Optimal Estimation of Object Pose from a Single Perspective View

Thai-Quynh Phong, Radu Horaud, Adnan Yassine, Dinh-Tao Pham

Optimal Estimation of Object Pose from a Single Perspective View

T. Q. Phong[†‡], R. Horaud[‡], A. Yassine[§], and D. T. Pham[†]

[‡]LIFIA-IRIMAG
46, ave. F. Viallet
38031 Grenoble

[†]INSA-Rouen
BP 08
76131 Mont-Saint-Aignan

[§]Université de Nancy I
BP 239
54506 Vandoeuvre-les-Nancy

Abstract

In this paper we present a method for robustly and accurately estimating the rotation and translation between a camera and a 3-D object from point and line correspondences. First we devise an error function and second we show how to minimize this error function. The quadratic nature of this function is made possible by representing rotation and translation with a dual number quaternion. We provide a detailed account of the computational aspects of a trust-region optimization method. This method compares favourably with Newton's method which has extensively been used to solve the problem at hand, with Faugeras-Toscani's linear method [3] for calibrating a camera. Finally we present some experimental results which demonstrate the robustness of our method with respect to image noise and matching errors.

1 Introduction

The problem of determining the position and orientation of an object with respect to a camera has many relevant applications in computer vision: object positioning, camera calibration, hand-eye calibration, docking for land and space mobile robots, and cartography. This problem is also known as the *perspective n-point problem*, *exterior (or extrinsic) camera calibration problem*, or *camera location and orientation problem* and can be stated more formally as follows: Given a set of points that are described in an *object centered frame*, given the projections of these points onto an image, and given a projection model and the parameters of this model, determine the rigid transformation (rotation and translation) between the object centered frame and the camera centered frame.

Previous approaches attempting to solve this problem fall into two categories: (i) *Closed-form solutions* and (ii) *Numerical solutions*. In this paper we concentrate on numerical solutions.

Since the object pose from a single view problem is nonlinear, choices for (i) the mathematical representation of the problem, (ii) the error function to be minimized, and for (iii) the optimization method are crucial.

Yuan [14] proposed to separate the rotational component of the problem from the translational one and he concentrated on the estimation of the rotation parameters. The rotation is represented by an orthonormal matrix and the solution is given by the common root(s) of six quadratic equations. The common root(s) is then found using Newton's iterative gradient method. However the author noticed that local optima occur when gradient techniques are used. Several local minima correspond to the nonlinear nature of the problem. The global minimum can be reached only by properly initializing the iterative algorithm.

Lowe [8] used Newton's method as well for estimating the orientation and location of an object with respect to a camera. As with Yuan's method, Lowe noticed some problems with Newton's method and in a subsequent paper he suggested how to deal with the initialisation and stability problems [9].

Liu & al. [7] examined alternative iterative approaches to solving for the viewing parameters. The rotation is represented by the Euler angles. The authors linearize the error function. They noticed that their method worked well only when the three Euler angles are less than 30^0.

Using the mathematical formulation suggested by Liu & al. [7], Kumar & Hanson [6] examined two minimization methods: an iterative technique that linearizes the error function and which requires a good initial estimate and a least median of squares technique which is combinatorial in complexity.

In the light of the above discussion a robust and accurate method is still to be proposed. In this paper we

0-8186-3870-2/93 $3.00 © 1993 IEEE

devise a method for solving the object pose problem. The method is tailored as follows:

- Section 2 – each line correspondence (or equivalently each pair of point correspondences) provides two constraints which express that the object line, its corresponding image projection, and the center of projection of the camera are coplanar. This approach has already been used by Horaud & al. [5], Dhome & al. [2], Liu & al. [7] and Kumar & Hanson [6].

