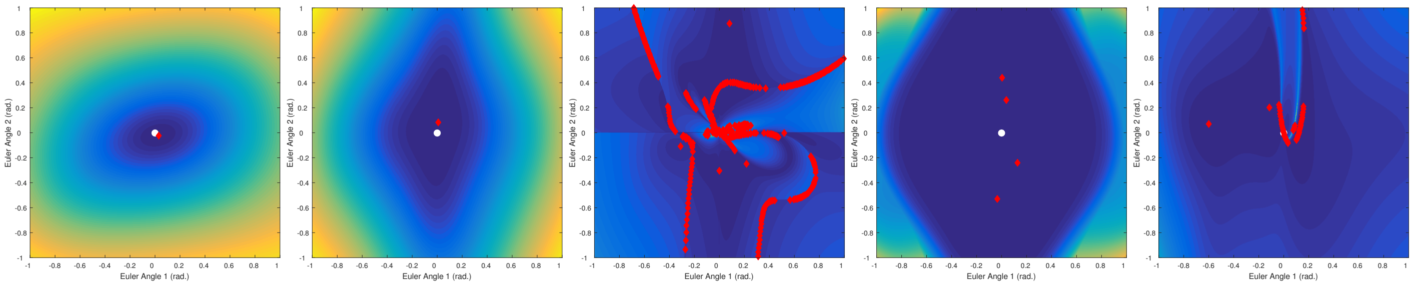
HARD-PnP: PnP Optimization Using a Hybrid Approximate Representation

Simon Hadfield, *Member, IEEE,* Karel Lebeda, Richard Bowden, *Senior Member, IEEE*

Abstract—This paper proposes a Hybrid Approximate Representation (HAR) based on unifying several efficient approximations of the generalized reprojection error (which is known as the *gold standard* for multiview geometry). The HAR is an over-parameterization scheme where the approximation is applied simultaneously in multiple parameter spaces. A joint minimization scheme "HAR-Descent" can then solve the PnP problem efficiently, while remaining robust to approximation errors and local minima.
The technique is evaluated extensively, including numerous synthetic benchmark protocols and the real-world data evaluations used in previous works. The proposed technique was found to have runtime complexity comparable to the fastest $O(n)$ techniques, and up to 10 times faster than current state of the art minimization approaches. In addition, the accuracy exceeds that of all 9 previous techniques tested, providing definitive state of the art performance on the benchmarks, across all 90 of the experiments in the paper and supplementary material.

Index Terms—PnP, perspective-n-point, camera resectioning, overparameterization, multiview geometry



(a) Cost surface (b) Proposed warp (c) Euler axis warp (d) Rot. vec. warp (e) Quaternion warp

Fig. 1: Visualization of the PnP cost surface in the presence of noise and outliers, when optimized using different parameterizations. Color indicates the reprojection error from low (blue) to high (yellow). Fig. 1a plots this against initial camera orientation (defined by 2 Euler angles). Remaining subplots show the resulting error from a single refinement in various parameterization spaces. Diamonds indicate local minima and the white circle is the ground truth pose.

1 Introduction

Estimating the pose of a camera using observations of n points from the environment is one of the most fundamental problems in multi-view geometry. It's often referred to as the Perspective-n-Point (PnP) problem, and has been investigated since the 1980s [1]. The PnP problem can be seen as a special case of the Bundle Adjustment problem, when there is only 1 camera and the 3D point cloud is not modified. Recent applications of the PnP problem include robotics [2], augmented reality [3], [4], 3D tracking [5], structure from motion [6] and action recognition [7]. Despite excellent progress in recent years, it is extremely challenging to develop a fast and accurate approach, which is resistant to noisy observations and near-singular point configurations. We propose a direct minimization approach, operating on an extremely efficient approximation to the theoretically optimal cost function. We also introduce a Hybrid Approximate Representation (HAR) and corresponding joint optimization framework HAR-Descent, which make it possible to unify the representations from several of state-of-the-art algorithms, improving robustness and accuracy. Figure 1 provides an illustration of this idea. The topology of the cost surface (i.e. the extrema) does not change when transformed to different parameter spaces. However, the *shape* of the cost surface, and thus the path taken during optimization, changes dramatically. Because of this, the refined cost surfaces shown in Figures 1b-e demonstrate vastly different convergence properties and even different probabilities of encountering local minima. By jointly selecting an optimization path which is simultaneously suitable for all these parameter spaces, we greatly reduce the likelihood of encountering a local minimum in the cost surface. To verify these findings and motivate the remainder of the paper, we provide the Matlab code necessary to generate Figure 1 as supplementary material.

Direct minimization methods are widely employed for PnP problems, either as a complete solution or as a final "polishing" stage in the pipeline [8]. However, the characteristics of minimization approaches depend heavily on the choice of error function to be minimized. The reprojection error is generally considered the *Gold standard* error function for multi-view geometry [9], however it is very difficult to optimize, leading to a fractional programming problem (*i.e.* relying on the ratio of two general non-convex functions). As a result, direct minimization methods either have prohibitive computational complexity, such as the branch-and-bound method of [10], or optimize an alternative algebraic error via local-optimization [11], [12], reducing the accuracy of the results and leading to issues with local minima. In contrast, the direct minimization scheme derived in this paper is solved efficiently using convex combination descent [13].

2 Related work

Previous work on efficient direct minimization techniques for PnP have explored various algebraic cost functions. Lu *et al*. [11] proposed an iterative minimization of the object space error, and proved that their approach was globally convergent. Schweighofer and Pinz [12] explored ambiguities in the object space error for planar targets, and later approximated the problem using a Semidefinite program relaxation [14]. Garro *et al*. [15] more recently proposed an algebraic image-space error, which they minimized iteratively.

