2012 Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization & Transmission

Solving the PnP Problem with Anisotropic Orthogonal Procrustes Analysis

Valeria Garro
Dept. of Computer Science
University of Verona, Italy
valeria.garro@univr.it

Fabio Crosilla
DICA
University of Udine, Italy
fabio.crosilla@uniud.it

Andrea Fusiello
DIEGM
University of Udine, Italy
andrea.fusiello@uniud.it

Abstract

In this paper we formulate the Perspective-n-Point (a.k.a. exterior orientation) problem in terms of an instance of the anisotropic orthogonal Procrustes problem, and derive its solution. Experiments with synthetic and real data demonstrate that our method reaches the best trade-off between speed and accuracy. The MATLAB code reported in the paper testifies that it is also exceedingly simple to implement.

1. Introduction

The problem of estimating the position and orientation of a perspective camera given its intrinsic parameters and a set of world-to-image correspondences is known as the *Perspective-n-Point* camera pose problem (PnP) in computer vision or *exterior orientation* problem in photogrammetry. In both communities it has received much attention, being one of the building blocks for many applications like structure from motion, object/camera localization, robotics, just to mention a few. An extensive literature exists on this topic, which will be summarized in Sec. 2.

The trade-off that all the methods face is between speed and accuracy. Direct methods are usually faster but less accurate, as they do not minimize a significant cost function, whereas iterative methods, that explicitly minimize a meaningful geometric error are more accurate but slower. Our method reaches the best trade-off between speed and accuracy and is easier to implement than its closest competing algorithm [21].

On the theoretical side, the contribution of this paper lies in casting the PnP problem as an instance of the *Orthogonal Procrustes* (OP) problem, namely an *anisotropic* OP [4] where each measurement may have a different scaling factor. This particular version of the problem has never been considered before in the literature, hence we will derive its general solution.

In this respect, the closest work to our is [10]: the author recognizes that, if the depths of the 3D points are known, the exterior orientation problem can be reduced to an *absolute orientation* problem. Hence, he first finds the depth with an handcrafted method and then recovers the absolute orientation by solving an orthogonal Procrustes problem.

2. Background and Related Work

In literature several solutions for the PnP problem have been presented. They can be classified along several dimensions, the most relevant being: iterative/closed form, minimal ($n = 3$ or 4)/general ($\forall n \geq 3$), linear/non-linear solution. Table 1 gives an overview of this classification, whereas each method will be briefly described in the following.

Iterative approaches ([27], [18], [25],[22]) generally formulate the pose estimation as a non-linear least-squares problem minimizing a cost function related to a geometric (e.g. reprojection residual) or algebraic error [20]. They usually achieve better accuracy compared to the closed-form methods at a price of a significant computational load. They generally need a good initial guess in order to converge properly to the correct solution. In [28] the authors minimize an error based on collinearity in 3D space, they successively improve an estimate of the rotation matrix enforcing the orthogonal constraint and then compute the translation vector. The method proposed in [7] (POSIT) applies iteratively a linear closed-form solver for a scaled orthographic projection camera. It alternates between computing the (orthographic) pose and computing correction factors that are directly connected to the points depth. The final *perspective* pose is obtained from a sequence of scaled orthographic approximations. More recently, [31] presented an algorithm that transforms in linear time the minimization problem into a semi definite positive program (SDP) and than solves it using SeDuMi [34], an iterative solver that yield a global optimal solution.

Conversely, closed-form algorithms directly compute the camera pose formulating the problem into a system of equations. The first closed-form methods [17, 9, 11] dealt with the so-called P3P problem, which aims at finding the exterior orientation for the minimal case $n = 3$. The P3P

978-0-7695-4873-9/12 $26.00 © 2012 IEEE
DOI 10.1109/3DIMPVT.2012.40

Table 1. Classification of state-of-the-art methods

	iterative	closed form	$n = 3$	n general	linear	nonlinear
Lowe [27]	x			x		x
Hesch [21]		x		x		x
Fiore [10]		x		x	x	
Schweighofer [31]	x			x	x	
Lepetit [26]		x		x	x	
Kneip [24]		x	x			x
Ansar [1]		x		x	x	
Lu [28]	x			x	x	
Gao [13]		x	x			x
DeMenthon [7]	x			x	x	

problem implies the computation of multiple solutions, up to four as first explained in [9, 17]. An extensive review of these works published before 1994 can be found in [19]. More recently, [13] presented an algebraic approach combined with a set of analytical criteria to determine the number of real positive solutions. They also provided a more intuitive geometric approach that involves linear inequalities. In [24] the authors proposed a novel parameterization of the P3P problem which computes in one stage the orientation and the position of the camera in the world reference frame. Several solutions [11, 30, 10, 1, 26] have been proposed for the general case $n \geq 3$. In [26] the main idea is to represent the n points in the object space as a weighted sum of only four control points and consequently to solve the problem in terms of the camera reference coordinates of these four virtual points. This transformation can be done in $O(n)$. Hesh *et al.* [21] recently presented a closed-form method that minimize a nonlinear least-squares cost function whose size is not dependent on the number of points.