  The rigid transformation whose parameters are the unknowns of the problem is represented by a *dual number quaternion*. With this representation the constraints mentioned above become quadratic equations. The advantage of using a dual number quaternion representation is that rotation and translation are estimated simultaneously rather than sequentially. Walker & al. [13] introduced dual number quaternions in computer vision and they solved for the 3-D/3-D pose problem. At our knowledge there is no attempt to use dual number quaternions in conjunction with the exterior camera calibration (2-D/3-D pose) problem.

- Section 3 – a non linear numerical optimization method is described and used for estimating the best rigid transformation. The error function to be minimized over the pose parameters is the sum of squares of the quadratic constraints just described. Unlike most of previous approaches in computer vision, we use a second order approximation of the error function. More specifically we use a *trust region* optimization method. The idea of using a trust region goes back to Soresen [11] and Moré [10] (See also Clermont & al.[1], Pham D.T. & al.[12]). We provide a complete description of the algorithm that we implemented.

- Section 4 – in order to check the validity of the solution thus obtained we compare our results with the results obtained using the camera calibration method proposed by Faugeras & Toscani [3]. We analyse the accuracy and robustness of our method with respect to the number of correspondences, the image noise, and matching errors.

2 Object pose from line correspondences

We consider a pin-hole camera model and we assume that the parameters of the projection (the intrinsic camera parameters) are known. The origin of the camera frame is at F – the center of projection, the z-axis is parallel to the optical axis, and the xy-plane is parallel with the image plane. We assume that the optical axis is perpendicular onto the image plane.

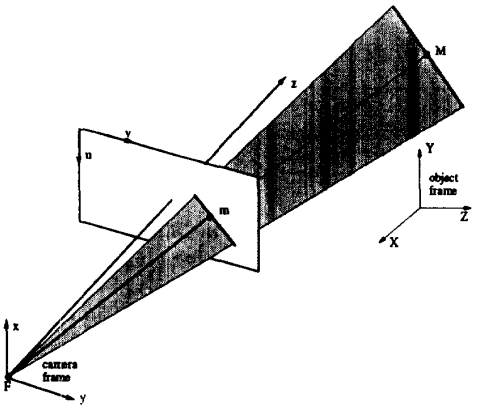

Figure 1: The object line, its projection onto the image, and the center of projection F are coplanar and this plane is shown in grey. $\vec{n}$ is the unit vector normal to this plane.

We consider now an object line. In the object frame this line is described parametrically by its direction $\vec{v}$ and by a point vector $\vec{p}$ and it can be expressed in the camera frame as well:

$$\vec{p}' = R\vec{p} + \vec{t}, \quad \vec{v}' = R\vec{v}$$

where the 3×3 rotation matrix R and the translation vector $\vec{t}$ describe the rigid transformation from the object frame to the camera frame and are precisely the parametrs associated with the object pose problem. The correspondence constraints express the fact that an object line belongs to the plane defined by the center of projection and the image line, i.e.,:

$$\vec{n} \cdot (R\vec{v}) = 0, \quad \vec{n} \cdot (R\vec{p} + \vec{t}) \tag{1}$$

where $\vec{n}$ is the vector normal to this plane, Figure 1.

Therefore, each line correspondence provides 2 constraints. In the general case if N line correspondences are available, the pose problem becomes the problem of solving for a set of $2N$ non linear constraints, or equivalently, the problem of minimizing the following error function:

$$f(R,t) = \sum_{i=1}^{N} (\vec{n}_i \cdot (R\vec{v}_i))^2 + \sum_{i=1}^{N} (\vec{n}_i \cdot (R\vec{p}_i + \vec{t}))^2 \tag{2}$$

2.1 Rotation, translation, and dual number quaternions

A rigid transformation may be represented by a *dual number quaternion* which has a real part and a dual part:

$$\hat{q} = \mathbf{r} + \epsilon \mathbf{s}$$

where $\mathbf{r}$ and $\mathbf{s}$ are quaternions and $\epsilon^2 = 0$, [13].

Let a rigid transformation be represented by a vector $\vec{k}$, a point vector $\vec{l}$, and two scalars, θ and d. Many will recognize here a screw representation of rotation and translation: $\vec{k}$ and $\vec{l}$ define the screw's axis, θ is the angle of rotation about this axis and d is the length of the translation along the axis.

