

# PARAMETER AND CONFIGURATION ANALYSIS FOR NON-LINEAR POSE ESTIMATION WITH POINTS AND LINES

Bernhard Reinert, Martin Schumann and Stefan Mueller

*Institute of Computational Visualistics, University of Koblenz, Koblenz, Germany*

Keywords: Camera Pose Tracking, Model Features, Correspondences, Non-linear Optimization.

Abstract: In markerless model-based tracking approaches image features as points or straight lines are used to estimate the pose. We introduce an analysis of parametrizations of the pose data as well as of error measurements between 2D image features and 3D model data. Further, we give a review of critical geometrical configurations as they can appear on the input data. From these results the best parameter choice for a non-linear pose estimator is proposed that is optimal by construction to handle a combined input of feature correspondences and works on an arbitrary number and choice of feature type. It uses the knowledge of the 3D model to analyze the input data for critical geometrical configurations.

## 1 INTRODUCTION

Model-based camera pose tracking is the process of estimating the viewing position and orientation of a camera by establishing 2D-3D correspondences between features of the model and the camera image. Estimating the pose can be done by using linear or non-linear solutions. Most pose algorithms originate from the domain of linear solutions and operate on point or line features. Combining both feature types has been done, but no complete comparison of parametrization and error measurements for non-linear solutions is available. Further, a priori model knowledge should be used to improve stability of a combined solution.

We provide a discussion on possible parametrizations of the camera pose, including its constraints which must be taken into account during or after the optimization process as well as on several error measurements for the feature correspondences to be minimized by the optimization process. Further, we give a review of critical configurations as they are known to cause ambiguous results on the input geometry of points and lines. From these results we propose the best parameter choice for a non-linear pose estimator that is optimal by construction for the acceptance of point and line correspondences in arbitrary combination and number. The estimator takes account of critical configurations of the input data by using the knowledge of the 3D model and selecting the best minimal correspondence set for a stable solution. We

investigate the behaviour on variable input types and numbers of correspondences, as well as the influence of noise on the robustness of the pose.

## 2 RELATED WORK

The problem of camera pose estimation with 2D-3D correspondences of points is known as the perspective-n-point problem (PnP) (Fischler and Bolles, 1981). There are several approaches to the solution of P3P surveyed and compared for their numerical stability by (Haralick et al., 1994). While these linear solutions work on a fixed number of three point correspondences, (Lepetit et al., 2009) introduce virtual control points to make possible a solution for arbitrary numbers of point correspondences (PnP) with linear complexity. Another linear solution for any number of point or line correspondences is given by (Ansar and Daniilidis, 2003) but does not work on both types of features simultaneously.

Most of the non-linear solutions are based on classical iterative algorithms of optimization, like the Gauss-Newton- or Levenberg-Marquardt-Method. Non-linear pose with line correspondences has been estimated by (Kumar and Hanson, 1994) who analyzed the influence of certain line representations on the optimization process. A solution using arbitrary line traits on an object is shown by (Lowe, 1991). Possible error measurements for point corre-

spondences were analyzed by (Lu et al., 2000), who also proposed a new possibility requiring fewer iterations. In (Dornaika and Garcia, 1999) points and lines are combined under an extension of the well-known POSIT algorithm with additional optimization.

Joint usage of point and line correspondences exists in the field of linear solutions and to a lesser extent in non-linear approaches. But most authors analyze one parametrization and a single error measurement only, when solving for the camera pose. So in sections 4, 5 and 7 we will first give a full analysis of error measurements and parametrizations to find the best fitted parameters for constructing combined non-linear pose. An overview of critical configurations for points is given by (Fischler and Bolles, 1981), where (Wolfe et al., 1991) prove the number of solutions for P3P geometrically as (Hu and Wu, 2002) and (Gao and Tang, 2006) do for P4P. An example for critical line configurations is given by (Christy and Horaud, 1999). We will address the problem of recognizing such configurations in section 9.