Orthogonal Procrustes analysis is a valuable tool to perform the direct least squares solution of similarity transformation problems in any dimensional space. At first, it was used for the multidimensional rotation and scaling of different matrix configuration pairs [32, 33]. Successively, the solution was generalized for the simultaneous least squares matching of more than two corresponding matrices [16, 35]. Procrustes analysis was originally applied in factor analysis. Only more recently, this technique became popular and applied for the direct transformation problem solution in many other scientific fields, like for instance shape analysis [15, 8], shape registration [29], medicine [14].

Some applications of the OP analysis in photogrammetry, for matching different 3D object models from images or matching 3D laser point clouds, are due to [5] and [2] respectively. In these cases the authors applied the so called generalized version of the OP problem to simultaneously match more than two coordinate matrices of corresponding points expressed in different reference systems.

Very recently, an interesting extension of the OP analysis with anisotropic scaling has been proposed by [4], also in its generalized form [3]. The algorithm deals with anisotropic scaling along space dimensions and is based on an iterative block relaxation technique [6] that starts with an uninformed initialization.

3. Method

Given a number of 2D-3D point correspondences $\mathbf{m}_i \leftrightarrow \mathbf{M}_i$ and the intrinsic camera parameters K, the PnP problem requires to find a rotation matrix R and a translation vector $\mathbf{t}$ (which specify attitude and position of the camera) such that:

$$\zeta_i \tilde{\mathbf{m}}_i = K[R|\mathbf{t}]\tilde{\mathbf{M}}_i \quad \text{for all } i. \tag{1}$$

where ζ_i denotes the depth[1] of $\mathbf{M}_i$, and the $\tilde{}$ denotes homogeneous coordinates (with a trailing "1").

After some rewriting, (1) becomes:

$$\underbrace{\begin{bmatrix} \zeta_1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \zeta_n \end{bmatrix}}_{Z} \underbrace{\begin{bmatrix} \tilde{\mathbf{p}}_1^\mathsf{T} \\ \vdots \\ \tilde{\mathbf{p}}_n^\mathsf{T} \end{bmatrix}}_{P} R + \underbrace{\begin{bmatrix} \mathbf{c}^\mathsf{T} \\ \vdots \\ \mathbf{c}^\mathsf{T} \end{bmatrix}}_{\mathbf{1c}^\mathsf{T}} = \underbrace{\begin{bmatrix} \mathbf{M}_1^\mathsf{T} \\ \vdots \\ \mathbf{M}_n^\mathsf{T} \end{bmatrix}}_{S}. \tag{2}$$

where $\tilde{\mathbf{p}}_i = K^{-1}\tilde{\mathbf{m}}_i$, $\mathbf{c} = -R^\mathsf{T}\mathbf{t}$, and $\mathbf{1}$ is the unit vector. In matrix form:

$$S = ZPR + \mathbf{1c}^\mathsf{T} \tag{3}$$

where

P is the matrix by rows of (homogeneous) image coordinates defined in the camera frame,

S is the matrix by rows of point coordinates defined in the external system,

Z is the diagonal (positive) depth matrix,

[1] The depth of a point is its distance from the focal plane of the camera.

$\mathbf{c}$ is the coordinate vector of the projection centre,

R is the orthogonal rotation matrix.

One can recognize an instance of the classical OP model [33], generalized by the fact that the usual (isotropic) scale factor is substituted by an anisotropic scaling characterized by a diagonal matrix Z of different scale factors. Indeed this can be defined as an *anisotropic* OP problem with *data* scaling.

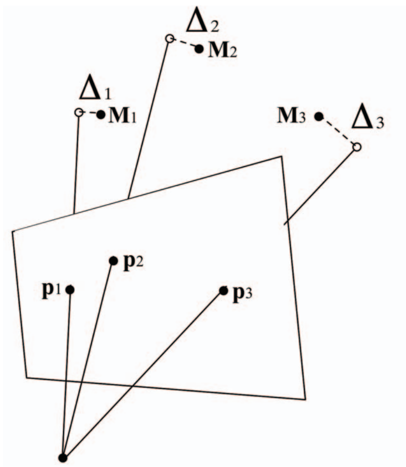

Figure 1. The position and attitude of the camera plus the depth of the points are estimated in such a way to minimize the length of the Δs for all the points, in a least squares sense. The estimated depth defines a 3D point along the optical ray of the image point $\mathbf{p}$. The segment (perpendicular to the optical ray) joining this point and the corresponding reference 3D point $\mathbf{M}$ is Δ.

3.1. Analytical Solution of the Anisotropic Orthogonal Procrustes

Anisotropic OP comes in – at least – three flavors: pre/post scaling on the variables (or space dimensions) [4], and data scaling, where each data point or measurement can be scaled independently of the others. The latter version – which fits our formulation of the PnP problem – has never been considered in the literature, hence we derive its general solution here along the same line as in [33].

