January 1998

A New Metric for Object Pose Estimation

Jeffrey Mendelsohn
University of Pennsylvania

A New Metric for Object Pose Estimation

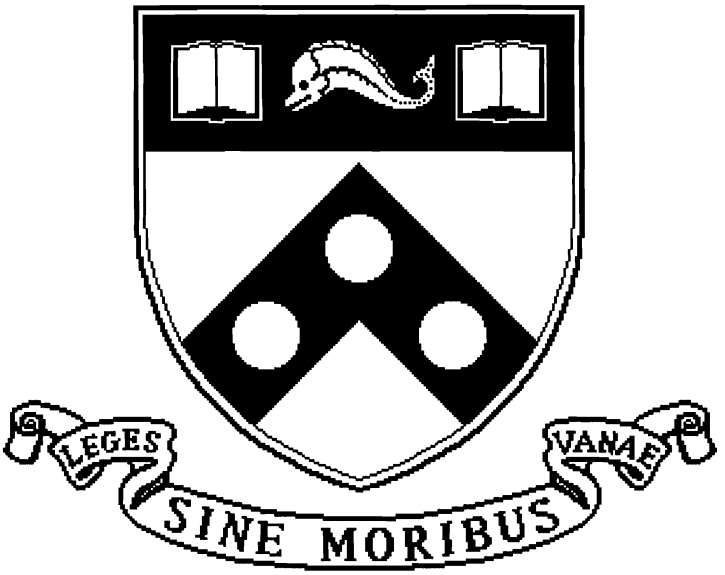
Abstract
Object pose estimation is a difficult task due to the non-linearities of the projection process; specifically with regard to the effect of depth. To overcome this complication, most algorithms use an error metric which removes the effect of depth. Recently, two new algorithms have been proposed based upon iteratively improving pose estimates obtained with weak-perspective or paraperspective approximations of the projection equations. A simple technique for improving the estimates of the two projection approximation algorithms is presented and a new metric is proposed for use in 'polishing' these object pose estimates. At all distances, the new algorithm reduces the estimated orientation error by over ten percent. At short distances, the orientation improvement is about seventeen percent and the position error is reduced by twelve percent.

Comments
University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-98-17.

A New Metric for Object Pose Estimation

MS-CIS-98-17

Jeffrey Mendelsohn



University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

1998

A New Metric for Object Pose Estimation*

Jeffrey Mendelsohn

University of Pennsylvania, Philadelphia PA 19104-6228, USA

Abstract. Object pose estimation is a difficult task due to the non-linearities of the projection process; specifically with regard to the effect of depth. To overcome this complication, most algorithms use an error metric which removes the effect of depth. Recently, two new algorithms have been proposed based upon iteratively improving pose estimates obtained with weak-perspective or paraperspective approximations of the projection equations. A simple technique for improving the estimates of the two projection approximation algorithms is presented and a new metric is proposed for use in 'polishing' these object pose estimates. At all distances, the new algorithm reduces the estimated orientation error by over ten percent. At short distances, the orientation improvement is about seventeen percent and the position error is reduced by twelve percent.

1 Introduction

The problem of object pose estimation will be defined as finding the rigid transformation from the object frame to the camera frame given the camera projection model and calibration information, a set of points described in an object frame, and the projection of these points.

Object pose estimation has many uses in computer vision: object positioning, docking (moving the camera to a set transformation from the target), camera calibration, and cartography. The key difficulties in solving the problem stem from the constraints of a rotation matrix and having only the projection of the object points for data. Furthermore, as in all real world situations, the data has noise. For the object pose problem, a primary source of noise is the localization error; a measure of how well the data points correspond to the true projection of the object points.

Previous approaches to solving this problem can be classified into two broad categories: closed-form solutions and numerical solutions. Closed-form solutions make use of a finite number of correspondences and solve the object pose problem by directly solving for the transformation parameters in the set of projection equations. Such solutions exist for three points [2], four coplanar points [7], and four points in general position [4, 5]. While it is possible to derive closed-form solutions to overconstrained pose estimation problems, it is exceedingly difficult since the equations involve non-linear constraints. This is addressed by the numerical solutions.