### 3 DEFINITIONS

The pose problem can be described as the estimation of the extrinsic camera parameters relative to a known reference coordinate system, i.e. the world coordinate system, from given correspondences between 2D features  $\vec{q}$  of a camera image and 3D features  $\vec{p}$  of a synthetic model. The coordinate system of a value is identified by a superscript  $w$  for world-,  $c$  for camera-,  $i$  for image- and  $p$  for pixel-coordinate-system. For all coordinate systems, the normalized vector of  $\vec{p}$  is denoted by  $\vec{\hat{p}}$ . The camera pose is represented in combination as a tuple consisting of a rotation matrix  $R \in \mathbb{R}^{3 \times 3}$  and a translation vector  $\vec{t} \in \mathbb{R}^3$  as  $C : (R, \vec{t})$  and represents the transformation of a point  $\vec{p}$  between world- and camera-coordinate-system as  $\vec{p}^c = R\vec{p}^w + \vec{t}$ . The intrinsic camera matrix  $K$  is assumed to be known by calibration. Hence, given a world point  $\vec{p}^w$  and  $K$ , its pixel coordinates  $\vec{p}^p$  can be calculated by perspective projection. A *point correspondence* between the world point  $\vec{p}^w$  and the pixel point  $\vec{q}^p$  is represented by a tuple  $k_p : (\vec{p}^w, \vec{q}^p)$ .

The line features in the camera images are defined as straight lines  $l : (\phi^p, \rho^p)$  with infinite extent.  $\phi^p$  represents the angle between the line normal and the y-axis of the pixel coordinate system and  $\rho^p$  is the orthogonal distance of the line from the image origin. Hence, for each point  $\vec{p}^p \in l$  it holds

$$\cos \phi^p p_x^p + \sin \phi^p p_y^p = \rho^p. \quad (1)$$

A straight *line correspondence* is represented by two world points  $\vec{s}^w$  and  $\vec{e}^w$  (typically start- and endpoint)

of the model line and an image line  $l^p : (\phi^p, \rho^p)$  by a tuple  $k_l : ((\vec{s}^w, \vec{e}^w), l)$ .

### 4 ERROR MEASUREMENTS FOR POINTS

Since the aim is to minimize the distance between the transformed model feature and camera image feature, error measurements, i.e. residuals, for correspondences have to be defined. We investigated three different point residuals which measure the error in different coordinate-systems (Figure 1).

The well-known **Reprojection-Error** measures the distance between the image feature  $\vec{q}^p$  and the model feature  $\vec{p}^p$  after its projection to the image plane in pixel-coordinates. Hence, the residual for each correspondence  $k_p$  becomes

$$\vec{r}^{RE} = \vec{p}^p - \vec{q}^p = \begin{pmatrix} p_x^p - q_x^p \\ p_y^p - q_y^p \end{pmatrix}. \quad (2)$$

The **Object-Space-Error** measures the distance in camera-coordinates and was introduced by (Lu et al., 2000). To recover the depth for the image feature, its normalized sight vector  $\vec{q}^c$  (i.e. the connecting vector from the camera center to the feature) is projected onto the sight vector of the model  $\vec{p}^c$ . The resulting scalar is used to scale the normalized sight vector of the image  $\vec{q}^c$  and both sight vectors are then compared based on their camera coordinates. The residual per correspondence  $k_p$  becomes

$$\vec{r}^{OE} = \begin{pmatrix} p_x^c - \langle \vec{q}^c, \vec{p}^c \rangle \hat{q}_x^c \\ p_y^c - \langle \vec{q}^c, \vec{p}^c \rangle \hat{q}_y^c \end{pmatrix}. \quad (3)$$

It has to be noted that points whose projections on the image plane are the same produce greater residuals for greater distances from the image plane, i.e. are not depth invariant.

