four, in contrast to those in the EPnP that were of degree two; (2) the variables being computed differ in several orders of magnitude and small inaccuracies in the input data may propagate to large errors in the estimation; and (3) the number of possible combinations in the solution subspace explodes combinatorially for large kernel sizes. All these issues make that a naive selection of equations for back substitution after linearization produces unreliable results. Moreover, a least squares solution of the

- *The three authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, 08028, Spain.*
E-mail: {apenate,cetto,fmoreno}@iri.upc.edu

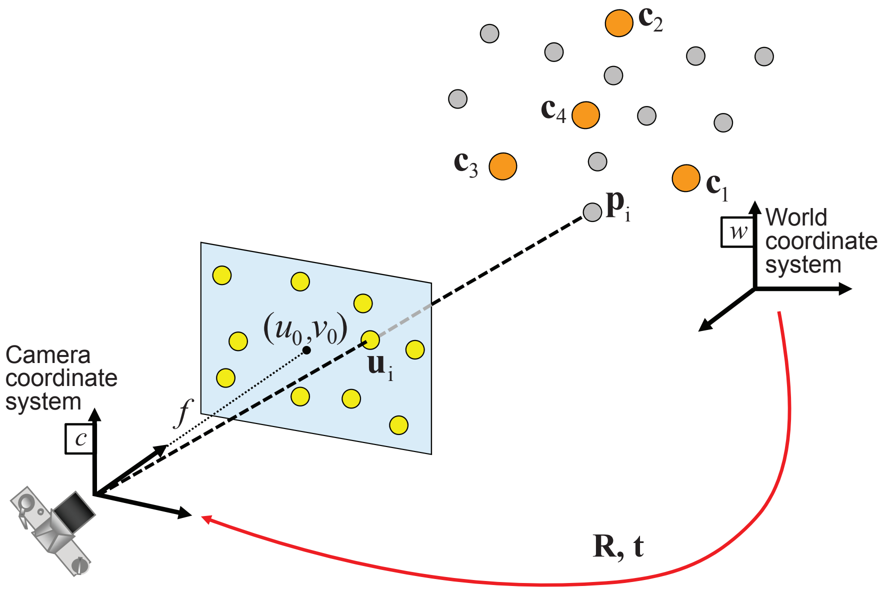

Fig. 1. **Problem Formulation:** Given a set of correspondences between 3D points $\mathbf{p}_i$ expressed in a world reference frame, and their 2D projections $\mathbf{u}_i$ onto the image, we seek to retrieve the pose ($\mathbf{R}$ and $\mathbf{t}$) of the camera w.r.t. the world and the focal length f.

kernel weights is also not viable since it will equally ponder constraints that involve variables with different orders of magnitude. We propose alternative solutions, which we call exhaustive linearization and exhaustive relinearization that circumvent these limitations by systematically exploring the solution subspace.

As will be shown in the results section, our method, called Uncalibrated PnP (UPnP), compares favorably in terms of accuracy to the DLT algorithm, the only closed-form solution we are aware that is applicable for an arbitrary number of correspondences. This is because the least squares solution of the DLT algorithm chooses an optimal solution only in the direction along the vector associated with the smallest singular value of the linear system of equations built from the 3D-to-2D correspondences. In contrast, our method considers all directions of the kernel of the system, which for the ideal case is of size one [23], but for noisy overconstrained systems grows in size [20]. Our method also yields better accuracy and efficiency than [15] and [16], which are algorithms that guarantee maximum error tolerance, but which are computationally expensive. In fact, the accuracy of our results is even comparable with that of the EPnP, which assumes known calibration parameters.

2 Problem Formulation

In this section we formulate the problem of recovering the camera pose and focal length from a set of n 3D-to-2D point correspondences. We first show that these matches yield a rank-deficient linear system, which requires additional constraints to be solved. We then introduce distance constraints that convert the original linear system into a set of polynomial equations of degree four. In Sec. 3 we introduce novel linearization techniques that help solve this polynomial set of equations.

2.1 Linear Formulation of the Problem

We assume that we are given a set of 3D-to-2D correspondences between n reference points $\mathbf{p}_1^w, \ldots, \mathbf{p}_n^w$ expressed in a world coordinate system w, and their 2D projections $\mathbf{u}_1, \ldots, \mathbf{u}_n$ in the image plane. We further assume a camera with square pixel size and with the principal point (u_0, v_0) at the center of the image, although we do not know its focal length. Under these assumptions, we formulate the problem as that of retrieving the focal length f of the camera, and the rotation $\mathbf{R}$ and translation $\mathbf{t}$, that align the world and the camera coordinate systems (see Fig. 1).

We will address this problem by minimizing the following objective function based on the reprojection error:

$$\underset{f,\mathbf{R},\mathbf{t}}{\text{minimize}} \sum_{i=1}^{n} \|\mathbf{u}_i - \tilde{\mathbf{u}}_i\|^2 , \tag{1}$$

where $\tilde{\mathbf{u}}_i$ is the projection of point $\mathbf{p}_i^w$:

$$k_i \begin{bmatrix} \tilde{\mathbf{u}}_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} , \tag{2}$$

with k_i a scalar projective parameter.

Following [20], we rewrite each 3D point in terms of the barycentric coordinates of 4 control points. This turns the problem into that of finding the solution of a linear system of $2n$ equations in 12 unknowns.

Let $\mathbf{c}_j^w, j = 1, \ldots, 4$, be the coordinates of these four control points defining an arbitrary basis in the world coordinate system. Without loss of generality, we choose this basis to be centered at the mean of the reference points and aligned with the principal directions of the data. Each reference point $\mathbf{p}_i^w$ then becomes

$$\mathbf{p}_i^w = \sum_{j=1}^{4} a_{ij}\mathbf{c}_j^w \; . \tag{3}$$

The a_{ij} terms indicate the barycentric coordinates of the i-th reference point and may be computed from the position of the reference and control points in the world coordinate system, with the normalization constraint that $\sum_{j=1}^{4} a_{ij} = 1$. Note that these barycentric coordinates are independent from the coordinate system we use, i.e., the same points in the camera referential c become $\mathbf{p}_i^c = \sum a_{ij}\mathbf{c}_j^c$.

Therefore, replacing $\mathbf{R}\mathbf{p}_i^w + \mathbf{t}$ with $\mathbf{p}_i^c$ into Eq. 2, produces the two following perspective projection equations for each 3D-to-2D correspondence:

$$\sum_{j=1}^{4}\left(a_{ij}x_j^c + a_{ij}(u_0 - u_i)\frac{z_j^c}{f}\right) = 0\,, \tag{4}$$

$$\sum_{j=1}^{4}\left(a_{ij}y_j^c + a_{ij}(v_0 - v_i)\frac{z_j^c}{f}\right) = 0\,, \tag{5}$$

where $\mathbf{u}_i = [u_i, v_i]^\top$ and $\mathbf{c}_j^c = [x_j^c, y_j^c, z_j^c]^\top$. These equations can be jointly expressed for all the n correspondences as a linear system

$$\mathbf{M}\mathbf{x} = \mathbf{0}\,, \tag{6}$$

where $\mathbf{M}$ is a $2n \times 12$ matrix made of the coefficients a_{ij}, the 2D points $\mathbf{u}_i$, and the principal point; and $\mathbf{x}$ is our vector of 12 unknowns containing both the 3D coordinates of the control points in the camera reference frame and the camera focal length, dividing the z terms:

$$\mathbf{x} = [x_1^c, y_1^c, z_1^c/f, \ldots, x_4^c, y_4^c, z_4^c/f]^\top \; . \tag{7}$$

Note that by using the barycentric coordinates we have converted the pose estimation problem to that of estimating the position of the four control points $\mathbf{c}_i^c$ in the camera coordinate system. The two problems, though, are equivalent, since given $\mathbf{c}_i^w$ and $\mathbf{c}_i^c$, we can then apply standard techniques to compute the orientation and translation between the world and camera referentials [13].

Equation 6 tells us that the solution lies on the null-space of $\mathbf{M}$. We can therefore write $\mathbf{x}$ as a weighted sum of the null eigenvectors $\mathbf{v}_k$ of $\mathbf{M}^\top\mathbf{M}$, which can be computed using Singular Value Decomposition (SVD). Hence, we write

$$\mathbf{x} = \sum_{k=1}^{N} \beta_k \mathbf{v}_k \; , \tag{8}$$

where the weights β_k become our new unknowns and N is the rank of the kernel of $\mathbf{M}^\top\mathbf{M}$. It can be shown that, for $n \geq 6$, and with noise-free correspondences, $N = 1$. In practice, though, noise makes no eigenvalue exactly zero and the matrix $\mathbf{M}^\top\mathbf{M}$ has full rank. Nonetheless, the matrix loses rank numerically and the effective dimension of the null space increases. Thus, we have to consider the effective dimension of the kernel being greater than one, and to cope with this situation, we follow a similar strategy as in [20], and compute the solution for various values of N, picking the one that minimizes Eq. 1. There is no clear criterion on the value of N to choose, as this will depend on the focal length magnitude and on the amount of noise in our input data. Yet, setting $N \leq 3$ has proven adequate in all our experiments.