To obtain the least squares solution for model (3), let us make explicit the residual matrix Δ:

$$S = ZPR + \mathbf{1}\mathbf{c}^{\mathsf{T}} + \Delta. \tag{4}$$

The geometric interpretation of Δ is a matrix whose rows are difference vectors between reference 3D points (S) and the the back-projected 2D points (P) based on their estimated depths (Z) and the estimated camera attitude and position ($R, \mathbf{c}$). The solution of the anisotropic OP problem finds Z, R and $\mathbf{c}$ in such a way to minimize the sum of squares of the residual Δ, i.e., the sum of the squared norm of the difference vectors mentioned above (see Fig. 1). This can be written as

$$\min \|\Delta\|_F^2 \text{ subject to } R^{\mathsf{T}}R = I \tag{5}$$

The problem is equivalent to the minimization of the the Lagrangian function

$$F = \operatorname{tr}\left(\Delta^{\mathsf{T}}\Delta\right) + \operatorname{tr}\left(L\left(R^{\mathsf{T}}R - I\right)\right) \tag{6}$$

where L is the matrix of Lagrangian multipliers. This can be solved by setting to zero the partial derivatives of F with respect to the unknowns R, $\mathbf{c}$ and the diagonal matrix Z. The derivation of the formulae is reported in Appendix A.

Whereas in the classical solution of the extended OP problem [33] one can recover first R, that does not depend on the other unknowns, then the isotropic scale (that depends only on T) and finally $\mathbf{c}$, in the anisotropic case the unknowns are entangled in such a way that one must resort to the so called "block relaxation" scheme [6], (of which EM is the most famous instance), where each variable is alternatively estimated while keeping the others fixed. The algorithm can be summarized as follows:[2]

Algorithm 1 PROCRUSTEAN PNP (PPNP)

Input: a set of 2D-3D correspondences (P, S)
Output: position $\mathbf{c}$ and attitude R of the camera

1. Start with $Z = 0$ (or any guess on the depth);
2. Compute $R = U\operatorname{diag}\left(1{,}1{,}\det(UV^{\mathsf{T}})\right)V^{\mathsf{T}}$ with $UDV^{\mathsf{T}} = P^{\mathsf{T}}Z\left(I - \mathbf{1}\,\mathbf{1}^{\mathsf{T}}/n\right)S$;
3. Compute $\mathbf{c} = (S - ZPR)^{\mathsf{T}}\,\mathbf{1}/n$;
4. Compute $Z = \operatorname{diag}\left(PR(S^{\mathsf{T}} - \mathbf{c}\mathbf{1}^{\mathsf{T}})\right)\operatorname{diag}\left(PP^{\mathsf{T}}\right)^{-1}$;
5. Iterate over steps 2, 3, 4 until convergence.

In step 2 we use $R = U\operatorname{diag}\left(1{,}1{,}\det(UV^{\mathsf{T}})\right)V^{\mathsf{T}}$ instead of $R = UV^{\mathsf{T}}$ in order to guarantee that R is not only orthogonal but has positive determinant [23]. Moreover, the non-negativity constraint on Z must be enforced a-posteriori in step 4 by clipping to zero negative values, if any.

One might recognize in this alternation between computing pose and computing depth a resemblance of the POSIT algorithm [7].

4. Experimental validation

We compared our Procrustean PnP algorithm (**PPnP**) with state of the art, recent algorithms whose implementation is available on the WWW, namely:

[2]The diag operator is overloaded: when applied to a vector it returns a matrix with a given diagonal; when applied to a matrix it suppresses off-diagonal elements.

DLS: the direct least-squares method of Hesch and Roumeliotis [21];

EPnP: the non-iterative linear solution of Lepetit *et al.* [26];

gOp: the Semi Definite Program approach of Schweighofer and Pinz [31];

Fiore: the SVD-based method by Fiore [10][3].

The MATLAB code of PPnP is reported in Appendix B. In all the experiments the initial depths were set to zero, as specified in the algorithm, and the method always converged to the correct solution. The algorithm terminates when the difference between the values of the residual matrix of two consecutive iterations is smaller than a threshold (set to 10^{-5} in our experiments).

We run both synthetic and real experiments. In the synthetic case we randomly distribute $n = \{6, \ldots, 50\}$ 3D points at a distance between 0.5 and 1.5 meters from the camera. The focal length of the camera were set to 600 pixel and the image resolution is 800×600. The image coordinates obtained from the projection of the 3D points have been perturbed with different values of noise $\sigma = \{1, \ldots, 10\}$ [pixel]. For each value of n the test has been run 100 times and the average error norm has been computed.

Results are reported in Figs. 2, 3, and 4. As a figure of merit the rotation and translation errors are shown. The former is the angle of the residual rotation, computed as $\|\log(R^\mathsf{T}\hat{R})\|_F$, where R is the ground truth, $\hat{R}$ is the actual rotation and $\|\cdot\|_F$ is the Frobenius norm.

Please note that, although Fiore has the closest formulation to ours, its accuracy is consistently worse. This is because in the depth recovery step an algebraic residual is implicitly minimized, hence the final solution is not statistically optimal.

The rundown of the experiments is that, among the three more accurate methods (PPnP, gOP and DLS), our method is the fastest, although to a small degree. It should be noticed, however, that comparing MATLAB implementations for performances gives only rough figures (as pointed out by [21]). Apart from that, there are other positive features that need to be considered, namely:

- casting PnP in the procrustean framework is interesting *per-se*;
- the PPnP algorithm is particularly easy to implement (see Appendix B) for it only needs matrix algebra and SVD (cf. [21, 31]);
- being iterative, the speed vs accuracy trade-off can be controlled with the termination threshold;
- albeit iterative, PPnP always converged with an uninformed initialization ($Z = 0$) in all our experiments.

