In Proc. 1st Austrian Cognitive Vision Workshop,
volume 186 of OCG Schriftenreihe, pages 51-58, 2005

Robust Pose Estimation from a Planar Target

Gerald Schweighofer and Axel Pinz
Inst. of El. Measurement and Measurement Signal Processing
Graz University of Technology, Austria
email: gschweig@emt.tugraz.at, axel.pinz@tugraz.at

Abstract:
This paper presents a novel way to visualise the ambiguity of the pose estimation problem. By doing this we show that there exist two local minima of the error function when using planar targets. These two minima are the reason why pose jumps are observed in many tracking applications. We further explain these minima and link them to the solutions of the P3P algorithm. Based on this we present a new "Robust Pose Estimation Algorithm for Planar Targets" which takes these two minima into account to give a robust pose.

keywords: tracking, pose ambiguity

1 Introduction

There are many applications of pose estimation, where 6 degrees of freedom of a camera's pose have to be calculated from known correspondences with known scene structure. This can be done from a single image or from an image sequence. In photogrammetry this problem is known as space resectioning and it is often solved offline by bundle adjustment techniques, achieving very high precision and at the same time high robustness against outliers. In general, there are many good solutions, as long as many points can be used, and when the pose can be calculated offline (e.g. taking information from frames $n + k$ into account to calculate the pose for frame n). Recent success has also been reported for online structure *and* motion estimation [7], where many interest points are extracted, frame-to-frame correspondence is rather easy, and no a priori reference to a scene coordinate system is required (for early work in this direction see [2]).

Pose tracking for cognitive vision and for similar applications requires a system which works online, identifies a limited number of visual landmarks on each of the objects, estimates the pose for each object and the background (which could also be an object), and relates the camera poses to a scene coordinate system. In most cases the background is planar (table, wall, points at infinity), but also the objects are sometimes planar (depending on the object and on the viewpoint of the camera).

Examples of vision-based tracking systems, which are often based on planar targets, include ARToolkit ([3], widely used for many applications), Malik et al. ([6], they claim to be more precise than ARToolkit due to an improved target design), and our own developments ([1], fusing inertial and vision sensors to increase robustness). Users of such systems observe that vision-based pose is not very precise, which results in significant jitter, and not very robust (pose jumps, gross pose outliers). Interest points are extracted from a target, and the camera pose is calculated relative to the target's pose in the scene. In theory, pose can be calculated from 4 or more coplanar and not collinear points, if the intrinsic parameters of the camera are known. For an uncalibrated camera there remain at least two solutions, even if many coplanar points are available [5]. In the Computer Vision literature, several approaches to pose estimation are known. Most of them work for coplanar points, some have been extended to use points and lines, and some work for arbitrary 3D target point configurations.

We have performed a number of experiments comparing several pose algorithms and planar target configurations. Even if we use a well calibrated camera and avoid known critical configurations, we observe pose jumps, which should not occur, according to the literature. This paper explains the reasons for these pose ambiguities, gives a comprehensive interpretation of the different solutions (mirrored symmetry), and proposes a new algorithm for a unique and robust solution of the planar pose problem.

2 The Pose Estimation Problem

Camera pose estimation problem is the problem of finding the exterior parameters of the camera (orientation $R = f_1(\alpha, \beta, \gamma)$ and position $\mathbf{t} = [t_x, t_y, t_z]^T$), which minimise an error function. Thus, a point $\mathbf{p}_i$ in scene coordinates is transformed to camera coordinates $\mathbf{v}_i$ by

$$\mathbf{v}_i = R\mathbf{p}_i + \mathbf{t}. \tag{1}$$

While bundle-adjustment methods mostly use the *image-space* error function (equation 2), there are also pose estimation algorithms which use an *object-space* error function (equation 3). $\hat{v}$ denotes measurements in image space.

