Markerless Camera Pose Estimation - An Overview

Tobias Nöll[1], Alain Pagani[1], and Didier Stricker[1]

1 **Department of Augmented Vision**
German Research Center for Artificial Intelligence GmbH
Trippstadter Str. 122, 67663 Kaiserslautern, Germany
{Tobias.Noell, Alain.Pagani, Didier.Stricker}@DFKI.de

Abstract

Human perception shows that a correct interpretation of a 3D scene on the basis of a 2D image is possible without markers. Solely by identifying natural features of different objects, their locations and orientations on the image can be identified. This allows a three dimensional interpretation of a two dimensional pictured scene. The key aspect for this interpretation is the correct estimation of the camera pose, i.e. the knowledge of the orientation and location a picture was recorded. This paper is intended to provide an overview of the usual camera pose estimation pipeline as well as to present and discuss the several classes of pose estimation algorithms.

Keywords and phrases Pose Estimation

Digital Object Identifier 10.4230/OASIcs.VLUDS.2010.45

1 Introduction

A large number of different markerless pose estimation algorithms already exist [5] [8] [3] [17] [6] [12] . Usually the camera pose for a given image is estimated solely using a set of correspondences. A correspondence is a match of an object's natural feature (3D) detected on the image (2D). With a sufficient number of correspondences the camera pose can be determined uniquely. In this paper the different classes of pose estimation algorithms will be presented along with their advantages and disadvantages. The classes can be roughly categorized into the following:

- **Direct Linear Transformation:** These algorithms intend to directly estimate the camera pose ignoring certain restrictions regarding the solution space.
- **Perspective n-Point:** These algorithms intend to directly estimate values for the parameters parameterizing the solution space of all valid camera poses.
- **A priori information estimators:** Beside the point correspondences, additional information regarding the camera pose is often available before estimating. The algorithms belonging to this class use this information in order to provide more reliable or faster results.

Figure 1 gives a possible application of a camera pose estimation algorithm: Once the camera pose is known, the recorded image is extended with important 3D information. Not only can all objects whose three dimensional positions are known be identified on the image, the image can also be enriched with virtual information, allowing a fusion of reality and virtuality (virtual reality). This way images can be extended with additional virtual objects.

Visualization of Large and Unstructured Data Sets– IRTG Workshop, 2010.
Editors: Ariane Middel, Inga Scheler, Hans Hagen; pp. 45–54
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

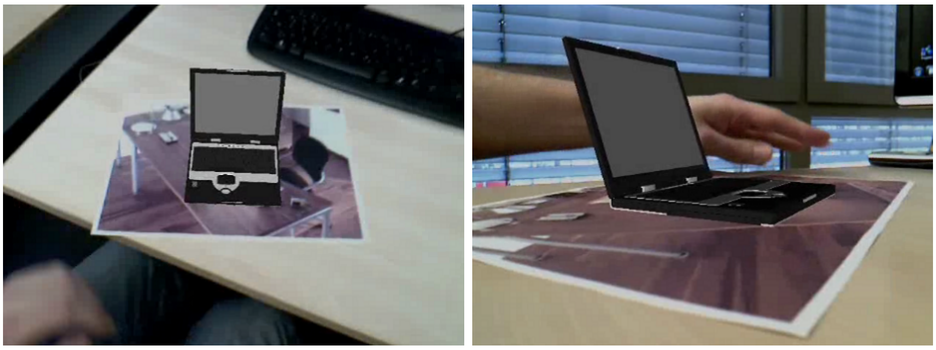

Figure 1 By estimating the camera pose, the live recorded image can be added with additional virtual objects (in this case a notebook). Even though the camera moves, the impression of the mixed reality and virtuality is still unbiased.

2 Background

The mathematical camera model which is used in the following is built on the principles of the pinhole camera. It models the central projection of 3D points through the center of projection onto the image plane. This mapping depends on the following parameters:

- The *internal parameters* of the camera: $K \in \mathbb{R}^{3\times3}$ which depends on certain properties of the manufactured camera (i.e. focal length, lens properties). These parameters can be estimated using camera calibration techniques such as [18] and are assumed to be known. K maps 3D points in camera coordinates to 2D image points.
- The *external parameters* of the camera: $\left[R \mid \mathbf{t}\right] \in \mathbb{R}^{3\times4}$. This mapping is commonly referred to as the *camera pose* which relates 3D points from their representation in a world coordinate system to their representation in the camera coordinate system by use of a rotation R and a translation $\mathbf{t}$.