Ganapathy [3] proposed a linear solution on the assumption that the constraints on the rotation matrix need not be imposed; the rotation matrix is given nine degrees of freedom. This algorithm is extremely susceptible to noise mainly because the orthogonality constraints are ignored. Numerical methods that correctly constrain the problem [5, 8, 10] require a good initial estimate of the transformation parameters. This is a major limitation created, primarily, by the minimization technique. A state-of-the-art pose estimator by Phong et al. [9] uses a trust-region minimization technique which provides an excellent convergence rate and, essentially, removes the need for an initial estimate.

A relatively new sub-class of the numerical solutions can be defined as solving the pose estimation problem under an approximation to the desired projection model [1, 6]. For instance, using weak-perspective

* J. Mendelsohn is supported by NSF grant GER93-55018.

Univ. of Pennsylvania Computer Information Science Technical Report MS-CIS-98-17. June 17, 1998.

or paraperspective projection as approximations to perspective projection. Clearly, it is unlikely for an approximation method to provide as accurate results as a method based on the true projection model. This is mostly overcome by iteratively moving the image points towards those that would have been produced by the approximation model's projection equations. While these algorithms do not converge for very close objects, in practice this is not a concern. Furthermore, these algorithms do not enforce the constraints of the transformation's rotation matrix. In total, the execution time required for pose estimation is dramatically reduced at the expense of accuracy.

2 Projection Models

Three projection models will be discussed: perspective, weak perspective, and paraperspective. Perspective projection is the 'standard' model. The other two are linear approximations to perspective projection.

The following notation will be used:

$\mathbf{x_i}$	object frame coordinates of point i,
u_i, v_i	perspective projection of point i,
u_i^w, v_i^w	weak-perspective projection of point i,
u_i^p, v_i^p	paraperspective projection of point i,
$\mathbf{R}$	rotation matrix,
$\mathbf{r_1}$, $\mathbf{r_2}$, $\mathbf{r_3}$	the rows of $\mathbf{R}$,
$\mathbf{t}$	translation vector,
t_x, t_y, t_z	components of the translation vector,
and $\hat{\mathbf{z}}$	unit vector along the optical axis.

The conversion from the imaged points to $[u_i, v_i]^T$ is dependent only upon the calibration model and calibration parameters. It has been assumed that these are known and hence the conversion can be performed to provide the data points for the object pose estimation problem.

Sections 2.1 through 2.3 describe the projection models with regard to pose estimation. A pictorial comparison of the projection models is provided in Figure 1.

2.1 Perspective Projection

Combining the perspective projection equations with the change of coordinate system from the object frame to the camera frame yields:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} (\mathbf{R}\mathbf{x_i} + \mathbf{t}) \frac{1}{\hat{\mathbf{z}}^\mathbf{T} (\mathbf{R}\mathbf{x_i} + \mathbf{t})}$$

The effect of depth makes these equations difficult to use in minimization techniques. To overcome this, both sides are multiplied by the depth term and then the equation is simplified:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -u_i \\ 0 & 1 & -v_i \end{bmatrix} (\mathbf{R}\mathbf{x_i} + \mathbf{t})$$

Phong et al. [9] uses the square of these equations, plus two terms with Lagrange multipliers to enforce the constraints of the transformation, as an error metric.

For comparison with the other algorithms, these equations will be rewritten by dividing through by t_z and rearranging the terms:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & -u_i \\ 0 & 1 & -v_i \end{bmatrix} \mathbf{R}\mathbf{x_i} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right) \frac{1}{t_z} \tag{1}$$

2.2 Weak Perspective Projection

Obtaining the weak perspective projection of a set of points is a two step process. First the object points are projected onto a plane, herein called the reference plane, that is frontal parallel to the image plane. This projection is done by finding the intersection of the line parallel to the optical axis through the object point with the reference plane. These new points are then projected onto the image plane as per the perspective projection model; by dividing by the depth. In practice, the reference plane is chosen to contain the transformation of the object frame's origin.

$$\begin{bmatrix} u_i^w \\ v_i^w \end{bmatrix} = \left(\begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \end{bmatrix} \mathbf{R}\mathbf{x_i} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right) \frac{1}{t_z} \tag{2}$$

2.3 Paraperspective Projection

The paraperspective projection of a set of points is obtained in a similar manner. The only difference is instead of using lines parallel to the optical axis to reach the reference plane, lines parallel to the translation vector are used.

