A closed-form solution for paraperspective reconstruction

Etienne Grossmann* José Santos Victor
Instituto Superior Técnico, ISR
Av. Rovisco Pais, 1,
1049-001 Lisbon, Portugal
{etienne,jasv}isr.ist.utl.pt

We address the problem of 3D reconstruction from image features tracked along a sequence. The most precise algorithms compute the Maximum Likelihood (ML) estimate and are iterative. They need an approximate 3D reconstruction as starting position. For that purpose, we propose a closed-form expression of paraperspective reconstruction.

A matrix that approximately verifies the properties of a paraperspective projection matrix is first built, as in Christy and Horaud [1] or Poelman and Kanade [3]. Our contribution lies in showing how to transform this matrix so that it exactly verifies the properties of paraperspective projection matrices. This is done by a closed form expression, in which the depth of the camera is also retrieved. The camera position is then found directly, instead of being obtained as the solution of a non-linear optimization problem, like in [3].

As in [1, 5, 3], we assume that calibration is known.

1 Introduction

If an orthonormal basis $[\mathbf{i}, \mathbf{j}, \mathbf{k}]$ defines the orientation of a camera, and we consider a 3D point with coordinates $\mathbf{s}$ in a frame located at coordinates $\mathbf{t}$ in the camera frame, then the perspective projection of $\mathbf{s}$ (Figure 1) is the well-known expression :

$$\left[\begin{array}{c} u \\ v \end{array}\right] = \frac{1}{\mathbf{k}\cdot(\mathbf{s}+\mathbf{t})}\left[\begin{array}{c} \mathbf{i}\cdot(\mathbf{s}+\mathbf{t}) \\ \mathbf{j}\cdot(\mathbf{s}+\mathbf{t}) \end{array}\right] \quad (1)$$

The paraperspective projection is the first-order approximation, at $[0, 0, 0]$, of the function that associates (u, v) to $\mathbf{s}$ (for given $\mathbf{i}, \mathbf{j}, \mathbf{k}$ and $\mathbf{t}$.)

When one has tracked the projections of some 3D points $\mathbf{s}_1 \dots \mathbf{s}_P$ through a sequence of F images, one obtains the observation data $(u_{fp}, v_{fp})_{f=1\dots F, p=1\dots P}$. The problem of structure and motion is that of estimating the unknown quantities $\mathbf{i}_f, \mathbf{j}_f, \mathbf{k}_f, \mathbf{t}_f$ and $\mathbf{s}_p$, from the observations.

*This work has been supported by PRAXIS XXI grant BD / 19594 / 99

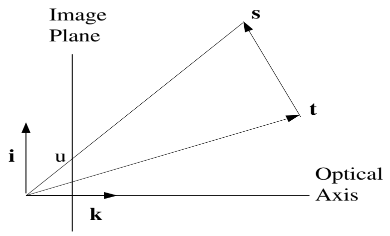

Figure 1. Perspective projection model (top view)

Using the notations of [3], the paraperspective projection (u, v) of the 3D point with coordinates $\mathbf{s}$ in the world frame, when the origin of the world frame has coordinates $\mathbf{t}$ in the camera frame is :

$$\left[\begin{array}{c} u \\ v \end{array}\right] = \left[\begin{array}{c} \mathbf{m} \\ \mathbf{n} \end{array}\right]\mathbf{s} + \left[\begin{array}{c} x \\ y \end{array}\right] + \text{Noise} \quad (2)$$

$$\text{where } \left\{\begin{array}{rcl} z &=& \mathbf{t}\cdot\mathbf{k} \\ x &=& \mathbf{t}\cdot\mathbf{i}/z \\ y &=& \mathbf{t}\cdot\mathbf{j}/z \end{array}\right. \quad (3)$$

$$\text{and } \left[\begin{array}{c} \mathbf{m} \\ \mathbf{n} \end{array}\right] = \frac{1}{z}\left[\begin{array}{c} \mathbf{i}-x\mathbf{k} \\ \mathbf{j}-y\mathbf{k} \end{array}\right].$$

We derive the paraperspective reconstruction in the next section and show in Section 3 its performance on real and synthetic data.