$$E_{is}(R, \mathbf{t}) = \sum_{i=1}^{n} \left[\left(\frac{\hat{v}_{ix}}{\hat{v}_{iz}} - \frac{R_x^t \mathbf{p}_i + t_x}{R_z^t \mathbf{p}_i + t_z} \right)^2 + \left(\frac{\hat{v}_{iy}}{\hat{v}_{iz}} - \frac{R_y^t \mathbf{p}_i + t_y}{R_z^t \mathbf{p}_i + t_z} \right)^2 \right] \tag{2}$$

$$E_{os}(R, \mathbf{t}) = \sum_{i=1}^{n} \left\| \left(I - \frac{\hat{\mathbf{v}}_i \hat{\mathbf{v}}_i^t}{\hat{\mathbf{v}}_i^t \hat{\mathbf{v}}_i} \right) (R\mathbf{p}_i + \mathbf{t}) \right\|^2 \tag{3}$$

The main problem in comprehension of the pose estimation problem is that there exists no visualisation method for these error functions.

In this paper we show that there is a way to visualise the *object-space* error function (equation 3). By choosing $R = R_z(\gamma)R_y(\beta)R_x(\alpha)$ ($R_i(j)$ describes a rotation of j degrees about axis i) we can show that the optimal values w.r.t to the minimisation problem for the four parameters (γ, t_x, t_y, t_z) only depend on α and β.

$$E_{os}(\gamma, \beta, \alpha, t_x, t_y, t_z) \Longrightarrow F_{os}(\alpha, \beta) \tag{4}$$

The function E_{os} which depends on six parameters and thus cannot easily be visualised, can be written as a function F_{os} which depends only on two variables and can be visualised in 3D. Full derivation of F_{os} is given in Appendix A.

2.1 Visualisation of the Pose Estimation Problem

For our experiments we randomly generated 20 points in a plane. They are then projected onto an image with known internal camera parameters. (external camera parameters: $[\alpha, \beta, \gamma] = [17.6822, -55.5935, -5.4510]$ and $[t_x, t_y, t_z] = [0.0028, -0.2409, 1.0145]$).

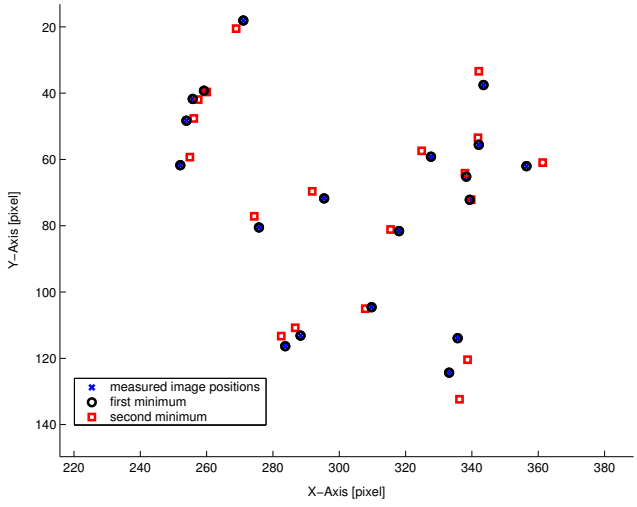
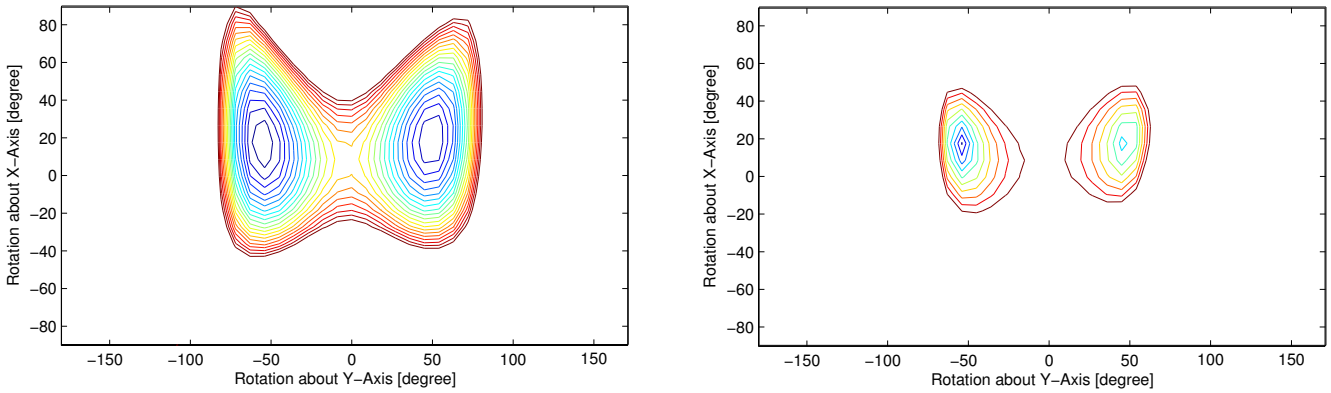