Altogether a homogeneous representation $\mathbf{P} \in \mathbb{R}^4$ of a 3D point in world coordinates is mapped by the camera to the point $\mathbf{p} \in \mathbb{R}^3$. $\mathbf{p}$ itself is a homogeneous representation of the corresponding 2D point on the image. This mapping is defined by:

$$\mathbf{p} = K\left[R \mid \mathbf{t}\right]\mathbf{P} \tag{1}$$

Estimation of the camera pose from correspondences then refers to searching the camera pose $\left[R \mid \mathbf{t}\right]$ which best relates a set of given correspondences $C = \{\mathbf{P_i} \longleftrightarrow \mathbf{p_i}\}$ using Equation 1 under a given camera calibration K.

2.1 Correspondence generation

The set of correspondences C is usually automatically generated using feature detectors. Given a set of images, feature detectors try to identify and match several features of the captured objects on those images (see Figure 2). In context of camera pose estimation, one image is referred to as the *reference image*. Features on this image have – manually or automatically – been assigned to 3D coordinates in the world coordinate frame. This way, once a feature from the reference image with assigned 3D coordinate $\mathbf{P_i}$ is detected on position $\mathbf{p_i}$ in the captured image, a correspondence $\mathbf{P_i} \longleftrightarrow \mathbf{p_i}$ is generated. Popular feature detectors are SIFT [13] and Randomized Trees [11].

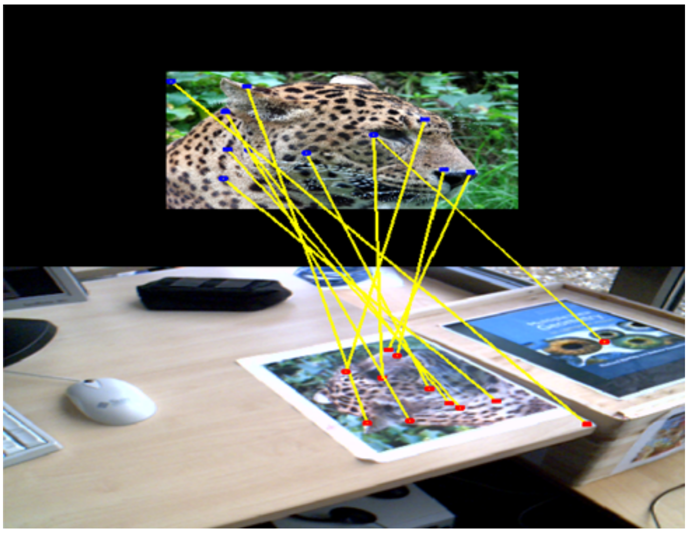

Figure 2 Two images (top: reference image, bottom: captured image) and their matched features. Some features have been misleadingly matched.

Since the correspondences are generated automatically, C will also contain a certain percentage of outliers, i.e. untrue correspondences $\mathbf{P_i} \longleftrightarrow \mathbf{p_i}$. These outliers are problematic because they potentially can have a huge negative impact on the quality of the solution. Therefore, methods such as RANSAC [5] need to be used to identify the inliers in order to perform the camera pose estimation only using this set.

2.2 Exclusion of outliers

RANSAC uses a stochastic approach in order to identify the inliers. Therefore this algorithm is of non-deterministic nature in the sense that it produces a reasonable result only with a certain probability. This probability increases as more iterations are allowed. The algorithm operates as follows:

1. A random subset C_S of the provided correspondences C is selected. It contains only the minimal number of correspondences required for camera pose estimation using an arbitrary pose estimation algorithm M.
2. A camera pose Q is estimated using M along with C_S.
3. All remaining correspondences $C \setminus C_S$ are then checked for integrity with Q. Therefore the points $\mathbf{P_i}$ are projected using the known camera matrix K and Q to the 2D points $\mathbf{p'_i}$. If $\|\mathbf{p'_i} - \mathbf{p_i}\| \leq \tau$, then the correspondence $\mathbf{P_i} \longleftrightarrow \mathbf{p_i}$ fits well to the estimated camera pose and will therefore also be considered as a hypothetical inlier.
4. Q is reasonably good if a sufficient number of correspondences has been classified as hypothetical inliers. Q is then reestimated from all hypothetical inliers using M, because it has only been estimated from the initial set of hypothetical inliers C_S.