The **Normal-Space-Error** also measures the distance in the camera-coordinates. From the image feature  $\vec{q}^c$  two orthogonal normals  $\vec{n}_1^c$  and  $\vec{n}_2^c$  can be created, describing the direction of the sight vector of the image in camera-coordinates. The dot product of the normalized normals and the sight vector of the model  $\vec{p}^c$  provides a measurement between image and model features and the residual per correspondence  $k_p$  becomes

$$\vec{r}^{NE} = \begin{pmatrix} \langle \vec{n}_1^c, \vec{p}^c \rangle \\ \langle \vec{n}_2^c, \vec{p}^c \rangle \end{pmatrix}. \quad (4)$$

The Normal-Space-Error is not depth invariant as well.

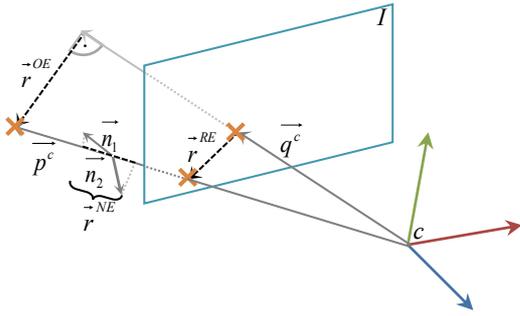


Figure 1: Error measurements for points. Shown are the camera coordinate system  $c$ , the image plane  $I$ , the vectors for model and image feature  $\vec{p}^c$  and  $\vec{q}^c$ , the normals of the Normal-Space-Error  $\vec{n}_1$  and  $\vec{n}_2$  together with their projection on  $\vec{p}^c$  and the error measurements  $\vec{r}^{RE}$ ,  $\vec{r}^{OE}$  and  $\vec{r}^{NE}$ .

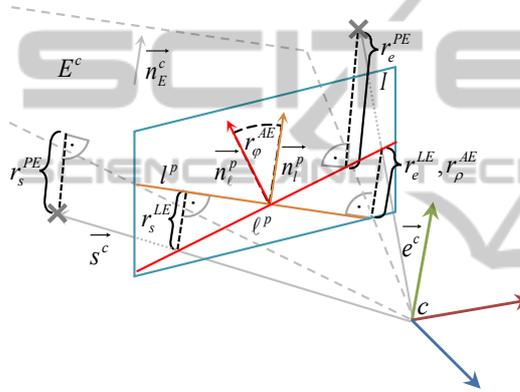


Figure 2: Error measurements for lines. Shown are the camera coordinate system  $c$ , the image plane  $I$ , the model and image feature  $\vec{s}^c$ ,  $\vec{e}^c$  and  $l^p$ , the line  $l^p$ , their normals  $\vec{n}_1^p$  and  $\vec{n}_1^c$ , the plane  $E^c$  with its normal  $\vec{n}_E^c$  and the components of the error measurements  $\vec{r}^{AE}$ ,  $\vec{r}^{LE}$  and  $\vec{r}^{PE}$ .

## 5 ERROR MEASUREMENTS FOR LINES

We investigated three different line residuals which measure the error in different coordinate-systems (Figure 2).

The **Angle-/Distance-Error** measures the differences of the angles and the distances of the image and model feature. The start- and endpoint of the model feature are transformed into their pixel points  $\vec{s}^p$ ,  $\vec{e}^p$  to compute the conjunctive line  $\ell: (\phi^p, d^p)$ , which is compared to the image line  $l^p$ . The residual per correspondence  $k_l$  therefore becomes

$$\vec{r}^{AE} = \begin{pmatrix} \phi^p - \phi^p \\ d^p - \rho^p \end{pmatrix}. \quad (5)$$

Noteable for this error measurement is that the dimensions of the angle and the distance are not equal.