Recall that a quaternion may also be viewed as a 4-vector:

$$r = r_0 + ir_x + jr_y + kr_z = (r_0, \vec{r}) = (r_0\ r_x\ r_y\ r_z)^T$$

Given $\hat{q}$ such that $r \cdot r = 1, r \cdot s = 0$, the rotation matrix R and the translation vector $\vec{t}$ can be easily derived from $\mathbf{r}$ and $\mathbf{s}$ using the following formulae:

$$\begin{pmatrix} 1 & 0^T \\ 0 & R \end{pmatrix} = W(r)^T Q(r), \quad \begin{pmatrix} 0 \\ t \end{pmatrix} = 2W(r)^T s \quad (3)$$

where $W(r)$ and $Q(r)$ are two 4×4 matrices associated with a quaternion:

$$Q(r) = \begin{pmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & -r_z & r_y \\ r_y & r_z & r_0 & -r_x \\ r_z & -r_y & r_x & r_0 \end{pmatrix} \quad (4)$$

$$W(r) = \begin{pmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & r_z & -r_y \\ r_y & -r_z & r_0 & r_x \\ r_z & r_y & -r_x & r_0 \end{pmatrix} \quad (5)$$

2.2 The error function

If 3-vectors are treated as purely imaginary quaternions, that is: $v = (0\ ,\ \vec{v})$ then the constraint of eq. (1) can be written as:

$$\begin{aligned} \vec{n} \cdot (R\vec{v}) &= r^T Q(n)^T W(v) r \\ \vec{n} \cdot (R\vec{p} + \vec{t}) &= r^T Q(n)^T W(p) r + r^T (2Q(n)^T) s \end{aligned}$$

In other terms, each correspondence i provides two quadratic constraints:

$$r^T A_i r = 0, \quad r^T B_i r + r^T C_i s = 0$$

with A_i, B_i, and C_i being three 4×4 matrices.

We can now write a new expression of the error function associated with our problem, i.e., eq. (2):

$$\begin{aligned} f(r,\ s) &= \sum_{i=1}^{N} ((r^T A_i r)^2 + (r^T B_i r + r^T C_i s)^2) \\ &\quad + \lambda (r^T r - 1)^2 + \lambda (r^T s)^2 \end{aligned} \quad (6)$$

where the parameters to be estimated are $\mathbf{r}$ and $\mathbf{s}$. λ is a positive number which must be taken very large in order to guarantee that the penalization constraints ($r^T r = 1$ and $r^T s = 0$) are satisfied (for our application we took $\lambda = 50$).

Notice that an alternative to this error function may be to consider the estimation of $\mathbf{r}$ and $\mathbf{s}$ separately. One may estimate the rotation first using the following error function:

$$f(\mathbf{r}) = \sum_{i=1}^{N} (r^T A_i r)^2 + \lambda (r^T r - 1)^2 \quad (7)$$

Once the optimal value of $\mathbf{r}$ is found, the computation of the optimal value of $\mathbf{s}$ is trivial.

3 The trust-region method

It is clear that the minimization of the error functions described by equation (6), equation (7) equivalent to the following non linear least squares problem:

$$0 = \min\{f(x) = \frac{1}{2} \sum_{j=1}^{m} \Phi_j^2(x) : x \in I\!R^n\} \quad (8)$$

with $\Phi_j(x)$ being twice continuously differentiable from $I\!R^n$ to $I\!R$.