$$\begin{bmatrix} u_i^p \\ v_i^p \end{bmatrix} = \left(\begin{bmatrix} 1\,0\,-\frac{t_x}{t_z} \\ 0\,1\,-\frac{t_y}{t_z} \end{bmatrix} \mathbf{R}\mathbf{x_i} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right) \frac{1}{t_z} \tag{3}$$

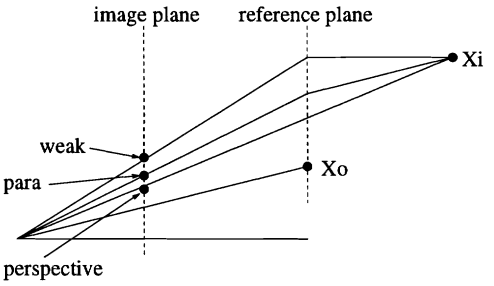


Fig. 1. Comparison of the Projection Models

3 Proposed Method

Equations 1 through 3 provide three linearizations of the perspective projection model. On the surface, the only difference between the three sets of equations are the measurements on the left hand side and how the point is projected onto the frontal-parallel plane through the translation vector. If there were no noise, Equation 1 correctly projects the points to the plane and then to the sensor. In the presence of noise, Equation 1 can be interpreted two ways. The first is that the noisy measurements are coupled with the known data; this would lead to a very difficult estimation problem which is clearly not solved in the literature. The second is that the projection is an approximation to perspective projection where, instead of one vector being used to project all the points to the frontal-parallel plane, a set of points are used to project the points to the frontal-parallel plane. In essence, if these points are closer to the 'true' projection than the projection of

the translation vector, a better approximation to the perspective projection equations is obtained relative to paraperspective projection. However, the projection model does not iteratively improve to the perspective model implying an algorithm based upon such a metric can not 'correctly' solve the problem in the presense of localization error. To correctly solve the problem, the equations must be modified so that the projection values in the right hand side reflect the values obtained by the current model. In contrast, Equation 2 and Equation 3 can be used to iteratively obtain an exact solution in theory.

Also, there is - in a sense - a greater need to constrain the rotation matrix correctly in Equation 1 than in Equation 2 and Equation 3. If the matrix is unconstrained, Equation 1 uses nine unknowns to represent four values while Equation 2 and Equation 3 use only six unknowns for four values.

Theoretically, it is preferable to use the weak-perspective or paraperspective method due to the expected behavior given localization noise and their low execution time. In practice, the non-linear algorithm provides more accurate results. The projection approximation algorithms have two weaknesses that can be readily identified and improved: distortion of the true perspective projection error metric and maintenance of the rotation matrix constraints.

3.1 Simulation Method

Before evaluating and comparing the algorithms in a quantitative manner, the framework for comparison must be defined.

The object used in the simulations is a cube of width one hundred millimeters. The eight corners of the cube plus the centroid of the cube are the data points and the object is assumed to be a wire-frame so that all nine points are always imaged. The correspondence between the model points and the image points is assumed known.

The camera used to image the object had a focal length of 8.5 millimeters. The imaging device has dimensions of 320×240 pixels; one pixel is equivalent to 0.0275 millimeters.

The object is given a random rotation by choosing an axis of rotation and an angle. The axis is chosen by first taking a vector comprised of three components selected from a uniform distribution over the range $[0, 1)$ and then the vector is normalized. The angle is chosen from a uniform distribution over $[-\pi, \pi)$.

The depth of the object's transformed origin is the parameter varied over in the simulations. The other two components of the translation vector are chosen from a uniform distribution over the imaging device, and scaled by the depth divided by the focal length.

The projection of the points is then performed. If any point of the object is not on the imaging sensor, the translation and rotation parameters are reselected.

The localization error is assumed to be Gaussianly distributed with a standard deviation set in the simulation. The noise is added in the plane containing the imaging device.

Finally, each point is divided by the focal length to provide the measurements used in pose estimation.

In the literature, two metrics for determining goodness-of-fit are used; relative position error and orientation error. Relative position error is defined as taking the magnitude of the vector between the true and estimated translation vectors and dividing by the the magnitude of the true translation vector. Orientation error is computed by finding the angle of the rotation between the true and estimated rotation matrices. For this purpose, the resultant rotation matrices of the algorithms are always orthonormalized.

For each depth value, one thousand trials are performed and the average for both metrics is reported. For each algorithm at each depth, twenty iterations are performed in the minimization; this is significantly more than is needed for convergence.

