

Decoupling Orientation Recovery from Position Recovery with 3D–2D Point Correspondences*

Rodrigo L. Carceroni

Christopher M. Brown

University of Rochester
Computer Science Department
Rochester, NY – 14627 – USA

Abstract

We propose a new algorithm for model–based extrinsic camera calibration that allows one to separate the recovery of the relative orientation of the camera from the recovery of its relative position, given a set of at least three correspondences between model and image points. The key idea is to replace each (real) model point whose correspondence is known by two (virtual) model edges, and then to use the fact that these edges have pairwise intersections in 3D space to derive a set of alignment constraints. We provide a proof that the resulting technique is essentially more powerful than any of the traditional methods for decoupled orientation and position recovery based uniquely on line correspondences. We also present a detailed example of a real–life application that benefits from our work, namely autonomous navigation using distant visual landmarks. We use simulation to show that, for this specific application, our algorithm, when compared to similar techniques, is either significantly more accurate at the same computational cost, or significantly faster with roughly the same average–case accuracy.

1 Introduction

The recovery of the relative orientation and position (pose) of a camera in a scene, given the resulting 2D image, is a central problem in computer vision known as *extrinsic (external) camera calibration*. Applications in which this problem arises are numerous: cartography, tracking, object recognition, hand–eye coordination, augmented reality and visual navigation. In many cases, a description of the three–dimensional geometry of the scene is available *a priori*, allowing the use of *model–based* techniques. These techniques typically use known correspondences between model features such as vertices, edges and angles, and their respective images, in order to create a set of constraints that can be used to invert the 3D to 2D projective transformation performed by the camera.

Efficiency is a critical requirement in applications such as tracking and navigation, for instance, and it is well known that if *line correspondences* are used, the process of orientation recovery can be completely separated from position recovery. This allows one to reduce a six–DOF problem to two simpler three–DOF problems that can be solved in a more efficient way. Actually, after orientation recovery is completed, position recovery can be performed by solving a single linear system and this fact has been exploited in several techniques [4; 9; 8].

Unfortunately, as we are going to show later, line correspondences are intrinsically less powerful than *point correspondences*, in the sense that any alignment constraint that can be expressed as a set of line

*The support of CAPES (Proc. BEX 0591/95-2), of NSF (Infrastructure Grant CDA-94-01142), and of DARPA (VSAM Contract DAAB07-97-C-J027) are gratefully acknowledged.

correspondences can also be expressed as an equivalent set of point correspondences, but the converse is not true. On the other hand, if point correspondences are used, then the separation between orientation and position is apparently no longer possible because the description of any individual model point in the camera coordinate system depends directly on the translational components of the camera pose with respect to the scene. However, in the present work, we propose an efficient way of achieving this separation while still preserving the full power of point correspondences. We also present a formal proof that the resulting technique is strictly more powerful than any of the techniques based on line correspondences and we characterize exactly the situations in which this superiority arises.

The idea of decoupling rotation recovery from translation recovery with point correspondences was originally proposed by Joseph Yuan [18]. But in Yuan’s formulation the problem instances in which point correspondences are theoretically more powerful than line correspondences result in a singularity that is treated as a special case. As a consequence, stability problems arise in the neighborhood of these instances. Thus, the algorithm that we propose here is, to the best of our knowledge, the first method for independent orientation recovery that allows the theoretical superiority of point correspondences over line correspondences to be actually exploited. To support this claim, we suggest a specific application in which this superiority can be translated into practical benefits, and we present promising results obtained in extensive experiments performed with synthetic data.

In the next section we briefly review the literature on model-based pose recovery. In section 3 we discuss the details of a technique based on line correspondences that we use as a basis for the derivation of our own algorithm. In section 4 we introduce and analyze our contribution from a theoretical point of view. In section 5 we present an example of a practical application that may benefit from our work and we also present empirical evidence to support our claims. Finally, in section 6, we present our concluding remarks.

2 Background

A number of different solutions have been proposed to the problem of model-based pose recovery from an arbitrary set of known correspondences between scene and image features with a perspective camera model. The most traditional, straightforward, and widely used approach consists of defining a measure of discrepancy between the actual image measurements and the measurements that would be expected given a perspective camera model and an arbitrary estimate for the unknown pose. Then, by replacing the chosen error measure (which is a non-linear function of the pose parameters) with a local linear approximation around the point corresponding to the current pose estimate, one can compute a correction that in general yields a better pose estimate. This process can be iterated until (ideally) the error function is locally minimized and the current pose estimate converges to the actual pose, within a predefined desired precision.

David Lowe [11; 10] proposed a classic solution along this line. Given a certain pose estimate, his algorithm computes the expected values for a vector of measurements (positions or orientations) in the resulting image, using a non-linear projective model. Then, Newton’s iterative gradient method is employed to minimize this error vector in a least-squares sense. Lowe’s algorithm was later improved by Araújo *et al* [2] through the use of a more accurate projective model. Similar methods that use models composed by more generic features [15] and that perform orientation recovery independently from position recovery [9; 18] have also been developed. The basic problem with all these approaches is that the use of first-order optimization techniques makes them oversensitive to the initial conditions, causing convergence problems in several instances.

A more recent approach, suggested by DeMenthon and Davis [4], consists of computing an initial estimate for the pose with a weak-perspective camera model and then refining this model numerically, in order to account for the perspective effects in the image. The key idea is to isolate the non-linearity of the perspective projection equations with a set of parameters that explicitly quantify the degree of perspective distortion in

different parts of the scene. By artificially setting these parameters to zero, one can then generate an affine estimate for the pose. Then, the resulting pose parameters can be used to estimate the distortion parameters and this process can be iterated until the resulting camera model (presumably) converges to full perspective. Oberkampff *et al* [13] extend DeMenthon–Davis’s original algorithm to deal with planar objects (the original formulation is not able to handle that particular case) and Horaud *et al* [7] propose a similar approach that starts with a paraperspective rather than a weak–perspective camera model.

