An Alternative Approach to Recovering 3D Pose Information from 2D Data

Gareth Loy, Rhys Newman, Alex Zelinsky and John Moore
Department of Systems Engineering
Research School of Information Sciences and Engineering
The Australian National University
Gareth.Loy@syseng.anu.edu.au

Abstract

A novel approach is presented for real-time pose estimation of a known object being observed through a single camera. The future aim is to develop the method outlined here into a robust closed-form solution suitable for implementation on a monocular face tracking system. However, the method outlined in this paper offers a generic 2D to 3D pose estimation technique which is not limited to face tracking applications. The approach requires the observed object to consist of at least six 3D points. It uses true perspective projection and provides a means towards a direct solution, without the need for iteration. While the limited information available from a monocular vision system suggests that stereo vision is more suitable for pose recognition problems these stereo camera must still be calibrated. Camera calibration is an equivalent problem to monocular pose estimation, and must be performed on each camera individually, and so the approach discussed will also be useful in the calibration of a stereo system.

1 Introduction

When a person sees a photograph of a familiar object, such as a human face, he or she can immediately tell which direction the face is oriented and approximately how far away the face was from the camera when the photograph was taken. This is an example of the human visual processing system estimating the pose of a familiar three dimensional object (a face) from a two dimensional image of the object (the photograph). This report presents a new idea with the potential for increasing the robustness with which a robotic vision system is able to perform this task.

Estimating the three-dimensional pose of a known object from a two-dimensional image is a classical problem in computer vision. The 'known object' is described by a set of three dimensional points, of which there must be at least four (non-coplanar) points in order for there to be a unique solution. If only three (non-collinear) points are taken the problem becomes the classical photogrammetry three-point resection problem, which has multiple solutions and was solved by Grunert in 1841; numerous other direct solutions have been developed since and are discussed and evaluated in [Haralick *et al.*, 1991]. One way of solving the four-point pose estimation problem considered here is to divide the object into four sets of three points, solve each of the resulting three-point resection problems, and take the answer which is common to all four sets of solutions. However, the closed form solutions presented for the three-point resection problem are typically numerically unstable and highly sensitive to noise.

In addition to estimating the pose of an observed 3D object, the algorithm can estimate the pose of the *camera* given an observed object with a known pose. This is the same problem (with the appropriate choice of reference frames). [Ganapathy, 1984] presented a close-form solution for this problem requiring the observed 3D object to consist of at least six points.

Thus the problem is equivalent to the camera calibration problem, which in turn can be used in robotic navigation, where the position of a mobile robot could be estimated from the familiar objects that the robot observes. The camera calibration problem involves estimating the 3D location of the camera centre, which is typically difficult to determine accurately. Even in the calibration of stereo vision systems, each camera centre is determined from data observed by that camera alone, and so a monocular 2D to 3D camera pose estimation algorithm is still relevant.

The approach discussed in this paper uses true perspective projection to provide a direct solution, without the need for iteration. However, it does require the observed three dimensional object to consist of at least six points. The system is reduced to a set of underconstrained linear equations where the nonlinear geometric constraints which normally ensure that the rotation is a true rotation are ignored.

In the noise free case the solution to this system is easily found, however, in the presence of noise the estimated rotation becomes a rotation plus a distortion - since it has not been constrained to be a pure rotation. Thus it becomes necessary to compute a true rotation which best fits the observed data. It is this final step which is the most challenging aspect of the pose recovery problem.

This paper introduces an approach which has the potential to increase the robustness of the estimated rotation with respect to noise in the observed image.

2 Problem formulation

The set of n three dimensional model points (x_{mi}, y_{mi}, z_{mi}) describing the known object is written as a matrix

$$M = \begin{bmatrix} \mathbf{m}_1 & \mathbf{m}_2 & \dots & \mathbf{m}_n \end{bmatrix} \quad (1)$$

where $\mathbf{m}_i = [x_{mi}, y_{mi}, z_{mi}]'$. This object can then be rotated and translated anywhere in three dimensional space by pre-multiplication with a homogeneous transformation matrix T.

$$T = \begin{bmatrix} \Theta & \mathbf{p} \\ \mathbf{0}' & 1 \end{bmatrix}$$

where Θ is a rotation matrix describing a rotation about an axis passing through the origin, $\mathbf{p}$ is a vector describing the translation of the rotated object and $\mathbf{0}' = [0, 0, 0]$.