The **Line-Error** measures the distances of the projected start- and endpoint  $\vec{s}^p$ ,  $\vec{e}^p$  to the image line  $l^p$  and is used, among others, by Lowe (Lowe, 1991). Hence, from Eq. 1 the residual for each correspondence  $k_l$  becomes

$$\vec{r}^{LE} = \begin{pmatrix} \cos \phi^p s_x^p + \sin \phi^p s_y^p - \rho^p \\ \cos \phi^p e_x^p + \sin \phi^p e_y^p - \rho^p \end{pmatrix}. \quad (6)$$

The **Plane-Error** can be regarded as the Object-Space-Error extension for straight lines, i.e. the depth recovery of the image feature, and is used e.g. by (Kumar and Hanson, 1994). The optical center and the image line in camera coordinates  $l^c$  define a plane  $E^c$  with normal  $\vec{n}_E^c = (\cos \phi^i \sin \phi^i -\rho^i)^T$ . For start- and endpoint in camera coordinates  $\vec{s}^c$ ,  $\vec{e}^c$  the distance to this plane can be computed. The residual for each correspondence  $k_l$  becomes:

$$\vec{r}^{PE} = \begin{pmatrix} \langle \vec{n}_E^c, \vec{s}^c \rangle \\ \langle \vec{n}_E^c, \vec{e}^c \rangle \end{pmatrix}. \quad (7)$$

Similar to the Object-Space-Error, the Plane-Error is not depth invariant.

## 6 OPTIMIZATION

The residuals of all correspondences are joined in one combined residual  $\vec{r}$ . The pose parameters for translation  $\vec{a}^t$  and rotation  $\vec{a}^R$  are combined in the parameter vector  $\vec{a}$ . The objective is to minimize the sum of the squared entries of this combined residual in terms of the pose parameters  $\vec{a}$ .

$$\operatorname{argmin}_{\vec{a}} \sum_{j=0}^{2m} r_j^2 \quad (8)$$

To solve this problem standard optimization techniques are used so that the parameter vector is turned into a sequence  $\langle \vec{a}_i \rangle$ . Since the Gauss-Newton optimization is comparatively fast but does not guarantee convergence and Gradient-Descent optimization is slow but guarantees convergence, we used the Levenberg-Marquardt optimization that is commonly known as a technique which guarantees convergence but is still comparatively fast. The iteration rule is  $\vec{a}_{i+1} = \vec{a}_i + \vec{\delta}$  with

$$\vec{\delta} = \left( J_{\vec{r}}(\vec{a}_i)^T J_{\vec{r}}(\vec{a}_i) + \lambda D \right)^{-1} J_{\vec{r}}(\vec{a}_i) \vec{r}(\vec{a}_i), \quad (9)$$

$D = \operatorname{diag} \left( J_{\vec{r}}(\vec{a}_i)^T J_{\vec{r}}(\vec{a}_i) \right)$  and  $J_{\vec{r}}$  the Jacobian matrix of  $\vec{r}$ . We found the initial value of  $\lambda = 10^{-3}$  and an alteration rule of  $\lambda_i = \lambda_{i-1}/10$  for a residual decrease and  $\lambda_i = \lambda_{i-1} * 10^w$  with  $w \in \mathbb{N}$  for a residual increase of the last step to work best.

## 7 PARAMETRIZATION

We also investigated the influence and conditions of different parametrizations of the pose. Different parametrizations of the translation vector  $\vec{t}^c$  do not hold any major differences and it is therefore represented by three scalars  $t_x^c$ ,  $t_y^c$  and  $t_z^c$ , describing the translation in camera coordinates; the translation parameter vector is  $\vec{a}^t = \vec{t}^c$ .

For rotation parametrization we investigated four different alternatives. All rotations are described by their corresponding transformation matrices  $R$  and have to fulfill certain rotation properties since  $R \in SO(\mathbb{R}, 3)$ , i.e. the special orthogonal group properties (SOP). In general, these properties can be assured at various stages of the optimization process:

### I. prior to optimization

By choosing an appropriate rotation parametrization the SOP can be assured partly or completely prior to the optimization process.

### II. during optimization

By adding the SOP as additional elements to the residual vector, these properties are optimized with the other residuals. A drawback of this alternative is that with the presence of imperfect or false correspondences the properties are not enforced but only minimized according to the least squares approach of the optimization.

### III. after optimization

By employing the singular value decomposition, a rotation matrix  $\tilde{R}$  can be computed from the estimated transformation matrix  $R$  that assures the SOP and is similar to  $R$ .

These three possibilities are not exclusionary; alternative II. has to be combined with alternative III. for the SOP to be fulfilled.