The main advantage of this kind of approach is its efficiency. Like the optimization–based techniques, each iteration of the algorithms based on initial affine approximations demands the resolution of a possibly overconstrained system of linear equations. However, in the latter methods, the coefficient matrix of this system depends only on the scene model and thus its pseudo–inverse can be computed off–line, while the optimization–based techniques must necessarily perform this expensive operation at every single iteration. [4].

But on the other hand, optimization–based techniques are more generic in the sense that they can be easily extended to recover some unknown intrinsic camera parameters, and to deal with non–rigid motion [11; 10]. Furthermore, in scenes with very significant perspective effects (for instance, those generated with wide–angle lenses), affine solutions may not be reasonable initial approximations for full perspective at all, causing methods such as DeMenthon–Davis’s (as originally stated) to diverge. In this case, the ability of the optimization–based approaches to exploit the temporal coherence of a sequence of images may be a crucial requirement to achieve stability in pose estimation [6].

3 Phong–Horaud–Tao’s Original Algorithm

Ideally, one would like to combine the generality of the optimization–based techniques such as Lowe’s algorithm with a decreased sensitivity to the initial conditions, but still keep the new formulation efficient enough for real–time usage. This problem was addressed by Phong, Horaud and Tao [14]. Like the other optimization–based schemes, their technique starts with an arbitrary initial estimate for the pose and then refines this estimate numerically. However, instead of computing the pose corrections from successive local linearizations of the error function, Phong–Horaud–Tao’s solution uses a second–order trust–region optimization technique, with better global convergence properties. For a detailed description of this trust–region technique and an intuitive comparison between it and Levenberg–Marquardt’s method, we refer the reader to [14].

The drawback of this approach is that the computational cost per iteration tends to increase, since the trust–region optimization requires the computation of a relatively expensive quadratic local approximation for the error function. In order to ameliorate this problem, Phong, Horaud and Tao use two powerful ideas. One of them is the previously mentioned fact that, if one uses line rather than point correspondences, it is possible to separate the recoveries of orientation and position completely, through the use of the concept of *interpretation plane* [14; 12; 8].

If the imaging process is modeled as a perspective transformation, then a model edge can be the right correspondence for a given image edge if and only if it lies on the plane defined by the image edge and the optical center, which is called the interpretation plane. This plane can be also defined as the geometrical locus of all possible model edge positions that result in a certain image edge, hence the name. The nice property of this geometrical entity is that its description in camera–centered coordinates (and thus the description of its normal vector) does not depend on the model at all.

So, the problem of recovering the orientation can be cast into the equivalent (but easier) problem of enforcing the orthogonality between the description of each model edge in the camera coordinate system and the unit vector normal to the interpretation plane of the corresponding image edge. Similarly, position recovery can be carried out by selecting an arbitrary point on each edge and requiring its description in

camera coordinates to be aligned with the proper interpretation plane. If the pose is represented by a 3×3 rotation matrix \mathbf{R} and a three–element translation vector \vec{t} , then these constraints, for an arbitrary correspondence i , can be stated as:

$$\vec{n}_i \cdot (\mathbf{R} \vec{e}_i) = 0, \quad (1)$$

$$\vec{n}_i \cdot (\mathbf{R} \vec{p}_i + \vec{t}) = 0, \quad (2)$$

where \vec{e}_i and \vec{p}_i are the object–frame descriptions of an arbitrary model edge i and of an arbitrary point belonging to that edge, respectively, and \vec{n}_i is the camera–frame description of the normal to the corresponding interpretation plane.

The problem with this formulation is that the rotation matrix \mathbf{R} has nine elements but only three DOF. Obviously, performing an optimization on a nine–dimensional space to solve a three DOF problem is not an efficient alternative. At this point the other powerful idea of Phong–Horaud–Tao’s algorithm comes into play. It is a well–known fact that any rigid transformation can be represented as a *screw*, that is, “the composition of a rotation about a unique axis not passing through the origin and a translation along the same axis” [14]. Walker *et al* [17] proposed a convenient parametrization for this representation, using the concept of *dual number quaternions*. A quaternion is an extension proposed by Hamilton to the concept of complex number, that has one real and three imaginary components [1]. A dual quaternion $\hat{\mathbf{q}}$, on its turn, is an entity composed by two quaternions \mathbf{r} and \mathbf{s} , such that: $\hat{\mathbf{q}} = \mathbf{r} + \epsilon \mathbf{s}$, where $\epsilon^2 = 0$.

Phong, Horaud and Tao specify the screw axis using a unit vector \vec{r} to represent its orientation, and a vector \vec{l} , such that $\vec{l} \cdot \vec{r} = 0$, to represent the location of the point belonging to it which is closest to the origin. If the angle of the rotation about the screw axis is denoted by ϕ and the magnitude of the translation is denoted by t , then this representation can be readily translated into dual quaternion form, through the following equations:

$$\mathbf{r} = \left[\cos \frac{\phi}{2}, \sin \frac{\phi}{2} \vec{r} \right], \quad \mathbf{s} = \left[-\frac{t}{2} \sin \frac{\phi}{2}, -\frac{t}{2} \cos \frac{\phi}{2} \vec{r} + \sin \frac{\phi}{2} (\vec{l} \times \vec{r}) \right], \quad (3)$$

where, by definition, \mathbf{r} has unit norm and is orthogonal to \mathbf{s} .