So, the transformed object X is described by

$$\begin{bmatrix} X \\ \mathbf{1}' \end{bmatrix} = T \begin{bmatrix} M \\ \mathbf{1}' \end{bmatrix} \quad (2)$$

where $\mathbf{1}$ is an $n \times 1$ vector of 1's, and X is again a series of Cartesian co-ordinates

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ z_1 & z_2 & \dots & z_n \end{bmatrix}$$

The transformed object X is observed by a monocular pinhole camera located at the origin and looking along the positive z-axis. This two dimensional image is used to estimate the three dimensional pose of the object.

X is decomposed into *known* quantities which can be measured from the two dimensional camera view and *unknown* quantities. The unknown quantities are the depths r_i, where r_i is the distance of the i^{th} point from the origin.

The decomposition is as follows. X is written in spherical polar form using the transformation

$$x = r sin\psi_2 cos\psi_1$$

$$y = r sin\psi_2 sin\psi_1$$

$$z = r cos\psi_2$$

where

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\psi_1 = \begin{cases} \tan^{-1}\left(\frac{y}{x}\right) & \text{if } x \neq 0 \\ \frac{\pi}{2} & \text{if } x = 0 \text{ and } y \geq 0 \\ \frac{-\pi}{2} & \text{if } x = 0 \text{ and } y < 0 \end{cases}$$

$$\psi_2 = \begin{cases} \tan^{-1}\left(\frac{\sqrt{x^2+y^2}}{z}\right) & \text{if } z \neq 0 \\ \frac{\pi}{2} & \text{if } z = 0 \end{cases}$$

giving,

$$X = \begin{bmatrix} r_1 sin\psi_{2,1} cos\psi_{1,1} & r_2 sin\psi_{2,2} cos\psi_{1,2} & \dots & r_n sin\psi_{2,n} cos\psi_{1,n} \\ r_1 sin\psi_{2,1} sin\psi_{1,1} & r_2 sin\psi_{2,2} sin\psi_{1,2} & \dots & r_n sin\psi_{2,n} sin\psi_{1,n} \\ r_1 cos\psi_{2,1} & r_2 cos\psi_{2,2} & \dots & r_n cos\psi_{2,n} \end{bmatrix}$$

This can now be divided into a known component Y (which is independent of r_i) and an unknown component R containing r_i.

$$X = YR \quad (3)$$

where

$$Y = \begin{bmatrix} sin\psi_{2,1} cos\psi_{1,1} & sin\psi_{2,2} cos\psi_{1,2} & \dots & sin\psi_{2,n} cos\psi_{1,n} \\ sin\psi_{2,1} sin\psi_{1,1} & sin\psi_{2,2} sin\psi_{1,2} & \dots & sin\psi_{2,n} sin\psi_{1,n} \\ cos\psi_{2,1} & cos\psi_{2,2} & \dots & cos\psi_{2,n} \end{bmatrix} \quad (4)$$

and

$$R = \begin{bmatrix} r_1 & 0 & \dots & 0 \\ 0 & r_2 & & : \\ : & & . & 0 \\ 0 & \dots & 0 & r_n \end{bmatrix}$$

Substituting this back into equation 2 gives

$$\begin{bmatrix} YR \\ \mathbf{1}' \end{bmatrix} = \begin{bmatrix} \Theta & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} M \\ \mathbf{1}' \end{bmatrix}$$

$$\Rightarrow YR = \Theta M + \mathbf{p}\mathbf{1}'$$

which is equivalent to

$$YR - \Theta M - \mathbf{p}\mathbf{1}' = 0 \quad (5)$$

Y and M are known whilst R, Θ and $\mathbf{p}$ are unknown. Equation 5 is the key relationship which must be satisfied in order to find the true pose.

3 Description of the approach

3.1 Describing the system with a compact set of linear equations

To find a solution for R, Θ and $\mathbf{p}$ which satisfies equation 5 we start by defining the non-negative cost function

$$\Phi = tr(PP')$$

where

$$P = YR - \Theta M - \mathbf{p}\mathbf{1}'$$

$$\Theta = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \\ \theta_7 & \theta_8 & \theta_9 \end{bmatrix} \qquad (6)$$

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

which will be zero if and only if equation 5 is satisfied. We then seek to minimise this cost function, by choosing appropriate values for the unknowns $\theta_1, \theta_2, \ldots, \theta_9, p_1, p_2, p_3, r_1, r_2, \ldots, r_n$ (where n is the number of points being observed).