2.2 Introducing Distance Constraints

In order to solve for the weights β_k in Eq. 8 we add constraints that preserve the distance between control points. That is, for each pair of control points $\mathbf{c}_j$ and $\mathbf{c}_{j'}$,

$$\|\mathbf{c}_j^c - \mathbf{c}_{j'}^c\|^2 = d_{jj'}^2 \; , \tag{9}$$

where $d_{jj'}$ is the known distance between control points $\mathbf{c}_j^w$ and $\mathbf{c}_{j'}^w$ in the world coordinate system. Rewriting $\mathbf{c}_j^c$ and $\mathbf{c}_{j'}^c$ in terms of the β_k coefficients, from Eqs. 7 and 8 we obtain

$$\mathbf{c}_j^c = \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix} = \sum_{k=1}^{N} \begin{bmatrix} \beta_k v_{k,x}^{[j]} \\ \beta_k v_{k,y}^{[j]} \\ f\beta_k v_{k,z}^{[j]} \end{bmatrix} \; , \tag{10}$$

where $\mathbf{v}_k^{[j]} = [v_{k,x}^{[j]}, v_{k,y}^{[j]}, v_{k,z}^{[j]}]^\top$ is the sub-vector of $\mathbf{v}_k$ corresponding to the coordinates of the j-th control point. Observe that the unknown focal length has been moved to the right-hand side of Eq. 8 and now multiplies the z component of the control points. As a consequence, applying the distance constraints between all pairs for control points will now generate 6 polynomials of degree 4, in contrast to the quadratic equations appearing in the original EPnP formulation. As we will see in the following sections this will require a substantially different approach, especially when solving the cases $N = 2$ and $N = 3$.

3 Exhaustive Linearization and Relinearization

In this section we introduce novel closed-form linearization techniques to solve the systems of polynomial equations which result from combining Eq. 8 and the six distance constrains of Eq. 9. We will see that a standard linearization approach is only effective to solve the case $N = 1$ (when only two variables need to be estimated), but it fails to solve the cases $N = 2$ and $N = 3$, in which we have a larger number of unknowns while the number of equations remains the same.

3.1 Case $N = 1$: Linearization

For the case where $N = 1$, we only need to solve for β_1 and f. This case may be solved by simply linearizing the system of equations and introducing new unknowns for the quadratic and bi-quadratic terms. In particular, we will use $\beta_{11} = \beta_1^2$, and $\beta_{ff11} = f^2\beta_1^2$. Applying the six distance constraints from Eq. 9 between all pairs of control points, results in a system of the form

$$\mathbf{L}\mathbf{b} = \mathbf{d}\,, \tag{11}$$

where $\mathbf{b} = [\beta_{11}, \beta_{ff11}]^\top$ and $\mathbf{L}$ is a 6×2 matrix built from the known elements of $\mathbf{v}_1$, and $\mathbf{d}$ is a 6-vector of squared

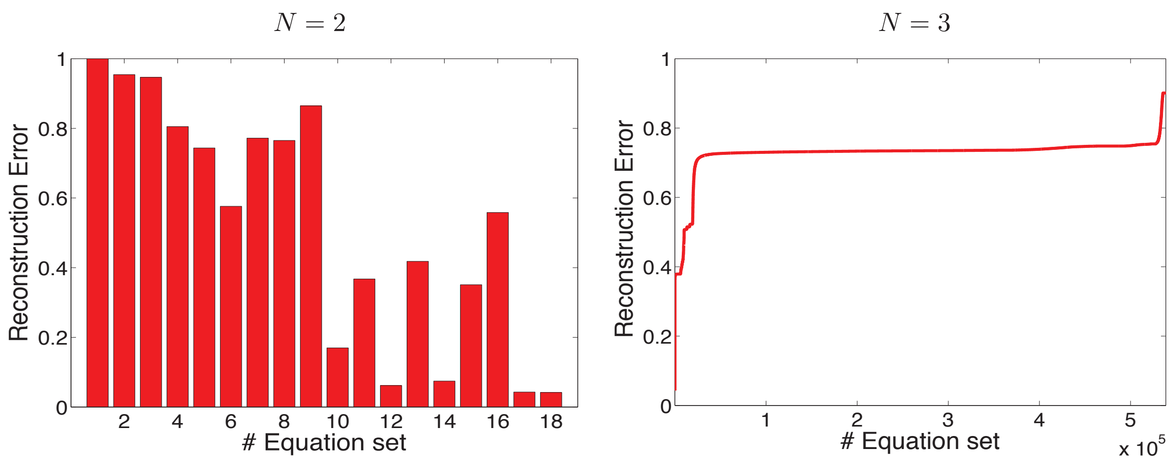

Fig. 2. Reconstruction error for all possible equation sets. The graphs plot the mean normalized reconstruction error over 1000 different experiments with random input correspondences, and different amounts of noise. **Left.** All 18 triplet combinations for a kernel of size $N = 2$. **Right.** All 538.704 quadruplet combinations for a kernel of size $N = 3$. In this case, the reconstruction errors have been sorted in increasing order of magnitude for viewing purposes. Observe that in both cases, the selection of one set of equations from another, results in significantly different reconstruction error (and hence pose and focal length estimation).

distances between the control points. We solve this overdetermined linearized system using least squares and estimate the magnitudes of β_1 and f by back substitution:

$$\beta_1 = \sqrt{\beta_{11}}\,, \qquad f = \sqrt{|\beta_{ff11}|/|\beta_1|}\,. \tag{12}$$

Finally, we select the sign of β_1 such that after computing the pose, all the points end up placed in front of the camera.

3.2 Case $N = 2$: Exhaustive Linearization

For the case $N = 2$ we need to solve for β_1, β_2 and f. Applying the six distance constraints we obtain again a linear system $\mathbf{Lb} = \mathbf{d}$, where now $\mathbf{L}$ is a 6×6 matrix built from substituting the known elements of the basis $\mathbf{v}_1$ and $\mathbf{v}_2$ into Eq. 9. The number of unknowns becomes a six dimensional vector

$$\mathbf{b} = [\beta_{11}, \beta_{12}, \beta_{22}, \beta_{ff11}, \beta_{ff12}, \beta_{ff22}]^\top. \tag{13}$$

Note that the entries in $\mathbf{L}$ become quadratic expressions on the elements of the orthogonal basis vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, and since these are made of control points which are by construction different from each other, $\mathbf{L}$ has full rank. Following a similar procedure as before, we can thus retrieve the vector of linear unknowns $\mathbf{b}$ computing the inverse of $\mathbf{L}$.

However, the simple backsubstitution scheme used to solve for each of the individual unknowns as in Eq. 12 is no longer valid. In fact, by simple observation of the vector $\mathbf{b}$ it can be seen that the individual variables may be computed, once $\mathbf{b}$ is known, applying backsubstitution over 18 different triplets, namely $(\beta_{11}, \beta_{12}, \beta_{ff11})$, $(\beta_{11}, \beta_{12}, \beta_{ff12})$, $(\beta_{11}, \beta_{12}, \beta_{ff22})$, $(\beta_{11}, \beta_{22}, \beta_{ff11})$ and so on. It turns out that in the absence of noise, all these triplets render the same solution, but when noise comes into play, each of the triplets has a different effect on the solution. This is depicted in Fig. 2-left, where we plot the mean reconstruction error of the solution obtained with each triplet, computed as the mean Euclidean distance between the 3D points aligned with respect to the ground truth camera coordinate system, and the same 3D points aligned using the estimated pose and focal length.

To choose the right equation set we propose what we call an *exhaustive linearization*, which is a strategy that generates and explores all possible triplets, and takes the one that minimizes the reprojection error of Eq. 1. Note that the number and form of each triplet is always the same, and independent of the input data. Therefore, this exploration can be efficiently executed in parallel.

To solve the monomial quadratic terms we rewrite bilinearities as logarithmic sums. That is, by applying logarithms on the absolute values of all the elements within the triplet, we can rewrite the terms β_{ij} as equations of the form $\log|\beta_{ij}| = \log|\beta_i| + \log|\beta_j|$. Doing this for all elements within the triplet produces a linear system of 3 equations and 3 unknowns that yields the magnitude of each individual variable. To determine the sign of each variable we check sign consistency with the components of $\mathbf{b}$ that have not been used, and also enforce the geometric constraints of positive focal length and 3D point location in front of the camera.

3.3 Case $N = 3$: Exhaustive Relinearization

For the case of $N = 3$ we need to solve for β_1, β_2, β_3 and f. Unfortunately, neither the linearization nor the exhaustive linearization techniques suffice to address this case, because the number of quadratic unknowns in the linearized system is larger than the number of equations. We have 12 linearized terms of the form β_{kl} and β_{ffkl} with k and $l \in \{1, 2, 3\}$, while the number of distance constraints remains equal to six. We solve this problem by using a relinearization technique [17] in conjunction with our exhaustive strategy described above.

We call the combination of both methods as *exhaustive relinearization*.