Along with absolute results, graphs representing comparisons between algorithms are presented; these graphs

have line labels such as "polished / weak". This denotes that the graph is of the relevant metric value produced by the polished algorithm divided by the metric value for the weak-perspective projection algorithm.

3.2 Reducing Error Metric Distortion

In both the weak-perspective and paraperspective algorithms, the projection approximation is calculated by multiplying the noisy image measurement by a factor and then, in the paraperspective case, adding an offset. Clearly, the noise in each point of the approximation projection model has been scaled and, by not removing this effect, the estimation is inappropriately biased towards reducing the error at points further away. For both algorithms, this can be accomplished by simply multiplying every equation by:

$$\frac{t_z}{t_z + \hat{\mathbf{z}}^{\mathbf{T}}\mathbf{R}\mathbf{x_i}}$$

before squaring for the error metric. Since the change to the metric is more significant when points are closer, it is expected that the improvement in performance will be more significant when the object distance to size ratio is small.

While this does not completely remove the error metric distortion, during simulations it improved the orientation and position metric results for both algorithms in the shortest distances by about ten percent.

3.3 Polishing the Estimate

If the error-free values of the imaged data were known, the error metric represented by Equation 1 can be modified so as to be ideal in terms of providing a metric for pose estimation. Possibly, by using sufficiently good estimates - by having good initial values for the pose estimation - of these imaged values, the modified metric can be used to iteratively obtain the true pose estimation parameters. Without a good initial guess, like most other non-linear algorithms, this algorithm will fail. From the results seen for the modified paraperspective algorithm, it is clear that the pose estimated by that algorithm can be used as an initial estimate.

Furthermore, if the residual rotation between the current estimate and the true value is small enough, a linearization of the rotation matrix in the estimation problem can be used. From the simulation results, this is clearly the case. The new estimate of the rotation matrix, $\mathbf{R}$, will be approximated by applying an approximation of a small rotation matrix to the current estimate of the rotation matrix $\mathbf{R_0}$:

$$\mathbf{R} \approx \begin{bmatrix} 1 & -w_z & w_y \\ w_z & 1 & -w_x \\ -w_y & w_x & 1 \end{bmatrix} \mathbf{R_0}$$

The estimated rotation matrix is orthonormalized after each estimation.

Labelling the current estimate of the projection of the object points as $\left[u_i^0, v_i^0\right]^T$, the error metric is derived from:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & -u_i^0 \\ 0 & 1 & -v_i^0 \end{bmatrix} \mathbf{R}\mathbf{x_i} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right) \frac{1}{t_z} \tag{4}$$

In the graphs, the modified paraperspective algorithm was executed for ten iterations and then this 'polished' algorithm was run for ten iterations.

By using the polishing technique, the orientation error for the modified paraperspective algorithm is decreased by over ten percent at all depths. The results for position showed essentially no change.

The absolute results for the polished algorithm are shown if Figure 2.