We recall that the gradient $\nabla f(x)$ and the Hessian $\nabla^2 f(x)$ can be calculated as follows:

$$\nabla f(x) = \sum_{j=1}^{m} \Phi_j(x) \nabla \Phi_j(x) = J(x)^T \Phi(x), \quad (9)$$

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^{m} \Phi_j(x) \nabla^2 \Phi_j(x) \quad (10)$$

where $\Phi(x) = (\Phi_1(x), \ldots, \Phi_m(x))^T$ and $J(x) = (\nabla\Phi_1(x), \ldots, \nabla\Phi_m(x))^T$ is the $m \times n$ Jacobian matrix of $\Phi(x)$. In practice the Gauss-Newton approximation of the Hessian is used, i.e.,$H(x) = J(x)^T J(x)$. This is based on the premise that the first-order term will eventually dominate the second-order term.

Let x_k denote the current estimate of the solution; a quantity subscripted by k will denote that quantity evaluated at the k^{th} iteration of the algorithm.

The basic idea of the trust-region optimization method consists of successively approximating the error function by a *local quadratic form* in a neighbourhood of the current solution x_k:

$$f(x_k + d) \approx f(x_k) + q_k(d), \quad \|d\| \leq \delta_k$$

where:

$$q_k(d) = g(x_k)^T d + \frac{1}{2} d^T H(x_k) d \qquad (11)$$

The error function will be reduced via the direction d_k, i.e., $x_{k+1} = x_k + d_k$, where d_k is the value of d which minimizes the local quadratic form over a restricted spherical region centered around x_k: *the trust region*:

$$\min\{q_k(d) : \|d\| \leq \delta_k\} \qquad (12)$$

The parameter δ_k is called the trust radius and is determined dynamically using a measure of the quality of the approximation; this is measured by a quality coefficient r_k:

$$r_k = \frac{f(x_k) - f(x_k + d_k)}{q_k(0) - q_k(d_k)} \qquad (13)$$

If r_k is too small it means that the approximation is not good and the trust region should be decreased. Otherwise the trust region should be increased. The local quadratic form depends on the gradient and the Hessian of the error function. Hence, the minimum thus found has "good" second-order properties. In a trust-region method the main difficulty resides in the minimization of the local quadratic form. Various trust region algorithms differ upon the method being used to minimize the local quadratic form inside the trust region.

3.1 Local quadratic problem

Local quadratic problem is the problem of minimizing a quadratic form inside a sphere:

$$\min\{\frac{1}{2} d^T H d + g^T d : \|d\| \leq \delta\} \qquad (LQP)$$

where $g \in \mathbb{R}^n$, H is a symmetric matrix and δ is a positive number. All existing methods for solving this problem are based on the following theorem:

Theorem 1 *d^* is a solution to (LQP) if and only if there exists $\mu \geq 0$ such that:*

(i) $H + \mu I$ is positive semidefinite,

(ii) $(H + \mu I)d^* = -g$,

(iii) $\|d^*\| \leq \delta$ and $\mu(\|d^*\| - \delta) = 0$.

Such a μ is unique.

For $\mu > 0$ the (LQP) problem has a solution on the boundary of its constraint set, i.e, $\|d^*\| = \delta$ and the problem is reduced to the problem of finding μ such that: $\phi(\mu) = \|d(\mu)\| = \delta$ where $d(\mu)$ is a solution of $(H + \mu I)d = -g$. In this case, μ and d^* (the optimal solution) can by found efficiently by the method by Hebden [4]. In fact, Hebden's algorithm can be viewed as Newton's method for the zero-finding problem:

$$\psi(\mu) = \frac{1}{\delta} - \frac{1}{\phi(\mu)} = 0, \quad \text{for } \mu \in]-\lambda_1, +\infty[\qquad (14)$$

The most important feature of Hebden's algorithm is that usually the number of iterations required to produce an acceptable approximation of solution μ^* is very small since ψ is convex, almost linear, and strictly decreasing on $]-\lambda_1, +\infty[$.

3.2 Practical trust-region algorithm

We propose to apply the following practical trust-region algorithm to our problem (see also Clermont & al. [1] and Pham & al. [12]).