1.1 Ambiguity in the reconstruction :

It is well known that with perspective projection, 3D reconstruction can be solved only up to a scaled rotation and translation. With paraperspective projection, the ambiguity is up to a scaled *orthogonal transformation* (rotation or reflection). Thus, even when the scale, origin of world coordinates and orientation are fixed, there are two possible solutions to the reconstruction problem[2].

2 Closed-form reconstruction :

We first introduce some notation : the observations are arranged in a matrix

$$
\begin{aligned}
W_0 &= \begin{bmatrix} u_{11} & \cdots & u_{1P} \\ \vdots & & \vdots \\ u_{F1} & \cdots & u_{FP} \\ v_{11} & \cdots & v_{1P} \\ \vdots & & \vdots \\ v_{F1} & \cdots & v_{FP} \end{bmatrix} \\
&= \underbrace{\overbrace{\begin{bmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_F \\ \mathbf{n}_1 \\ \vdots \\ \mathbf{n}_F \end{bmatrix}}^{M} \overbrace{[\mathbf{s}_1 \ldots \mathbf{s}_P]}^{S}}_{W} + \begin{bmatrix} x_1 \\ \vdots \\ x_f \\ y_1 \\ \vdots \\ y_F \end{bmatrix} \underbrace{[1 \ldots 1]}_{1\times P} \\
&\quad + \text{Noise}
\end{aligned} \tag{4}
$$

Given a matrix W, a closed-form expression of a paraperspective reconstruction is obtained by :

A Estimate the centered observations W by $\hat{W}$ and $(x_1, y_1, \ldots, x_F, y_F)$ by $(\hat{x}_1, \hat{y}_1, \ldots, \hat{x}_F, \hat{y}_F)$ (Sections 2.1 and 2.2).

B Compute the SVD of $\hat{W}$, and a matrix $\tilde{M}$ that, because of noise, *approximately* verifies the properties of paraperspective projection matrices (Section 2.3).

C Compute a matrix M that verifies exactly the properties of paraperspective projection matrices, by modifying the 3rd column of $\tilde{M}$. Compute $z_1, \ldots, z_F$ and the camera frames $[\mathbf{i}_f, \mathbf{j}_f, \mathbf{k}_f]$ (Sections 2.4 and 2.5).

D Compute $S = \arg\min_S ||MS - \hat{W}||^2$, using $S \triangleq \left(M^\top M\right)^{-1} M^\top \hat{W}$.

Step **D** do not need further explanation. Steps **A** and **B** appear also in [3] and [1]. Our contribution lies in step **C**, where we propose a closed-form expression for the camera frames. This is an improvement over [3] where a non-linear optimization problem is solved for each frame, and over [1] where the vectors of the camera frames are non-orthogonal.

2.1 Estimating the translation x, y :

From our choice of reference of the coordinate system, one has $\sum_p \mathbf{s}_p = 0_3$. For any frame f, if the noise has zero mean, independent terms and variance is σ^2 , then $\hat{x}_f = \frac{1}{P}\sum_p u_{fp}$ and $\hat{y}_f = \frac{1}{P}\sum_p v_{fp}$ are unbiased estimates of x_f and y_f , with variance σ^2/P.

2.2 Centering the observations :

By substracting from W the newly-found (estimated) terms $\hat{x}_f$ and $\hat{y}_f$, one gets an estimate of W, which we shall call $\hat{W}$. This matrix, if it weren't for the noise , would be of rank 3. If the amplitude of the noise is small with respect to that of W, one expects [6][1] $\hat{W}$ to have 3 "big" eigenvalues, all the others being smaller. Our algorithm, described in Section 2.4, like that of [5] is based on the assumption that noise is small and that the singular vectors of $\hat{W}$ will be nearly equal to that of W.

2.3 An approximately paraperspective projection matrix :

We show how to approximate the matrix $\hat{W}$ by a rank-3 matrix, which is an *approximate* paraperspective projection. We do this using the same method as in [3, 1].

It is well-known that the best rank-3 approximation of $\hat{W}$ is provided by its Singular Value Decomposition (SVD):

$$\hat{W} = AHH^{-1}DB^\top,$$

where matrices A and B are formed by the first three left and right singular vectors of W, D is the diagonal matrix of singular values and H is any invertible 3×3 matrix.

