

REAL-TIME CAMERA POSE ESTIMATION USING CORRESPONDENCES WITH HIGH OUTLIER RATIOS

Solving the Perspective N-Point Problem using Prior Probability

Tobias Nöll, Alain Pagani, Didier Stricker

Augmented Vision, DFKI, Trippstadterstr. 122, D-67663 Kaiserslautern, Germany

Tobias.Noell@dfki.de, Alain.Pagani@dfki.de, Didier.Stricker@dfki.de

Keywords: Real-time camera pose estimation, low quality correspondences.

Abstract: We present PPnP, an algorithm capable of estimating a robust camera pose in real-time, even if being provided with large sets of correspondences containing high ratios of outliers. For these situations, standard pose estimation algorithms using RANSAC are often unable to provide a solution or at least not in the required time frame. PPnP is provided with a probability distribution function which describes all valid possible camera pose estimates. By checking the correspondences for being compatible with the prior probability, it can be decided effectively at a very early stage, which correspondences can be treated as outliers. This allows a considerably more effective selection of hypothetical inliers than in RANSAC. Although PPnP is based on a technique called BlindPnP which is not intended for real-time computing, a number of changes in PPnP allows to estimate a camera pose with the same high quality as BlindPnP while being considerably faster.

1 INTRODUCTION

In this paper, we address the problem of camera pose estimation from correspondences. Our goal is to find a solution for the Perspective n-Point (PnP) problem for correspondences with high ratios of outliers in real-time.

Usually the camera pose for a given image is estimated solely using a set of n correspondences. A large amount of different markerless pose estimation algorithms already exists. Those algorithms presented in (Dhome et al., 1989), (Fischler and Bolles, 1981), (Gao et al., 2003), (Haralick et al., 1994), (Quan and Lan, 1999) typically search for the roots of an eight-degree polynomial with no odd terms. Their complexity varies from $O(n^2)$ to even $O(n^8)$. In (Lepetit et al., 2009) a method called EPnP (Efficient Perspective n-Point Camera Pose Estimation) is proposed which allows the computation of an accurate and unique solution in $O(n)$ for $n \geq 4$.

In practice, the estimation of the camera pose is problematic if solely relying on the correspondences: Often correspondences are established automatically using feature detectors and a certain amount will be

misleadingly classified (outliers). In order to identify and exclude the outliers, stochastic approaches such as RANSAC (Fischler and Bolles, 1981) have been developed. However RANSAC tends to fail or needs an unacceptable large iteration count especially if the outlier ratio grows.

In (Moreno-Noguer et al., 2008) a method to compute the camera pose from correspondences called BlindPnP is developed which integrates additional information beside the correspondences. BlindPnP assumes that only two sets of 3D points and 2D points are given without any correspondences. As additional input, BlindPnP uses a probability distribution regarding the final camera pose estimate (pose prior probability). This prior probability is then used to establish a camera pose and the corresponding correspondences in parallel.

BlindPnP delivers good results even if no correspondences are given at all. However due to its slow runtime performance, BlindPnP is not applicable in real-time reactive systems. The authors mentioned, that BlindPnP can be modified for using correspondences without providing results of this modification. We called this modified version BlindPnPc (BlindPnP with correspondences) and implemented and

evaluated it in order to check whether it is applicable for real-time camera pose estimation. We will show, that BlindPnP delivers high quality solutions, even if being provided with low quality correspondences. However the experiments also showed, that the runtime of BlindPnP highly depends on n . As n grows, BlindPnP is not able to provide solutions in real-time. To also provide solutions in real-time for large values of n , we developed a new algorithm called PPnP (Prior probability Perspective **n**-Point Camera Pose Estimation) based upon BlindPnP. We will show, that PPnP is capable of estimating a robust camera pose even though being provided with sets of noisy correspondences having high outlier ratios. We will also show, that PPnP reaches both higher precision and speed than the comparable conventional RANSAC+EPnP method as well as the prior probability based BlindPnP method.