One of the advantages of this representation over the mere use of the screw parameters \vec{r} , \vec{l} , ϕ and t is that the conversions between it and the original pose representation (\mathbf{R} and \vec{t}) can be performed in a simpler way. Given an arbitrary dual quaternion $\hat{\mathbf{q}} = \mathbf{r} + \epsilon \mathbf{s}$, \mathbf{R} and \vec{t} are determined by the following relations:

$$\begin{bmatrix} 1 & \vec{0} \\ \vec{0}^T & \mathbf{R} \end{bmatrix} = \mathbf{W}(\mathbf{r})^T \mathbf{Q}(\mathbf{r}), \quad \begin{bmatrix} 0 \\ \vec{t} \end{bmatrix} = 2 \mathbf{W}(\mathbf{r})^T \mathbf{s}, \quad (4)$$

where, for an arbitrary quaternion $\mathbf{q} = [q_0, q_x, q_y, q_z]^T$, the matrices \mathbf{Q} and \mathbf{W} are defined as:

$$\mathbf{Q}(\mathbf{q}) = \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & -q_z & q_y \\ q_y & q_z & q_0 & -q_x \\ q_z & -q_y & q_x & q_0 \end{bmatrix}, \quad \mathbf{W}(\mathbf{q}) = \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & q_z & -q_y \\ q_y & -q_z & q_0 & q_x \\ q_z & q_y & -q_x & q_0 \end{bmatrix}.$$

If we treat arbitrary three–vectors \vec{v} as quaternions whose first component is null ($\mathbf{v} = [0 \ \vec{v}]$), then using the equations above, the conditions for alignment of the model edges with their corresponding interpretation planes (originally expressed in Eqs. (1) and (2)) can be restated in quaternion form:

$$\vec{n} \cdot (\mathbf{R} \vec{e}) = \mathbf{n}^T \left(\mathbf{W}(\mathbf{r})^T \mathbf{Q}(\mathbf{r}) \mathbf{e} \right) = 0, \quad (5)$$

$$\vec{n} \cdot (\mathbf{R} \vec{p} + \vec{t}) = \mathbf{n}^T \left(\mathbf{W}(\mathbf{r})^T \mathbf{Q}(\mathbf{r}) \mathbf{p} + 2 \mathbf{W}(\mathbf{r})^T \mathbf{s} \right) = 0. \quad (6)$$

Notice that in the equations above, the pose quaternions \mathbf{r} and \mathbf{s} are “trapped” inside the matrices \mathbf{Q} and \mathbf{W} . Since the numerical algorithm is going to work directly with these quaternions, it is a good idea to “isolate” them, so that the computation of the derivatives of the rotation and translation error functions is simplified. Fortunately, it is easy to check from the definitions of \mathbf{Q} and \mathbf{W} that for two arbitrary quaternions \mathbf{a} and \mathbf{b} : $\mathbf{Q}(\mathbf{a})\mathbf{b} = \mathbf{W}(\mathbf{b})\mathbf{a}$. This “commutative” property, in addition to the associativity of matrix multiplication, allows one to separate the parameters \mathbf{r} and \mathbf{s} in the Eqs. (5) and (6).

Finally, it is important to notice that now there are eight parameters in the pose encoding, but of course the problem itself has only six DOF, as usual. Phong, Horaud and Tao suggest that this can be fixed by introducing extra terms in the error functions, so as to penalize solutions in which the quaternion \mathbf{r} either does not have unit norm ($\mathbf{r}^T\mathbf{r} \neq 1$) or is not orthogonal to the quaternion \mathbf{s} ($\mathbf{r}^T\mathbf{s} \neq 0$). Taking this into account, the error (discrepancy) functions to be minimized for decoupled orientation and position recovery, respectively, from line correspondences, are given by:

$$d_r^{(l)}(\mathbf{r}) = \sum_{i=1}^n \left(\mathbf{r}^T \mathbf{A}_i \mathbf{r} \right)^2 + \rho \left(\mathbf{r}^T \mathbf{r} - 1 \right)^2, \quad (7)$$

$$d_t^{(l)}(\mathbf{s}) = \sum_{i=1}^n \left(\mathbf{r}^T \mathbf{B}_i \mathbf{r} + \mathbf{r}^T \mathbf{C}_i \mathbf{s} \right)^2 + \rho \left(\mathbf{r}^T \mathbf{s} \right)^2, \quad (8)$$

$$\text{where: } \mathbf{A}_i = \mathbf{Q}(\mathbf{n}_i)^T \mathbf{W}(\mathbf{e}_i), \quad \mathbf{B}_i = \mathbf{Q}(\mathbf{n}_i)^T \mathbf{W}(\mathbf{p}_i), \quad \mathbf{C}_i = 2 \mathbf{Q}(\mathbf{n}_i)^T.$$

The parameter ρ in the equations above is used to regulate the relative strength between the alignment constraints derived from the known geometry of the target and the consistency constraints, which eliminate the two redundant DOF in the dual quaternion representation. According to Phong, Horaud and Tao, ρ “must be taken very large in order to guarantee that the penalization constraints are satisfied.”

As we already discussed, the motivation for separating the recovery of orientation from the recovery of position, as done in Eqs. (7) and (8), is to reduce the overall computational cost. But on the other hand, the fact that \mathbf{r} is treated as a constant in Eq. (8) implies that the values recovered for the orientation parameters are only optimized to guarantee that each model edge is parallel to the interpretation plane of the corresponding image edge, but not to guarantee that the model edges are actually *included* in the appropriate planes. This means that the solutions recovered with this scheme where orientation and position are decoupled tend to be (slightly) less accurate than those obtained when all the constraints available are used simultaneously. So, if one wants maximum accuracy at the expense of significantly increasing the computational cost of the pose recovery process, a simple solution is to work with a coupled error function $d_c^{(l)}(\mathbf{r}, \mathbf{s}) = d_r^{(l)}(\mathbf{r}) + d_t^{(l)}(\mathbf{s})$.