The parameter vector of the **Matrix-Parametrization** is  $\vec{a}^R = (i_x \ i_y \ i_z \ j_x \ j_y \ j_z \ k_x \ k_y \ k_z)^T$ . The transformation matrix is composed of three vectors  $\vec{i}, \vec{j}, \vec{k} \in \mathbb{R}^3$ :

$$R^M = \begin{pmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \\ k_x & k_y & k_z \end{pmatrix} \quad (10)$$

Advantageous is that the transformation matrix and its derivatives can efficiently be computed. However, the SOP are not enforced and alternatives II. and III. have to be employed: Matrix  $R$  should form an orthonormal basis and have determinant +1. Both constraints are added to the residual.

The parameter vector of the **Euler-Angles-Parametrization** is  $\vec{a}^R = (\phi_x \ \phi_y \ \phi_z)^T$ . The ro-

tation matrix is composed of three consecutive rotations around the coordinate axis:

$$R^E = R_{\phi_z} R_{\phi_x} R_{\phi_y} = \begin{pmatrix} c_y c_z - s_x s_y s_z & -c_x s_z & c_z s_y + c_y s_x s_z \\ c_y s_z + c_z s_x s_y & c_x c_z & s_y s_z + c_y c_z s_x \\ -c_x s_y & s_x & c_x c_y \end{pmatrix} \quad (11)$$

with  $c_x = \cos \phi_x$ ,  $c_y = \cos \phi_y$ ,  $c_z = \cos \phi_z$ ,  $s_x = \sin \phi_x$ ,  $s_y = \sin \phi_y$  and  $s_z = \sin \phi_z$ . The SOP are therefore enforced prior to optimization. Disadvantageous is the well-known gimbal lock and the extensive usage of trigonometric functions.

The parameter vector of the **Quaternion-Parametrization** is  $\vec{a}^R = (w \ x \ y \ z)^T$ . The transformation matrix is composed by using a unit quaternion  $q : (w, x, y, z) \in \mathbb{H}$  with rotation axis  $v = (x, y, z)^T$  and rotation angle  $\phi = 2 \arccos w$ :

$$R^Q = \begin{pmatrix} w^2 + x^2 - y^2 - z^2 & 2(xy - wz) & 2(xz + wy) \\ 2(xy + wz) & w^2 - x^2 + y^2 - z^2 & 2(yz - wx) \\ 2(xz - wy) & 2(yz + wx) & w^2 - x^2 - y^2 + z^2 \end{pmatrix} \quad (12)$$

Since  $q$  has to be a unit quaternion, the SOP are not completely enforced and alternatives II. and III. have to be employed with  $|q| = 1$  added to the residual. It is cogitable to substitute e.g.  $w$  in terms of  $x, y$  and  $z$  as  $w = \sqrt{1 - x^2 - y^2 - z^2}$ . Disadvantageous with quaternions is that there exist complex solutions for  $w$ .

The parameter vector of the **Rodrigues'-Formula-Parametrization** is  $\vec{a}^R = (x \ y \ z)^T$ . The rotation matrix is composed by a rotation axis  $\vec{v} = (x, y, z)^T$  and a rotation angle  $\phi = |\vec{v}|$ :

$$R^R = \begin{pmatrix} \cos \phi + x^2 c_\phi & xy c_\phi - zs_\phi & ys_\phi - xz c_\phi \\ zs_\phi + xy c_\phi & \cos \phi + y^2 c_\phi & yz c_\phi - xs_\phi \\ xz c_\phi - ys_\phi & xs_\phi + yz c_\phi & \cos \phi + z^2 c_\phi \end{pmatrix} \quad (13)$$

with  $c_\phi = \frac{1 - \cos \phi}{\phi^2}$ ,  $s_\phi = \frac{\sin \phi}{\phi}$  and  $\phi = \sqrt{x^2 + y^2 + z^2}$ , or in case of  $\phi = 0$  it holds  $R_R = I_3$ . The SOP are guaranteed by this parametrization.