Φ is minimum at the point where all of its partial derivatives are zero. This follows from the structure of Φ: it is the sum of the diagonals of PP', each of which is quadratic in the above unknowns.

We can rewrite Φ as

$$\Phi = \sum_{i=1}^{n} (r_i^2 \mathbf{y}_i'\mathbf{y}_i - 2r_i\mathbf{y}_i\Theta\mathbf{m}_i - 2r_i\mathbf{y}_i'\mathbf{p} + \mathbf{m}_i'\Theta'\Theta\mathbf{m}_i + 2\mathbf{p}'\Theta\mathbf{m}_i + \mathbf{p}'\mathbf{p}) \qquad (7)$$

where $\mathbf{y}_i$ denotes the i^{th} column of Y (as defined in equation 4).

We now define M_i and $\vec{\Theta}$

$$\vec{\Theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_9 \end{bmatrix}$$

$$M_i = \begin{bmatrix} \mathbf{m}_i' & 0 & 0 \\ 0 & \mathbf{m}_i' & 0 \\ 0 & 0 & \mathbf{m}_i' \end{bmatrix}$$

and replace each occurance of $\Theta\mathbf{m}_i$ in equation 7 with $M_i\vec{\Theta}$ giving

$$\Phi = \sum_{i=1}^{n} (r_i^2 \mathbf{y}_i'\mathbf{y}_i - 2r_i\mathbf{y}_i M_i\vec{\Theta} - 2r_i\mathbf{y}_i'\mathbf{p} + \vec{\Theta}'M_i'M_i\vec{\Theta} + 2\mathbf{p}'M_i\vec{\Theta} + \mathbf{p}'\mathbf{p}) \qquad (8)$$

To minimise Φ we take its partial derivatives with respect to each of the unknowns and set these to zero.

$$\frac{\partial\Phi}{\partial r_i} = 2r_i\mathbf{y}_i'\mathbf{y}_i - 2\mathbf{y}_i'M_i\vec{\Theta} - 2\mathbf{y}_i'\mathbf{p} = 0$$

$$\Rightarrow r_i = \frac{\vec{\Theta}'M_i'\mathbf{y}_i + \mathbf{p}'\mathbf{y}_i}{\| \mathbf{y}_i \|^2} \qquad (9)$$

$$\frac{\partial\Phi}{\partial\mathbf{p}} = \sum_{i=1}^{n}(-2r_i\mathbf{y}_i' + \vec{\Theta}'M_i' + 2\mathbf{p}') = 0$$

$$\Rightarrow \mathbf{p}' = \sum_{i=1}^{n}(r_i\mathbf{y}_i' - \vec{\Theta}'M_i) \qquad (10)$$

$$\frac{\partial\Phi}{\partial\vec{\Theta}} = \sum_{i=1}^{n}(-2r_i\mathbf{y}_i'M_i + 2\vec{\Theta}'M_i'M_i + 2\mathbf{p}'M_i = 0 \qquad (11)$$

Substituting equation 9 into equations 10 and 11 and rearranging give

$$\mathcal{A}\mathbf{p} + \mathcal{B}\vec{\Theta} = 0 \qquad (12)$$

and

$$\mathcal{B}'\mathbf{p} + \mathcal{C}\vec{\Theta} = 0 \qquad (13)$$

respectively where

$$\mathcal{A} = \sum_{i=1}^{n}\left(\frac{\mathbf{y}_i\mathbf{y}_i'}{\| \mathbf{y}_i \|^2} - I\right)$$

$$\mathcal{B} = \sum_{i=1}^{n}\left(\frac{\mathbf{y}_i\mathbf{y}_i'}{\| \mathbf{y}_i \|^2} - I\right) M_i$$

$$\mathcal{C} = \sum_{i=1}^{n} M_i'\left(\frac{\mathbf{y}_i\mathbf{y}_i'}{\| \mathbf{y}_i \|^2} - I\right) M_i$$

Assuming $\mathcal{A}$ is invertible, equation 12 can be rewritten as

$$\mathbf{p} = -\mathcal{A}^{-1}\mathcal{B}\vec{\Theta} \qquad (14)$$

which can then be substituted into equation 13 to eliminate $\mathbf{p}$ giving

$$\mathcal{D}\vec{\Theta} = 0 \qquad (15)$$

where

$$\mathcal{D} = (\mathcal{C} - \mathcal{B}'\mathcal{A}^{-1}\mathcal{B})$$

Thus we have a system of 9 linear equations describing the 9 elements of Θ, and once Θ is found the other unknowns $\mathbf{p}$ and r_i can be determined from equations 14 and 9 respectively. However, nowhere has Θ been constrained to be a rotation matrix, and as a consequence solving for Θ in a practical case (with slightly noisy data) is not a trivial task.