The remainder of this paper is organized as follows: Section 2 summarizes the underlying theory used for BlindPnP. Section 3 covers the concept and implementation of PPnP. In section 4 we provide quality and performance analyses of both algorithms using synthetic and real data scenarios. We finally conclude in section 5 with an outlook to future work.

2 PNP WITH PRIOR INFORMATION

PPnP has similar concepts as BlindPnP. This algorithm is explained in detail in (Moreno-Noguer et al., 2008). The underlying theory is summarized in this section.

Let C be the set of correspondences, containing a ratio of λ outliers. Our aim is to find the true camera pose $P = [R \mid \mathbf{t}]$ as well as the set of inliers. Each camera pose Q can also be parametrized as a 6 dimensional vector \mathbf{x}_Q . Let V be the pose prior probability which describes all valid parameterizations of possible solutions Q . V is modeled using a Gaussian mixture model with a number of g Gaussian components. Each of the Gaussian components consists of a mean value $\mathbf{x}_Q \in \mathbb{R}^6$ along with a covariance matrix $\Sigma_Q \in \mathbb{R}^{6 \times 6}$. Figure 1 gives an example of a possible modeling. For simplicity, only the translation uncertainty is visualized (green ellipsoids). The mean values \mathbf{x}_Q hereby specify the position, the covariances Σ_Q the shape of the ellipsoids. A real pose prior probability would normally consist of a 6D covariance.

Let $M_Q \subseteq \{\mathbf{P}_i \leftrightarrow \mathbf{p}_i \mid i \in \{1, \dots, n\}\}$ be the set of pairs that match under the assumption that Q is the correct pose. Let additionally $F_Q = C \setminus M_Q$ be



Figure 1: Modeling the camera pose prior probability by mixtures of Gaussians.

the set of correspondences for which no match can be established hypothesizing pose Q . The correct pose is found by minimizing the error function E

$$E(\mathbf{x}_Q) \stackrel{\text{def}}{=} \sum_{\mathbf{P}_i \leftrightarrow \mathbf{p}_i \in M_Q} \|\mathbf{p}_i - Proj_{\mathbf{x}_Q}(\mathbf{P}_i)\| + \theta |F_Q| \quad (1)$$

with $Proj_{\mathbf{x}_Q}(\mathbf{P}_i)$ defined as the projection of \mathbf{P}_i on the image using pose \mathbf{x}_Q and $\theta \in \mathbb{R}$ as a penalty term that penalizes unmatched points. The minimization is computed by utilizing the prior probability.

Roughly summarized, in each iteration BlindPnP hypothesizes consecutively three 3D to 2D point correspondences which are compatible with the prior probability. Since the camera pose \mathbf{x}_Q has an uncertain position indicated by the covariance Σ_Q , there exists also an uncertainty Σ_Q^i regarding the position of the projected points which can be calculated by error propagation using the Jacobian J_Q^i of $Proj_{\mathbf{x}_Q}(\mathbf{P}_i)$. A correspondence is marked as compatible with the prior probability if its projected 3D point lies within that uncertainty Σ_Q^i around \mathbf{p}_i . Hypothesizing of one correspondence is realized using a Kalman filter. Hereby the 6 dimensional pose parameter is interpreted as the system state and each correspondence is interpreted as a measurement. During this process, the camera pose \mathbf{x}_Q evolves and the assigned covariance Σ_Q (i.e. uncertainty) reduces. After three correspondences are hypothesized, the remaining correspondences can be checked for validity using this camera pose by projecting the 3D points \mathbf{P}_i on the image and checking the distance to their corresponding 2D point \mathbf{p}_i . This way the sets M_Q and F_Q can be constructed. The pose with the least error function value $E(\mathbf{x}_Q)$ is then chosen as result.