Unfortunately, in some cases, even this more expensive alternative is still relatively inaccurate. The problem is that, depending on the geometry of the object and on the actual pose, constraints based on line correspondences alone will not provide enough information to define a unique solution, regardless of how many correspondences are used. More specifically, consider the case where the object is planar and the plane that contains it also includes the optical center of the camera. In this case, the interpretation plane for any edge in the object is the same: the object plane itself. So, any pose that aligns the object with this unique interpretation plane (but not necessarily aligns all the object points in their proper positions) will satisfy all alignment constraints obtained from the orientations of the image edges (notice that by definition *line* correspondences do not include any *length* information). As a result of this, any pose recovery algorithm based uniquely on line correspondences will be unable to recover the orientation component about the axis normal to the object plane.

On the other hand, in this same situation, any instance of the *Perspective-3-Point* (P3P) problem in which the three model points are all distinct and have distinct images has at most four feasible solutions [5]. And, of course, the proper alignment of an arbitrary model edge with non-null length is always entailed

by the alignment of any two distinct points belonging to it. So, we can conclude that any set of alignment constraints based on line correspondences can be expressed as an equivalent set of alignment constraints obtained from point correspondences, but the converse is not true. In other words, point correspondences are strictly more powerful than edge correspondences, as we claimed in the introductory section.

Phong, Horaud and Tao proposed a variation of their original algorithm that exploits the geometrical constraints generated by point correspondences. For details, we again refer the reader to [14]. The most important aspect of this solution, from the point of view of our discussion here, is that it does not allow orientation recovery to be separated from position recovery. In experiments performed with synthetic data, Phong, Horaud and Tao found that it is up to an order of magnitude slower than the the solution based on line correspondences with orientation and position decoupled, if the same number of correspondences is used.

4 The Decoupled Solution for Point Correspondences

In principle the coupling between orientation and position recovery seems to be inherent to the use of point correspondences, because alignment constraints for individual points do not make sense unless translation is taken into account. However, it is possible to recover the rotational components of the pose of a rigid object in a way that is completely independent of the associated translational components, but still exploits the additional constraints provided by point correspondences in order to avoid the singularities that arise when only line correspondences are used. Our key idea to devise an algorithm that satisfies these requirements is to represent each model point implicitly, as the intersection of two (virtual) model edges. Then, the interpretation planes of these edges can be used to define a basic set of alignment constraints, exactly as before. However, the fact that the resulting pairs of edges have intersection points in the 3D space can now be used to derive an additional set of constraints, which add restrictive power to our formulation.

In order to simplify our analysis, let's initially focus on the P3P problem. The resulting solution can then be easily extended to overconstrained cases. We assume that the correct correspondences for three model points in general position (P_0 , P_1 and P_2) are known. In this case, each of these points is constrained to lie in a line-of-sight defined by the optical center and by the corresponding image point. Let \vec{l}_0 , \vec{l}_1 and \vec{l}_2 be the unit vectors along these lines-of-sight, as shown in Fig. 1. Then, the description of each model point in the camera frame, $P_i^{(c)}$, $0 \leq i \leq 2$, must be a vector parallel to \vec{l}_i . We express this constraint in parametric form as: $\vec{p}_i = \lambda_i \vec{l}_i$. Notice that the unit vectors \vec{l}_i can be obtained with a simple normalization of the coordinates of the corresponding image points in the 3D camera frame.

Each pair of points $[P_i, P_j]$, $i \neq j$, defines a virtual edge that is used to derive alignment constraints. As we already saw, one such constraint, for any (virtual) edge, is the inclusion in the interpretation plane of its (virtual) image. However, up to this point we still have not used the fact that the *length* of any model edge is known *a priori*. The reason for this is very simple: unless we know the translation parameters, the ratio between the length of a model edge and the length of its image provides almost no information about its true orientation, because different combinations of values for the translation and the angle about the normal to the interpretation plane can generate the same size ratio. For instance, if an edge were closer than it actually is, but were viewed more obliquely, its apparent size could still be the same.

However, if we consider two edges that have a common intersection point, then the relationship between the ratios of their actual and apparent sizes gives us some valuable information about the orientation of the target, even if the translational pose parameters are completely unknown. Let's consider the edges $\vec{e}_{01}^{(c)} = P_0^{(c)} P_1^{(c)}$ and $\vec{e}_{02}^{(c)} = P_0^{(c)} P_2^{(c)}$, in Fig. 1, for instance. Since the extreme point $P_0^{(c)}$, which is common to both of them, is located at a distance λ_0 from the origin, one might be tempted to say that the size ratios for both edges must be equal if the object is properly aligned. However, this is not the case, because the distance λ_1 is not necessarily equal to λ_2 . Indeed, it is possible to derive a more subtle alignment condition