The idea of the relinearization technique is to add constraints that enforce the algebraic nature of the elements β_{kl} and β_{ffkl}. We start by considering the following homogeneous linear system:

$$[\mathbf{L}| - \mathbf{d}] \begin{bmatrix} \mathbf{b} \\ \rho \end{bmatrix} = \mathbf{0} \quad \Rightarrow \quad \tilde{\mathbf{L}}\tilde{\mathbf{b}} = \mathbf{0} \,, \tag{14}$$

where $\tilde{\mathbf{L}}$ is now a 6×13 matrix, $\tilde{\mathbf{b}}$ is a 13-vector including the quadratic and biquadratic unknowns, and ρ is a scaling factor. The solution for $\tilde{\mathbf{b}}$ is then spanned by the null space of $\tilde{\mathbf{L}}$. That is,

$$\tilde{\mathbf{b}} = \sum_{i=1}^{M} \lambda_i \tilde{\mathbf{w}}_i \,, \tag{15}$$

where $\tilde{\mathbf{w}}_i$ are the right singular vectors of $\tilde{\mathbf{L}}$. As in the case $N = 2$, $\tilde{\mathbf{L}}$ is of rank 6 by construction, and thus $M = 7$. Finally, we solve for the λ_i-s setting $\rho = 1$ to remove the scale ambiguity, and using additional constraints coming from the commutativity of the multiplication of the β_{kl} and β_{ffkl} monomials, e.g.,

$$\beta_{klmf} = \beta_{kl}\beta_{mf} = \beta_{k'l'm'f'} \,, \tag{16}$$

where (k', l', m', f') represents any permutation of (k, l, m, f). After imposing these constraints, the coefficients λ_i are solved using linearization, and thus the name *relinearization*.

However, this second linearization suffers again from the problem we mentioned above for the case $N = 2$. That is, the coefficients λ_i may be retrieved from small sets of quadratic monomials $\lambda_{ij} = \lambda_i\lambda_j$, but due to noise, choosing each of these sets produces a different reprojection error, which is the function we are trying to minimize. Hence, we need to perform again an exploration of the possible minimal sets of λ_{ij} vectors. In addition, once the coefficients $\lambda_1, \ldots, \lambda_M$ have been recovered, we need to retrieve the coefficients β_1, β_2, β_3 and f by exploring the possible minimal sets of β_{kl} vectors. To filter out parasitic solutions we impose the additional constraints $\beta_{ii}\beta_{jk} = \beta_{ij}\beta_{ik}$.

3.3.1 Efficient Exploration of the Minimal Equation Sets

Note that the number of all possible sets of equations we have to explore grows exponentially with M. In our experiments we have observed that it is sufficient to explore only up to the 5th singular vector of $\tilde{\mathbf{L}}$, which produces 1548 different equation sets from which to retrieve the λ's, and for each of them we have 348 quadruplets from which to retrieve the β's. This yields a total of 538.704 possible combinations to explore. Exploring all possible combinations is computationally expensive (in the order of minutes on a standard PC). Fortunately, the right equation set to choose does not heavily depend on the point configuration nor the value of the focal length, more than on the algebraic combination of variables. For this reason, we devised a strategy to select off-line the best equation set from a large number (10^3) of synthetic experiments, without jeopardizing the computational efficiency of the overall method at run time.

ExploreSet($\mathbf{p}$,$\mathbf{u}$,E_1,E_2)
Input:
$\mathbf{p}$: 3D points.
$\mathbf{u}$: image correspondences.
E_1: reprojection error for the case $N = 1$.
E_2: reprojection error for the case $N = 2$.
Output:
$\mathbf{t}^*$: Camera translation.
$\mathbf{R}^*$: Camera rotation.
f^*: focal length.

$E^* \leftarrow \infty$
if $E_1 > E_{\text{min}}$ **and** $E_2 > E_{\text{min}}$ **then**
  for each equation set $\mathcal{Q}_i$ in decreasing rank order **do**
    $(\lambda's,\beta's,f) \leftarrow$ ExhaustiveRelinearization($\mathbf{p}$,$\mathbf{u}$,$\mathcal{Q}_i$)
    $(\mathbf{R},\mathbf{t}) \leftarrow$ RecoverPose($\mathbf{p}$,$\beta's$,f)
    $E \leftarrow$ ReprojectionError($\mathbf{p}$,$\mathbf{R}$,$\mathbf{t}$,f)
    if $E < E^*$ **then**
      $E^* \leftarrow E$, $\mathbf{R}^* \leftarrow \mathbf{R}$, $\mathbf{t}^* \leftarrow \mathbf{t}$, $f^* \leftarrow f$
    end if
    if $i > i_{\text{max}}$ **or** $E^* \leq E_{\text{min}}$ **then**
      Return($\mathbf{R}^*$, $\mathbf{t}^*$, f^*)
    end if
  end for
end if

Algorithm 1: Algorithm to explore the set of equations in the case $N = 3$.

The idea is to order the equation sets according to their weighted contribution in solving all experiments in the large training session. To do this, we run the complete algorithm over synthetic random input data and assign to each equation set $\mathcal{Q}_i$, a weight inversely proportional to the cumulative reconstruction error throughout all experiments. Fig. 2-right illustrates the normalized error distribution for each equation ordered using this weight.

At run time, this ordering is used to test each equation set searching for the one that minimizes Eq. 1, as shown in Alg. 1. Only in those cases when the reprojection is still not good enough (above a threshold E_{min}) for the cases $N = 1$ and $N = 2$, we enter the case $N = 3$ and iterate over the ordered list of equation sets, compute the λ's, β's and f for each set, and use these parameters to recover the pose parameters $\mathbf{R}$ and $\mathbf{t}$. The solution is updated should it improve the reprojection error. A stopping condition is set after exploring a reduced number of equation sets or once the reprojection error falls below E_{min}.

The parameter i_{max} defines the maximum number of equation sets to explore, and thus it is an upper bound in the time required by our algorithm. This parameter offers a trade-off between efficiency and optimality. While the computation time grows linearly with i_{max}, the residual error of the minimization rapidly falls after just a few iterations. In practice, as shown in the next section, by setting the maximum number of equations to validate to 500, the accuracy results are comparable to that of the calibrated case, while maintaining computational efficiency. In addition, E is usually good enough for the cases $N = 1$ or $N = 2$, preventing from having to evaluate the case $N = 3$ at all, a situation that happens roughly 80% of the time for noise levels between 1 and 3 pixels.

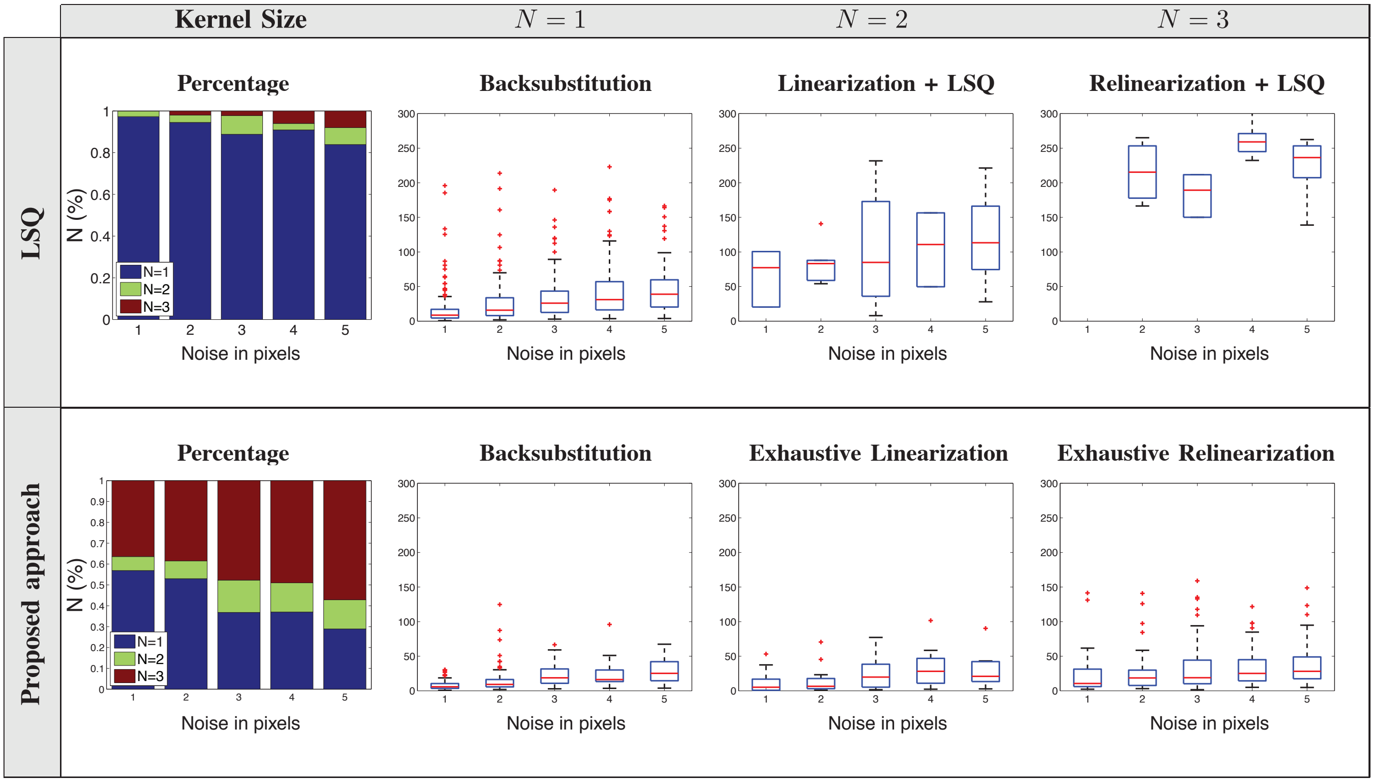

Fig. 3. Comparison of our approach that estimates the parameters via backsubstitution over minimal equation sets, against an approach that estimates these via backsubstitution over a least squares approximation using all equations for a set of 1000 experiments and varying image noise. **Left column:** Effective number N of null eigenvalues of $\mathbf{M}^\top\mathbf{M}$. **Second to fourth columns:** Reprojection error distributions (in pixels) for the $N = 1$, $N = 2$, and $N = 3$ cases. The box edges in the boxplots denote first and third quartiles, red lines inside the boxes indicate medians, dashed lines represent the extent of the statistical data, and red crosses are outliers.