Discussion: We ran a number of experiments with different ratios of outliers and compared BlindPnP to usual pose estimation approaches. As a representative common pose estimation approach

we chose RANSAC combined with EPnP (RANSAC+EPnP). The experiments conducted showed, that by effective usage of the prior probability, BlindPnP estimates high quality results, mostly independent of the outlier ratio. However for low outlier ratios, RANSAC+EPnP outruns BlindPnP in precision. Additionally, the runtime of BlindPnP grows very fast with the number n of correspondences used. This prohibits an application in real-time reactive systems. We present the results of these experiments in detail in section 4.

3 NEW APPROACH

To overcome the problems of BlindPnP, we developed an algorithm called PPnP which utilizes the prior probability similar to BlindPnP in order to effectively reduce the search space of the correspondence problem.

3.1 Idea

Similar to BlindPnP, in PPnP consecutively a number of correspondences which are valid with respect to the prior probability are hypothesized using a Kalman filter. Thereby the camera pose evolves from its initial position.

BlindPnP: Problematic in BlindPnP is that each consecutively selected correspondence has to be valid in order to converge towards the real pose. If once in BlindPnP a wrong hypothesis is made, the pose evolves in a bad way because future hypotheses chosen will be outliers with a higher probability.

Hypothesizing an outlier will always badly affect the current pose estimation Q . When projecting the 3D points \mathbf{P}_i and constructing the image projection covariance Σ_Q^i , many inliers previously correctly classified will now be marked as outliers and thereby not be considered in the next selection process. Additionally, by badly evolving the pose, outliers can now become compatible with the current pose prior probability and are therefore treated as hypothetical inliers. Combined, these effects lead to an increased outlier ratio for the hypothesizing possibilities in the next selection process. Thereby an outlier is also chosen in the next selection process with a higher probability, evolving Q even worse. BlindPnP tries to solve this issue by recursively hypothesizing all arguable sequences of compatible correspondences containing only three elements and selecting the one with the least error function value. Thus a very large number of consecutive hypotheses has to be made.

PPnP: PPnP tries to solve this issue by using a different approach: Consecutively a number of c correspondences are hypothesized. Similar to BlindPnP, hypothesizing a correspondence $\mathbf{P}_i \longleftrightarrow \mathbf{p}_i$ is realized using a Kalman filter. Different than in BlindPnP, c is usually a number much higher than three. While in BlindPnP the whole sequence of hypotheses has to be free of outliers, this is not a necessary condition for PPnP: At each step, all hypothesizing possibilities are stored for future use along with the uncertainty information Σ_Q^i and J_Q^i . When it comes to selecting a new hypothesizing candidate, it is randomly selected from all available hypothesizing possibilities (containing also the ones not hypothesized in the past). The key point is, that once an outlier is hypothesized, the number m_i of compatible candidates when just considering the actual pose probability at hypothesizing step i is relatively small compared to the number of all hypothesizing possibilities from the previous steps $m_{old} \stackrel{\text{def}}{=} \sum_{j=0}^{i-1} m_j$. Since in the past a sequence of correct hypotheses was made, the majority of all previous hypothesizing possibilities will contain correctly identified inliers. Since the new hypothesis is randomly chosen among all those $m_{new} \stackrel{\text{def}}{=} m_{old} + m_i$ possibilities with approximately m_{old} correct hypothesizing possibilities and only approximately m_i outliers misleadingly classified as inliers, a correct correspondence is selected with a relatively high probability. Hence, if an outlier is hypothesized at step i , PPnP selects with a high probability an inlier for the next candidate and thereby pushes the wrongly evolved pose back to a valid state.

To gain a similar precision as RANSAC+EPnP, the final camera pose estimate is only used in order to classify M_Q and F_Q . M_Q is then used in order to calculate a high precision solution using EPnP.

Combined, this allows evolving the pose from a relatively small fixed number of hypothesizing sequences containing c correspondences, instead of considering each permutation combination of three correspondences.