In contrast to these direct minimization schemes, there is a second class of solution to the PnP problem which is particularly popular in the literature. We refer to these as algebraic techniques, as they focus on developing alternative parameterizations of the PnP problem, which are then rewritten into polynomial form. For example, Hesch and Roumeliotis [16] develop a system of polynomial equations based on the rotation vector, and solve them using the Macaulay matrix. Recently, the field has had a great deal of success solving these algebraic techniques using the Gröbner basis method, partly due to the automated basis generator of Kukelova *et al*. [17]. The best known example is the 5 point Essential matrix algorithm of Stewénius *et al*. [18], but it has also been applied to the PnP problem using derivations from the Cayley representation [2], the unit-quaternion representation [19] and the non-unit quaternion representation [20]. Recently Wu [21] used the Gröbner basis method to solve the P4P problem (unknown focal length) using a hybrid representation of one Euler angle and a 2 DOF quaternion[1]. In brief, the Gröbner basis method requires a particular monomial ordering to be selected. New polynomials are then iteratively generated and reduced until a suitable set of polynomials have been generated for solving. This solution is computed offline using random values selected from a prime field, and the series of steps is recorded. The discovered procedure can then be applied to the real data at test time. For more details on the Gröbner basis method we refer the reader to [22].

Another interesting formulation was proposed by Lepetit *et al*. [23], where the solutions were found algebraically without the use of Gröbner bases. Instead a barycentric parameterization was used, where 3D points are defined as a weighted combination of 4 control points, which can be automatically selected to ensure the problem is well conditioned similar to the normalization of the Direct Linear Transform (DLT) method [24]. Unfortunately, the barycentric representation requires different solutions depending on the rank of the null-space for the control point weightings. This null-space estimation is very sensitive to outliers, thus Ferraz *et al*. [25] proposed an extension to the technique with integrated outlier rejection, by forcing the control point assignment to always have rank 1.

As most of these algebraic techniques are based on polynomial equations, they generally result in a large number of roots, depending on the complexity of the representation employed. Some recent techniques report as many as 81 solutions [20] with the lowest reported as 16 [19]. The number of solutions obtained also depends on the configuration of the points (for example if they are planar or quasi-singular). Unfortunately, although some of these solutions can often be rejected (for example any complex roots), it is generally necessary to introduce an additional stage which evaluates the various roots according to one of the direct minimization cost functions, in order to select the best. It is also important to note that although the roots of these equations are guaranteed to be "optimal" in some sense, this generally does *not* mean they minimize the gold-standard reprojection error, or that they elegantly handle noise and outliers. For many applications, better solutions may still exist.

In the remainder of the paper we start by formalizing the PnP problem, and deriving a cost function based on a generalized form of the reprojection error in Section 3. Then in Section 4 we describe an efficient approximation scheme, which allows us to obtain the solution faster than most competing state-of-the-art approaches, including several non-iterative $O(n)$ techniques. An overparameterization scheme is discussed in Section 5, which combines several representations to improve robustness. In Section 6 we perform an extensive evaluation of the proposed technique, firstly we compare different variances in Section 6.1. We then compare against 9 state of the art approaches, including both direct minimization and algebraic techniques (Section 6.2). Finally we examine robustness to outliers in Sections 6.3. We then summarize the findings in Section 7.

3 Derivation of the general cost function

To formalize the PnP problem, we begin by assuming that a collection of n points from the world are observed. This collection is defined as $\mathbf{P} \in \mathbb{R}^{n\times 3}$ and the i-th point is defined as $\mathbf{p}_i \in \mathbb{R}^3$. The observations of a point are defined by the normalized observation ray $\mathbf{f}_i \in \mathbb{R}^3$ (also known as the bearing vector). Parameterizing the observations using normalized rays (in the euclidean space) instead of pixel coordinates (in the projective space), makes the system more flexible, and applicable to a wider range of optical systems such as spherical cameras [2].

Given these definitions, it follows that

$$\lambda_i \mathbf{f}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t}, \quad i \in \{1 \ldots n\}, \tag{1}$$

where λ_i is the unknown depth of point i and $\mathbf{R}$,$\mathbf{t}$ define the rotation and translation, respectively, from the world co-ordinate frame to the camera co-ordinate frame. The PnP problem is then to estimate the unknown $\lambda_{1\ldots n}$,$\mathbf{R}$ and $\mathbf{t}$ from the known $\mathbf{f}_{1\ldots n}$ and $\mathbf{p}_{1\ldots n}$.

One of the most important issues when deriving solutions to the PnP problem is the choice of parameterization for the camera pose. This is also one of the primary differences between many recent techniques. The choice of parameterization for the translation vector $\mathbf{t}$ is straightforward, however there are many possible parameterizations of the rotation matrix $\mathbf{R}$ which enforce the important properties $\det(\mathbf{R}) = 1$ and $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$, with each parameterization having different advantages. In theory $\mathbf{R}$ has 3 degrees

[1] It is important to note that this "hybrid" representation is still minimal, and is not an *overparameterization*, unlike the proposed approach.

of freedom, but most 3 element parameterizations (*e.g.* Euler angles, Cayley transform, rotation vector) suffer from instabilities and singularities. As such, over-parameterizations (*e.g.* angle-axis, rotation matrix, unit-quaternion and the recent non-unit-quaternion [20]) are often used to improve stability and generality, at the cost of requiring additional constraints and making convergence more challenging.

By defining a general function $\mathbf{R} = \text{Rot}_j(\mathcal{R}_j)$, to convert any rotation $(\mathcal{R}_j)$ from parameterization j into a rotation matrix representation $(\mathbf{R})$, the remainder of the paper is general enough to be compatible with most existing PnP formulations. The new general PnP formulation is

$$\lambda_i \mathbf{f}_i = \text{Rot}(\mathcal{R})\, \mathbf{p}_i + \mathbf{t}, \quad i \in \{1 \dots n\}. \tag{2}$$

Note that the depth λ_i is equal to the magnitude of $\mathbf{p}_i$ after transformation to the camera co-ordinate frame. We can substitute this into Equation 2 to remove the unknown λ and rewrite as an error function

$$\epsilon_i(\mathcal{R}, \mathbf{t}) = \left\| \frac{\text{Rot}(\mathcal{R})\, \mathbf{p}_i + \mathbf{t}}{\|\text{Rot}(\mathcal{R})\, \mathbf{p}_i + \mathbf{t}\|_2} - \mathbf{f}_i \right\|_2. \tag{3}$$

The total error for a particular solution is then

$$E(\mathcal{R}, \mathbf{t}) = \sum_{i=1}^{n} \epsilon_i(\mathcal{R}, \mathbf{t})^2. \tag{4}$$

This can be seen as a generalization of the reprojection error, which is regarded as the *gold standard* cost [9]. Note that this is the error measure visualized in Figure 1.