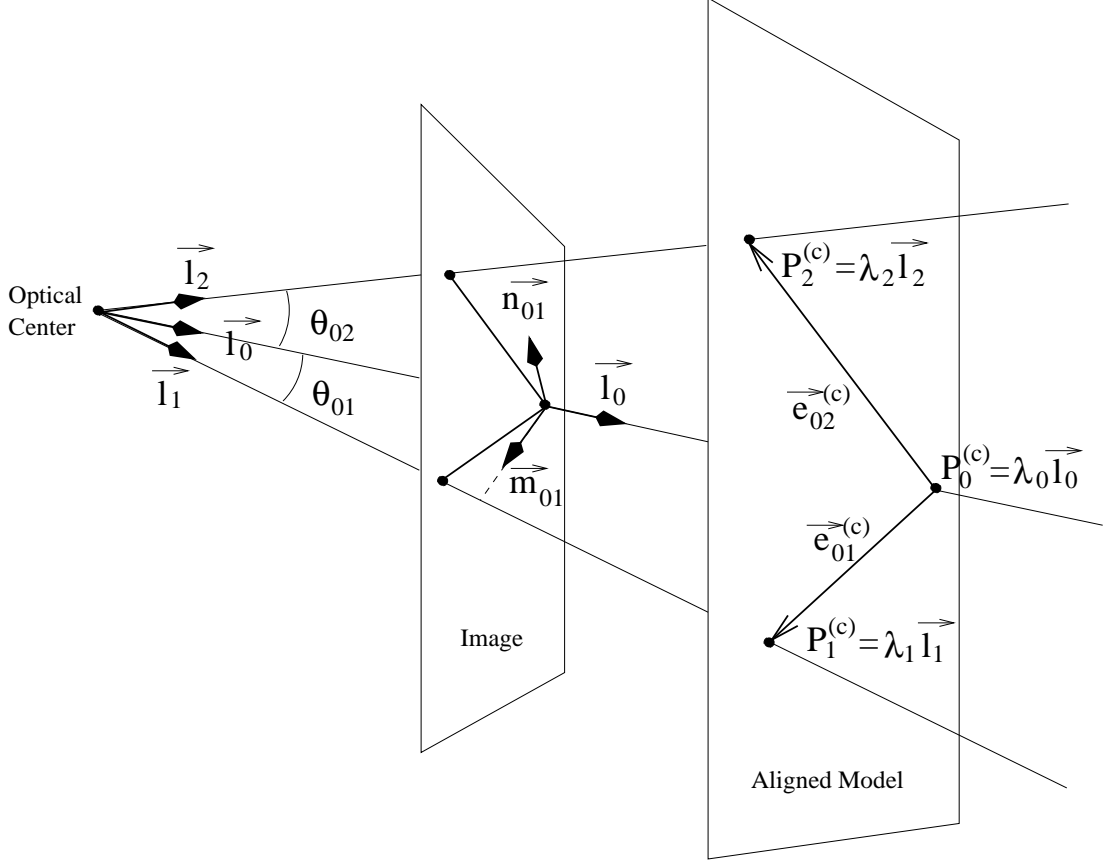


Figure 1: Geometry of our solution of the problem of independent orientation recovery from a set of point correspondences.

using the fact that each size ratio “induces” a value for the distance λ_0 . So, we can use the difference between the values induced by edges $\vec{e}_{01}^{(c)}$ and $\vec{e}_{02}^{(c)}$ as a measure of error in the alignment between the 3D model and the image.

In order to come out with a mathematical formalization of this idea, we start with a set of simple alignment conditions: we require the difference between the parametric descriptions (in terms of the line-of-sight unit vectors) for the extreme points of each edge to be equal to the description of the model edge itself, mapped to the camera coordinate system. Notice that the description of an edge is a vector invariant with respect to translations of the object. So, mapping the edge description from the model to the camera reference system involves only a premultiplication by the rotation matrix \mathbf{R} and we can write the alignment conditions for the pair \vec{e}_{01} and \vec{e}_{02} as:

$$\vec{e}_{01}^{(c)} = \mathbf{R} \vec{e}_{01} = \lambda_1 \vec{l}_1 - \lambda_0 \vec{l}_0, \quad (9)$$

$$\vec{e}_{02}^{(c)} = \mathbf{R} \vec{e}_{02} = \lambda_2 \vec{l}_2 - \lambda_0 \vec{l}_0. \quad (10)$$

Let’s focus on Eq. (9). We want to solve this for λ_0 , given an estimate of \mathbf{R} and thus we need to eliminate the extra parameter λ_1 . In order to do this, we define a new intermediate coordinate system “between” the object and the camera frames and express the three-dimensional alignment restriction at hand in terms of the axes of this new system, which we call the *Interpretation Reference Frame* (IRF). One of the axes of the IRF for edge \vec{e}_{01} is defined to be the unit vector \vec{l}_0 , another axis (\vec{n}_{01}) is defined to be normal to the interpretation plane of the corresponding edge in the image, and the remaining axis (\vec{m}_{01}) is defined as the

cross product of the previous two, as shown in Fig. 1. We express Eq. (9) in terms of these axes as follows:

$$\begin{bmatrix} \vec{l}_0, \vec{m}_{01}, \vec{n}_{01} \end{bmatrix}^T (\mathbf{R} \vec{e}_{01} + \lambda_0 \vec{l}_0 - \lambda_1 \vec{l}_1) = 0.$$

Then, we can use the fact that the three axes of the IRF form an orthonormal basis in order to simplify the expression above to the following form:

$$\begin{cases} \vec{l}_0 \cdot (\mathbf{R} \vec{e}_{01}) &= -\lambda_0 + \lambda_1 (\vec{l}_0 \cdot \vec{l}_1), \\ \vec{m}_{01} \cdot (\mathbf{R} \vec{e}_{01}) &= \lambda_1 (\vec{m}_{01} \cdot \vec{l}_1), \\ \vec{n}_{01} \cdot (\mathbf{R} \vec{e}_{01}) &= 0. \end{cases}$$

Notice that the last equation in the system above is equivalent to Eq. (1). So, the alignment constraints for our decoupled solution actually subsume those of Phong–Horaud–Tao’s algorithm for line correspondences. The other two equations can be solved for λ_0 in the following way:

$$\begin{aligned} \lambda_1 &= \vec{a}_{01} \cdot (\mathbf{R} \vec{e}_{01}), \quad \text{where: } \vec{a}_{01} = \frac{1}{\vec{m}_{01} \cdot \vec{l}_1} \vec{m}_{01}, \\ \lambda_0 &= \vec{b}_{01} \cdot (\mathbf{R} \vec{e}_{01}), \quad \text{where: } \vec{b}_{01} = (\vec{l}_0 \cdot \vec{l}_1) \vec{a}_{01} - \vec{l}_1 = \cotan \theta_{01} \vec{m}_{01} - \vec{l}_1, \end{aligned} \quad (11)$$

and θ_{01} is the angle between \vec{l}_0 and \vec{l}_1 (as shown in Fig. 1), in radians.