This procedure is repeated a fixed number of times, each time producing either a camera pose which is rejected because too few points are classified as inliers or a refined pose together with a corresponding error measure. In the second case the refined pose is kept if its error is lower than the last saved pose.

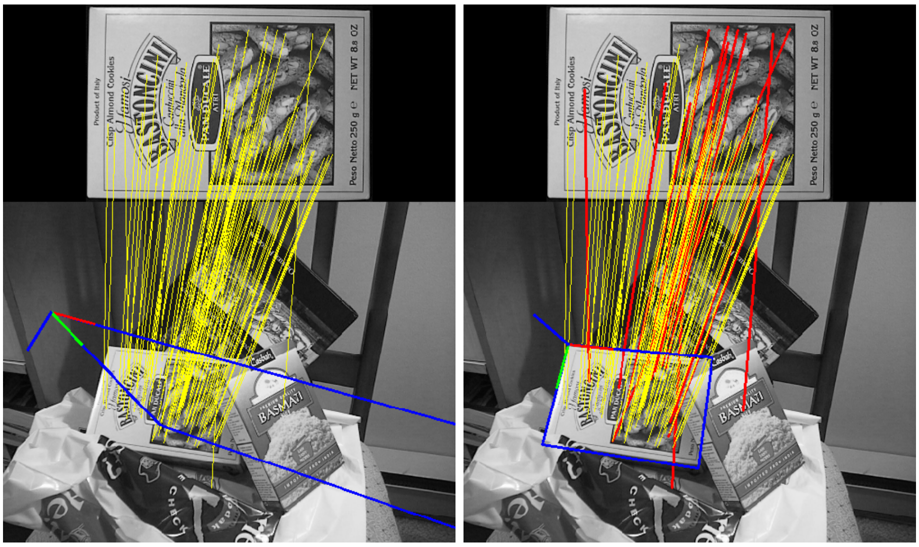

Figure 3 The yellow correspondences are computed using the SIFT feature detector [13] providing many good matches along with a small number of outliers. The borders of the reference image are reprojected in blue in the scene using the calculated camera pose. One can see that the quality of the solution is highly improved by identifying the outliers. Left: All yellow correspondences are used, including the outliers. Right: RANSAC is used in order to find the set of inliers (yellow correspondences). The red correspondences are treated as the outliers and not used in order to compute the final camera pose.

Figure 3 gives an example in the context of augmented reality, showing how important the identification of outlying point correspondences is in order to receive a reliable camera pose.

3 Direct Linear Transformation

For the Direct Linear Transformation (DLT) algorithm [9] it is assumed that the set of points $\mathbf{P_i}$ spans a real 3D space which means that those points are arbitrarily distributed in space an thus do not lie on a single point, line or plane. Let $\mathbf{P_i} = \begin{pmatrix} X_i & Y_i & Z_i & W_i \end{pmatrix}^T \longleftrightarrow \mathbf{p_i} = \begin{pmatrix} u_i & v_i & w_i \end{pmatrix}^T$ be n correspondences. A DLT $F \in \mathbb{R}^{3\times 4}$ now is a linear function $F : \mathbb{R}^4 \mapsto \mathbb{R}^3$ which maps the points $\mathbf{P_i}$ to the points $\mathbf{p_i}$. This can be expressed in the homogeneous context as

$$F\mathbf{P_i} \sim \mathbf{p_i} \Leftrightarrow \mathbf{p_i} \times F\mathbf{P_i} = \mathbf{0} \tag{2}$$