- *Initialization* : Let x_o, δ_o, ϵ,ϵ_g,ϵ_f be given. k=0.
- *Iteration k:=0,1,...*

  k.1 Compute $f_k = f(x_k), g_k = \nabla f(x_k)$ and $H_k = J(x_k)^T J(x_k)$.

  k.2 If $\|g_k\| \leq \epsilon_g$ or $\delta_k \leq \epsilon_\delta$ or $f_k \leq \epsilon_f$ then stop: x_k is a solution.

  k.3 Let d be a solution of the system $H_k d = -g_k$ If $\|d\| < \delta_k - \epsilon$ then $d_k = d$. Otherwise, using Hebden's algorithm to find a $\mu > 0$ so that the solution of $(H_k + \mu I)d = -g_k$ satisfies $|\|d\| - \delta_k| < \epsilon$, then $d_k = d$.

  k.4 Compute r_k using eq. (13).

  k.5 If $r_k \geq s$ then $x_{k+1} := x_k + d_k$. If $r_k \geq t$ then $\delta_{k+1} := 2\delta_k$. Otherwise $\delta_{k+1} := \delta_k$ Set $k := k + 1$ and return to k.1

  k.6 If $r_k < s$ then $\delta_k := \delta_k/2$ and return to k.3.

The parameter s must belong to the interval [0.1, 0.3] and the parameter t must belong to the interval [0.5, 0.8]. For our application, these parameters were set at: $s = 0.25$ and $t = 0.75$.

4 Experimental results

The trust region algorithm is particularly well-suited for solving the object pose from a single view problem because the error function is a sum of squares of quadratic constraints. Indeed, the trust region algorithm – generally applicable for any non linear constraints – is more robust and more efficient when these constraints are quadratic. The robutness and efficiency of the algorithm are du to the quadratic nature of the constaints. The experiments that we performed can be paraphrased as follows:

- A calibrating object with 500 calibrating points is viewed by a camera and point-to-point correspondences are established;
- The intrinsic and extrinsic camera parameters are determined using these 500 point correspondences and the method of Faugeras & Toscani [3];
- Subsets of point correspondences and hence line correspondences are randomly selected from the initial set of 500 points.
- The trust region algorithm is applied to these sets of line correspondences. The parameters thus found are compared with the extrinsic parameters previously found using the following by Faugeras-Toscani's method;
- Noise is sometimes added to the positions of the image points;

nb of lines	error in rotation	error in transl.	nb of iter.	CPU time	added noise
10	0.0044	0.0545	16	0.1	
50	0.0008	0.0320	12	0.4	
50	0.0003	0.0269	11		0.01
50	0.0009	0.0218	11		0.1
50	0.0040	0.0098	11		0.5
50	0.0080	0.0291	12		1.0
50	0.0158	0.0647	11		2.0
150	0.0001	0.0118	8	1.6	

Table 1: The experimental results obtained when the rotation and translation are estimated sequentially. The CPU time is measured in seconds on a SPARC-2 processor. The noise is in pixels, is random with maximum amplitude as indicated, and is added to the 2-D point positions.

- In a separate experiment we artificially mismatch some of the correspondences but this mismatch is done locally: a mismatch is defined as a set of two point correspondences that are inverted. This experiment validates the robustness of our method with respect to matching errors.

Table 1 summarizes the results obtained with our method when applied to eq. (7). Once the optimal rotation is thus found, we determine the optimal translation using linear optimization.

Table 2 summarizes the results obtained with our method when applied to eq. (6), that is, the optimal rotation and translation are estimated simultaneously.

nb of lines	error in rotation	error in transl.	nb of iter.	CPU time	added noise
10	0.0024	0.0498	125	1.9	
50	0.0001	0.0069	39	2.1	
50	0.0004	0.0354	38		0.01
50	0.0003	0.0345	38		0.1
50	0.0009	0.0315	38		0.5
50	0.0021	0.0302	39		1.0
50	0.0044	0.0353	39		2.0
150	0.0001	0.0093	20	3.0	
200	0.0005	0.0157	19	4.0	

Table 2: The experimental results obtained when the rotation and translation are estimated simultaneously.