Figure 1: Image used for the experiment.

In fig. 1 we see the image of our randomly generated model (crosses) visualised with the given external camera parameters.

Figure 2 (a) visualises the error function E_{os}. We see two local minima. To understand how these minima look like, we visualised in fig. 1 the projection of the model with the external parameters of both minima. Because we have a noise-free situation the first local minimum corresponds exactly to the correct pose $[\alpha, \beta] = [17.7, -55.6]$.



(a) (b)

Figure 2: Contour plots of the error functions: (a) E_{os} (b) E_{is}.

The other minimum of the error function is the reason why people observe pose jumps with standard pose estimation algorithms. Since pose estimation algorithms which minimise an error function are always iterative algorithms, they need start values for their parameters. If these start values are close to the second (wrong) pose, then the iterative algorithm will converge to the wrong result. In fig. 2 (b) we visualised the error function E_{is} to show that the second (wrong) local minimum does not depend on a specific error function.

3 Analysis of the two local minima

While doing a lot of experiments and trying different pose estimation algorithms we found that the two local minima of the error function correspond exactly to the two (out of max. four) solutions of the P3P algorithm. Imagine the situation where we have only three points. In this case we get two valid solutions which could both be a true configuration of the problem. Adding now a fourth point to this scenario we have four possibilities of choosing three points out of the four. Within the solutions of the P3P algorithm we recognise that one of the solutions is always the same for the different configurations. This is the true solution. What we also recognise is that the other solutions are not exatly the same, but similar. Which means that if we use an error function we will get a local minimum somewhere close to the second solution of the P3P algorithm.

To verify this we used the same setup as before. We used all points to estimate a first guess of the pose. In fig. 3 we visualised this pose with a black x. For all possibilities of choosing three points we caculated for each configuration the solutions of the P3P algorithm. One of the solutions coincides with the first pose. The other solutions are visualised as red circles.

In fig. 3 (a) we see the results without any error in the image corrdinates. By comparing with fig. 2, which shows the error functions, we observe that the results are compact and close to the second local minimum. By adding Gaussian noise to the image points (fig. 3 (b)-(f)) we

see that the results begin to move away from the second minmum, but they are always located around the minimum.