In real experiments we considered real data coming from structure-and-motion reconstructions, shown in Fig. 5, where cameras and points have been bundle adjusted. In the lack of a proper ground truth we took this as the reference. Basing on given 2D-3D correspondences we oriented each camera with respect to the structure and computed the errors. We choose randomly $n = \{20, \ldots, 50\}$ 2D-3D pairs from the set of correspondences of each camera. For these experiments the minimum value of n has been set to 20 instead of 6 in order to deal with the presence of noise. For each n value we run the test 100 times and the orientation and translation average error norms have been computed. In this case we could not test gOP algorithm because the code for the planar case is not longer available on the WWW. The first dataset is shown in Fig. 5, results are reported in Fig. 6. The second reported experiment has been run using the standard dataset "Herz-Jesu-P8" available on the WWW[4]. The related results are shown in Fig. 7. The results of the real experiments confirm the algorithms' performance in the synthetic case, PPnP achieves always the best accuracy, instead DLS has a slightly worse error, with an unexplainable glitch for camera n. 4 of the "RectStone" dataset.

5. Conclusions

The contribution of the paper is both theoretical and practical: on the theoretical side it provides a procrustean formulation of the PnP problem; on the practical side it describes an algorithm which is very easy to implement (MATLAB code provided in Appendix B) and achieves the optimal trade-off between speed and accuracy, as experiments have shown.

A. Derivation of the solution of the Anisotropic Procrustes Model

Let us start from (6), and substitute (4) for Δ:

$$\begin{aligned} F = &\operatorname{tr}\left(S^\mathsf{T} S\right) + \operatorname{tr}\left(R^\mathsf{T} P^\mathsf{T} Z^\mathsf{T} Z P R\right) + n \operatorname{tr}\left(\mathbf{c}^\mathsf{T}\mathbf{c}\right) - \\ &- 2\operatorname{tr}\left(S^\mathsf{T}\mathbf{1}\mathbf{c}^\mathsf{T}\right) - 2\operatorname{tr}\left(S^\mathsf{T} Z P R\right) + \\ &+ 2\operatorname{tr}\left(R^\mathsf{T} P^\mathsf{T} Z^\mathsf{T}\mathbf{1}\mathbf{c}^\mathsf{T}\right) + \operatorname{tr}\left(L\left(R^\mathsf{T} R - I\right)\right) \end{aligned} \tag{7}$$

The projection centre $\mathbf{c}$ can be obtained by equating to zero the partial derivative:

$$\frac{\partial F}{\partial \mathbf{c}} = 2n\mathbf{c} - 2S^\mathsf{T}\mathbf{1} + 2R^\mathsf{T} P^\mathsf{T} Z^\mathsf{T}\mathbf{1} = 0 \tag{8}$$

[3]The code for this method is not available on the WWW, so we used our implementation of the formulation found in [12].

[4]http://cvlab.epfl.ch/~strecha/multiview/denseMVS.html

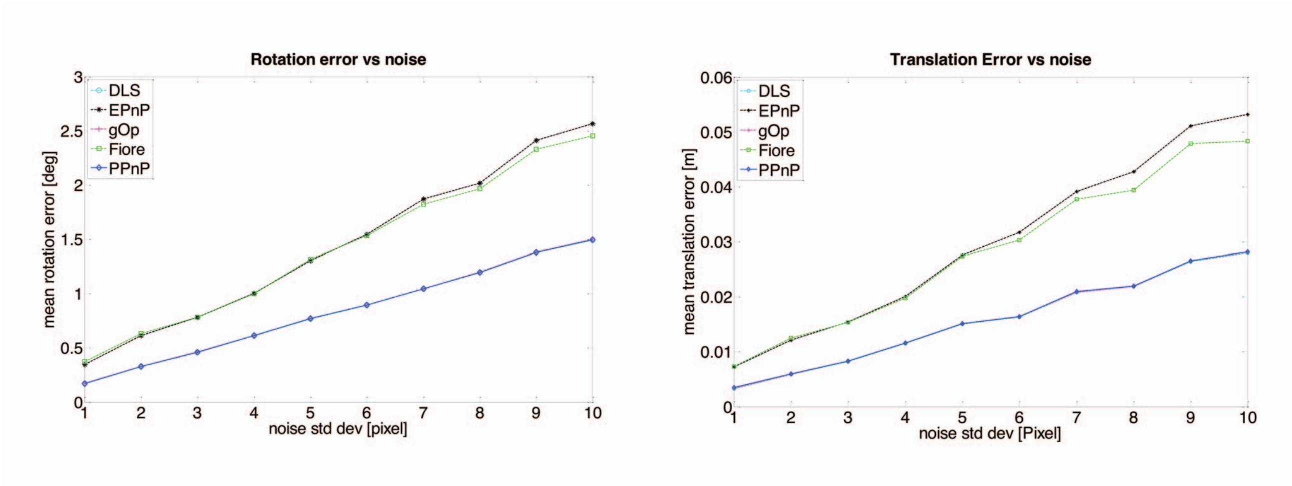

Figure 2. Rotation and translation errors vs noise using 30 correspondences. The average rotation error and the average translation error are plotted against the standard deviation of the noise added to image coordinates. PPnP, gOP, and DLS are practically superimposed with the lowest error.