F has 12 unknowns:

$$F = \begin{bmatrix} {f_1}^1 & {f_2}^1 & {f_3}^1 & {f_4}^1 \\ {f_1}^2 & {f_2}^2 & {f_3}^2 & {f_4}^2 \\ {f_1}^3 & {f_2}^3 & {f_3}^3 & {f_4}^3 \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{f^1}^T \\ \mathbf{f^2}^T \\ \mathbf{f^3}^T \end{bmatrix}$$

Rewriting Equation 2 using the fact that $\mathbf{f^i}^T\mathbf{P_i} = \mathbf{P_i}^T\mathbf{f^i}$ yields

$$\mathbf{p_i} \times F\mathbf{P_i} = \begin{pmatrix} v_i\mathbf{f^3}^T\mathbf{P_i} - w_i\mathbf{f^2}^T\mathbf{P_i} \\ w_i\mathbf{f^1}^T\mathbf{P_i} - u_i\mathbf{f^3}^T\mathbf{P_i} \\ u_i\mathbf{f^2}^T\mathbf{P_i} - v_i\mathbf{f^1}^T\mathbf{P_i} \end{pmatrix} = \underbrace{\begin{bmatrix} \mathbf{0}^T & -w_i\mathbf{P_i}^T & v_i\mathbf{P_i}^T \\ w_i\mathbf{P_i}^T & \mathbf{0}^T & -u_i\mathbf{P_i}^T \\ -v_i\mathbf{P_i}^T & u_i\mathbf{P_i}^T & \mathbf{0}^T \end{bmatrix}}_{\stackrel{\text{def}}{=} A'_i} \underbrace{\begin{pmatrix} \mathbf{f^1} \\ \mathbf{f^2} \\ \mathbf{f^3} \end{pmatrix}}_{\stackrel{\text{def}}{=} \mathbf{f}} \tag{3}$$

However each correspondence will only derive two *linearly independent* equations. The linearly independent system now reads

$$\underbrace{\begin{bmatrix} \mathbf{0}^T & -w_i\mathbf{P_i}^T & v_i\mathbf{P_i}^T \\ w_i\mathbf{P_i}^T & \mathbf{0}^T & -u_i\mathbf{P_i}^T \end{bmatrix}}_{\stackrel{\text{def}}{=} A_i} \mathbf{f} = \mathbf{0} \tag{4}$$

For $n \geq 6$ System 4 is overdetermined and hence F can be estimated by a *Singular Value Decomposition* (SVD). The camera pose can now be extracted from F:

$$F\mathbf{P_i} \sim \mathbf{p_i} \sim K\left[R \mid \mathbf{t}\right]\mathbf{P_i} \Rightarrow F \sim K\left[R \mid \mathbf{t}\right] \Rightarrow \left[R \mid \mathbf{t}\right] \sim K^{-1}F \tag{5}$$

Since by Equation 5 the pose is defined up to a scale, this scale factor should be chosen in a way that R mostly approximates a rotation matrix (i.e. $R^TR \approx RR^T \approx I, det(R) \approx 1$). Also an orthogonalization of R is advisable afterwards.

4 Perspective n-Point Problem

Formally, the Perspective n-Point (PnP) problem is defined as follows: Given a set of n 3D points $\mathbf{P_i}$ whose coordinates are known in some object coordinate frame O. Let $\mathbf{p_i}$ be a set of n 2D points which are the projections of the points $\mathbf{P_i}$ on the image plane I. Let $\vec{\mathbf{v_i}} = \overline{\mathbf{Cp_i}}$ be n directional vectors with $\mathbf{C}$ as the camera's center of perspective (Note: Since the camera is assumed to be calibrated, one can determine the vectors $\vec{\mathbf{v_i}}$ from the camera calibration matrix K). The PnP problem is defined as to determine the position of $\mathbf{C}$ and its orientation relative to O.

This problem was first solved directly in 1841 by the German mathematician Grunert [7]. In 1945 Church [2] first formulated an iterative solution algorithm which – if provided with a good starting value – constitutes an approximate solution. Since then, a great variety of different direct and iterative solution techniques were invented. However, for Computer Vision a good starting value of the camera pose is usually not known in advance and therefore the direct solutions of the PnP problem are of more interest. For those non-iterative algorithms like Dhome et al. [4], Fischler and Bolles [5], Gao et al. [6], Haralick et al. [8], Quan and Lan [17], it typically involves solving for the roots of an eight-degree polynomial with no odd terms, yielding up to four solutions, so that a fourth point is needed for disambiguation. The complexity of these algorithms varies according to [12] from $O(n^2)$ to $O(n^8)$. As for the iterative solution approaches the method POSIT by Dementhon et al. [3] has to be highlighted: Even though POSIT is an iterative algorithm with a precision increasing in each iteration step, no starting value is needed for initialization.