3.2 Optimization

Accelerating the hypothesizing process: Before a correspondence from all available possibilities is randomly selected for hypothesizing, all correspondences are checked for validity with the current pose prior probability. The information of each compatible correspondence is then added to the set of selectable options. If the number of correspondences grows, the procedure of testing each correspondence at each step for validity may lead to large overhead. Fortunately this procedure can be optimized: A correspondence

being invalid with the pose prior probability at step i is unlikely to become valid in later steps $i+1, i+2, \dots$ because the overall reprojection uncertainty reduces. This way, one can skip the successive testing for validity of a certain correspondence, once it has been declared as invalid. Also it became apparent that the correspondences chosen for hypothesizing at step i need not to be tested for validity again at later steps. Thereby the computation can be eased by keeping an exclusion list containing the correspondences which will not be checked for validity with the pose prior probability in later steps. The exclusion list contains the correspondences already hypothesized or once marked as invalid.

Optimizing the hypothesis picking process:

The more information from steps far beyond the current step is hypothesized, the slower the pose will converge to its correct position. In order to select the hypothesizing possibilities with the right balance between previous and current possibilities, all available possibilities are kept in a list, linearly ordered according to their degree of uncertainty (i.e. hypothesizing possibilities appearing at later steps are pushed at the end of the list). If a correspondence is randomly chosen from that list, this is not realized uniformly but with a probability linearly increasing towards the end of the list.

4 EXPERIMENTS

4.1 Synthetic Test Setting

In order to compare the different approaches, the algorithms are evaluated with respect to quality ϵ and runtime performance μ . ϵ is measured in terms of the mean reprojection error in pixel. μ is simply the time in milliseconds it needs to find any solution. In our scenario we assumed that the camera was located somewhere inside a torus around the object in focus and approximately directed towards it. The diameter of the torus hereby defines the degree of uncertainty with respect to the cameras' position. This scenario was modeled using a Gaussian mixture model of $g = 20$ Gaussian components. We then constructed a set C of n correspondences having an outlier ratio of λ . We added normal distributed values up to 5 pixel to the 2D points in order to simulate noise.

4.1.1 Results

Due to the large number of required hypothesizing combinations, BlindPnPC suffers from a high run-

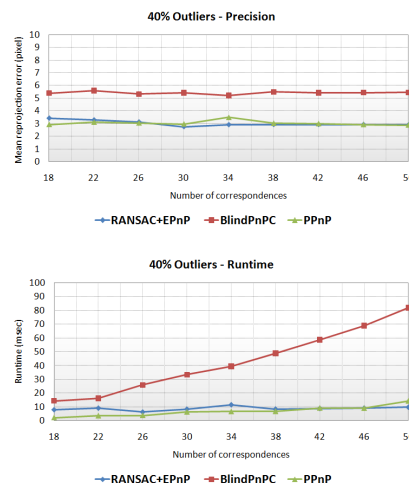


Figure 2: 40% outlier ratio measurement results.

time, allowing a profitable use of the pose prior probability only for outlier ratios of 60% and above. If an outlier ratio of 80% and above is reached, BlindPnPC and PPnP both are the only algorithms evaluated which still are able to estimate a robust camera pose. For these large outlier ratios the prior probability seems to be crucial in order to estimate a reliable camera pose.

For low outlier ratios, BlindPnPC consequentially returns camera pose estimations with larger reprojection errors than the corresponding estimations of RANSAC+EPnP. This is related to the fact, that only three Kalman filter iterations are applied to evolve the pose. However this unnecessary error is still tolerable since it could be easily decreased by numerical optimization techniques without significantly increasing the runtime.

PPnP uses EPnP in order to gain results of similar precision as RANSAC+EPnP for low outlier ratios. Additionally, PPnP is implemented in an iterative way, runs with a fixed number of iterations and is thereby considerably faster than BlindPnPC. This way the runtime can be lowered to a level which allows an efficient usage of pose prior probability for outlier ratios of 40% and above.

The measurements taken in the experiment are displayed in figures 2, 3 and 4.