4 Efficient approximation

In order to efficiently minimize this error function, we first reformulate it into an iterative scheme where $\mathcal{R}^0$ and $\mathbf{t}^0$ are the initial solution and $\Delta\mathcal{R}$ and $\Delta\mathbf{t}$ are the estimated update to the solution. To obtain the initial solution we randomly select 3 of the points and corresponding observations. The minimal P3P solver [26] is then applied to obtain an initial pose $\mathcal{R}^0$ and $\mathbf{t}^0$. Note that applying P3P to a random set of points is an extremely weak initialization cue, primarily serving to set the correct order of magnitude for the translation. More accurate results would be obtained by initializing from a competing PnP algorithm (as in many previous works with a final "polishing" stage), however this would adversely affect the speed of the approach. This idea is examined experimentally in Section 6.2 and in the supplementary material.

We perform a Taylor series expansion around the position $\mathcal{R}^0$,$\mathbf{t}^0$ to get the cost function

$$\begin{aligned} E(\mathcal{R}^0 + \Delta\mathcal{R}, \mathbf{t}^0 + \Delta\mathbf{t}) = \sum_{i=1}^{n} \Bigg(& \epsilon_i(\mathcal{R}^0, \mathbf{t}^0) \\ & + \mathbf{J}_E \begin{bmatrix} \Delta\mathcal{R} \\ \Delta\mathbf{t} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta\mathcal{R} \\ \Delta\mathbf{t} \end{bmatrix}^\top \mathbf{H}_E \begin{bmatrix} \Delta\mathcal{R} \\ \Delta\mathbf{t} \end{bmatrix} \dots \Bigg)^2. \end{aligned} \tag{5}$$

where $\mathbf{J}_E$ is the Jacobian of the error function, and $\mathbf{H}_E$ is its Hessian. These derivatives obviously depend, in part, on the rotation parameterization used. However, for any particular representation, they can be computed in closed form. Due to the size of the resulting equations, please see the supplementary material for details.

From this expansion, we can then obtain an efficient approximation to the cost function by taking a subset of the terms. The number of higher order terms which are included determines the trade-off between computational complexity and approximation accuracy. However, the gain in accuracy from using a more complex approximation is often negligible compared to the gain from performing additional iterations. In contrast the decrease in computation time when using fewer terms is significant. As such, in this paper we make use of the first order approximation and linear solvers to estimate the update at each iteration:

$$[\Delta\mathcal{R}^*, \Delta\mathbf{t}^*] = \underset{[\Delta\mathcal{R}, \Delta\mathbf{t}]}{\arg\min} \sum_{i=1}^{n} \left(\epsilon_i(\mathcal{R}^0, \mathbf{t}^0) + \mathbf{J}_E \begin{bmatrix} \Delta\mathcal{R} \\ \Delta\mathbf{t} \end{bmatrix} \right)^2. \tag{6}$$

However, by maintaining second order terms the following formalization could exploit Hessian based solvers (see supplementary material for additional formalization of the error Hessian).

In practice we find a single iteration is often sufficient to obtain a reasonably accurate solution, and that convergence (up to numerical precision) occurs in 5 iterations.

5 Robust overparameterization

Iterative estimation schemes are typically susceptible to local minima and the quality of the result depends on the initialization. In addition, the low order Taylor approximation introduces some inaccuracy. However, we can mitigate these effects by fusing estimates from multiple parameterizations. This is because different types of inaccuracy and different sets of local minima are found, when performing the approximation in different parameter spaces.

We perform a joint optimization, where the cost functions relating to the different representations are combined within a single framework, and a solution is obtained to satisfy all representations simultaneously. It is important to note that this is not a simple "late fusion" scheme where the problem is solved independently in every parameterisation and the results fused (e.g. by conversion to a single reference representation followed by averaging). Such a scheme would have little effect on the frequency of local minima, which would still impact the fused result. Instead, the proposed approach unifies the different parameterisations *during* the optimization process. In this case, the process will not halt unless it has hit minima in all representations simultaneously. This intuition can be verified by examining Figure 1 and the supplementary code provided. To this end, we define the overparameterization $\mathcal{R}$ which includes m different representations,

$$\mathcal{R} = \{\mathcal{R}_j | j \in 1 \dots m\}. \tag{7}$$

In order to make use of this overparameterization, we define a general conversion function ${}_i\underrightarrow{\text{Rot}}_j(\mathcal{R}_i) = \mathcal{R}_j$ which generates a rotation in parameterization j from a rotation in parameterization i. We can then redefine Equation 2:

$$\lambda_i \mathbf{f}_i = \text{Rot}_j\left({}_1\underrightarrow{\text{Rot}}_j(\mathcal{R}_1)\right) \mathbf{p}_i + \mathbf{t}, \quad i \in \{1 \dots n\}. \tag{8}$$

Note that even in this "early fusion" approach, a reference representation $\mathcal{R}_1$ is still required. However, in this case it is embedded *within* the cost function, being converted to each representation $(\mathcal{R}_j)$ in turn and having the PnP problem parameterized in this new representation. Note that if $j = 1$

then ${}_1\underrightarrow{\mathrm{Rot}}_j$ is an identity transformation and thus Equation 8 is equivalent to Equation 2 in this case.

We can now repeat the previous approximation starting from Equation 8, to obtain a new cost based on the Hybrid Approximate Representation $\mathcal{R}$ (i.e. combining approximations in various representations within a single cost),

$$[\Delta\mathcal{R}^*, \Delta\mathbf{t}^*] = \underset{[\Delta\mathcal{R},\Delta\mathbf{t}]}{\arg\min} \sum_{\mathcal{R}_j \in \mathcal{R}} \sum_{i=1}^{n} \left(\mathbf{J}_{j\,1}\underrightarrow{\mathbf{J}}_j \begin{bmatrix} \Delta\mathcal{R} \\ \Delta\mathbf{t} \end{bmatrix} \right. \\ \left. + \epsilon_i \left(\mathrm{Rot}_j \left({}_1\underrightarrow{\mathrm{Rot}}_j \left(\mathcal{R}_1^0 \right) \right), \mathbf{t}^0 \right) \right)^2. \tag{9}$$

Following the total derivative chain rule, the Jacobian $\mathbf{J}_j$ of the error function in the representation j should be augmented by multiplication with the Jacobian ${}_1\underrightarrow{\mathbf{J}}_j$ relating to the derivatives of the composed function ${}_1\underrightarrow{\mathrm{Rot}}_j$. Once again, note that if $j = 1$ then ${}_1\underrightarrow{\mathbf{J}}_j = \mathbf{I}$ and this term of the cost function matches that of Equation 6.