3.4 Why Using Minimal Sets of Equations?

One question that may naturally arise from our methodology is why exploring minimal sets of equations (triplets for solving the case $N = 2$ and quadruplets for the case $N = 3$). As an alternative to this, we could have also tried to take the logarithms of all the elements of the vector $\mathbf{b}$, and use least squares over the resulting overdetermined system to retrieve the variables $\log|\beta_i|$ and $\log f$. However, although this solution is faster than independently evaluating triplets or quadruplets and retaining the solution with minimum reprojection error, it is far less accurate. The reason is that the algebraic combination of variables with severe differences in order of magnitude, weights binomials that include focal length more heavily than other binomials, and a least squares solution would wrongly average such inconsistencies.

To see this effect, we compare the Exhaustive Linearization and Exhaustive Relinearization approaches, to linearization and relinearization implementations that use least squares to solve for the β's and λ's. Fig. 3 compares both alternatives in an experiment where pose and focal length are computed for $n = 6$ random 3D-to-2D correspondences with increasing amounts of noise in a 640×480 image. The leftmost plots show the effective number N of null singular values in $\mathbf{M}^\top\mathbf{M}$, i.e., the percentage of solutions in which the minimal reprojection error has been obtained for each specific value of N. Note that for $N = 1$, neither linearization nor relinearization come into play, since β_1 and f are obtained by simple backsubstitution in both cases, with the only difference being the percentage of solutions with minimal error. The differences between the methodologies can be assessed for those cases in which $N = 2$ or $N = 3$ improve the solution obtained with $N = 1$. When all minimal equation sets are independently explored, we observe a significant increase in the percentage of cases in which a solution with $N = 2$ or $N = 3$ produces smaller reprojection error than a solution with $N = 1$ (see Fig. 3 third and fourth columns), indicating that exhaustive linearization and exhaustive relinearization clearly outperform a least squares solution.

This result was in fact expected because the noise in the input 3D-to-2D correspondences is not homogeneously propagated through the SVD decomposition and linearization processes, and as seen in Fig. 2, it results in equation sets with very different accuracies. Simultaneously handling all equation sets in a least squares sense does not allow to filter out these large variations, and is only using a robust method like the algorithm we proposed in the previous section that we can optimally search for the right values for the β's and λ's.

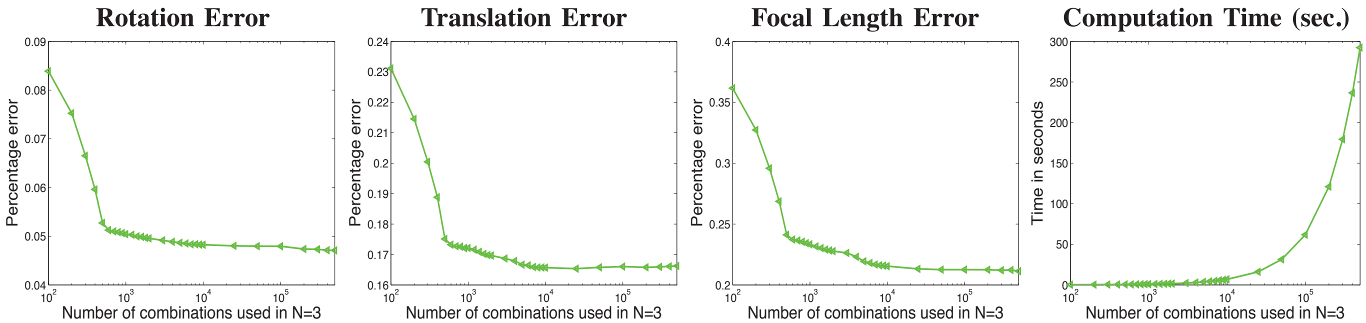

Fig. 4. Mean rotation, translation and focal length estimation errors when $N = 3$ is selected as the best solution (we refer the reader to Sec. 4.1 for a precise definition of these errors), and computation time for an increasing number of equation sets. Note that the horizontal axis is plotted in logarithmic scale, and the time scales linearly with the number of equation sets. Exploring 500 equation sets is a good trade off between accuracy and computation time. These graphs are generated with random experiments with $n = 7$ points, and large amounts of 2D noise (at the level of $\sigma_n = 5$), in order to ensure that exploration of the case $N = 3$ was meaningful.

3.5 Dealing with Planar Configurations

Like the EPnP algorithm [20], our approach can be easily adapted to address situations in which the 3D points lie on a plane. For these configurations, the n 3D reference points can be spanned using only three control points –instead of four–. The 3D to 2D projection of the point correspondences may then be written as a linear system equivalent to that of Eq. 6, but with a different dimensionality. Now, the matrix $\mathbf{M}$ of coefficients will be $2n \times 9$, and the vector of unknowns will contain the focal length and the coordinates of only three control points, $\mathbf{x} = [x_1^c, y_1^c, z_1^c/f, \ldots, x_3^c, y_3^c, z_3^c/f]^\top$.

We will solve this homogeneous linear system by independently resolving specific dimensionalities of the Kernel of $\mathbf{M}^\top\mathbf{M}$, as in the non-planar case. However, note that when using three control points, we can only define up to three constraints based on their inter-distances. These three equations will not be sufficient to solve for the six unknowns of the vector $\mathbf{b}$ in Eq. 13, for the case $N = 2$. As a consequence, for $N \geq 2$, we will need to make use of the additional equations provided by the extended relinearization technique explained above.

3.6 Iterative Refinement

Although the exhaustive linearization and relinearization techniques perform a sequential exploration of the collection of equation sets, the spirit of the whole algorithm is still non-iterative, as no initialization is required and the exploration can be performed in bounded time. We will now feed this result into a final iterative stage that will increase the accuracy in the estimation of both the camera pose and focal length at a very small additional cost.

Following [20], we iterate over the parameters β_1, β_2, β_3, and f to solve the problem

$$\underset{\beta_1,\beta_2,\beta_3,f}{\text{minimize}} \sum_{(i,j) \text{ s.t. } i<j} (\|\mathbf{c}_i^c - \mathbf{c}_j^c\|^2 - d_{ij}^2) \tag{17}$$

where the d_{ij}'s are the known distances between control points in the world coordinate system and, following Eq. 10, the $\mathbf{c}_i^c$ are expressed in terms of the β_k coefficients and focal length f. Their values are initialized to those estimated using the exhaustive linearization approaches, or to zero when they are not available. That is, when the effective rank of $\mathbf{M}^\top\mathbf{M}$ is found to be $N = 1$, then β_2 and β_3 are initialized to zero. When the rank is found to be $N = 2$, only β_3 is set to zero. We then perform the minimization using a standard Gauss-Newton optimization.

Note that the minimization is performed over the four dimensional space of the β's and f coefficients, and not over the seven dimensional space of the pose and focal length. In addition, since in general the initialization provided by the linearization approaches is usually very accurate, the optimization typically converges in about 10 iterations. Overall, the impact of this refinement on the method's computational time is of less than 5% of the total time.

4 Experimental Results

In this section we compare the accuracy of our algorithm with and without the final Gauss Newton optimization (we denote these cases *UPnP+GN* and *UPnP*, respectively) against the *DLT* [11], and the approaches [15] and [16], which search for a global solution. The first of these methods, denoted by *L2-L2*, is based on a branch and bound strategy that minimizes the L_2 norm of the reprojection error. The approach described in [16] shows that replacing the L_2 norm by the L_∞ norm yields a convex formulation of the problem with a unique minimum which is retrieved using second-order cone programming. In the following we will denote this method by *Linf* [1]. Note that DLT, L2-L2 and Linf retrieve the complete 3×4 projection matrix $\mathbf{P}$, while our approach separately estimates the orientation $\mathbf{R}$, translation $\mathbf{t}$ and focal length f. In order to perform a fair comparison, given $\mathbf{P}$ we will first retrieve the calibration matrix $\mathbf{A}$, using a Cholesky factorization of $\mathbf{P}_3\mathbf{P}_3^\top$,

1. For the L2-L2 method we have used the implementation from the Branch and Bound Optimization toolbox available at http://www.cs.washington.edu/homes/sagarwal/code.html. The code for the Linf method has been taken from the L-infinity toolbox available at http://www.maths.lth.se/matematiklth/personal/fredrik/download.html.