4.2 Real Data Test Setting

A real data test setting was constructed with the intention to acknowledge the results gained in the synthetic test setting:

Scenario: The scene filmed by the camera (640×480 , 30fps) was a desk. On the desk

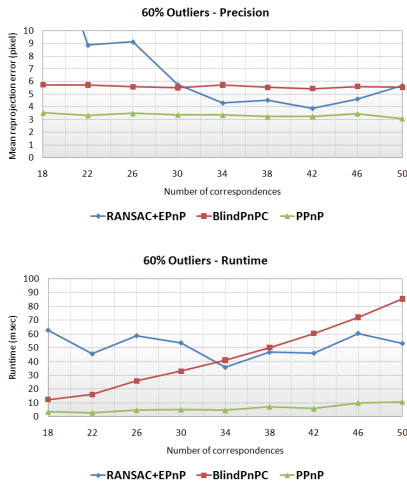


Figure 3: 60% outlier ratio measurement results.

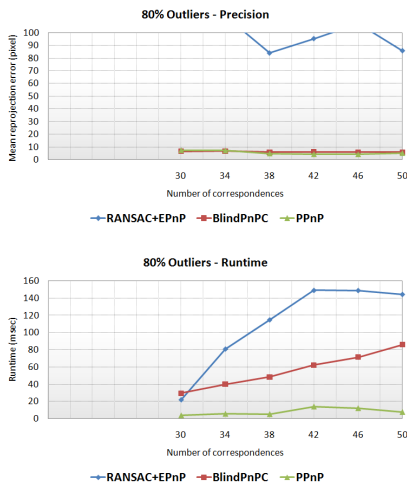


Figure 4: 80% outlier ratio measurement results.

two images with distinctive patterns were positioned whose coordinates were known with respect to some coordinate system. The prior probability V was established using a Gaussian mixture model of $g = 6$ components as sketched in figure 1.

Correspondences: The correspondences were established in real-time using randomized trees as classifiers. The technique is explained in detail in (Lepetit et al., 2005). As seen in figure 5 a threshold value τ allows to control the certainty of the correspondences.

Because no ground truth camera pose is available in a real data setting, we used the calculated set of inliers I in order to check the quality. Using I as provided by the respective algorithm, we used EPnP for calculating a camera pose estimate Q . The border

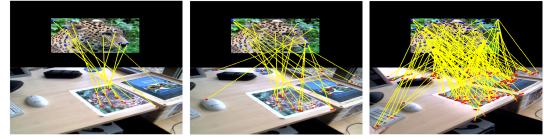


Figure 5: Threshold values τ from 0.1, over 0.05 to 0 control the quality of the correspondences. This results in outlier ratios ranging from $\lambda \approx 30\%$ over $\lambda \approx 60\%$ up to $\lambda \approx 80\%$.

rectangle of the object is then projected in yellow into the scene using Q . This resulted in correctly framing the object on the captured image if I was estimated correctly. Thus we were able to visually check the integrity of I . As evidence of robustness, we counted the frames where the inliers were correctly calculated. We consider an algorithm as ‘robust’, if I was correctly calculated for more than 90% of the frames. For values less than 50%, an algorithm is considered as ‘not robust’. Values between are considered as ‘intermediate robust’. Based on I we also calculated the average outlier ratio λ_{\approx} .

4.2.1 Results

Three experiments with different values for the correspondence generator threshold τ were made. The results are presented in table 1.

Summarized, the experiments using a real data test setting confirmed the results deduced from the synthetic test settings. Standard methods such as RANSAC+EPnP only deliver reliable results for small outlier ratios λ . As λ grows, standard methods tend to fail or require an unacceptable large RANSAC iteration count in order to deliver results at all.

BlindPnP and PPnP use the available pose prior probability and thereby are significantly less error prone and faster than the standard methods. The estimations computed by BlindPnP and PPnP are both robust and comparable. A difference however exists with respect to runtime performance: Especially if n is large, PPnP is able to provide results much faster. Images series taken from the test setting using low quality correspondences ($\tau = 0$) with PPnP are shown in figure 6. Hereby the correspondences detected by the feature detector are represented as red and green dots. Red dots have been declared as outliers using the respective algorithm, green dots as inliers.