Eq. (10) can be solved for λ_0 in an analogous way, yielding:

$$\lambda_0 = \vec{b}_{02} \cdot (\mathbf{R} \vec{e}_{02}), \quad \vec{b}_{02} = \cotan \theta_{02} \vec{m}_{02} - \vec{l}_2. \quad (12)$$

Now we finally use the assumption that edges \vec{e}_{01} and \vec{e}_{02} intersect each other at point P_0 . The alignment expressed by \mathbf{R} is correct if and only if the values for λ_0 generated by Eqs. (11) and (12) are the same. So, equating those two expressions, we obtain:

$$\vec{b}_{01} \cdot (\mathbf{R} \vec{e}_{01}) - \vec{b}_{02} \cdot (\mathbf{R} \vec{e}_{02}) = 0. \quad (13)$$

Using the dual quaternion representation for the pose parameters, this can be rewritten as:

$$\mathbf{b}_{01}^T (\mathbf{W}(\mathbf{r})^T \mathbf{Q}(\mathbf{r}) \mathbf{e}_{01}) - \mathbf{b}_{02}^T (\mathbf{W}(\mathbf{r})^T \mathbf{Q}(\mathbf{r}) \mathbf{e}_{02}) = 0.$$

Then, using the “commutative” property $\mathbf{Q}(\mathbf{a}) \mathbf{b} = \mathbf{W}(\mathbf{b}) \mathbf{a}$, as well as the associativity and distributivity of matrix multiplication, the pose parameters can be “isolated” in the equation above, yielding:

$$\mathbf{r}^T (\mathbf{Q}(\mathbf{b}_{01})^T \mathbf{W}(\mathbf{e}_{01}) - \mathbf{Q}(\mathbf{b}_{02})^T \mathbf{W}(\mathbf{e}_{02})) \mathbf{r} = 0. \quad (14)$$

This equation and Eq. (1) are the constraints that we use to guarantee the proper alignment of the three distinct model points with the lines-of-sight of their respective distinct image points.

Now, let’s turn our attention to the case of (possibly) overconstrained problems. Notice that, according to the analysis performed so far, each (virtual) edge alone yields one alignment constraint and each edge intersection yields another alignment constraint. So, if the problem at hand consists of recovering the pose based on a set of n point correspondences, one possibility is to organize these points in a ring. For each point i such that $1 \leq i \leq n$, we use the virtual edges to the predecessor of i , $p(i)$, and to successor of i , $s(i)$, in the ring, in order to derive the alignment constraints expressed in Eqs. (1) and (14). Then, the orientation recovery problem can be solved through the minimization of the following function:

$$d_r^{(r)}(\mathbf{r}) = \sum_{i=1}^n (\mathbf{r}^T \mathbf{A}_{(i)} \mathbf{r})^2 + \gamma \sum_{i=1}^n (\mathbf{r}^T \mathbf{B}_{(i)} \mathbf{r})^2 + \rho (\mathbf{r}^T \mathbf{r} - 1)^2, \quad (15)$$

$$\text{where: } \mathbf{A}_{(i)} = \mathbf{Q}(\mathbf{n}_{(i)(p(i))})^T \mathbf{W}(\mathbf{e}_{(i)(p(i))}),$$

$$\mathbf{B}_{(i)} = \mathbf{Q}(\mathbf{b}_{(i)(p(i))})^T \mathbf{W}(\mathbf{e}_{(i)(p(i))}) - \mathbf{Q}(\mathbf{b}_{(i)(s(i))})^T \mathbf{W}(\mathbf{e}_{(i)(s(i))}),$$

where γ and ρ are empirically-defined weighting factors that control the relative strength of the different types of constraints, all the quaternions with indices $(i)(p(i))$ are obtained from the virtual edge connecting the i -th point to its predecessor in the ring, and all the quaternions with indices $(i)(s(i))$ are obtained from the virtual edge connecting the i -th point to its successor in the ring.

Notice, however, that the approach described above increases the amount of work per iteration with respect to the original decoupled algorithm based on line correspondences. Another alternative, which does not increase the computational cost per iteration, is to group the object points with known correspondences in triangles and then apply our P3P solution individually on each triangle. Assume, without loss of generality, that the triangle i consists of points $3i-2$, $3i-1$ and $3i$. Then, the function that must be minimized to obtain the unknown orientation is:

$$d_r^{(t)}(\mathbf{r}) = \sum_{i=1}^{n/3} (\mathbf{r}^T \mathbf{A}_{(3i-1)} \mathbf{r})^2 + \gamma \sum_{i=1}^{n/3} (\mathbf{r}^T \mathbf{B}_{(3i-1)} \mathbf{r})^2 + \sum_{i=1}^{n/3} (\mathbf{r}^T \mathbf{C}_{(3i-1)} \mathbf{r})^2 + \rho (\mathbf{r}^T \mathbf{r} - 1)^2, \quad (16)$$