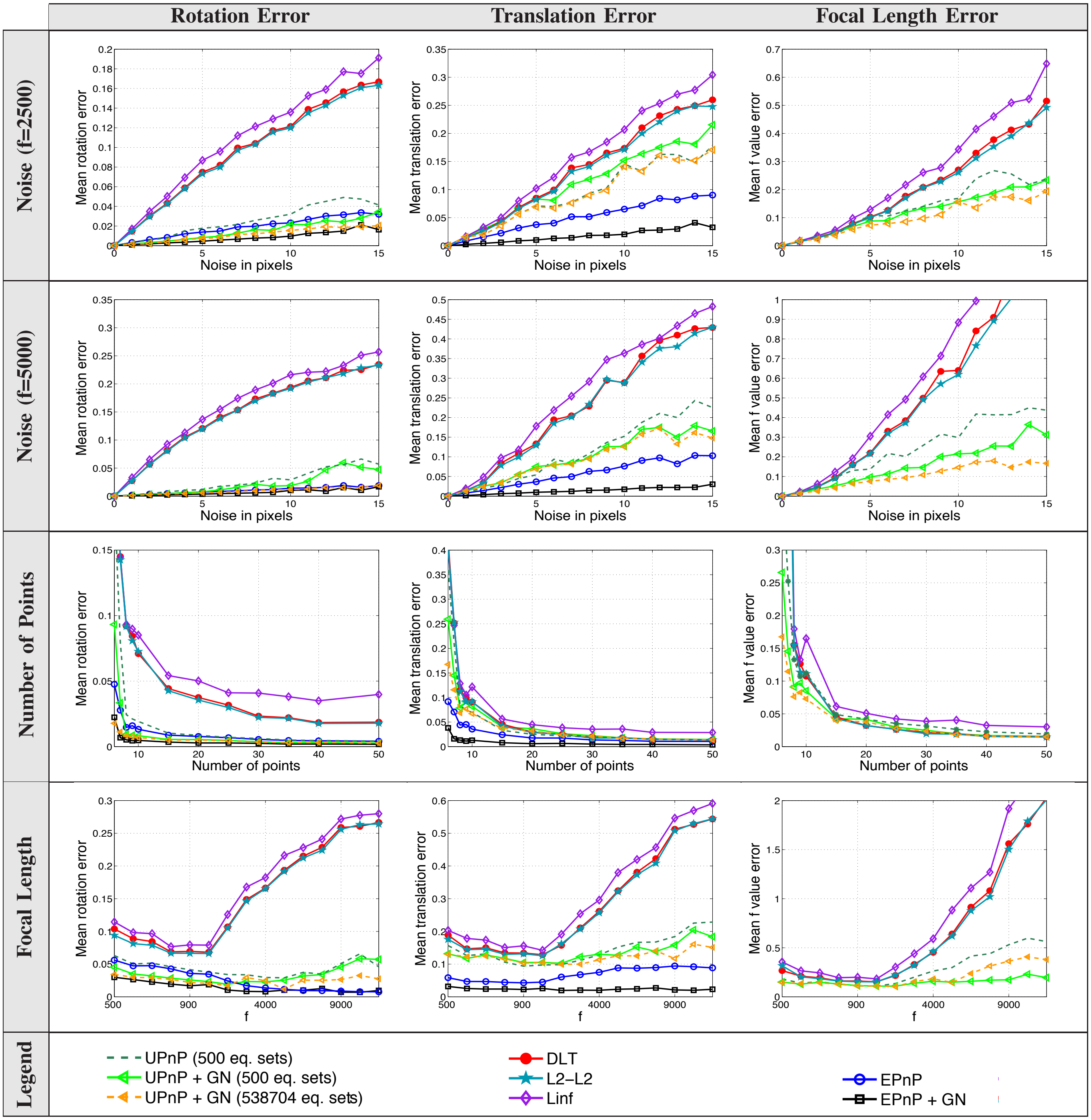

Fig. 5. Results on synthetic data for non-planar distributions of points. **Two upper rows:** Mean rotation, translation and focal length errors for: increasing levels of image noise on 10 2D-3D correspondences, and two different focal lengths. **Third row:** increasing number of 2D-3D correspondences. **Fourth row:** increasing focal length, also for 10 point correspondences. Each tick in the plot represents the average over 100 experiments with random points.

where $\mathbf{P}_3$ is the left 3×3 submatrix of $\mathbf{P}$ [29]. We will then fix the principal point to the ground truth value and estimate $\mathbf{R}$ by ortho-normalizing $\mathbf{A}^{-1}\mathbf{P}_3$. The translation vector $\mathbf{t}$ is directly estimated from the last column of $\mathbf{P}$.

We also include the results of the EPnP [22], and the EPnP with a Gauss-Newton refinement [20]. For both these approaches the true focal length is provided and obviously work better than the uncalibrated methods. We plot them here as a reference baseline.

One parameter that needs to be chosen beforehand in our algorithm, is the maximum number $i_{\max}$ of equations we want to explore for the case $N = 3$. In Fig. 4 we plot the pose and focal length estimation errors as a function of the number of equations, when we enforce our algorithm to compute

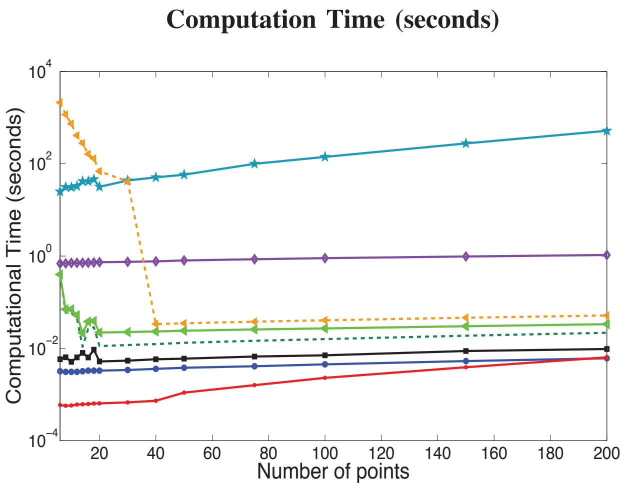

Fig. 6. Comparison of the computation time of our method against state of the art approaches with respect to the number of input points, and for fixed values of $f = 2500$ and $\sigma_n = 5$. The color codes and line styles are the same as those used in Fig. 5.

pose with only the case $N = 3$. In all cases we obtain reasonable results in relatively short time by exploring 500 sets of equations, which only represents a very small fraction of all possible 538.704 combinations. In the experiments, we will thus evaluate each of these situations, indicating the number of explored equations. When nothing is said, we will assume that only 500 equations are evaluated.

4.1 Non-Planar Synthetic Experiments

For the synthetic experiments, we simulated 3D-to-2D correspondences for sets of points of different size, uniformly distributed in the cube $[-2, 2] \times [-2, 2] \times [4, 8]$, and projected onto a 640×480 image using a virtual calibrated camera with squared pixels, and principal point at $(u_0, v_0) = (320, 240)$. Image points were corrupted with Gaussian noise.

For any given ground truth camera pose, $\mathbf{R}_{\text{true}}$ and $\mathbf{t}_{\text{true}}$, focal length f_{true}, and corresponding estimates $\mathbf{R}$, $\mathbf{t}$ and f, the relative rotation error was computed as $\mathrm{E}_{\text{rot}} = \|\mathbf{q}_{\text{true}} - \mathbf{q}\| / \|\mathbf{q}\|$, where $\mathbf{q}$ and $\mathbf{q}_{\text{true}}$ are the normalized quaternions of $\mathbf{R}$ and $\mathbf{R}_{\text{true}}$, respectively; the relative translation error was computed with $\mathrm{E}_{\text{trans}} = \|\mathbf{t}_{\text{true}} - \mathbf{t}\| / \|\mathbf{t}\|$; and the error in the estimation of the focal length was determined by $\mathrm{E}_f = |f_{\text{true}} - f|/f$. All errors reported in this section correspond to average errors estimated over 100 experiments with random positions of the 3D points.

The first and second rows in Fig. 5 show the robustness of all methods against image noise. For these experiments the 2D coordinates of the matches were corrupted with additive Gaussian noise with a growing standard deviation σ up to 15 pixels, and the number of correspondences was set to $n = 10$. Observe that our approach performs consistently better than other uncalibrated approaches, and even retrieves the rotation matrices with an accuracy comparable to that of the calibrated ones. Yet, the translation error is larger, and responds to the fact that the ambiguity between focal length and translation cannot be perfectly solved, specially for noisy 2D-to-3D correspondences. In fact, note that not even with the final refinement using Gauss Newton optimization and considering all equation sets we were able to completely solve this ambiguity. In any case, both the translation and focal lengths estimations we obtain are remarkably more accurate than those obtained by the rest of uncalibrated methods. It is worth to note that the L2-L2 algorithm guarantees a bound with respect to the global minimum solution below a certain tolerance ϵ, which we set to 0.05. Although improved accuracies might be achievable choosing smaller tolerances, we found it prohibitive as the computational burden at $\epsilon = 0.05$ was already too high.

The third row in Fig. 5 shows the robustness of the method for varying sizes of the point correspondence set. Fixing the image reprojection noise at $\sigma = 5$, and varying the number of points in the set from 6 to 50, the method again outperforms the other uncalibrated methods, and turns to be very similar to the EPnP. In particular, using all equation sets and only six points, pertains to the situation depicted in Fig. 3. Note that although in this case there is a clear difference in exploring all or just the reduced set of equations, for point sizes $n \geq 8$, the solutions recovered using the reduced equation set are as good as the solution using all equations, with a significant advantage in computational cost.