The paraperspective projection model imposes some constraints on the projection matrix. From eq. (3), one sees that a pair of rows $(\mathbf{m}_f, \mathbf{n}_f)$ in a paraperspective projection matrix must verify :

$$
\left\{ \begin{array}{rcl} \mathbf{m}_f \cdot \mathbf{m}_f &=& \frac{1+x_f^2}{z_f^2} \\ \mathbf{n}_f \cdot \mathbf{n}_f &=& \frac{1+y_f^2}{z_f^2} \\ \mathbf{m}_f \cdot \mathbf{n}_f &=& \frac{x_f y_f}{z_f^2} \end{array} \right. \tag{5}
$$

If those constraints hold, then so does the following system of $2F$ linear equations[2] :

$$
\begin{aligned}
\frac{\mathbf{a}_f^\top Q \mathbf{a}_f}{1+x_f^2} &= \frac{\mathbf{b}_f^\top Q \mathbf{b}_f}{1+y_f^2} \\
\mathbf{b}_f^\top Q \mathbf{a}_f &= \frac{1}{2} x_f y_f \left(\frac{\mathbf{a}_f^\top Q \mathbf{a}_f}{1+x_f^2} + \frac{\mathbf{b}_f^\top Q \mathbf{b}_f}{1+y_f^2} \right)
\end{aligned} \tag{6}
$$

Where $Q = H^\top H$ and the $\mathbf{a}_f^\top$ and $\mathbf{b}_f^\top$ are the rows of A. This equations in the six distinct components of Q can be solved in the least-squares sense, with the constraint that $||Q||_F = 1$. If the resulting Q is positive definite, one finds easily, e.g. with the Cholesky decomposition of Q, a real

[1] Wilkinson provides bounds on the norm of the difference of the eigenvectors of the matrices with and without an error term, assuming that the error term has known bounds. Mathematical rigor would require that we know some statistical properties of the eigenvectors of the noisy matrix W + Noise.

[2] We do not detail the derivation because it is found in [3, 1].

matrix H such that $Q = H^\top H$. However, it may happen that Q is not positive semidefinite, and [3] (Section 4.3 or Appendix B.3) do not say what to do when that happens. Quan [4] proposes obtain directly for H by non-linear optimization, which does not necessarily yield the global optimum. The problem of obtaining an optimal H in a fail-less manner is left open.

2.4 The exact paraperspective matrix

The estimates of the motion parameters $\hat{x}_1, \ldots, \hat{x}_F, \hat{y}_1, \ldots, \hat{y}_F$ and the approximate paraperspective matrix $\tilde{M}$ allow us to compute an exact paraperspective matrix M and the depths z_f . We obtain M, which verifies exactly eq. (5), by modifying one single column of $\tilde{M}$.

We show that any two columns of a paraperspective matrix M and the values of x and y, determine a finite number of possibilities for the remaining column of M.

Lets assume that M has e.g. the first two columns in common with $\tilde{M}$. We use the notation :

$$\tilde{M} = \begin{bmatrix} \tilde{\mathbf{m}}_1 \\ \vdots \\ \tilde{\mathbf{m}}_F \\ \tilde{\mathbf{n}}_1 \\ \vdots \\ \tilde{\mathbf{n}}_F \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1 & \tilde{\mu}_1 \\ \vdots & \vdots \\ \mathbf{M}_F & \tilde{\mu}_F \\ \mathbf{N}_1 & \tilde{\nu}_1 \\ \vdots & \vdots \\ \mathbf{N}_F & \tilde{\nu}_F \end{bmatrix}.$$

where $\mathbf{M}_f, \mathbf{N}_f$ are 1×2 and $\tilde{\mu}_f, \tilde{\nu}_f$ are scalars. We compute μ_f and ν_f independently for each pair of rows $(\tilde{\mathbf{m}}_f, \tilde{\mathbf{n}}_f)$, and of values $(\hat{x}_f, \hat{y}_f)$.

We consider a single camera, and may omit the index f. We are given the first two columns of a 2-by-3 projection matrix $[\mathbf{m}; \mathbf{n}]$, along with the position components x and y. We will see how this data defines almost uniquely (up to a sign ambiguity) the 3rd column of the projection matrix, as well as the depth z. We write $\mathbf{m} \equiv [\mathbf{M}, \mu]$ and $\mathbf{n} \equiv [\mathbf{N}, \nu]$, where $\mathbf{M}$ and $\mathbf{N}$ are 1×2 and known, and μ and ν are scalars which will be determined.