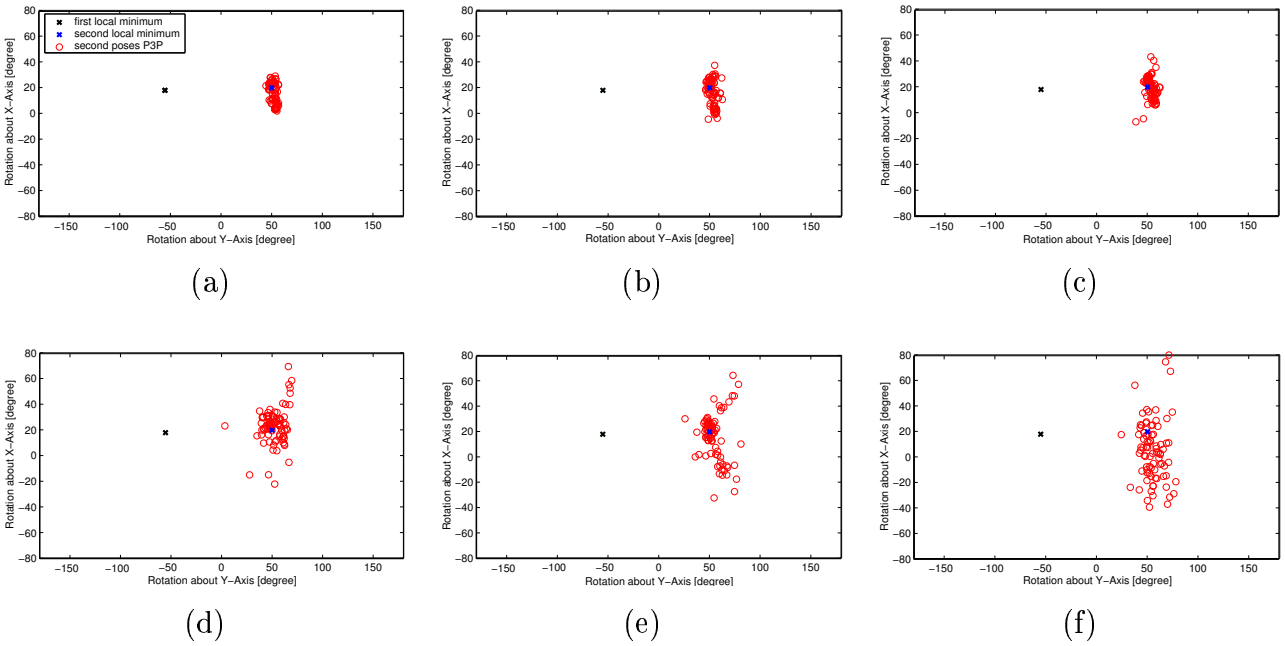


Figure 3: Estimation of second solution with P3P. Different gaussian noise e [pixel]: (a) $e = 0$; (b) $e = 2$; (c) $e = 4$; (d) $e = 6$;(e) $e = 8$; (f) $e = 10$.

4 Robust Pose Estimation Algorithm

Our new algorithm is based on the two major observations discussed above: There are two local minima, and the correct solution to the pose estimation problem has the lower error value. We thus propose the following "Robust Pose Estimation Algorithm for Planar Targets":

1. Estimate a first pose $P_1 = (R_1, \mathbf{t}_1)$ with any existing pose estimation algorithm. In our experiments we used the iterative algorithm proposed by [4].

2. From this pose $P_1 = (R_1, \mathbf{t}_1)$ estimate a guess for the second pose P_2^* with the P3P algorithm. This pose will be close to the second local minimum of the error function.

3. Use pose P_2^* as a start value for the iterative pose estimation algorithm to get P_2.

4. The final and correct pose is the one which has the lowest error.

4.1 Experiment

We used the same setup as in section 2.1. Gaussian noise was added to the image points. For each *noise-level* the experiment was repeated 100 times. The iterative algorithm was initialised with a randomly generated pose. Figure 4 displays the results of the experiment.

Using the algorithm proposed by [4] we have only a probability of about 50 % to get the correct pose. Using our proposed algorithm we get the correct solution up to a noise level of 2.5 pixel. By increasing the noise over 2.5 pixel the probability of finding the correct solution decreases to 80 % at 10 pixel noise in the image.