Previous work has shown that some representations are in general more valuable for PnP problems. As such, it may be possible to introduce weightings for the various representations (examined in the supplementary material), or even to introduce a more intelligent fusion scheme which favors certain representations based on the situation. It is also trivial in our approach, to introduce weightings for the individual points, if confidences in the observations are available (*e.g.* a point matching score). However, we leave these ideas for future work and the results in the remainder of this paper use equal weightings.

At every iteration, we wish to solve Equation 9 to obtain the optimal value (in a least squares sense) of $[\Delta\mathcal{R}^*\Delta\mathbf{t}^*]$. Helpfully, $\mathcal{R}$ is naturally bounded, and we can introduce sufficiently large bounds on $\mathbf{t}$ to define a compact solution space. In the limit, bounds on $\mathbf{t}$ may be equal to the limits of numerical precision, and thus do not constrain the possible accuracy of the approach. We can therefore solve the PnP problem via Convex Combination Descent [13] in the Hybrid Approximate Representation space, which we term HAR-Descent. This relates to iteratively finding the solution within the compact subspace, which minimizes the least squares error of the first order Taylor-approximation (e.g. Eq. 9) with decreasing steps.

6 EVALUATION

We follow the evaluation protocol which has become standard in recent years [8], [19], [20], [23] for comparing various classes of PnP solver, including algebraic methods, direct minimization methods, and combinations of the two. The algorithm performance is evaluated with varying numbers of points, varying levels of input noise, and in varying configurations (general, planar and quasi-singular). The quasi-singular configuration is when the points are poorly scaled and near degenerate. This means that algorithms with special handling for the planar case may attempt to use the non-planar solution, and suffer from numerical instabilities. Good performance across all point configurations is desirable for a general PnP algorithm. For each test, a number of 3D points are randomly generated uniformly in the ranges

$$\mathbf{p} \in \begin{cases} [-2,2] \times [-2,2] \times [4,8] & \text{general config.} \\ [-2,2] \times [-2,2] \times [6] & \text{planar config.} \\ [1,2] \times [1,2] \times [4,8] & \text{quasi-singular} \end{cases} \tag{10}$$

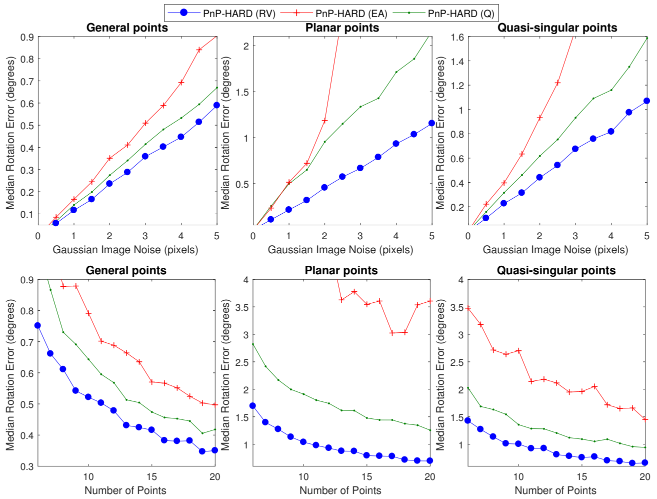


Fig. 2: Comparison of different variants of the **HARD-PnP** algorithm. **RV** = Rotation Vector, **EA** = Euler Axis+angle and **Q** = non-unit-quaternion. Top shows performance against observation noise (for 20 points). Bottom shows performance against the number of points (for 3px of noise).

These points are observed by a camera with a focal length of 800 pixels. The observations are then corrupted by Gaussian noise of a particular standard deviation. $\mathcal{R}$ and $\mathbf{t}$ are then estimated, and performance is measured by the orientation error (the maximum angle between any corresponding basis vectors from the estimated, and true camera orientation) in degrees. Each test is repeated 1000 times, and the median error is recorded. We also computed the translation error and reprojection error as specified in the benchmark. Note that the rotation and translation errors offer the fairest comparison against algebraic techniques. The final error measure is equivalent to Equation 4 and Figure 1 which the proposed technique attempts to optimize directly through its hybrid approximation. Regardless, the conclusions are similar for all three performance measures, and so for conciseness the latter 2 are relegated to the supplementary material.

For these tests, the Hybrid Approximate Representation is chosen as a combination of 3 different parameterizations, the minimal rotation vector **RV**, the Euler axis+angle **EA** and the non-unit-quaternion **Q** (used by Zheng *et al.* [20]).

6.1 Evaluation of proposed techniques

We first examine the effect of the reference representation on the proposed technique. All 3 variants of the technique include all 3 representations (**RV**, **EA** and **Q**), but each variant uses a different representation as the reference. As described in Section 5, the reference representation is the one which all equations are converted to and solved in (*i.e.* $\mathcal{R}_1$ in Equations 8 and 9). Results for this comparison are plotted in Figure 2. The top row shows the noise resilience of the algorithms, while the bottom row shows the performance against the number of points. From left to right the columns relate to the general, planar, and quasi-singular configurations, respectively.

For the **RV** variant even the worst performance (in the planar configuration with a noise level of 5 pixels) gives a median orientation error of slightly over 1 degree. The choice of reference representation has a clear affect on the performance (even though all variants include all 3 representations) confirming what has been found by previous work in the field. Using the 4 element representations (**EA** and **Q**) as a reference is less accurate than using the minimal **RV** representation in all cases. **EA** generally performs the worst. This agrees with the numbers of local minima found in the initial motivation for the paper (Figure 1).

For all representations, and in all point configurations, the accuracy of the technique with respect to the amount of observation noise (the top row of plots) is approximately linear. However, the impact of the noise (i.e. the slope of the trend) depends on the point configuration, with the same amount of noise causing roughly twice as much error in the planar case as in the general configuration. It is also interesting to note that as the noise level decreases, the performance of HAR using different reference representations converges.