The last row in Fig. 5 plots simulation results for varying focal length values. The number of 3D-to-2D correspondences and their 2D noise are set to constant values of $n = 10$ and $\sigma = 5$, respectively. Note that while for low values of f, our UPnP method performs slightly better than other approaches, as projection becomes orthographic, the difference becomes more drastic. The UPnP algorithm remains stable whereas the accuracy of the other uncalibrated algorithms degenerates. This is because DLT, L2-L2 and Linf assume a projective camera model, which leads to failure when the camera gradually comes close to turning orthographic. In contrast, our approach can naturally handle this situation, as the effect of moving from a fully perspective to an orthographic camera is to increase the dimensionality of the kernel of $\mathbf{M}^\top\mathbf{M}$, and thus for large values of the focal length, the UPnP method automatically finds the most accurate solutions at $N = 2$ or $N = 3$.

Fig. 6 shows the computation time of all algorithms for an increasing number of input correspondences and fixed values of $\sigma = 5$ and $f = 2500$. All algorithms are implemented in Matlab, although the Linf and L2-L2 methods use compiled C functions. Among the uncalibrated methods, only the DLT algorithm is faster than our algorithm, although as shown in Fig. 5, the DLT performs comparatively very poorly in terms of accuracy. Surprisingly, our approach happens to be slower for a small number of input correspondences. This is because when the number of input points is small, the pose and focal length estimates become very sensitive to noise. This requires evaluating all kernel dimensionalities, i.e., $N = 1, 2$ and 3, where the latter may be quite expensive, especially when testing all equation sets. In particular, observe that the difference in computation time of having to test 500 or all 538.304 equation sets is of more than two orders of magnitude,

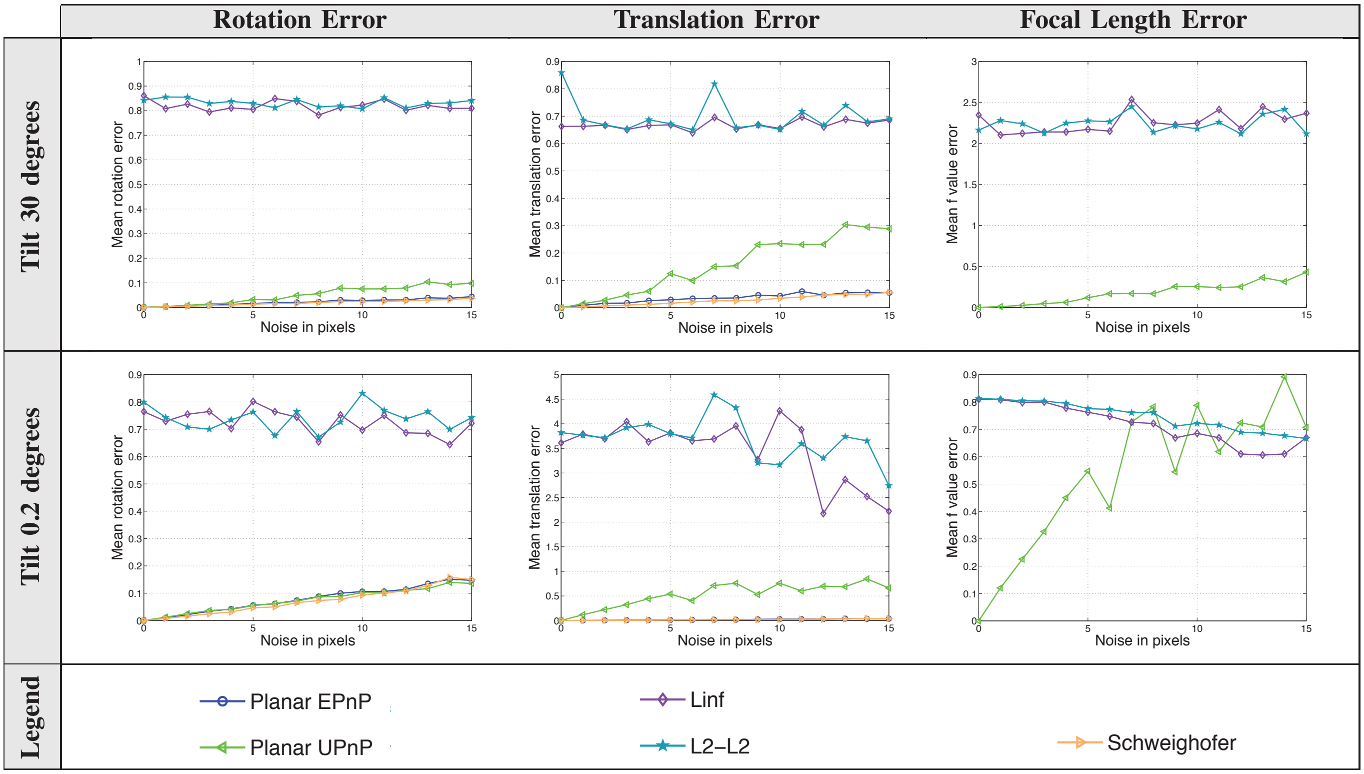

Fig. 7. Results on synthetic data for planar distributions of points. Mean rotation, translation and focal length errors for increasing levels of image noise, and two different tilt values.

while the performances reported in Fig. 5 of both alternatives are pretty similar.

Yet, when the size of the correspondence set increases ($n \geq 9$), ambiguities and instabilities induced by noise are reduced, and small reprojection errors are generally obtained by just evaluating $N = 1$ and $N = 2$. In fact, for a large number of points, the computation time of our approach is very similar to that of the EPnP, which assumes a calibrated camera. In addition, the cost of our algorithm could be further improved by exploiting the fact that the equations sets that need to be explored are independent and known in advance, and thus, their exploration could be easily parallelized.

4.2 Planar Synthetic Experiments

We now present the results obtained on planar scenes. The DLT has been removed from this analysis as it is not directly applicable to planar distributions of points. By contrast, we have included the approach of Schweighofer and Pinz [24], which is a calibrated method specifically designed to handle planar scenes. Jointly with the EPnP, this method is used as a baseline to evaluate the magnitude of the error of the non-calibrated approaches.

These experiments have been performed for a constant number $n = 10$ of 3D-2D correspondences, corrupted using Gaussian noise with a standard deviation σ ranging from 0 to 15 pixels. In addition, we have considered two different situations, one in which the points lie on a quasi frontoparallel plane, and another in which this plane has a tilt of 30 degrees w.r.t to the optical axis of the camera. Fig. 7 summarizes the results. Note that the pose and focal length estimates obtained using the UPnP clearly outperform those of the Linf and L2-L2 methods. The accuracy of our approach only falls when noisy input data is combined with a frontoparallel distribution of points. In this case, the ambiguity between focal length and translation is magnified and cannot be resolved by any of the non-calibrated methods. Yet, our approach yields an estimation of the rotation matrix which is almost as accurate as that of the calibrated algorithms.

4.3 Real Images

The method was also tested on a real image sequence taken with a Canon EOS 550D digital camera. The camera was manually moved around an object of interest with known geometry and the focal length was changed from 600 to 2000 pixels. Ground truth focal lengths were read from the *exif jpeg* image headers, and ground truth poses were computed by applying the EPnP+GN to a set of 3D-to-2D matches manually selected. We then manually registered the 3D model to one reference image, from which we extracted approximately 500 SIFT feature points. After backprojecting these 2D points onto the model we obtained a set of reference 3D points, with an associated SIFT descriptor.

At runtime, 2D feature points and their corresponding SIFT descriptors were automatically extracted from each input image, and matched to the set of reference 3D points. This provided an initial set of 3D-to-2D hypotheses. To filter out

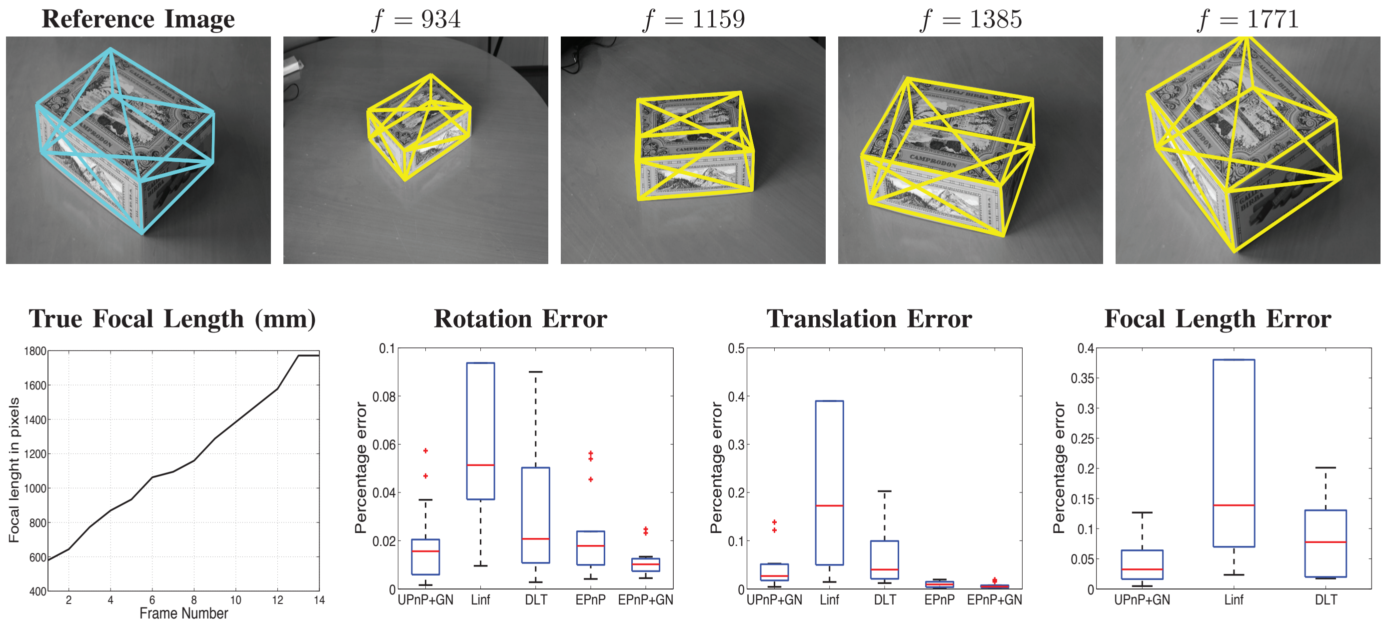

Fig. 8. Results on a real sequence with increasing focal length. **Top.** 3D model reprojected onto the reference and input images using the pose and focal length retrieved with the UPnP. **Bottom.** Ground truth focal length and performance comparison of all methods.