3.2 Resolving the rotation

In a noise free environment the solution to equation 15 is easily found using least squares. However, in the presence of noise in the observed 2D image (the top two lines of Y in equation 4) it is no longer possible to find a homogeneous transformation which exactly satisfies equation 5. Thus, the least squares solution is *not* a homogeneous transformation and the estimated rotation is in fact a rotation plus a distortion.

There are several well known closed-form methods which estimate a true rotation matrix from the distortion matrix, see [Kanatani, 1993] chapter 5.2. However, the approach introduced here does not use the distortion matrix.

Our aim is to find a $\vec{\Theta}$ which minimises $\mathcal{D}\vec{\Theta}$ and whose elements are constrained to form the rows of a rotation matrix.

Therefore $\vec{\Theta}$ must have the form

$$\vec{\Theta} = \begin{bmatrix} a^2+b^2-c^2-d^2 \\ 2(bc-ad) \\ 2(bd+ac) \\ 2(bc+ad) \\ a^2-b^2+c^2-d^2 \\ 2(cd-ab) \\ 2(bd-ac) \\ 2(cd+ab) \\ a^2-b^2-c^2+d^2 \end{bmatrix} \tag{16}$$

where

$$a^2+b^2+c^2+d^2 = 1$$

and a, b, c and d are real numbers. This follows from the quarternion representation of a rotation matrix, which states that any rotation matrix Θ can be written as

$$\Theta = \begin{bmatrix} a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2 \end{bmatrix}$$

where a, b, c and d are constrained as above.

The problem is to estimate a, b, c and d as accurately as possible from the available noisy data, so that $\mathcal{D}\vec{\Theta}$ is minimised.

3.3 Restructuring $\vec{\Theta}$

By altering the structure of the $\vec{\Theta}$ vector it is theoretically possible to generate estimates for linear combinations of the quarternion parameters a, b, c and d.

This is done as follows. Define a new state vector

$$\vec{\Theta}_{new} = P\vec{\Theta}$$

where

$$P = QD$$

is a change-of-basis matrix consisting of two components Q and D. D is chosen so that

$$D\vec{\Theta} = \begin{bmatrix} a^2-d^2 \\ b^2-d^2 \\ c^2-d^2 \\ ab \\ ac \\ ad \\ bc \\ bd \\ cd \end{bmatrix}$$

and is therefore defined as

$$D = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & -0.5 \\ 0 & 0 & 0 & 0 & 0 & -0.25 & 0 & 0.25 & 0 \\ 0 & 0 & 0.25 & 0 & 0 & 0 & -0.25 & 0 & 0 \\ 0 & -0.25 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0.25 & 0 \end{bmatrix}$$

Q is defined so that $QD\vec{\Theta}$ contains a perfect square, and the determinant of Q is one. It is possible to take complex linear combinations of the rows of $\vec{\Theta}_{new}$ to construct perfect squares. For example

$$[1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 2i \quad 0 \quad 0 \quad 0]\ \vec{\Theta}_{new} = (a+id)^2$$

In fact we can define an infinite number of different Q matrices which have the property that $QD\vec{\Theta}$ contains a perfect square. However, the following two sets should suffice

$$Q = \begin{bmatrix} 1 & t & 0 & \sqrt{t} & 0 & 2i\sqrt{1+t} & 0 & 2i\sqrt{1+t} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$Q = \begin{bmatrix} 1 & 0 & s & 0 & 2i\sqrt{1+s} & 0 & 2i\sqrt{1+s} & 0 & \sqrt{s} \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

where s and t are arbitrary. Which lead to

$$\vec{\Theta}_{new} = \begin{bmatrix} (a+\sqrt{t}b+i\sqrt{1+t}d)^2 \\ b^2-d^2 \\ c^2-d^2 \\ ab \\ ac \\ ad \\ bc \\ bd \\ cd \end{bmatrix}$$

and

$$\vec{\Theta}_{new} = \begin{bmatrix} (a+\sqrt{s}c+i\sqrt{1+s}d)^2 \\ b^2-d^2 \\ c^2-d^2 \\ ab \\ ac \\ ad \\ bc \\ bd \\ cd \end{bmatrix}$$

respectively.