$$\begin{aligned} \text{where: } \mathbf{A}_{(i)} &= \mathbf{Q}(\mathbf{n}_{(i)(i-1)})^T \mathbf{W}(\mathbf{e}_{(i)(i-1)}), \\ \mathbf{B}_{(i)} &= \mathbf{Q}(\mathbf{b}_{(i)(i-1)})^T \mathbf{W}(\mathbf{e}_{(i)(i-1)}) - \mathbf{Q}(\mathbf{b}_{(i)(i+1)})^T \mathbf{W}(\mathbf{e}_{(i)(i+1)}), \\ \mathbf{C}_{(i)} &= \mathbf{Q}(\mathbf{n}_{(i)(i+1)})^T \mathbf{W}(\mathbf{e}_{(i)(i+1)}). \end{aligned}$$

This new formulation explicitly guarantees only the alignment of *two* edges of each triangle with their respective interpretation planes and the agreement on the induced value for the distance of their intersection. However, these conditions necessarily entail the alignment of the third edge with respect to its interpretation plane, and thus, the complete alignment of the triangle:

Theorem 1 *Let \vec{l}_0 , \vec{l}_1 and \vec{l}_2 be unit vectors describing the orientations of the optical rays through any three distinct points in the image plane and let $\vec{e}_{01}^{(c)}$, $\vec{e}_{02}^{(c)}$ and $\vec{e}_{12}^{(c)}$ be vectors describing the edges of an arbitrary triangle in the camera reference frame (Fig. 1). If the linear system composed by the constraints below (where $\lambda_0^{(01)}$, $\lambda_1^{(01)}$, $\lambda_0^{(02)}$ and $\lambda_2^{(02)}$ are the unknowns)*

$$\vec{e}_{01}^{(c)} = \lambda_1^{(01)} \vec{l}_1 - \lambda_0^{(01)} \vec{l}_0, \quad (17)$$

$$\vec{e}_{02}^{(c)} = \lambda_2^{(02)} \vec{l}_2 - \lambda_0^{(02)} \vec{l}_0, \quad (18)$$

$$\lambda_0^{(01)} = \lambda_0^{(02)} \quad (19)$$

admits at least one solution, then the linear constraint below (where $\lambda_1^{(12)}$ and $\lambda_2^{(12)}$ are the unknowns)

$$\vec{e}_{12}^{(c)} = \lambda_2^{(12)} \vec{l}_2 - \lambda_1^{(12)} \vec{l}_1. \quad (20)$$

also admits at least one solution.

Proof: If $\vec{e}_{01}^{(c)}$, $\vec{e}_{02}^{(c)}$ and $\vec{e}_{12}^{(c)}$ describe the edges of a triangle (under the convention that $\vec{e}_{ij}^{(c)}$ starts at vertex $P_i^{(c)}$ and ends at vertex $P_j^{(c)}$), then they must satisfy

$$\vec{e}_{12}^{(c)} = \vec{e}_{02}^{(c)} - \vec{e}_{01}^{(c)}. \quad (21)$$

Substituting Eqs. (17) and (18) into Eq. (21), yields

$$\vec{e}_{12}^{(c)} = \lambda_2^{(02)} \vec{l}_2 - \lambda_1^{(01)} \vec{l}_1 + (\lambda_0^{(01)} - \lambda_0^{(02)}) \vec{l}_0. \quad (22)$$

Finally, substituting Eq. (19) into Eq. (22), yields $\vec{e}_{12}^{(c)} = \lambda_2^{(02)} \vec{l}_2 - \lambda_1^{(01)} \vec{l}_1$, which shows that, given a solution $\{\lambda_0^{(01)}, \lambda_1^{(01)}, \lambda_0^{(02)}, \lambda_2^{(02)}\}$ to Eqs. (17) to (19), the assignment $\lambda_0^{(12)} = \lambda_0^{(01)}$ and $\lambda_2^{(12)} = \lambda_2^{(02)}$ is a solution to Eq. (20). \square

Theorem 1 implies that the alignment constraints defined in Eq. (16) entail those of Phong–Horaud–Tao’s decoupled algorithm, when both are applied the same set of triangles. To see this, notice that Eq. (17), for instance, is satisfied if and only if the 3D edge $\vec{e}_{01}^{(c)}$ is parallel to the Interpretation Plane of the image edge defined by the optical rays \vec{l}_0 and \vec{l}_1 . Analogous observations are valid for Eqs. (18) and (20). Eq. (19) expresses the agreement on the induced distance for intersection of edges $\vec{e}_{01}^{(c)}$ and $\vec{e}_{02}^{(c)}$. Thus, our approach is at least as powerful as Phong–Horaud–Tao’s decoupled algorithm.

Furthermore, it is possible to show that the alignment of the *three* edges of each triangle with their respective interpretation planes does *not* entail the agreement on the induced values for the distances between the edge intersections and the center of projection. In other words, our approach is actually strictly more powerful than Phong–Horaud–Tao’s decoupled algorithm, with the same cost per iteration:

Theorem 2 *Let \vec{l}_0 , \vec{l}_1 and \vec{l}_2 be unit vectors describing the orientations of the optical rays through any three distinct and collinear points in the image plane. For every possible choice of these vectors, there is an infinite number of 3D triangles (with edges $\vec{e}_{01}^{(c)}$, $\vec{e}_{02}^{(c)}$ and $\vec{e}_{12}^{(c)}$, in the camera reference system), for which the system composed by Eqs. (17), (18) and (20) admits a solution that does not satisfy Eq. (19).*