UPnP+GN	Linf	DLT	EPnP	EPnP+GN
1.67	1052.04	2.72	0.15	0.13

TABLE 1
RANSAC Computation Times (seconds)

outliers, we then independently ran RANSAC using each of the algorithms until obtaining a consensus of 200 inlier correspondences. The UPnP, EPnP and DLT performed quite efficiently while Linf required a considerable additional amount of time. The L2-L2 was not applicable within a RANSAC framework as its convergence rate was even two orders of magnitude larger than that of Linf. Table 1 reports the mean computation time per frame required for each method.

The accuracies of all approaches are depicted in Fig. 8-bottom. The images on the top show the reprojection obtained with UPnP.

Finally, as a test case, the method was also used to register 12 images available on Flickr of the Cheverny Castle with its GoogleEarth 3D model. Feature correspondences were manually matched in both the reference and input to obtain pose ground truths. The true focal lengths were obtained from the camera settings available in the Flickr images. The test was again performed after using RANSAC to filter out mismatches between the SIFT features of the reference and input images. As shown in the box plots at the bottom of Fig. 9, our method compares again favorably with the DLT and Linf algorithms. Some of the reprojection results are shown in the top of the figure.

4.4 Comparison with Minimal Solutions

The UPnP provides an efficient solution to estimate pose and focal length from an arbitrary large number of 3D-to-2D correspondences. As discussed in Section 2.1, the minimum number of correspondences which are required to solve the underlying linear system of Eq. 6 is 6. In fact, we could even solve when only 5 noise-free correspondences are given, as in this case the rank of the kernel of $\mathbf{M}^\top\mathbf{M}$ would be $N = 2$. Solving the minimal case with 4 correspondences requires considering larger kernel dimensionalities, and the complexity of the exhaustive relinearization would become impractical. In order to solve the minimal case with four correspondences there exist specialized algorithms such as [2], which takes advantage of the constraints introduced by all the possible pairs of distances between 3D points. These constraints generate a system of 15 polynomial equations, solved using hidden variable or Gröbner basis methods. Note however, that this is only feasible for the minimal case, as the number of pairs of distances between points explodes with n. In Fig. 10 we compare the performance of [2], which we denote as P4Pf, with the UPnP for $n = 6$ and for an increasing amount of noise. As expected, P4Pf is more sensitive to noise, and by just considering two additional correspondences UPnP yields significantly more accurate results.

One advantage of taking minimal subsets is that it increases the chances of picking an *all-inliers* subset in a RANSAC-based algorithm. Yet, while this may accelerate the outlier removal process when considering noise free data, it can have an opposite effect when the data, besides containing outliers, is corrupted by noise [26]. In this case, the hypotheses fitted on minimal subsets may be severely biased, even when only containing inliers, and many true inliers may not be included

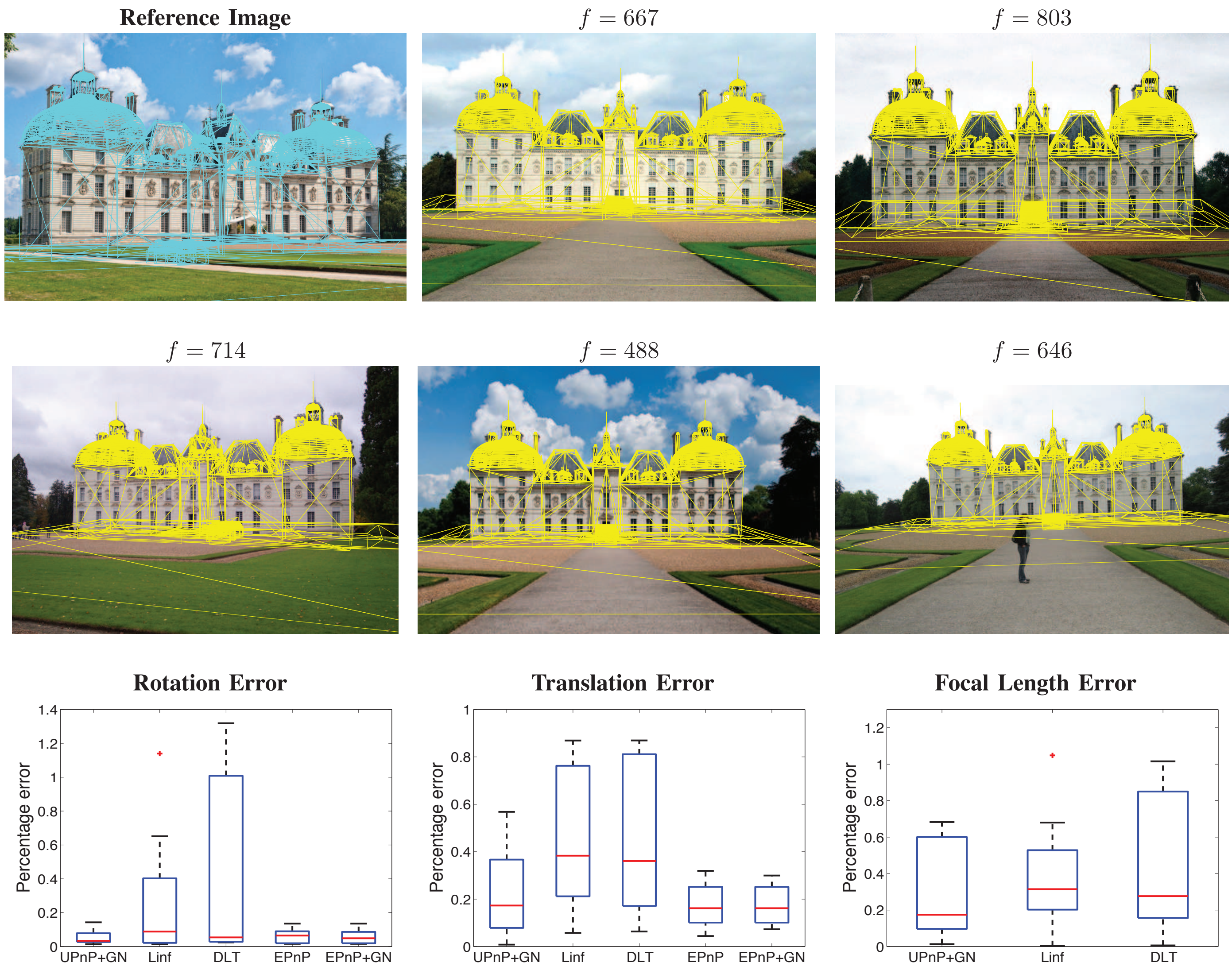

Fig. 9. Results on a real set of images obtained from $Flickr$ with a 3D model obtained from $GoogleEarth$. **Top.** 3D model reprojected onto the reference and input images using the pose and focal length retrieved with the $UPnP$. **Bottom.** Comparing the accuracy of our approach against Linf, DLT, EPnP and EPnP+GN. The last two methods assume a calibrated camera.

in the final consensus set, leading to accuracy errors. In order to put evidence on this, we have performed an experiment where the P4Pf and the UPnP have been used within a RANSAC scheme. We have considered a set of 5000 3D-to-2D correspondences, corrupted by 2D noise with $\sigma = 5$ pixels, and different percentages p_o of outliers, going from 10 to 60%. Taking minimal subsets of size $n = 4$ for the P4Pf and $n = 6$ for the UPnP, we have then followed an hypothesize-and-test approach, until reaching a maximum number of iterations $i_{\text{max}}^{\text{ransac}}$, that ensures with a confidence level P an outlier-free hypothesis. This threshold is computed as [9]

$$i_{\text{max}}^{\text{ransac}} = \frac{\log(1 - \mathrm{P})}{\log(1 - (\mathrm{p}_i)^n)} \tag{18}$$

where $\mathrm{p}_i = 1 - \mathrm{p}_o$ is the percentage of inliers, and P has been set to a 98% level. Fig. 11-left shows this theoretical number of iterations for the different percentages of outliers. Fig. 11-center and right represent the rotation, translation and focal length errors for each method and level of outliers. In each of these graphs, we plot both the error computing the pose and focal length with the *best* minimal subset for each algorithm, and the error computing the pose and focal length using the whole consensus. The latter has been computed with the UPnP in both cases, as the P4Pf can only be used in the minimal case. Observe that although the P4Pf requires a smaller number RANSAC iterations, the UPnP consistently yields better estimations of the pose. In fact, there are levels of outliers for which the number of theoretical iterations is very similar in both algorithms, while the gain in accuracy is still significant in favor of the UPnP.