Now it remains to estimate $\vec{\Theta}_{new}$ such that $\mathcal{D}\vec{\Theta}$ is minimised.

3.4 Estimating quarternion parameters

We wish to minimise $\mathcal{D}P\vec{\Theta}$ which is defined by equation 15. This equation can be rewritten as

$$\mathcal{D}P^{-1}(P\vec{\Theta}) = \mathcal{D}P^{-1}\vec{\Theta}_{new} = 0$$

Multiplying on the left by $(P^{-1})'$ gives an expression of the form

$$S\vec{\Theta}_{new} = 0$$

where $S = (P^{-1})'WP^{-1}$ is symmetric.

We now minimise

$$\vec{\Theta}'_{new}S'S\vec{\Theta}_{new}$$

with the constraint

$$\vec{\Theta}'\vec{\Theta} = 1$$

$$\Leftrightarrow \vec{\Theta}'(P^{-1})'P^{-1}\vec{\Theta} = 1$$

Using the method of Lagrange multipliers

$$E = \vec{\Theta}'_{new}S'S\vec{\Theta}_{new} + \lambda(\vec{\Theta}'_{new}(P^{-1})'P^{-1}\vec{\Theta}_{new} - 1) \quad (19)$$

Taking the partial derivatives with respect to the elements of $\vec{\Theta}_{new}$ and setting to zero gives a system of 9 linear equations which can be written in the form

$$S'S\vec{\Theta}_{new} + \lambda(P^{-1})'P^{-1}\vec{\Theta}_{new} = 0$$

Multiplying on the left by PP' gives

$$(PP'S'S - (-\lambda))\vec{\Theta}_{new} = 0$$

This is now an eigenvector-eigenvalue problem with the non-trivial solutions for $\vec{\Theta}_{new}$ being the eigenvectors of $PP'S'S$. [Horn, 1987] showed that to minimise E in equation 19 $\vec{\Theta}_{new}$ should be chosen to be the eigenvector corresponding to the minimum eigenvalue. Thus a bounded estimate of $\vec{\Theta}_{new}$ can be determined.

So by choosing numerous different Q's of the forms described by equations 17 and 18 it is possible to obtain a set of estimates for a series of perfect squares of the quarternion parameters. Square-rooting these gives us a set of estimates for linear combinations of the quarternion parameters. The true values of the individual parameters can then be estimated by a least squares solution.

At the time of writing the technique had not been tested due to numerical problems when attempting to determine the eigenvectors of $PP'S'S$. If these difficulties can be overcome then this method will offer a novel approach to analysing noisy rotation matrix data.

A solution to this problem will be sought and results presented at the conference. In addition the results will be made available on the Internet at `http://wwwsyseng.anu.edu.au/~gareth`

4 Conclusion

This paper has presented a new method for 2D to 3D pose estimation for a monocular vision system. While there are of yet no results the technique suggests a promising new avenue for the development of monocular pose estimation techniques and camera calibration algorithms. By effectively 'sampling' the noisy data in many different forms the robustness of the estimation should be improved.

References

[Ganapathy, 1984] S. Ganapathy. Decomposition of Transformation Matrices for Robotic Vision. *Journal of Optical Society of America: A*, 4(4), pp.629-642, 1987.

[Haralick *et al.*, 1991] Robert M. Haralick, Chung-nan Lee, Karsten Otterberg and Michael Nölle. Analysis and Solutions of The Three Point Perspective Pose Estimation Problem. *IEEE conference on Computer Vision and Pattern Recognition*, pp.592-598, Hawaii, June, 1991.

[Horn, 1987] B. P. K. Horn. Closed-form solution of absolute orientation using unit quarternions. *Journal of Optical Society of America: A*, 4(4), pp.629-642, 1987.

[Kanatani, 1993] K. Kanatani.. *Geometric Computation for Machine Vision.* Oxford University Press, 1993.