One sees from eqs. (3) that that the paraperspective projection matrix must verify the equations :

$$\left\{ \begin{array}{lcl} \mathbf{M}\cdot\mathbf{M} + \mu^2 & = & \frac{1+x^2}{z^2} \\ \mathbf{N}\cdot\mathbf{N} + \nu^2 & = & \frac{1+y^2}{z^2} \\ \mathbf{M}\cdot\mathbf{N} + \mu\nu & = & \frac{xy}{z^2} \end{array} \right. \quad (7)$$

The first two equations allow us to express μ^2 and ν^2 as functions of z^{-2} and of known quantities. Substituting in the 3rd equation, one gets a polynomial equation of second degree in z^{-2}. This equation always has two solutions for z^{-2}. Only one of them corresponds to positive values of μ^2 and ν^2 and is kept. There are then two possible solutions for μ and ν, of the form $\{(\mu, \nu), (-\mu, -\nu)\}$. We keep the one that minimizes $||[\mu - \tilde{\mu}, \nu - \tilde{\nu}]||$.

2.5 Recovering motion :

The camera motion is obtained by first noting that

$$\mathbf{k} = \frac{-1}{1 + x^2 + y^2} \left[zx\mathbf{m} + zy\mathbf{n} - z^2 \mathbf{m} \times \mathbf{n} \right], \quad (8)$$

and then computing $\mathbf{i}$ and $\mathbf{j}$ from eq. (3). Eq. (8) is simpler than the expressions used by either [3] or [1]. Moreover, because the matrix M that they use is does not verify eqs. (5) exactly, the resulting vectors $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are not orthogonal. This is not mentioned in [1], which focuses on reconstruction; non-linear optimization is used in [3] (Section 4.4, page 211) to obtain orthogonal vectors.

3 Experimental results

This section presents experimental results. We use synthetic data, for which ground-truth is available, to measure the error committed by our algorithm. We compare with the error obtained with two Maximum-Likelihood (ML) estimators, using the paraperspective and perspective camera model, respectively. Finally, we illustrate results obtained on real-world data.

3.1 Reconstruction of synthetic data

The two following experiments show how the performance of our reconstruction algorithm is affected by the the camera depth and error in the observations. For this purpose, we built fifty datasets and run the proposed algorithm on them. Each setup consists in $P = 20$ points $S = \mathbf{s}_1, \ldots, \mathbf{s}_P$, produced by a $N(0, 1)$ (normal Gaussian) generator and then scaled, shifted and trimmed so that $\mathrm{Tr}(S^\top S) = 3$, $\sum_p \mathbf{s}_p = 0$ and for all f, $||\mathbf{s}_f|| \leq 2$. In the first experiment, $F = 10$ camera frames are defined by a rotation around a random $N(0_3, I_3)$ vector (its norm is the angle). The values of x_f and y_f are $N(0, 1)$, trimmed so that $|x_f|, |y_f| \leq 2$ and then scaled by one-fifth of the depth z. The maximum viewing angle is thus atan$(2/5) \simeq 22$. The depth z is varied, from 3 to 35. The observations are obtained by perspective projection, to which 40dB Gaussian white noise was added. In the second experiment, three cameras have their optical axis coincident with the $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ axis of the world reference frame; x and y are chosen like above and z is fixed to 10. The observations, obtained by perspective projection, are corrupted by noise, with standard deviation varying from 0 to 3.15% (30 dB).

For each data set, we computed the closed-form, paraperspective Maximum-Likelihood and perspective ML reconstructions (labeled **“closed-form”, “ML para”** and **“ML persp”** in the figures). The error E on the reconstructed structure is measured up to the ambiguity described in Section 1.1, that is, $E = \min_{R,t,h} \sqrt{\frac{1}{P} \sum_p ||\mathbf{s}_p - hR\hat{\mathbf{s}}_p - t||^2}$ where h is a real number, t a translation and R is an orthogonal matrix (paraperspective algorithm) or rotation matrix

(ML algorithm). Figure 2 shows the median value of E, taken over all fifty setups. These curves show

- As expected, the error decreases when the distance increases and paraperspective projection better approximates perspective projection.
- The paraperspective ML estimator is not very much better than the closed-form estimator.
- The perspective ML reconstruction is much more accurate, in most situations.