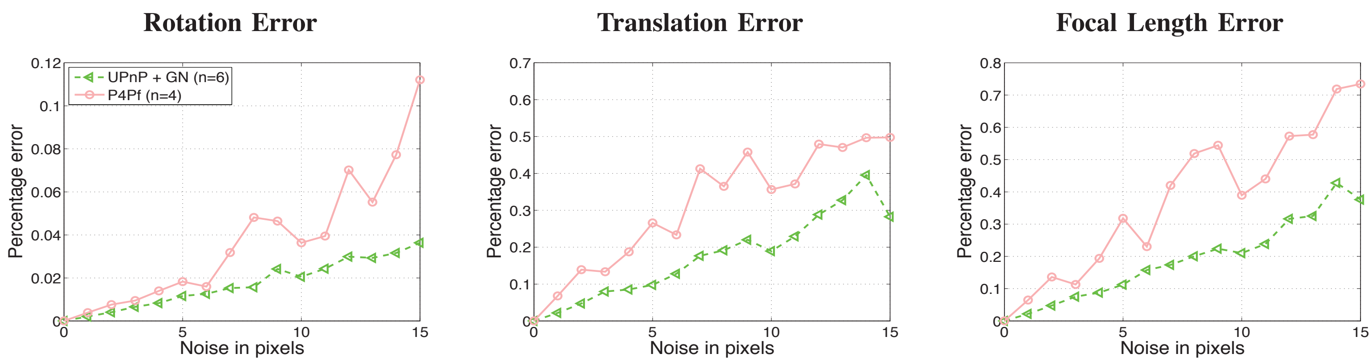

Fig. 10. Comparison of the UPnP using 6 correspondences vs. the minimal approach proposed in [2], which estimates pose and focal length from four 3D-to-2D correspondences.

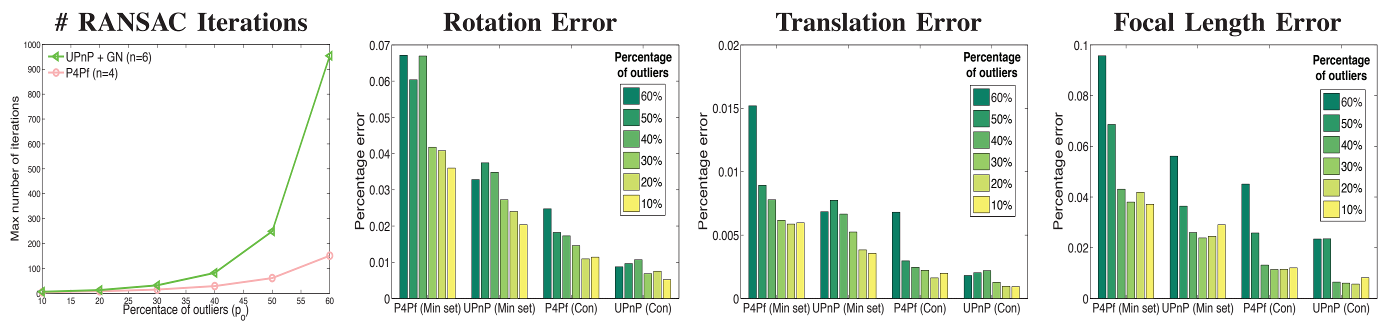

Fig. 11. Comparison of UPnP vs P4Pf within a RANSAC scheme. **Left:** Number of iterations required to retrieve a hypothesis of n point correspondences free of outliers, where $n = 4$ for the P4Pf and $n = 6$ for the UPnP. **Other three frames:** Rotation, Translation, and Focal length errors for different levels of outliers. Min set: Errors obtained when computing pose using the best minimal subset. Con: Error after computing pose using all the correspondences within the *inlier* set. Since the P4Pf does not generalize to more than four correspondences, the error of the consensus is computed using the UPnP in both cases.

5 Conclusions

In this paper, we have presented a fast solution to the problem of recovering the pose and focal length of a camera, given n 3D-to-2D correspondences. We have shown that our approach can be expressed as the solution of a fixed-size linear set of equations independent of the number of points, similar to the EPnP algorithm for the fully calibrated case. However, dealing with uncalibrated cameras required the introduction of new approaches to handle higher degree polynomials under noisy input data. To this end, this paper presents the *extended linearization* and *extended relinearization* techniques, which overcome the limitations of current linearization-based approaches. An extensive evaluation of the method shows remarkable improvement when compared to competing methods, and also to algorithms for pose recovery that make use of calibrated cameras.

An unexploited advantage of the approach is that it is highly parallelizable for large kernel sizes since the sets of equations that need to be exhaustively explored are known in advance. Another alternative to speed up the process would be to use strategies such as Kernel voting [3] to directly pick a solution from the set of minimal equations, based on how well they satisfy the distance constraints. This would remove the need to repetitively calculate and test reprojection error. We leave this as an unexplored venue for further research.

6 Acknowledgments

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness under project PAU+ DPI2011-27510, and by the EU projects INTELLACT FP7-269959 and ARCAS FP7-287617. A. Penate-Sanchez is the recipient of a JAE-Predoc scholarship funded by the European Social Fund.

References

[1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003.

[2] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the P4P problem for camera with unknown focal length. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[3] M. Bujnak, Z. Kukelova, and T. Pajdla. Robust focal length estimation by voting in multi-view scene reconstruction. In *Asian Conference on Computer Vision. Vol. 5994 of Lecture Notes in Computer Science*, pages 13–24, 2009.

[4] M. Bujnak, Z. Kukelova, and T. Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In *Asian Conference on Computer Vision. Vol. 6492 of Lecture Notes in Computer Science*, pages 11–24, 2010.

[5] M. Byrod, Z. Kukelova, K. Josephson, T. Pajdla, and K. Astrom. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[6] K. Choi, S. Lee, and Y. Seo. A branch-and-bound algorithm for globally optimal camera pose and focal length. *Image and Vision Computing*, 28(9):1369–1376, 2010.

[7] D. DeMenthon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141, 1995.

[8] P.D. Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 23(2):140–148, 2001.

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

[10] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[12] R. Horaud, F. Dornaika, B.t Lamiroy, and S. Christy. Object pose: The link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, 22(2):173–189, 1997.

[13] B. K. P Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.

[14] K. Josephson and M. Byrod. Pose estimation with radial distortion and unknown focal length. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2419–2426, 2009.

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

[16] F. Kahl and R. Hartley. Multiple-view geometry under the L_∞-norm. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 30(9):1603–1617, 2008.

[17] A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Annual International Cryptology Conference*, pages 19–30, 1999.

[18] Z. Kukelova, M. Bujnak, and T. Pajdla. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *British Machine Vision Conference*, pages 56.1–56.10, 2008.

[19] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[20] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate O(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):151–166, 2008.

[21] C.P. Lu, G.D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 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 *International Conference on Computer Vision*, pages 1–8, 2007.

[23] M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[24] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 28(12):2024–2030, 2006.

[25] H. Stewenius, D. Nister, F. Kahl, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 789–794, 2005.

[26] T. Thang-Pham, T.J. Chin, J. Yu, and D. Sutter. The random cluster model for robust geometric fitting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 710–717, 2012.

[27] B. Triggs. Camera pose and calibration from 4 or 5 known 3D points. In *International Conference on Computer Vision*, pages 278–284, 1999.

[28] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. In *International Conference on Computer Vision*, pages 1–8, 2007.

[29] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 1998.

Adrian Penate-Sanchez (S'09-M'10) received the Computer Engineering and Master in Cybernetics and Telecommunications degrees from the University of Las Palmas de Gran Canaria, Spain. He is a PhD student at the Institut de Robòtica i Informàtica Industrial, Barcelona, Spain, since 2010. His current research interest is on geometric computer vision.

Juan Andrade-Cetto (S'94-M'95) received the BSEE degree from CETYS Universidad, Mexico, in 1993, the MSEE degree from Purdue University, USA, in 1995, and the PhD degree in Systems Engineering from the Universitat Politécnica de Catalunya, Barcelona, Spain, in 2003. He received the EURON Georges Giralt Best PhD Award in 2005. He is Associate Researcher of the Spanish Scientific Research Council at the Institut de Robòtica i Informàtica Industrial, Barcelona, Spain. His current research interests include state estimation and computer vision with applications to mobile robotics.

Francesc Moreno-Noguer received the MSc degrees in industrial engineering and electronics from the Technical University of Catalonia (UPC) and the Universitat de Barcelona in 2001 and 2002, respectively, and the PhD degree from UPC in 2005. From 2006 to 2008, he was a postdoctoral fellow at the computer vision departments of Columbia University and the École Polytechnique Fédérale de Lausanne. In 2009, he joined the Institut de Robòtica i Informàtica Industrial in Barcelona as an associate researcher of the Spanish Scientific Research Council. His research interests include retrieving rigid and nonrigid shape, motion, and camera pose from single images and video sequences. He received UPC's Doctoral Dissertation Extraordinary Award for his work.