A non-iterative technique called EPnP (**E**fficient **P**erspective **n-P**oint Camera Pose Estimation) developed by Lepetit, Moreno-Noguer and Fua [12] allows the computation of an accurate and unique solution in $O(n)$ for $n \geq 4$. As in most of the existing PnP solution techniques, the idea in the implementation of EPnP is to retrieve the locations of $\mathbf{P_i}$ relative to the camera coordinate frame. Then retrieving the camera orientation and translation which aligns both sets of coordinates is a simple task [1]. The key of the algorithm's efficient $O(n)$ implementation is, to represent the $\mathbf{P_i}$ as a weighted sum of $m \leq 4$ control points $\mathbf{C_1} \ldots \mathbf{C_m}$ and perform all further computation only on these points. For large n this yields a much smaller number of unknowns compared to other algorithms and therefore accelerates further computations.

5 A priori information

Any information imaginable restricting the set of valid camera poses can be used whenever there exists a way of appropriately integrating it in a pose estimation algorithm. Two types of *a priori* information are considered:

- **Inclination information:** The inclination of the camera can be measured by inclination sensors. This reduces the three degrees of freedom for the rotation matrix R to only one degree of freedom.
- **Prior probability:** Often some subset of all possible camera poses is more probable or can be assumed. This information can be used in order to reduce the search space.

5.1 Inclination information

Inclination, in general, is the angle between a reference plane and another plane or axis of direction. The inclination relative to a surface can be measured using acceleration sensors which measure the inclination relative to each axis. Usually the sensing element consists of three acceleration sensitive masses. Daisuk Kotake et al. [10] propose a pose estimation algorithm similar to DLT being capable of integrating inclination information in the computation process. Both the sensor and the camera have their own coordinate frame. For the case where both coordinate frames are not identical, a calibration known as 'Hand-Eye Calibration' is necessary. For simplicity it is assumed that the rotation matrix R is parameterized using Euler parametrization and the sensor's world coordinate frame is identical to the camera coordinate frame. Thus the rotation R can be written as

$$R = R_x R_y R_z \stackrel{\text{def}}{=} R_{incl} R_z \tag{6}$$

This means that R can be represented as the product of three rotations R_x, R_y, R_z around each axis $\vec{\mathbf{x}} = \begin{pmatrix}1 & 0 & 0\end{pmatrix}^T, \vec{\mathbf{y}} = \begin{pmatrix}0 & 1 & 0\end{pmatrix}^T, \vec{\mathbf{z}} = \begin{pmatrix}0 & 0 & 1\end{pmatrix}^T$. The rotation around the X and Y axis R_{incl} can be measured using the inclination sensor, the remaining unknown part of the rotation is R_z. Since R_z is a rotation around the Z axis, one can write it explicitly

$$R_z = \begin{bmatrix} cos(\varphi) & -sin(\varphi) & 0 \\ sin(\varphi) & cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

At this stage one can already see that the overall rotation R, having three Euler angles as parameters, is reduced to a representation only having one degree of freedom $R_{incl}R_z(\varphi)$.

5.2 Prior probability

The a priori information can also be used to formulate probabilities regarding the final camera pose estimate (e.g. '*The camera will most likely have a small distances from the earths' surface and will not be positioned below it*', '*The camera will be flipped upside down*

Figure 4 Modeling the camera pose prior probability by mixtures of Gaussians. For simplicity, only the translation uncertainty is visualized (green ellipsoids). A real pose prior probability would normally consist of a 6D covariance. The mean values $\mathbf{x_Q}$ hereby specify the position, the covariances Σ_Q the shape of the ellipsoids.

only with a very low probability'). In context of probability theory, this prior probability is modeled using probability distributions. Depending on the pose parametrization, the pose $Q \stackrel{\text{def}}{=} [R \mid \mathbf{t}]$ can be written as a $m \geq 6$ dimensional parameter vector $\mathbf{x_Q}$ along with an invertible transformation function $T : \mathbb{R}^m \mapsto \mathbb{R}^{3\times 4}$, which transforms the parametrization of the pose to the real camera pose. Mathematically the *pose prior probability* is modeled in the 6D pose parametrization space using *mixture of Gaussians* 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* $\Sigma_Q \in \mathbb{R}^{6\times 6}$ (see Figure 4).