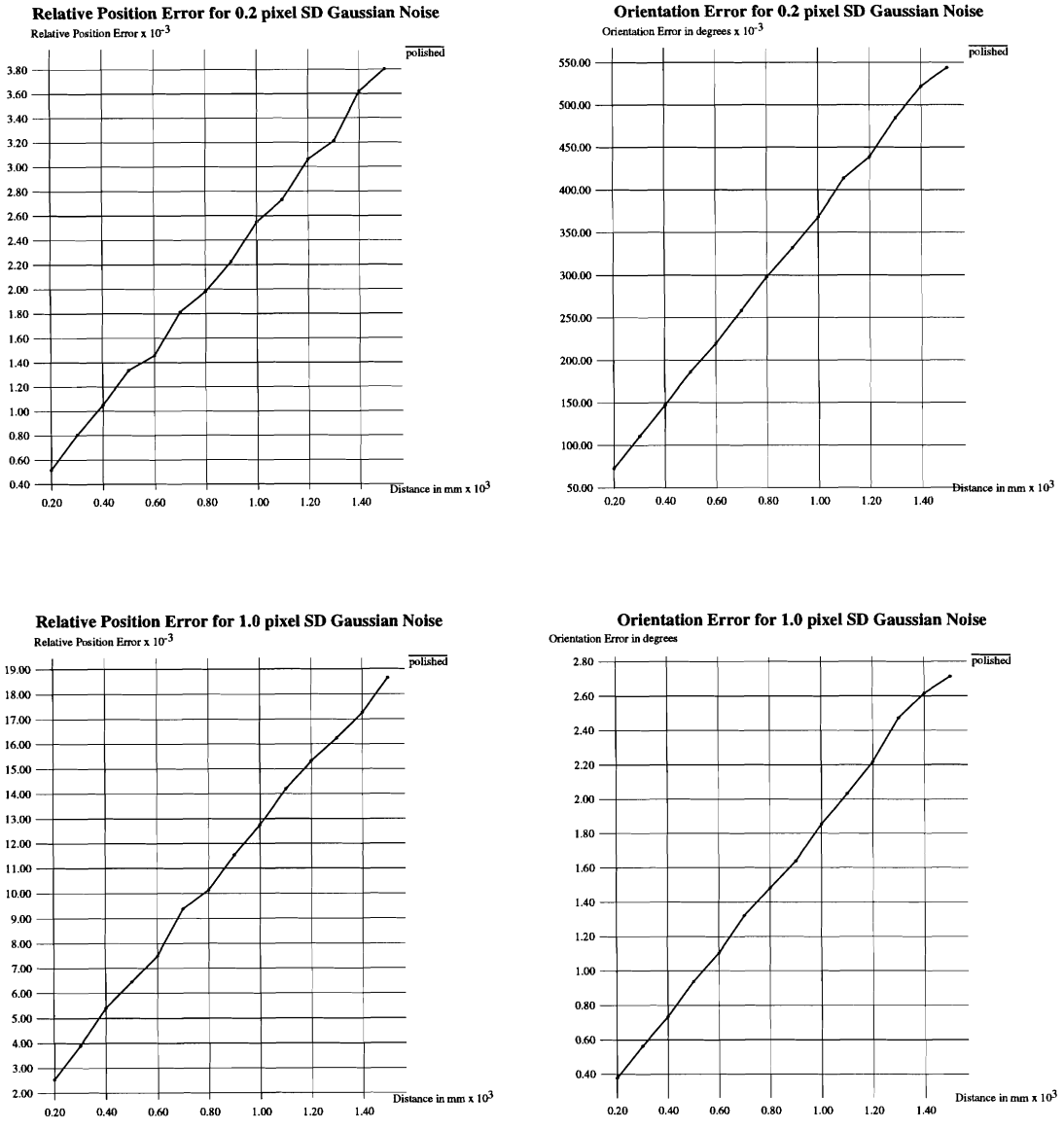


Fig. 2. Absolute Results for the Polished Algorithm

4 Conclusion

The projection approximation algorithms reviewed in this paper are viable techniques for object pose estimation. They are reasonably accurate and, relative to non-linear techniques, very fast. A simple modification to the algorithms was presented that removes a large portion of the error observed in these algorithm when the object is close. Finally, a polishing technique was suggested that dramatically improves the accuracy of the estimate. The overall improvements to the algorithms are shown in Figure 3. The polished algorithm does not require a significant increase in execution time and, as such, is believed to be a complete improvement over the reviewed algorithms. Furthermore, the observed accuracy results in Figure 2 suggest that the algorithm

can be used in nearly all applications.