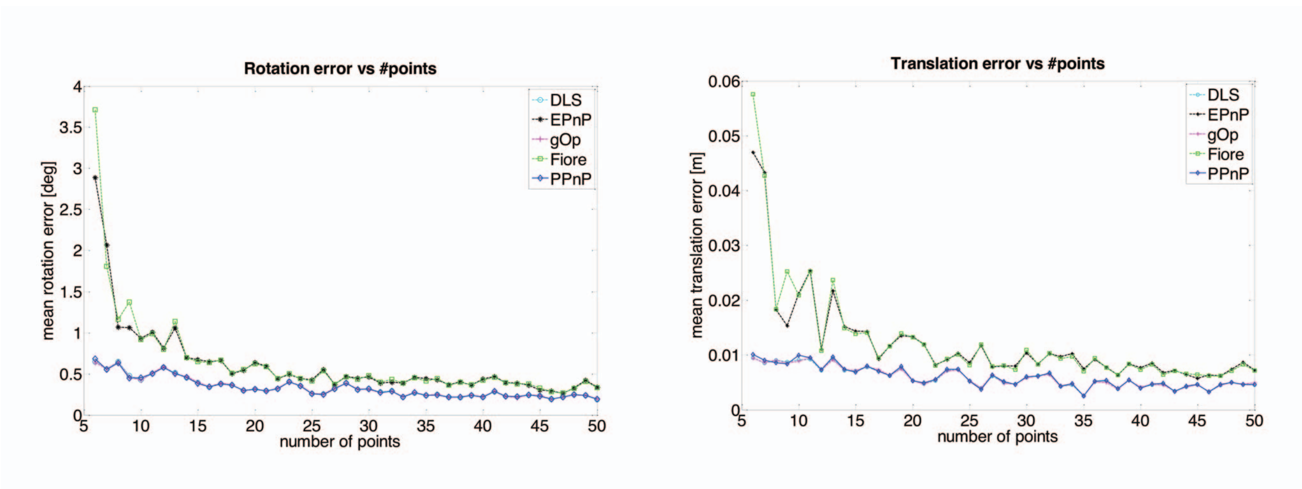

Figure 3. Rotation and translation error vs number of points. The average rotation and translation error are plotted against the number of points that have been used.

Hence:

$$\mathbf{c} = (S - ZPR)^\mathsf{T}\,\mathbf{1}/n \tag{9}$$

Once the derivatives of F with respect to R and $\mathbf{c}$ are set to zero, it results:

$$\begin{aligned}\frac{\partial F}{\partial R} =& P^\mathsf{T}Z^\mathsf{T}ZPR - P^\mathsf{T}Z^\mathsf{T}S + P^\mathsf{T}Z^\mathsf{T}\mathbf{1}\mathbf{c}^\mathsf{T} + \\ & + R(L + L^\mathsf{T})/2 = 0\end{aligned} \tag{10}$$

where $Q = (L + L^\mathsf{T})/2$.

Let us multiply (10) on the left by R^T:

$$\begin{aligned}& R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}ZPR - R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}S + R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}\mathbf{1}\mathbf{c}^\mathsf{T} + \\ & + R^\mathsf{T}R(L + L^\mathsf{T})/2 = 0\end{aligned} \tag{11}$$

Since matrices $R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}ZPR$ and $(L + L^\mathsf{T})/2$ are symmetric, then

$$\operatorname{sym}\left[R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}S - R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}\mathbf{1}\mathbf{c}^\mathsf{T}\right]. \tag{12}$$

Substituting (9) in (12), it results [5]

$$\operatorname{sym}\left[R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}S - R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}\left(\mathbf{1}\,\mathbf{1}^\mathsf{T}/n\right)(S - ZPR)\right] \tag{13}$$

which is equivalent to

$$\begin{aligned}&\operatorname{sym}[R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}S - R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}\left(\mathbf{1}\,\mathbf{1}^\mathsf{T}/n\right)S + \\ &\qquad + R^\mathsf{T}P^\mathsf{T}Z^\mathsf{T}\left(\mathbf{1}\,\mathbf{1}^\mathsf{T}/n\right)ZPR]\end{aligned} \tag{14}$$

[5]The predicate sym[ ] is true when the argument is a symmetric matrix.