Moreno-Noguer et al. [14] propose a method called BlindPnP where the prior probability is integrated in a pose estimation algorithm. Even though this algorithm was not intended to work with correspondences it can be easily modified for doing so, an optimization of BlindPnP is presented in [16]. In BlindPnP the prior probability is used primarily in order to increase the efficiency while identifying the outliers using an approach similar to RANSAC. The correct pose is found by minimizing the error functional

$$E(\mathbf{x_Q}) \stackrel{\text{def}}{=} \sum_{\mathbf{P_i} \longleftrightarrow \mathbf{p_i} \in M_Q} \|\mathbf{p_i} - Proj_{\mathbf{x_Q}}(\mathbf{P_i})\| + \theta |F_Q| \tag{8}$$

with $Proj_{\mathbf{x_Q}}(\mathbf{P_i})$ defined as the projection of $\mathbf{P_i}$ on the image using pose $T(\mathbf{x_Q})$ and $\theta \in \mathbb{R}$ as a penalty term that penalizes unmatched points. The minimization is computed by utilizing the prior probability: Roughly summarized, BlindPnP hypothesizes three 3D to 2D point correspondences consecutively which are compatible with the prior probability (see Figure 5). During this process, the camera pose $\mathbf{x_Q}$ evolves and the assigned covariance Σ_Q (i.e. uncertainty) reduces. In one cycle of BlindPnP all correspondences are projected on the image and checked for compatibility with the prior probability. All compatible correspondences are hypothesized in turn or marked as outliers. After three consecutive hypothesizing steps, the remaining correspondences can be checked for validity using this camera pose by projecting

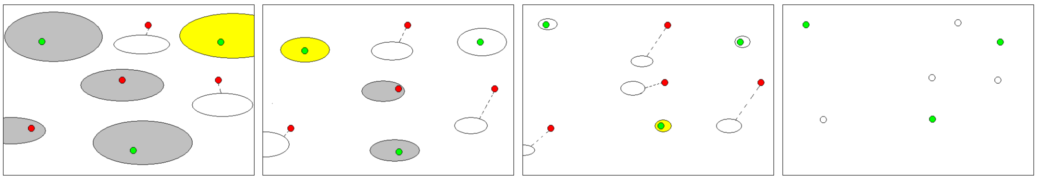

Figure 5 BlindPnP hypothesizing process: Red dots represent the 2D points which are assigned to wrong 3D correspondences (outliers), the green dots are correctly classified correspondences (inliers). The ellipses represent the region of uncertainty of the projected 3D points with the current camera pose estimate. If the ellipses are drawn in gray, the according correspondences are classified by BlindPnP as valid candidates for the hypothesizing process in the next step. The correspondence regarding the yellow marked ellipse is then selected for the current hypothesizing process. Thereby the camera pose changes (i.e. dots move) and the assigned uncertainty reduces (i.e. ellipses shrink).

the 3D points $\mathbf{P_i}$ on the image and checking their distance to their corresponding 2D point $\mathbf{p_i}$. This way the sets M_Q and F_Q can be constructed. Recursively iterating through all arguable subsets containing three correspondences valid for hypothesizing, the pose $T(\mathbf{x_Q})$ with the least error function value $E(\mathbf{x_Q})$ is then chosen as result.

6 Discussion

Nöll [15] showed that the weakness of the DLT algorithm lies in its unconstrained minimization. In the SVD solution method the vector $\mathbf{f}$ (i.e. the matrix F) is treated as unknown in all of its components with a total number of 11 degrees of freedom. However, a valid camera pose F has only 6 true degrees of freedom (three Euler angles for R and three coordinates for $\mathbf{t}$, K is known). It becomes clear that the minimization of $\mathbf{f}$ in Equation 4 does not provide a good solution F in all cases, especially if n is small or the correspondences are significantly affected by noise.