6.2 Comparison to State of the Art

We now compare the proposed algorithm against many previous state-of-the-art techniques including direct minimization (which are closest to the proposed technique) and algebraic methods. We follow the protocol of [19], comparing against a total of nine other approaches in their full "as released" form. Note that five of these techniques include both an algebraic and an iterative stage.

- **LHM** The current state-of-the-art iterative minimization technique of Lu *et al.* [11]. In the planar case, the planar variant **SP+LHM** of Schweighofer and Pinz [12] is used.
- **EPnP+GN** The non-iterative $O(n)$ approach of Lepetit *et al.* [23] followed by Gauss-Newton minimization of the solution (non-planar tests only).
- **DLT** The classic direct linear transform method [24] (non-planar tests only).
- **HOMO** The homography method of Malik *et al.* [27] (only for planar tests).
- **RPNP** The $O(n)$ solution of Li *et al.* [8], designed to be robust to planar and quasi-singular configurations, using an algebraic approach followed by a minimization step.
- **DLS$_{+++}$** The non-degenerate version of the Direct Least Squares technique of Hesch *et al.* [16] (one of the few algebraic techniques with no following minimization).
- **SOS** The Sum-Of-Squares technique solved via semidefinite programming by Schweighofer and Pinz [14].
- **OPnP** The recent $O(n)$ solver of Zheng *et al.* [20] using non-unit-quaternions and including a polishing stage.
- **UPnP** The Unified PnP approach of Kneip *et al.* [19], in its central PnP mode, with a final minimization stage.
- **REPPnP** The approach of Ferraz *et al.* [25] with integrated outlier detection.

Note that the primary comparison of our proposed technique is against **LHM** which is generally considered to be the state of the art iterative technique. Most other state of the art techniques in this comparison calculate a set of solutions algebraically, and perform a smaller amount of minimization or "polishing", in order to achieve accuracy comparable to **LHM** but with less computational overhead. It is also interesting to note that due to the representation used in **DLS**, the initial release suffered from a degeneracy in the case of 180 degree rotations around any of the 3 axes, with significantly decreased accuracy when the pose approaches these configurations (this was solved in a later release by running the algorithm multiple times). This is also part of the reasoning behind the move to a quaternion representation in **OPnP**. This provides further motivation for our HAR overparameterization, where issues with any one parameterization are balanced out by the other parameterizations.

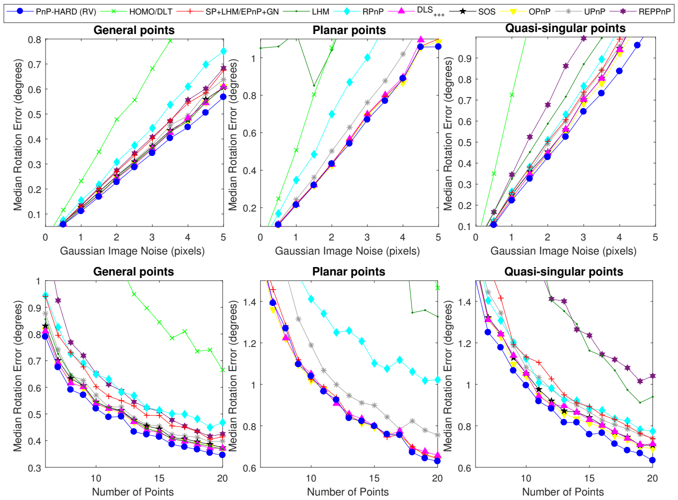


Fig. 3: Comparison of **HARD-PnP** against previous SOTA.

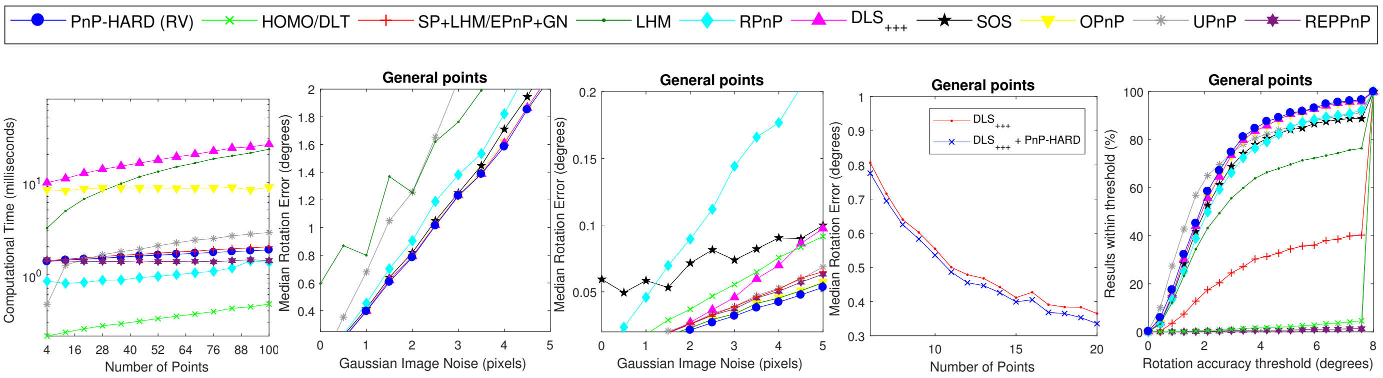
The evaluation is shown in Figure 3. As in Figure 2, the columns relate to the general, planar and quasi-singular configurations respectively, while the top row examines noise resilience and the bottom row varies the number of points.

The **HARD-PnP** algorithm compares very favorably against state-of-the-art and is the most accurate out of the ten techniques, at every point in all the graphs (i.e. it has the lowest error for all numbers of points and noise levels, in all three point configurations) apart from a brief region of the bottom-middle plot (the planar configuration). We also note that as the image noise approaches zero, most techniques are able to reliably recover the ground truth pose (with a few exceptions for challenging point configurations). This indicates that with perfect observations, minimization techniques such as **HARD-PnP** are only marginally disadvantaged by the lack of global optimality guarantees, which algebraic techniques can provide.