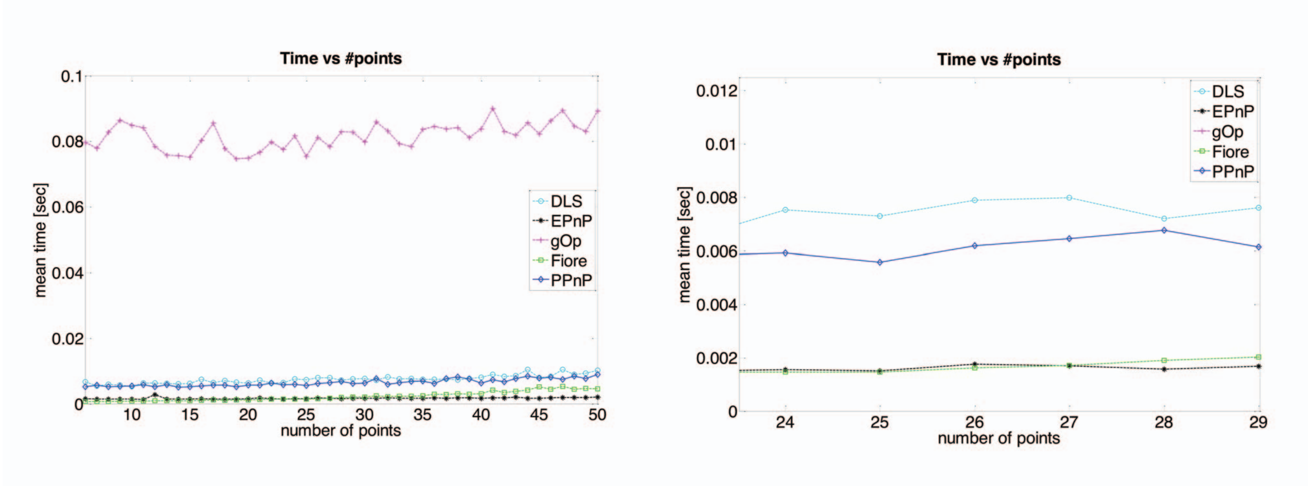

Figure 4. Execution time vs number of points. The average execution time is plotted against the number of points involved. gOP is by far the slowest algorithm. On the right a zoom-in shows that PPnP is slower than EPnP and Fiore, but slightly faster than DLS.

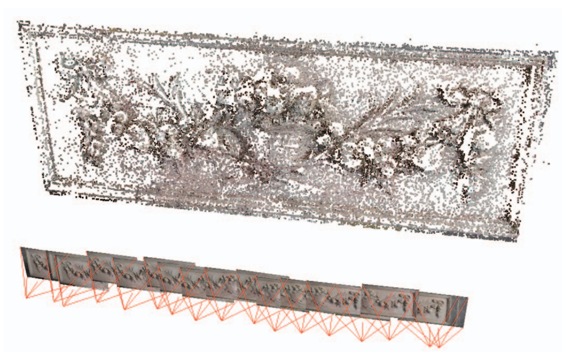
Figure 5. The "RectStone" dataset. The figure depicts point and cameras obtained by a structure-and-motion pipeline.

and finally:

$$\mathrm{sym}[R^\top P^\top Z^\top \left(I - \mathbf{1}\,\mathbf{1}^\top/n\right) S + \\ + R^\top P^\top Z^\top \left(\mathbf{1}\,\mathbf{1}^\top/n\right) ZPR]. \quad (15)$$

Since $R^\top P^\top Z^\top \left(\mathbf{1}\,\mathbf{1}^\top/n\right) ZPR$ is symmetric, also the first term must be symmetric, i.e.,

$$\mathrm{sym}\left[R^\top P^\top Z^\top \left(I - \mathbf{1}\,\mathbf{1}^\top/n\right) S\right] \quad (16)$$

is also symmetric.

Let us define the matrix T equal to

$$T = P^\top Z^\top \left(I - \mathbf{1}\,\mathbf{1}^\top/n\right) S \quad (17)$$

Matrix $R^\top T$ is symmetric, therefore the following condition must be satisfied

$$R^\top T = T^\top R \quad (18)$$

that is equivalent to

$$TT^\top = RT^\top TR^\top \quad (19)$$

Let $T = UDV^\top$ be the SVD of T, with matrices V, U orthonormal. Substituting into (19) yields:

$$UD^2U^\top = RVD^2V^\top R^\top \quad (20)$$

From (20) $U = RV$ and finally $R = UV^\top$.

The least squares solution for the diagonal matrix Z can be obtained by setting to zero the partial derivatives of (7) with respect to Z.

$$\begin{aligned}
\frac{\partial F}{\partial Z} &= \frac{\partial}{\partial Z}\,\mathrm{tr}\,(R^\top P^\top Z^\top ZPR) - 2\frac{\partial}{\partial Z}\,\mathrm{tr}\,(S^\top ZPR) + \\
&\quad + 2\frac{\partial}{\partial Z}\,\mathrm{tr}\,(R^\top P^\top Z^\top \mathbf{1}\mathbf{c}^\top) \\
\frac{\partial F}{\partial Z} &= \frac{\partial}{\partial Z}\,\mathrm{tr}\,(ZPRR^\top PZ^\top) - 2\frac{\partial}{\partial Z}\,\mathrm{tr}\,(PRS^\top Z) + \\
&\quad + 2\frac{\partial}{\partial Z}\,\mathrm{tr}\,(PR\mathbf{c}\mathbf{1}^\top Z) \\
\frac{\partial F}{\partial Z} &= Z(2PRR^\top P^\top) - 2PRS^\top + 2PR\mathbf{c}\mathbf{1}^\top \\
\frac{\partial F}{\partial Z} &= 2ZPP^\top - 2PRS^\top + 2PR\mathbf{c}\mathbf{1}^\top
\end{aligned} \quad (21)$$

By setting the derivatives to zero one obtains:

$$ZPP^\top = PR(S^\top - \mathbf{c}\mathbf{1}^\top) \quad (22)$$

hence

$$Z = \mathrm{diag}\left(PR(S^\top - \mathbf{c}\mathbf{1}^\top)\right) \mathrm{diag}\left(PP^\top\right)^{-1} \quad (23)$$

where $\mathrm{diag}(\cdot)$ returns a diagonal matrix that has the same diagonal elements of its argument.