5 CONCLUSION

In this paper we developed and evaluated a new algorithm called PPnP, capable of estimating a robust camera pose in real-time even though being provided with large sets of noisy correspondences having high

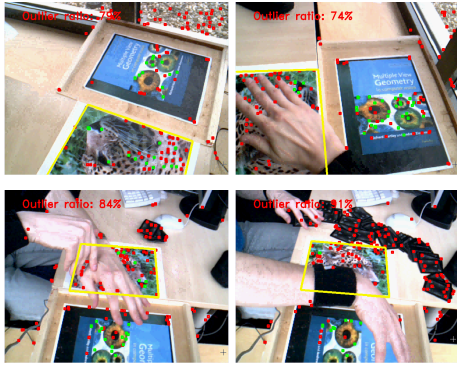


Figure 6: PPnP: High number of correspondences, high outlier ratio. A correct camera pose estimation is possible in 95% of the cases within at most 20 iterations. Even for outlier ratios larger than 90% and in occurrence of partial occlusions, PPnP delivers reliable results. Speed: 25 fps.

Table 1: Real data test setting results.

Algorithm	λ_{\approx}	Robustness	fps
RANSAC+EPnP	0.3	Yes	30
	0.6	Intermediate	6
	0.8	No	2
BlindPnP	0.3	Yes	30
	0.6	Yes	15
	0.8	Yes	4
PPnP	0.3	Yes	30
	0.6	Yes	30
	0.8	Yes	25

outlier ratios. PPnP is based upon BlindPnP which we also implemented and evaluated. Both algorithms use a probability distribution as additional information beside the correspondences in order to handle correspondences with high ratios of outliers.

In both synthetic and real data test settings we have shown, that as the ratio of outliers grows, standard pose estimation approaches using RANSAC fail in providing a robust camera pose estimate. In contrast to this, BlindPnP and PPnP provide reliable results independent of the outlier ratio in the correspondences. The pose prior probability allows BlindPnP and PPnP to ease the direct dependency of the estimated camera poses' quality on the ratio of outliers. This direct dependency on the input data represents the major weakness of standard methods.

In contrast to BlindPnP – as the number of correspondences used is raised – PPnP still is able to provide reliable results in real-time for these situations. This is related to the optimization techniques introduced in PPnP which allow to evolve the camera pose requiring a much smaller number of consecutive hypothesizing steps than in BlindPnP.

5.1 Future Work

In our experiments both BlindPnP and PPnP showed good results. We would like to investigate, in how far these algorithms can replace standard pose estimation techniques in practice. Comparing to standard methods, BlindPnP and PPnP depend on a large number of variables which have to be assigned for each situation accordingly (e.g. pose prior probability, threshold values, iteration count, ...). Since these variables are mutually dependent, the assignment is not intuitive and usually a certain effort has to be put into testing different assignments before using the algorithms appropriately. Hence additional techniques should be developed, intending in improving the usability.

ACKNOWLEDGEMENTS

This work has been partially funded by the project CAPTURE (01IW09001) and the German BMBF project AVILUSplus (01M08002).

REFERENCES

- Dhome, M., Richetin, M., Lapresté, J.-T., and Rives, G. (1989). Determination of the attitude of 3-d objects from a single perspective view. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, Vol. 24.
- Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F. (2003). Complete solution classification for the perspective–three–point problem. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25.
- Haralick, R. M., Lee, C.-N., Ottenberg, K., and Nölle, M. (1994). Review and analysis of solutions of the three point perspective pose estimation problem. In *International Journal of Computer Vision*, Vol. 13.
- Lepetit, V., Lagger, P., and Fua, P. (2005). Randomized trees for real-time keypoint recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2.
- Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). EPnP: An accurate O(n) solution to the PnP problem. In *International Journal of Computer Vision*, Vol. 81.
- Moreno-Noguer, F., Lepetit, V., and Fua, P. (2008). Pose priors for simultaneously solving alignment and correspondence. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*.
- Quan, L. and Lan, Z. (1999). Linear n–point camera pose determination. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21.