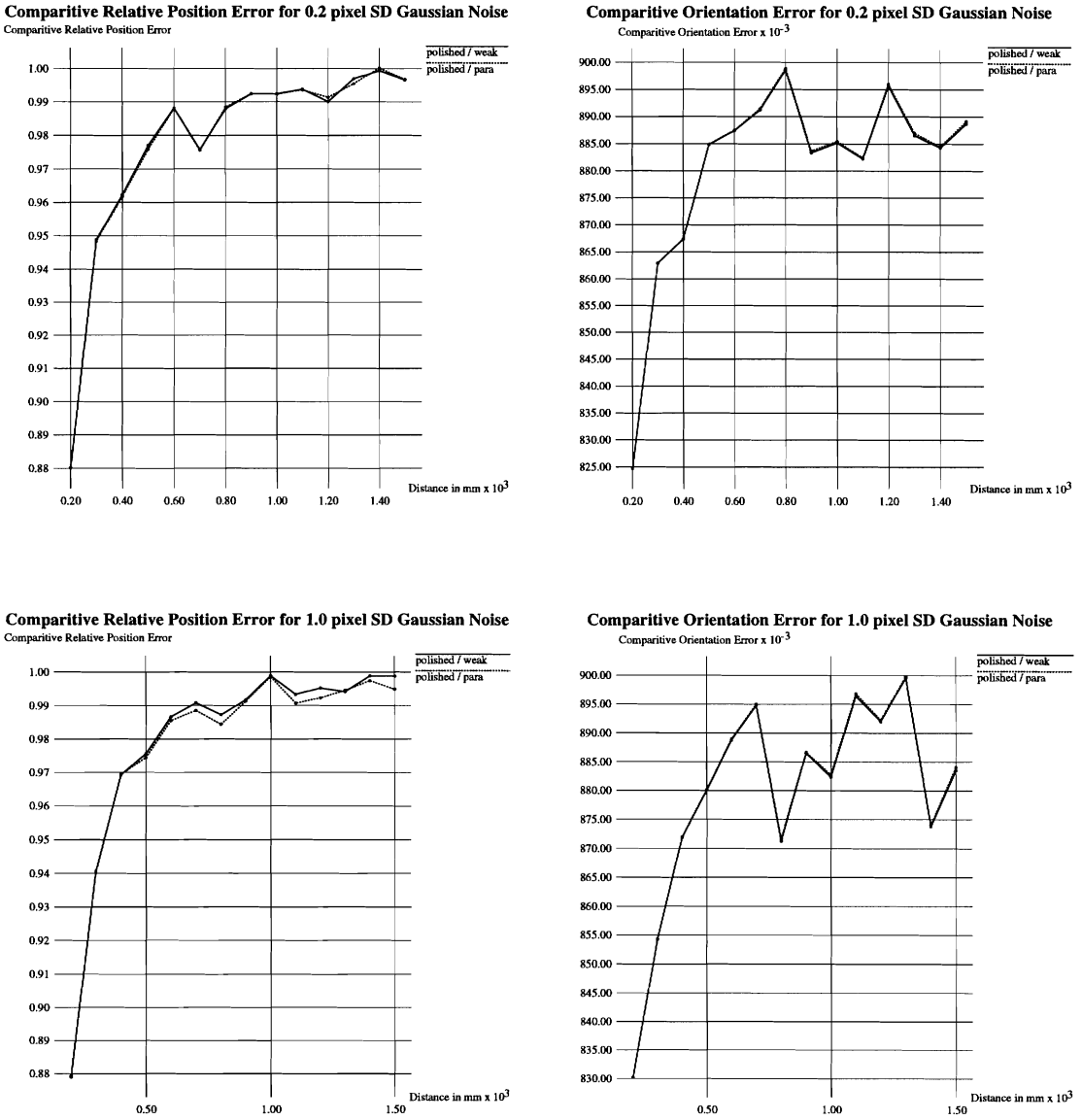


Fig. 3. Comparison between the Weak-Perspective and Paraperspective with the Polished Algorithm

References

1. D. DeMenthon and L. Davis. Model-Based Object Pose in 25 Lines of Code. *IJCV*, 15, 123–141, 1995.
2. M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24, 381–395, 1981.

3. S. Ganapathy. Decomposition of Transformation Matrices for Robot Vision. *Pattern Recognition Letters*, 2, 401–412, 1984.
4. R. Holt and A. Netravalli. Camera Calibration Problem: Some New Results. *CGVIP - Image Understanding*, 54, 368–383, 1991.
5. R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. An Analytic Solution for the Perspective 4-Point Problem. *Computer Vision, Graphics, and Image Processing*, 47, 33–44, 1989.
6. R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. Object Pose: The Link between Weak Perspective, Paraperspective, and Full Perspective. *IJCV*, 22, 173–189, 1997.
7. Y. Hung, P. Yeh, and D. Harwood. Passive Ranging to Known Planar Point Sets. *Proc. IEEE Int. Conf. on Robotics and Automation*, 80–85, 1985.
8. D. Lowe. Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, 31, 355–395, 1987.
9. T. Phong, R. Haraud, A. Yassine, P Tao. Object Pose from 2-D to 3-D Point and Line Correspondences. *IJCV*, 15, 225–243, 1995.
10. J. Yuan. A General Photogrammetric Method for Determining Object Position and Orientation. *IEEE Transactions on Robotics and Automation*, 5, 129-142, 1989.

This article was processed using the LaTeX macro package with LLNCS style.