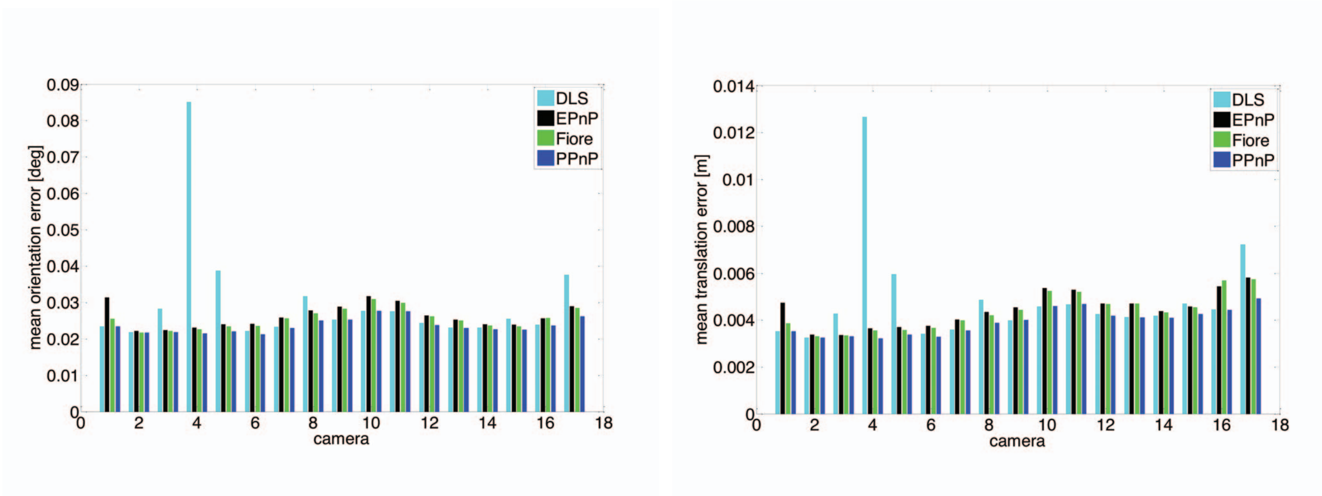

Figure 6. Mean rotation and translation error for each of the 17 cameras of the "RectStone" dataset.

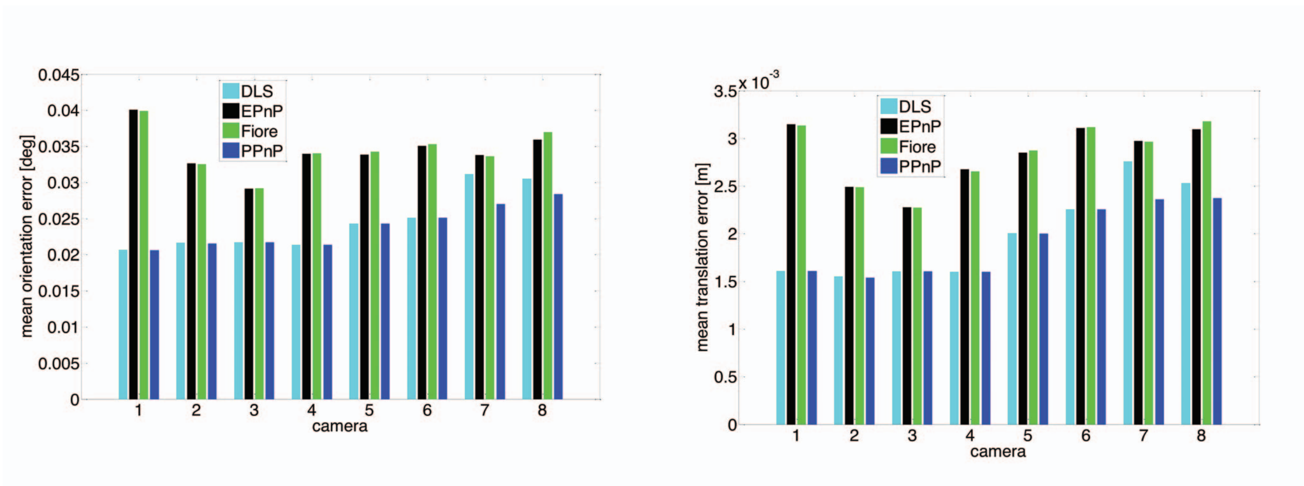

Figure 7. Mean rotation and translation error for each of the 8 cameras of the "Herz-Jesu-P8" dataset.