Algorithms belonging to the PnP class usually search the solution only in the valid solution space. As a consequence these algorithms are significantly more robust especially in these situations. If the correspondences contain a certain ratio of outliers, standard methods (i.e. DLT and PnP algorithms) usually utilize RANSAC in order to exclude those prior to computing the camera pose solely using the remaining consistent set of correspondences. In [15] it has been shown that the quality of RANSAC reduces as the ratio of outliers increases. At outlier ratios of 60%, in both synthetic and real data test settings, it was often not possible to estimate a robust camera pose within a reasonable time.

Two different methods have been presented that include a priori information in the camera estimation process, intending to thereby eliminate the weaknesses of the standard methods. In [10] the information is included directly in the equations which reduces the number of unknowns. Experiments in [15] showed that this results in a better runtime performance. Problematic in this approach is that in reality the inclination cannot be measured with infinite precision. This way errors are included in the estimation process. In order to receive robust estimations nevertheless, numerical optimizations such as Levenberg Marquardt have to be applied. Under those assumptions, experiments showed that this results in an even higher runtime and lower quality compared to standard methods. For correspondences containing low outlier ratios standard pose estimation algorithms such as RANSAC+EPnP are fast and robust enough even without using any a priori information. In [14] and [16] information in form of pose prior probability is used prior in order to exclude the outliers.

It was shown in [15] that for each scenario with appropriate a priori information available, there always exists an outlier ratio where these algorithms provide both more reliable and faster results than standard methods. In the evaluated scenarios this was the case for outlier ratios of 60% resp. 40% and above. A positive effect of the a priori information included in form of pose prior probability intending to identify the outliers could clearly be verified in both synthetic and real data test settings. However comparing to standard methods, BlindPnP and PPnP depend on a large number of variables which have to be assigned for each situation accordingly. 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.

References

1. J. Challis. A procedure for determining rigid body transformation parameters. *Journal of Biomechanics*, 28(6):733–737, 1995.
2. E. Church. Revised geometry of the aerial photograph. volume 15, Syracuse, NY, 1945. Syracuse University Press.
3. D. Dementhon and L. Davis. Model-based object pose in 25 lines of code. *Int. J. Comput. Vision*, 15(1-2):123–141, 1995.
4. M. Dhome, M. Richetin, and J.-T. Lapreste. Determination of the attitude of 3d objects from a single perspective view. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(12):1265–1278, 1989.
5. Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
6. X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):930–943, 2003.
7. J. A. Grunert. Das pothenotische Problem in erweiterter Gestalt nebst über seine Anwendung in der Geodäsie. In *Grunerts Archiv für Mathematik und Physik*, 1:238–248, 1841.
8. R. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *Int. J. Comput. Vision*, 13(3):331–356, 1994.
9. R. Hartley and A. Zisserman. *Multiple view geometry in computer vision.* Cambridge University Press, New York, NY, USA, 2000.
10. D. Kotake, K. Satoh, S. Uchiyama, and H. Yamamoto. A hybrid and linear registration method utilizing inclination constraint. In *ISMAR '05: Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 140–149, Washington, DC, USA, 2005. IEEE Computer Society.
11. V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2:775–781, Washington, DC, USA, 2005. IEEE Computer Society.
12. V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision*, 81(2):155–166, 2009.
13. D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
14. F. Moreno-Noguer, V. Lepetit, and P. Fua. Pose priors for simultaneously solving alignment and correspondence. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 405–418, Berlin, Heidelberg, 2008. Springer-Verlag.

15 T. Nöll. Robust camera position estimation by integration of a priori information. Master's thesis, University of Kaiserslautern, 2009.

16 T. Nöll, A. Pagani, and D. Stricker. Real-time camera pose estimation using correspondences with high outlier ratios. In *VISAPP 2010: International Conference on Computer Vision Theory and Applications*, pages 381–386, 2010.

17 L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):774–780, 1999.

18 Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.