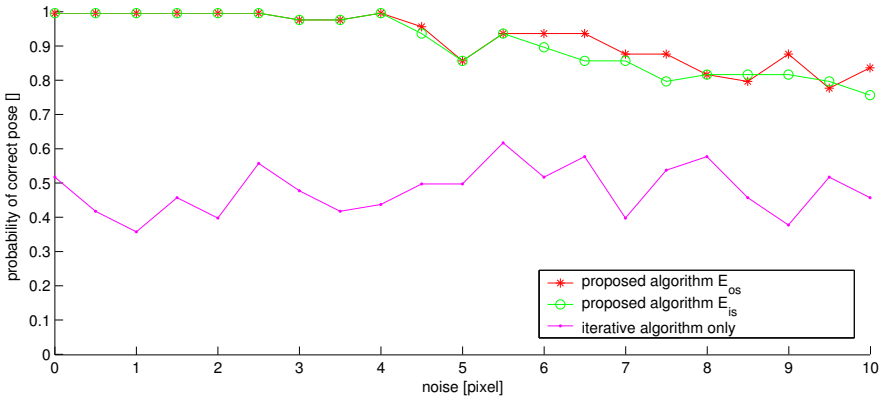


Figure 4: Probability of choosing the correct Pose.

5 Conclusion

We have presented a way to visualise the ambiguity of the pose estimation problem. By doing this we have shown that there exist two local minima of the error function when using planar targets. These two minima are the reason why pose jumps are observed in many tracking applications. We then explained these minima and linked them to the solutions of the P3P algorithm. Based on these findings, we presented a new "Robust Pose Estimation Algorithm for Planar Targets" which takes the two minima into account to give a robust pose. To our knowledge and experimental evidence this is the first algorithm which is robust against pose jumps in real applications.

Acknowledgements

This work was supported by the Austrian Science Foundation (FWF, project S9103-N04) and (FWF, project P15748).

Appendix A

Starting with the *object-space* error function (equation 3), we can solve for the optimal translation $\mathbf{t}$ w.r.t. to the minimisation of E_{os} by derivating equation 6 (equation 7) .

$$\hat{V}_i = \frac{\hat{\mathbf{v}}_i \hat{\mathbf{v}}_i^t}{\hat{\mathbf{v}}_i^t \hat{\mathbf{v}}_i} \tag{5}$$

$$E_{os}(R,\mathbf{t}) = \sum_{i=1}^{n} \|(I-\hat{V}_i)(R\mathbf{p}_i+\mathbf{t})\|^2 \tag{6}$$

$$\frac{\partial E_{os}}{\partial \mathbf{t}} = 0 \;\Rightarrow\; \mathbf{t}_{opt}(R) = \frac{1}{n}\left(I - \frac{1}{n}\sum_j \hat{V}_j\right)^{-1} \sum_j \left(\hat{V}_j - I\right) R\mathbf{p}_j = G\sum_j \left(\hat{V}_j - I\right) R\mathbf{p}_j \tag{7}$$

In equation 7 G is a constant and depends only on measured image positions $\hat{\mathbf{v}}_i$.
Inserting $\mathbf{t}_{opt}(R)$ into (equation 6) we get an error function which only depends on the rotation matrix R:

$$\begin{aligned} E_{os}(R,\mathbf{t}_{opt}(R)) &= \sum_{i=1}^{n}\|e_i\|^2 = \sum_{i=1}^{n}\|(I-\hat{V}_i)(R\mathbf{p}_i + t(R))\|^2 \\ &= \sum_{i=1}^{n}\|(I-\hat{V}_i)(R\mathbf{p}_i + G\sum_{j=1}^{n}(\hat{V}_j - I)R\mathbf{p}_j)\|^2 = E_{os}(R) \end{aligned} \tag{8}$$

Parameterising R as a composition of two rotation matrices R_{yx} and R_z

$$R = R_z(\gamma)R_{yx}(\alpha,\beta)$$

$$R_{yx}(\alpha,\beta) = R_y(\beta) * R_x(\alpha) = \begin{bmatrix} r_1 & r_2 & r_3 \\ 0 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \qquad R_z(\gamma) = \begin{bmatrix} z_1^2 - z_2^2 & -2z_1z_2 & 0 \\ 2z_1z_2 & z_1^2 - z_2^2 & 0 \\ 0 & 0 & z_1^2 + z_2^2 \end{bmatrix}$$

and inserting this into (equation 8) gives us:

$$E_{os}(R) = \begin{bmatrix} z_1^4 & z_1^3z_2 & z_1^2z_2^2 & z_1^1z_2^3 & z_2^4 \end{bmatrix} \cdot \begin{bmatrix} f_1(\hat{V}_{1\ldots n}, r_{1\ldots 9}) \\ f_2(\hat{V}_{1\ldots n}, r_{1\ldots 9}) \\ f_3(\hat{V}_{1\ldots n}, r_{1\ldots 9}) \\ f_4(\hat{V}_{1\ldots n}, r_{1\ldots 9}) \\ f_5(\hat{V}_{1\ldots n}, r_{1\ldots 9}) \end{bmatrix} \tag{9}$$

with functions $f_1..f_5$ only depending on measured image positions $\hat{V}_{1\ldots n}$ and the two rotation parameters $r_{1\ldots 9} = f_{ri}(\alpha,\beta)$.

To be $R_z(z_1,z_2)$ a valid rotation, it must fulfill

$$z_1^2 + z_2^2 = 1 \Leftrightarrow z_1 = sin(\gamma); z_2 = cos(\gamma).$$

Writing $sin(\gamma)$ and $cos(\gamma)$ as a function of tangents and substituting we get:

$$\gamma_t = \tan\frac{1}{2\gamma} \tag{10}$$

$$z_1 = \sin\gamma = \frac{2\gamma_t}{1+\gamma_t^2} \tag{11}$$

$$z_2 = \cos\gamma = \frac{1-\gamma_t^2}{1+\gamma_t^2} \tag{12}$$

By plugging this into (equation 9) we get a function $E(\gamma_t)$ which now only depends on γ_t.

$$\begin{aligned}
\frac{\partial E_{os}(\gamma_t)}{\partial \gamma_t} &= 0 \Rightarrow \\
0 &= 2f_2\gamma_t^8 + (16f_1 - 8f_3)\gamma_t^7 + (-32f_2 + 24f_4)\gamma_t^6 \\
&+ (-48f_1 - 64f_5 + 56f_3)\gamma_t^5 + (-80f_4 + 60f_2)\gamma_t^4 \\
&+ (64f_5 - 56f_3 + 48f_1)\gamma_t^3 + (-32f_2 + 24f_4)\gamma_t^2 + (8f_3 - 16f_1)\gamma_t + 2f_2
\end{aligned} \tag{13}$$

Given a pose-estimation problem, ie. model points $\mathbf{p}_i$ and image points $\mathbf{v}_j$, it is now possible to solve equation 13 for each given pair of α and β to get an optimal value for γ_t. Which means solving for the best rotation about the optical axis of the camera to get the final rotation R_f. By using equation 7 we get the translation vector $\mathbf{t}_f$ for this rotation. Evaluating equation 6 gives us an error-value of function E_{os} which only depends on α and β.

References

[1] M. K. Chandraker, Ch. Stock, and A. Pinz, *Real-time camera pose in a room*, 3rd Int Conf. Comp. Vision Systems ICVS (2003), 98–110.

[2] Chris Harris, *Geometry from visual motion*, Active Vision (Andrew Blake and Alan Yuille, eds.), MIT Press, 1992.

[3] H. Kato, K. Tachibana, M. Billinghurst, and M. Grafe, *A registration method based on texture tracking using artoolkit*, Augmented Reality Toolkit Workshop (2003), 77–85.

[4] C.P. Lu, G.D. Hager, and E. Mjolsness, *Fast and globally convergent pose estimation from video images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (June 2000), no. 6, 610–622.

[5] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *Euclidian reconstruction and reprojection up to subgroups*, The Proceedings of the Seventh IEEE International Conference on Computer Vision **2** (1999), 773–780.

[6] S. Malik, C. McDonald, and G. Roth, *Hand tracking for interactive pattern-pased augmented reality*, ISMAR. Proceedings (2002), 117–126.

[7] D. Nister, O. Naroditsky, and J. Bergen, *Visual odometry*, IEEE Conference on Computer Vision and Pattern Recognition **1** (2004), 652–659.