B. MATLAB code

```
function [R T] = ppnp(P,S,tol)
% input
% P : matrix (nx3) image coordinates in camera
%     reference [u v 1]
% S : matrix (nx3) coordinates in world
%     reference [X Y Z]
% tol: exit threshold
% output
% R : matrix (3x3) rotation (world-to-camera)
% T : vector (3x1) translation  (world-to-camera)

n = size(P,1); Z = zeros(n); e = ones(n,1);
A = eye(n)-((e*e')./n); II = e./n;
err = +Inf; E_old = 1000*ones(n,3);

while err>tol
    [U,~,V] = svd(P'*Z*A*S);
    VT = V';
    R=U*[1 0 0; 0 1 0; 0 0 det(U*VT)]*VT;
    PR = P*R;
    c = (S-Z*PR)'*II;
    Y = S-e*c';
    Zmindiag = diag(PR*Y')./(sum(P.*P,2));
    Zmindiag(Zmindiag<0)=0; Z = diag(Zmindiag);
    E = Y-Z*PR;
    err = norm(E-E_old,'fro'); E_old = E;
end
T = -R*c;
end
```

Acknowledgments

The authors are indebted with Dr. Mohammed Bennani Dosse for providing relevant suggestions about the problem solution. Thanks to the anonymous reviewer for pointing out the connection with POSIT.

References

[1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578 – 589, 2003. 2

[2] A. Beinat and F. Crosilla. Generalized procrustes analysis for size and shape 3d object reconstruction. In *Optical 3-D Measurement Techniques*, pages 345–353. 2001. 2

[3] M. Bennani Dosse, H. A. L. Kiers, and J. Ten Berge. Anisotropic generalized procrustes analysis. *Computational Statistics & Data Analysis*, 55(5):1961–1968, 2011. 2

[4] M. Bennani Dosse and J. Ten Berge. Anisotropic orthogonal procrustes analysis. *Journal of Classification*, 27(1):111–128, 2010. 1, 2, 3

[5] F. Crosilla and A. Beinat. Use of generalised procrustes analysis for the photogrammetric block adjustment by independent models. *ISPRS Journal of Photogrammetry & Remote Sensing*, 56(3):195–209, 2002. 2

[6] J. de Leeuw. Block-relaxation algorithms in statistics. In *Information Systems and Data Analysis*, page 308325. Springer-Verlag, 1994. 2, 3

[7] D. F. DeMenthon and L. S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995. 1, 2, 3

[8] I. L. Dryden and K. Mardia. *Statistical shape analysis*. J. Wiley, 1998. 2

[9] S. Finsterwalder and W. Scheufele. *Das Rückwärtseinschneiden im Raum*. Verlag der K. Akademie, 1903. 1, 2

[10] P. D. Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):140–148, 2001. 1, 2, 4

[11] M. A. Fischler and R. C. Bolles. Random Sample Consensus: a paradigm model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 2

[12] A. Fusiello. A matter of notation: Several uses of the Kronecker product in 3-D computer vision. *Pattern Recognition Letters*, 28(15):2127–2132, 2007. 4

[13] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:930–943, 2003. 2

[14] M. S. Gelfand, A. A. Mironov, and P. A. Pevzner. Gene Recognition Via Spliced Sequence Alignment. *Proc. of the National Academy of Sciences of the United States of America*, 93(17):9061–9066, 1996. 2

[15] C. Goodall. Procrustes Methods in the Statistical Analysis of Shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(2):285–339, 1991. 2

[16] J. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 2

[17] G. Grunert. Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in geodäsie. *Archiv für Mathematik und Physik*, 1841. 1, 2

[18] R. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426 –1446, 1989. 1

[19] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13:331–356, 1994. 2

[20] R. I. Hartley. Minimizing algebraic error in geometric estimation problems. In *Proc. of the International Conference on Computer Vision*, pages 469–476, 1998. 1

[21] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) solution for PnP. In *Proc. of the International Conference on Computer Vision*, 2011. 1, 2, 4

[22] R. Horaud, F. Dornaika, and B. Lamiroy. Object pose: The link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, 22:173–189, 1997. 1

[23] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993. 3

[24] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2

[25] R. Kumar and A. R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP: Image Understanding.*, 60:313–342, 1994. 1

[26] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate $o(n)$ solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009. 2, 4

[27] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441 –450, 1991. 1, 2

[28] C. ping Lu, G. D. Hager, I. C. Society, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:610–622, 2000. 1, 2

[29] D. Pizarro and A. Bartoli. Global optimization for optimal generalized procrustes analysis. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2409–2415, 2011. 2

[30] L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774 –780, 1999. 2

[31] G. Schweighofer and A. Pinz. Globally optimal o(n) solution to the pnp problem for general camera models. In *Proc. of the British Machine Vision Conference*, pages 55.1–55.10, 2008. 1, 2, 4

[32] P. Schnemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 2

[33] P. Schnemann and R. Carroll. Fitting one matrix to another under choice of a central dilation and a rigid motion. *Psychometrika*, 35(2):245–255, 1970. 2, 3

[34] J. F. Sturm. Using SEDUMI 1.02, a MATLAB* toolbox for optimization over symmetric cones, 2001. Downloadable from `http://sedumi.mcmaster.ca`. 1

[35] J. Ten Berge. Orthogonal procrustes rotation for two or more matrices. *Psychometrika*, 42(2):267–276, 1977. 2