## 8 SCALING

In general, due to discretization and erroneous matching we cannot rely on perfect correspondences. Therefore, the weight of each correspondence becomes crucial for the estimated result as unequal weights shift the minima of the squared sum (Eq. 8) to different positions. In order to find the optimal least-squares value for all given correspondences, the dimensions of all entries of the residual have to be equal. Except for the Angle-/Distance-Error, all error measurements and SOP have a dimension of either pixel or camera points, and scaling factors for the corresponding other dimension can easily be derived using the given camera matrix.

## 9 GEOMETRICAL CONFIGURATION

The pose problem can be solved with a finite number of solutions when at least three point correspondences are known (Fischler and Bolles, 1981), but there may exist up to four possible solutions. Using four correspondences the pose can be solved with a unique solution only when the correspondences are in coplanar configuration (Gao and Tang, 2006). Otherwise up to five solutions are possible. For five correspondences still up to two solutions may occur when they are not in coplanar configuration. Thus, when using a minimal number of four or five correspondences a coplanar configuration should be assured to prevent multiple solutions. Additionally, the correspondences should not be in degenerate configuration, i.e. they are linearly dependent. For six or more linearly independent point correspondences in arbitrary configuration the pose problem becomes unambiguous. For coplanar lines a finite solution is also feasible with at least three lines. For a non-coplanar configuration at least four lines are required. They must not be in degenerate configuration, i.e. more than two lines are parallel or intersect in one point (Christy and Horaud, 1999). Due to the known 3D data of the model, for each correspondence the geometrical relationship to the others in the correspondence set can be surveyed.

Table 1: Possible number of solutions for several numbers of point correspondences and their configuration.

#correspondences	non-coplanar	coplanar
3	-	4
4	5	1
5	2	1
$\geq 6$	1	1

To check for *linear dependency*, i.e. three points located on one straight line, the dot product of the normalized direction vectors from one point to the other two is calculated. If it becomes close to 1, the correspondence will be rejected for pose estimation. To check for *coplanarity*, a plane is calculated from the normalized direction vectors of three points. The vector of the remaining fourth point is then inserted into the equation of the plane. If the four points are in coplanar configuration, the solution will be unique and the correspondences are accepted as input data.

For straight lines linearly dependent configurations occur if more than two lines are parallel or intersect in one point. *Parallelism* is checked for by calculating the dot product of the normalized direction vectors of the lines. If all products are close to 1, the lines are linearly dependent and the line correspon-

dences will be rejected. To check for a *common intersection point*, the intersection point of each line pair is calculated respectively. If two or more intersection points exist and the euclidean distance between these points is below a predefined threshold, the lines are collinear and will not be candidates for input data.

## 10 RESULTS

We tested combinations of error measurements and parametrizations using points, lines and mixed correspondences for the non-linear estimator. For the experiments with synthetical data at an image resolution of 640 x 480 pixel we used a unit cube around the origin; the initial camera pose was  $(0, 0, -5)^T$ . In the test sequence the camera was rotated by  $30^\circ$  about the axes of the world coordinate system and translated by the vector  $(1, 1, 1)^T$ . Using this pose the correspondences were established by perspective projection of the 3D model features to the image plane, followed by adding uniformly distributed noise in the range of  $[0, 10]$  pixel of displacement in arbitrary direction. We regarded mean error and standard deviation between real and estimated pose for translation, rotation and number of iterations over 100.000 tests.

On points the comparison of error measurements indicates that the image based Reprojection-Error performs more accurately than the Object-Space-Error and Normal-Error for all possible combinations with pose parametrizations. This can be explained by a potentially higher weighting of points in the object space when they are more distant from the camera center. The number of iterations is comparable for all types of error measurement. Concerning parametrization, Euler computation is slightly slower while quaternions perform somewhat better, but lead to less accurate translation estimation. For line correspondences, the Angle-Distance-Error causes bad pose results and a higher computation time. Line- and Plane-Error are comparable for all parametrizations, regarding accuracy and speed. Quaternions show a slightly better computational performance.