We noticed that the rotation is relatively robust with respect to matching errors. The translation is robust too but to a least extent. The rotation and translation experiment allows up to 5% of "locally" mismatched points. The rotation then translation experiment is more sensitive to matching errors.

5 Discussion

The method that we presented in this paper for estimating the exterior parameters of a camera from line and point correspondences may be evaluated with respect to the following items:

- *Initialisation* – the final result is independent of the initialisation. This is a dramatic improvement with respect to other approaches using Newton's method.
- *Number of correspondences* – the results are also robust with respect to the number of matchings.

- *Accuracy* – The algorithm nicely resists when noise is injected in the image.
- *Efficiency* – The rotation then translation implementation is more efficient than the rotation and translation implementation. In fact there is a compromise between efficiency and accuracy. One may be interested in a fast algorithm which will provide a less accurate result. Ideally, with 30 line correspondences, the algorithm converges in less than 1 second.
- *Matching errors* – The algorithm allows for matching errors. In this case we noticed that the rotation and translation implementation is more robust with respect to matching errors. We are not aware of many experiments testing robustness and accuracy in the presence of matching errors.

To conclude, we believe that the method that we presented in the paper has those properties that make it suitable to be used whenever robustness, accuracy, and efficiency are needed. We also believe that the trust-region method could beneficially be used to solve for other non-linear minimization problems in computer vision such as hand/eye calibration and structure from motion.

Acknowledgements. The authors acknowledge Roger Mohr, Long Quan, and Thomas Skordas for their insightful comments. Financial support is from the Basic Research Esprit programme (the SECOND project).

References

[1] J. R. Clermont, M. E. De La Lande, P. D. Tao, and A. Yassine. Analysis of plane and axis-symmetric flows of incompressible fluids with the stream tube method: numerical simulation by trust-region algorithm. *International Journal for Numerical Methods in Fluids*, 13:371–399, 1991.

[2] M. Dhome, M. Richetin, J.T. Lapreste, and G. Rives. Determination of the Attitude of 3D Objects from a Single Perspective View. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, December 1989.

[3] O. D. Faugeras and G. Toscani. The Calibration Problem for Stereo. In *Proc. Computer Vision and Pattern Recognition*, pages 15–20, Miami Beach, Florida, USA, June 1986.

[4] M. D. Hebden. An algorithm for minimization using exact second derivatives. Technical Report TP 515, Atomic Energy Research Establishment, Harwell, England, 1973.

[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(1):33–44, July 1989.

[6] R. Kumar and A. R. Hanson. Robust estimation of camera location and orientation from noisy data having outliers. In *Proc. Workshop on Interpretation of 3-D Scenes*, pages 52–60, Austin, Texas, USA, November 1989.

[7] Y. Liu, T. S. Huang, and O. D. Faugeras. Determination of camera location from 2-d to 3-d line and point correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, January 1990.

[8] D. Lowe. Three-dimensional Object Recognition from Single Two-dimensional Images. *Artificial Intelligence*, 31:355–395, 1987.

[9] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[10] J. J. Moré. Recent developments in algorithms and software for trust region methods. In A Bchem, M. Grötschel, and B. Korte, editors, *Mathematical Programming: The State of the Art*, pages 258–287. Springer Verlag, Berlin, 1983.

[11] D. C. Soresen. Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982.

[12] P. D. Tao, S. Wang, and A. Yassine. Training multi-layered neural networks with a trust-region based algorithm. *Mathematical Modelling and Numerical Analysis*, 24(4):523–553, 1990.

[13] M. W. Walker, L. Shao, and R. A. Volz. Estimating 3-d location parameters using dual number quaternions. *CGVIP-Image Understanding*, 54(3):358–367, November 1991.

[14] J. S.-C. Yuan. A general photogrammetric method for determining object position and orientation. *IEEE Transactions on Robotics and Automation*, 5(2):129–142, April 1989.