When compared to the previous state-of-the-art minimization technique (**LHM**), **HARD-PnP** is slightly more accurate in the general and planar configurations. However, **LHM** is unreliable in the quasi-singular configuration as it uses a separate technique in the planar case. In contrast **HARD-PnP** performs equally well for this configuration.

We also see that the proposed technique consistently outperforms algebraic approaches such as **OPnP** and **DLS**, despite their theoretical optimality guarantees. This indicates better robustness to measurement noise.

In Figure 4 we perform a comparison of runtimes (Figure 4a), and also of the accuracy with both extremely small and large numbers of points (figures 4b and 4c). Speed tests for **HARD-PnP** were performed in Matlab using a single thread at 2.4 GHz. The runtime graph indicates that the complexity of the **HARD-PnP** algorithm compared to the number of points is drastically improved compared to the previous state-of-the-art minimization technique **LHM**. Indeed, complexity is comparable to the $O(n)$ algebraic solutions such as **EPnP**, **RPnP** and **UPnP** (and significantly better than **DLS** which is also $O(n)$). Note that **SOS** cannot be seen on



(a) Runtimes (b) 4 points (c) 2000 points (d) Polishing (e) Error distribution

Fig. 4: Additional evaluation of the proposed HARD-PnP technique, including runtime comparisons, examining performance with very few and very many points, and combining the proposed technique with existing state-of-the-art.

the plot, but has average runtimes of around 250 ms.

With extremely low numbers of points, the ranking of the algorithms changes significantly (although **HARD-PnP** is still the top performing algorithm). **DLT**, **EPnP** and **REPPnP** are no longer visible on the range of competitive plots. Additionally **UPnP** and **LHM** (the previously state of the art minimization technique) are no longer competitive with the best techniques. The difference in accuracy between **HARD-PnP**, **OPnP** and **DLS** is negligible with only 4 points, although as already mentioned **HARD-PnP** is significantly faster.

All techniques perform well with extremely large numbers of points (note the largest error is around 0.2°), the ranking of the algorithms again changes, but again **HARD-PnP** is the most accurate. The robust **RPnP** appears to be unable to exploit the additional information and actually performs worse than all other techniques including the **DLT** baseline. This is likely due to a limitation of their approximate cost function. In contrast, our Hybrid Approximate Representation does not suffer from this limitation.

In addition, as **HARD-PnP** is a minimization approach, it can be used as an alternative "polishing" step for existing algebraic PnP techniques. In Figure 4d we compare the most accurate competing technique (**DLS**) in its standard form, and when using **HARD-PnP** refinement. The refinement provides a consistent 5 % reduction in errors. Even larger gains (up to 50 %) are seen when combining **HARD-PnP** with other techniques (see the supplementary material). This experiment demonstrates that the proposed technique is extremely robust to it's initialization. The accuracy is similar when initialized randomly, or using a state-of-the-art algebraic technique (however a good initialization likely reduces the number of iterations necessary to converge).

We next examine in greater detail the distribution of performance for one of the data points from the previous experiments. We selected the most challenging experimental setup for exploration; the 4 point experiment with a noise level of 5. Rather than simply displaying the mean or median of the errors, Figure 4e displays a cumulative histogram of the errors (i.e. how frequently each technique achieved a result within a particular threshold of the ground truth). This is useful for exploring the frequency of local minima or suboptimal solutions. UPnP is able to most frequently exceed very tight success thresholds (<3 degrees), with the proposed technique having the second best success rates. At looser success thresholds the proposed technique overtakes **UPnP**, being able to get within 5 degrees of the true solution in 88% of cases compared to 84%. This indicates that the proposed technique suffers fewer catastrophic failures than **UPnP** (i.e. it gets stuck in distant local minima less often), while the solutions it finds are also generally more accurate than all other approaches.

6.3 Performance with outliers

Although this is a widely used standard benchmark, it has one significant drawback. Noise is assumed to be Gaussian distributed (i.e. caused by localization errors) with none of the outliers due to incorrect correspondences, which are ubiquitous in real PnP applications. Traditionally a RANSAC [1] framework is used to deal with outliers, hence the focus on "inlier performance" in the standard benchmark. However, it's still interesting to examine how algorithms behave in a more realistic setting. This is particularly true as recent techniques such as **REPPnP** [25] have been developed to handle outlier rejection internally.

For this experiment we follow the protocol of [25]. As in the previous experiment, data is generated in 3 different configurations, including Gaussian noise with a standard deviation of 3 pixels. Additionally, a varying number of outliers are generated by duplicating random 3D points and observations, creating invalid correspondences. As in [25] we compare against various combinations of "minimal" and "non-minimal" techniques, however we follow a slightly different approach which better exploits the non-minimal solvers. The previous benchmark employed a two stage process, where RANSAC was first run using the minimal solver, and the non-minimal solver was then run on the result of RANSAC. Instead we use a Locally-Optimized-RANSAC framework [28]. In brief, every time a new optimal solution is found by the minimal solver, the solution is iteratively refined using the non-minimal solver on the inlier set, with decaying inlier thresholds. The primary advantage for this experiment is that the non-minimal solvers are exploited to much greater effect, and their behaviors can be more easily analyzed. For further details we refer the reader to [28], but we should point out that LO-RANSAC is often faster than standard RANSAC (despite repeated calls to the

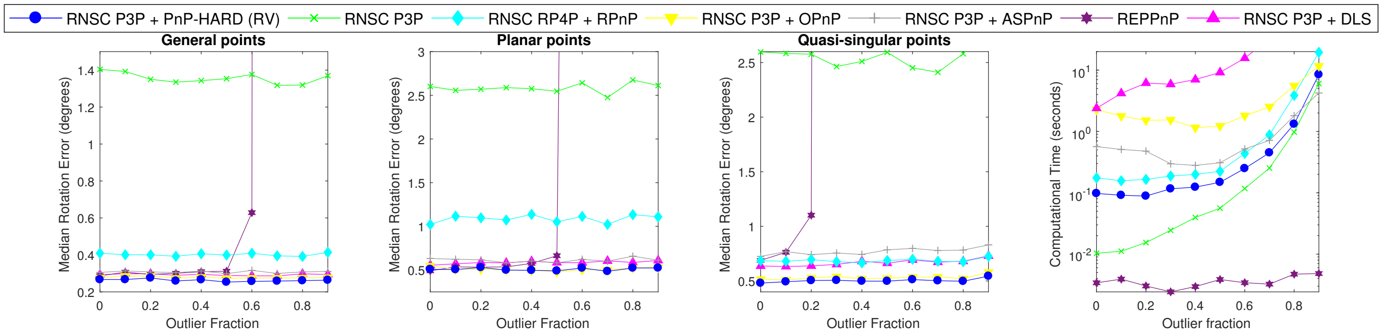


Fig. 5: Comparison of the proposed **HARD-PnP** algorithm against the previous state-of-the-art, with varying numbers of outliers (100 inliers and noise sigma 5). The first 3 columns show performance in the general, planar and quasi-singular configurations respectively. Right compares runtimes.

non-minimal solver) because it generally finds larger inlier sets (and thus terminates after fewer iterations).