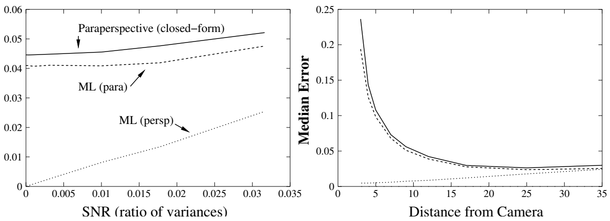

Figure 2. Median error in the structure parameters.

3.2 Reconstruction from real-world data

Figure 3 show the first and last images of the real-world sequences that were used. Some points have been tracked by hand : the forty eight corners of the black squares on the calibration grid, and the forty-four points marked on the noodle drain. The images are taken from less than a meter, thus from close-range. Figure 4 shows the reconstructions.

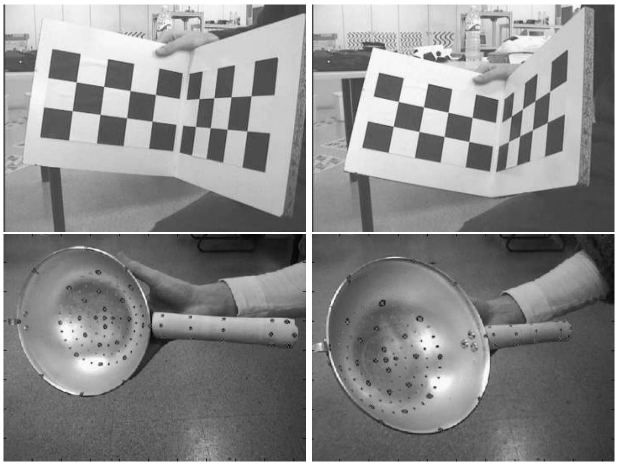

Figure 3. First and last images.

Although there is severe distortion, the overall shape and camera positions are realistic, and sufficient for starting a ML estimator.

4 Conclusions

We have presented a closed-form expression for 3D reconstruction using the paraperspective projection model.

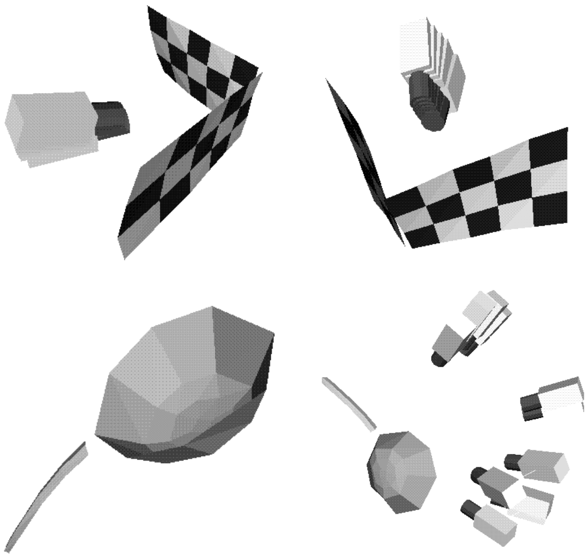

Figure 4. Real-world results.

It's main advantage is that it enforces -exactly- the constraints of paraperspective projection matrices, thus allowing for direct computation of camera position, through a purely algebraic procedure. This new method has been shown, experimentally, to provide an appropriate starting point for a more sophisticated reconstruction algorithm.

References

[1] Stéphane Christy and Radu Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. Technical Report 2421, INRIA, 1994.

[2] Stéphane Christy and Radu Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *PAMI*, 18(11):1098–1104, November 1996.

[3] Conrad J. Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE PAMI*, 19(3):206–218, March 1997.

[4] Long Quan. Self-calibration of an affine camera from multiple views. *IJCV*, 19(1):93–105, 1996.

[5] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams : a factorization method. Technical Report TR 92-1270 and CMU-CS-92-104, Cornell University and Carnegie Mellon University, 1992.

[6] J. H. Wilkinson. *The algebraic eigenvalue problem.* Oxford: Clarendon, 1965,1988.