Therefore, we propose Rodrigues'-Formular-Parametrization together with Reprojection- and Line-Error for combined non-linear pose estimation. This combination shows the smallest pose error without noticeable decrease in speed. In addition, using pixel space error for both feature types, scaling is not necessary to compensate for depth differences between the correspondences. The Rodrigues'-Formula proves to be the best parametrization because independent of the chosen error measurement it does not influence pose accuracy. It is gimbal-lock free and

complies with the constraints of a rotation matrix by definition. Thus, there is no need for additional entries in the residual vector to be minimized during optimization and no singular value decomposition has to be run afterwards to correct the result.

We also tested the influence of combining both feature types under our proposed parameters. We combined a minimal number of six correspondences of one feature type with several numbers of the other one and added several levels of noise. A pose estimated by points is improved in rotation and translation by adding line correspondences independently of the displacement error in the given point correspondences. The improvement of the estimate converges when at least six line correspondences are added to the point correspondences. In the opposing case additional point correspondences show a positive impact on the exactness of a pose estimated with lines only when the underlying line correspondences are very noisy. This is due to a generally higher stability of line features concerning displacement error. Point features will be affected more strongly by growing correspondence errors than lines. We can state that a combination of point and line features is useful in practical application to stabilize the estimated pose, especially when there is only a minimal set of correspondences available.

The resulting optimal estimator was successfully run on a real test scene for verification of real-time capabilities. The estimator pass for combined input with point and line correspondences took about 1 ms CPU time. The combination of both feature types did not reduce computational speed, thus real time application is ensured. Further, it could be seen that the required number of iterations depends to a large degree on the error level of the correspondences, while it is hardly influenced by the total number of correspondences.

## 11 CONCLUSIONS

We analyzed the best parameter choice for a non-linear pose estimator when using a combined input of point and line correspondences. Test results show that the error measurement in pixel coordinates is superior to the error in object space for points as well as for lines. Further, it is proved that a parametrization may be chosen which fulfills the constraints of a rotation matrix without requiring additional computational load. For points and lines this is the case with Rodrigues parametrization. An optimal non-linear estimator can be constructed by these propositions working on an arbitrary number and choice of feature

type with a minimal set of three correspondences. The estimator will improve the pose by considering the configuration of the combined input data and selecting those point and line correspondences only, which are proved to deliver unambiguous results.

## ACKNOWLEDGEMENTS

This work was supported by grant no. MU 2783/3-1 of the German Research Foundation (DFG).

## REFERENCES

- Ansar, A. and Daniilidis, K. (2003). Linear Pose Estimation from Points or Lines. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 25, pages 578–589.
- Christy, S. and Horaud, R. (1999). Iterative Pose Computation from Line Correspondences. *Computer Vision and Image Understanding*, 73:137–144.
- Dornaika, F. and Garcia, C. (1999). Pose Estimation using Point and Line Correspondences. *Real-Time Imaging*, 5:215–230.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395.
- Gao, X.-S. and Tang, J. (2006). On the Probability of the Number of Solutions for the P4P Problem. *Journal of Mathematical Imaging and Vision*, 25:79–86.
- Haralick, B., Lee, C.-N., Ottenberg, K., and Nille, M. (1994). Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356.
- Hu, Z. Y. and Wu, F. C. (2002). A Note on the Number of Solutions of the Noncoplanar P4P Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:550–555.
- Kumar, R. and Hanson, A. (1994). Robust methods for estimating pose and a sensitivity analysis. *Computer Vision Graphics and Image Processing: Image Understanding*, 60(3):313–342.
- Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). EPnP: An Accurate  $O(n)$  Solution to the PnP Problem. *International Journal of Computer Vision*, 81:155–166.
- Lowe, D. (1991). Fitting parameterized three-dimensional models to images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13(5), pages 441–450.
- Lu, C.-P., Hager, G., and Mjolsness, E. (2000). Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622.
- Wolfe, W. J., Mathis, D., Sklair, C. W., and Magee, M. (1991). The Perspective View of Three Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:66–73.