Note that we evaluate only the subset of the techniques above which were included in the recent [25] benchmark. The benchmark omits some techniques with higher complexity, which scale poorly to the large numbers of points involved in the experiments. The combinations of minimal/non-minimal techniques evaluated are: **RNSC P3P** using the minimal sampling of [26] without a non-minimal solver. **RNSC P3P/OPnP** including OPnP [20] as the non-minimal solver. **RNSC P3P/ASPnP** including ASPnP [29] as the non-minimal solver. **RNSC P3P/DLS**$_{+++}$ including the non-degenerate DLS variant [16] as a non-minimal solver. **RNSC RP4P/RPnP** using RPnP [8] as both the minimal and non-minimal technique (unlike other techniques, a minimal sample of 4 is required here). **REPPnP** using the technique of [25] which handles outliers, with no minimal solver or RANSAC. **RNSC P3P/HARD-PnP (RV)** using the proposed algorithm as the non-minimal solver.

In Figure 5 the performance is plotted for various levels of outlier contamination. In every case there were 100 inliers as in [25] (so 10 % outliers corresponds to 110 total points, and 90 % outliers corresponds to 1000 total points). The runtime plot shows that **REPPnP** is the fastest approach and because it does not require RANSAC the runtime does not change with the number of outliers. However, we also see that in the general point configuration, the **REPPnP** technique breaks down when the number of outliers is greater than the number of inliers (i.e. when the outlier fraction exceeds 0.5) while the other RANSAC based techniques provide consistent accuracy all the way up to 90 % outlier contamination. This breakdown point agrees with the findings in [25], however we also examine the performance for points in the planar and quasi-singular configurations. This has little effect on RANSAC based methods, but causes **REPPnP** to break down as early as 20 % outlier contamination.

In terms of runtime, most RANSAC techniques behave similarly. As mentioned previously, the LO-RANSAC techniques which repeatedly execute the non-minimal solver are still able to achieve similar runtimes to pure P3P RANSAC (but with greatly improved accuracy) as they can terminate earlier. However, the **RPnP** RANSAC scales poorly at the higher outlier ratios; it requires a larger minimal sample of 4 which greatly increases the number of RANSAC iterations required for a pure sample. **HARD-PnP** RANSAC proves to be the most accurate approach, followed by **OPnP** RANSAC, however there appears to be significant overhead in this technique causing runtimes significantly slower than any other approach except **DLS** when the outlier fraction is less than 0.8.

In addition to these experiments, all the tests from the previous sections (i.e. evaluation against noise level, number of points and different variants of **HARD-PnP**) are repeated in the presence of outliers in the supplementary material.

6.4 Evaluation on real data

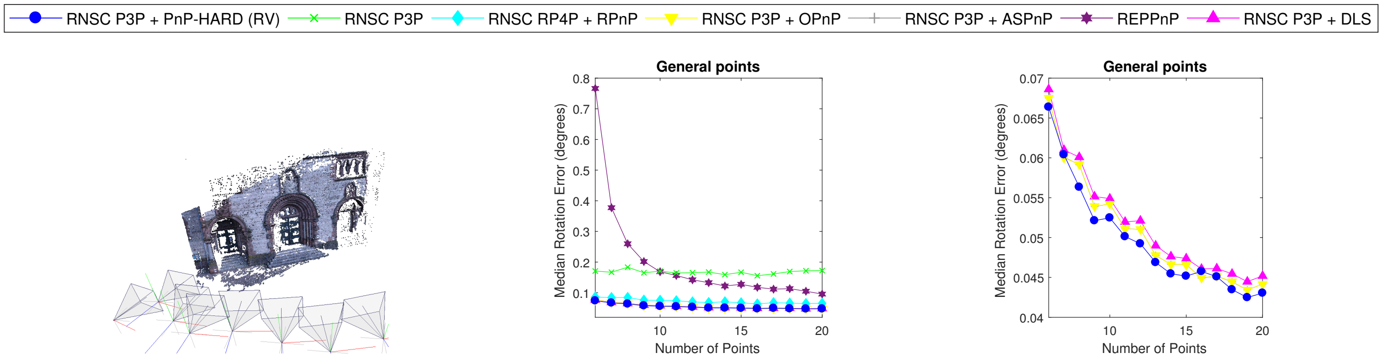
Finally, we take this realistic evaluation a step further. In the supplementary material we perform a qualitative examination of results obtained using real data obtained via SIFT point matching between images (including match outliers and feature localization noise) following [20] and [25]. In Figure 6 we present a similar quantitative evaluation, following the protocol of Garro *et al.* [15]. We first perform multiview stereo reconstruction [30], [31] on the entire Herz-Jesu-P8 dataset[2] (shown in Figure 6a). We then take random subsets of the reconstructed 3D point cloud, and the corresponding 2D feature detections from a single input image. These noisy 2D-3D correspondences are then provided to the various PnP techniques from the previous section, and the accuracy of the estimated camera pose is examined.

Clearly the trivial P3P technique performs poorly, and REPPnP has difficulty when the number of points is very low. The other techniques are able to achieve excellent accuracy, with median orientation errors less than a tenth of a degree even on realistic data. As shown in the zoomed subplot, the proposed technique has the best performance overall, particularly with smaller numbers of points available. OPnP comes a close second in terms of accuracy. However, as highlighted in Figures 4 and 5, OPnP is orders of magnitude slower than the proposed technique.

7 Conclusions

From these results we can conclude that using an overparameterized representation, such as our HAR, during PnP can greatly improve accuracy and robustness to noise. Our hybrid representation outperforms all 9 state of the art techniques in a huge range of experiments over 3 different types of point configuration. We have also shown that our HARD-PnP efficient approximation scheme is extremely robust to planar and near-planar point

[2] http://cvlabwww.epfl.ch/data/multiview/denseMVS.html



(a) Reconstruction (b) Overall view (c) Detailed view

Fig. 6: The reconstructed Herz-Jesu-P8 dataset (left) and two views (overall and zoomed in) of the the accuracy of different PnP techniques using different sized subsets of the data.

configurations. The approximation scheme, in conjunction with the convex combination descent solver, also provides runtimes which are up to 10 times faster than the current state-of-the-art minimization technique, and is even comparable to several recent $O(n)$ techniques.

Interestingly, the non-unit-quaternion representation (which has recently become popular in the field) performed significantly worse as a reference representation than the minimal rotation vector representation. This implies that the requirements for a good *reference* parameterization are different to the requirements for a good parameterization.

In the future, it would be interesting to investigate techniques to combine the Hybrid Approximate Representation with global direct solvers (such as the semidefinite programming of [14]). It is also likely that overparameterized representations may be useful within algebraic (rather than minimization based) PnP algorithms, or even in other areas of multi-view geometry. With additional derivation, the proposed technique could also be extended to Hessian based optimization schemes.

Acknowledgments: Supported by EPSRC project "Learning to recognise dynamic visual content" (EP/I011811/1) and the SNSF grant CRSII2 160811 "SMILE".

References

[1] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.

[2] L. Kneip, P. Furgale, and R. Siegwart, "Using multi-camera systems in robotics: Efficient solutions to the NPnP problem," in *ICRA*. IEEE, 2013, pp. 3770–3776.

[3] W. E. L. Grimson, G. Ettinger, S. J. White, T. Lozano-Perez, W. Wells Iii, and R. Kikinis, "An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization," *Medical Imaging, IEEE Transactions on*, 1996.

[4] G. Hirota, D. T. Chen, W. F. Garrett, M. A. Livingston *et al.*, "Superior augmented reality registration by integrating landmark tracking and magnetic tracking," in *Proceedings of the conference on Computer graphics and interactive techniques*, 1996.

[5] K. Lebeda, S. Hadfield, and R. Bowden, "2D or not 2D: Bridging the gap between tracking and structure from motion," in *Proc. ACCV*, ser. LNCS, vol. 9006. Singapore: Springer International Publishing, Nov. 1–5 2014, pp. 642–658.

[6] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *Proc. ECCV*. Heraklion, Crete: Springer, Sep. 5 – 11 2010, pp. 29–42.

[7] S. Hadfield, K. Lebeda, and R. Bowden, "Natural action recognition using invariant 3D motion encoding," in *Proc. ECCV*, Zurich, Switzerland, Sep. 6 – 13 2014, pp. 758 – 771.

[8] S. Li, C. Xu, and M. Xie, "A robust $O(n)$ solution to the perspective-n-point problem," *PAMI*, vol. 34, no. 7, July 2012.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University press, 2000.

[10] C. Olsson, F. Kahl, and M. Oskarsson, "Branch-and-bound methods for Euclidean registration problems," *PAMI*, vol. 31, 2009.

[11] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *PAMI*, vol. 22, 2000.

[12] G. Schweighofer and A. Pinz, "Robust pose estimation from a planar target," *PAMI*, vol. 28, no. 12, pp. 2024–2030, 2006.

[13] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 427–435.

[14] G. Schweighofer and A. Pinz, "Globally optimal O(n) solution to the PnP problem for general camera models." in *Proc. BMVC*, Leeds, UK, Sep. 1 – 4 2008, pp. 1–10.

[15] V. Garro, F. Crosilla, and A. Fusiello, "Solving the PnP problem with anisotropic orthogonal procrustes," in *3DIMPVT*, 2012.

[16] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (DLS) method for PnP," in *Proc. ICCV*, Nov. 6 – 13 2011.

[17] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in *Proc. ECCV*. Marseille, France: Springer, Oct. 12 – 18 2008, pp. 302–315.

[18] H. Stewenius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–294, 2006.

[19] L. Kneip, H. Li, and Y. Seo, "UPnP: An optimal $O(n)$ solution to the absolute pose problem with universal applicability," in *Proc. ECCV*. Zurich, Switzerland: Springer, Sep. 6 – 13 2014.

[20] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the PnP problem: A fast, general and optimal solution," in *Proc. ICCV*. Sydney, Australia: IEEE, Dec. 3 – 6 2013.

[21] C. Wu, "P3.5P: Pose estimation with unknown focal length," in *Proc. CVPR*, Boston, USA, Jun. 8 – 10 2015.

[22] D. A. Cox, J. Little, and D. Oshea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2007.

[23] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate $O(n)$ solution to the PnP problem," *IJCV*, vol. 81, no. 2, 2009.

[24] Y. Abdel-Aziz and H. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry," in *Proceedings of the Symposium on Close-Range Photogrammetry*, 1971, pp. 1–18.

[25] L. Ferraz, X. Binefa, and F. Moreno-Noguer, "Very fast solution to the PnP problem with algebraic outlier rejection," in *Proc. CVPR*, Columbus, USA, Jun. 24 – 27 2014, pp. 501–508.

[26] 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 *Proc. CVPR*. IEEE, 2011, pp. 2969–2976.

[27] S. Malik, G. Roth, and C. McDonald, "Robust corner tracking for real-time augmented reality," in *Proc. Conf. Vision Interface*, 2002.

[28] K. Lebeda, J. Matas, and O. Chum, "Fixing the locally optimized RANSAC," in *Proc. BMVC*, Surrey, UK, Sep. 3 – 7 2012.

[29] Y. Zheng, S. Sugimoto, and M. Okutomi, "ASPnP: An accurate and scalable solution to the perspective-n-point problem," *Transactions on Information and Systems*, vol. 96, no. 7, pp. 1525–1535, 2013.

[30] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Proc. CVPR*. IEEE, 2011.

[31] C. Wu, "Towards linear-time incremental structure from motion," in *3D Vision*. IEEE, 2013.