Proof: Given any \vec{l}_0 , \vec{l}_1 and \vec{l}_2 , consider the family of triangles with vertices $P_0^{(c)} = d_0 \vec{l}_0$, $P_1^{(c)} = d_1 \vec{l}_1$ and $P_2^{(c)} = d_2 \vec{l}_2 + c_0 \vec{l}_0$, where d_0 , d_1 , d_2 and c_0 can be any real number except zero. Then

$$\vec{e}_{01}^{(c)} = d_1 \vec{l}_1 - d_0 \vec{l}_0, \quad (23)$$

$$\vec{e}_{02}^{(c)} = d_2 \vec{l}_2 - (d_0 - c_0) \vec{l}_0, \quad (24)$$

$$\vec{e}_{12}^{(c)} = d_2 \vec{l}_2 - d_1 \vec{l}_1 + c_0 \vec{l}_0. \quad (25)$$

Since \vec{l}_0 , \vec{l}_1 and \vec{l}_2 are distinct and coplanar by hypothesis, there is a unique pair of real values $c_1 \neq 0$ and $c_2 \neq 0$ such that $\vec{l}_0 = c_1 \vec{l}_1 + c_2 \vec{l}_2$. Substituting this into Eq. (25) yields

$$\vec{e}_{12}^{(c)} = (d_2 + c_0 c_2) \vec{l}_2 - (d_1 - c_0 c_1) \vec{l}_1. \quad (26)$$

From Eqs. (23), (24) and (26), it can be concluded that the linear system of Eqs. (17), (18) and (20) admits the following unique solution: $\lambda_0^{(01)} = d_0$, $\lambda_1^{(01)} = d_1$, $\lambda_0^{(02)} = d_0 - c_0$, $\lambda_2^{(02)} = d_2$, $\lambda_1^{(12)} = d_1 - c_0 c_1$ and $\lambda_2^{(12)} = d_2 + c_0 c_2$. Since $c_0 \neq 0$ (by definition), $\lambda_0^{(01)} \neq \lambda_0^{(02)}$. Hence the solution to Eqs. (17), (18) and (20) shown above does not satisfy Eq. (19). \square

Notice that Theorem 2 not only shows that our technique is more powerful than Phong–Horaud–Tao’s decoupled solution, but also defines a class of problem instances in which this superiority arises. These instances (namely, scenes where all features and the camera lie on a single plane) are exactly those in which no technique based on line correspondences can guarantee complete alignment. This, along with the observation that our technique can include arbitrary line–based constraints via the use of Eq. (1), shows that it is actually strictly more powerful than any method based uniquely on line correspondences.

5 Experimental Results

This section provides some evidence that the theoretical advantages of our solution do make a difference in practice. Initially, we stress the fact that our technique does not need actual edges to work. It is based on

point correspondences and the edges \vec{e}_{ij} mentioned in its formulation are only virtual edges defined by pairs of points. Furthermore, the camera’s optical center does not need to be located exactly at the object plane and, in fact, the object does not need to be exactly planar in order for our solution to produce accuracy gains with respect to solutions based uniquely on line correspondences. If the smallest singular value on the SVD of the $n \times 4$ matrix composed by the image points (in *homogeneous coordinates*) is much smaller than the other three and the optical center is relatively close to the plane that fits the model points in a least-squares sense, then the proximity of a singularity will affect the line-based algorithms and the additional alignment constraints derived by our solution will have a significant impact on the overall convergence speed and accuracy of the numerical pose recovery.

A real-life scenario where these conditions arise is on off-road navigation of a mobile robot using landmarks located at relatively distant positions (in comparison to the maximum height variation of the visible terrain). An interesting aspect of this application is the fact that, in many situations, accurate *models* of the terrain are available, allowing the use of model-based techniques. For instance, the United States Geographical Survey offers *Digital Elevation Maps* (DEMs) with different resolutions, commercially and for free download on the Internet.

The traditional model-based approaches for recovering the position of off-road vehicles from images of relatively distant visual landmarks consist of performing exhaustive searches either in a space of all possible interpretations of the scene, or in the space of all possible renderings of the model (map), or in a discretized space of all possible poses of the camera [3].

In scenes that include distant features such as mountain peaks, this type of approach seems to be suitable for position recovery, in spite of its elevated computational cost. However, even in these situations, the relative *orientation* of the vehicle with respect to the features of interest can presumably change in a relatively fast way in an obstacle avoidance operation, or simply as a result of changes in the inclination of the terrain. So, the techniques mentioned above are certainly not suited for full 6D pose recovery in real time. We conjecture that they are useful in order to establish periodically the right correspondences between pairs of map (model) and image features, but a tracking technique based on some perspective inversion algorithm is needed to perform real-time orientation recovery. Furthermore, some of those traditional exhaustive techniques rely on mosaicing to obtain a reasonable number of feature correspondences, and it seems to us that a more practical alternative is to use wide-angle lenses. But of course, this requires the ability to deal with very significant perspective distortion, which the traditional techniques (as well as affine pose recovery methods) do not possess.

In order to test the impact of our contribution in this type of scenario, we performed extensive experiments with synthetic data. The process of scene generation involved initially the choice of the number of visible landmarks. Assuming independence in the probability of occurrence of individual landmarks in the visual field, this number was drawn from a Poisson distribution with mean three. All the underconstrained scenes (two visible features or less) were immediately discarded. In the remaining scenes, the positions of individual features along the ground plane were uniformly sampled on a 1×1 km square initially centered at the camera’s optical center. The heights of the features with respect to the ground plane were chosen from a normal distribution with mean zero and standard deviation explicitly controlled by a parameter h_s . The same number of scenes was generated with eight possible values for h_s , ranging exponentially from 1 to 128 m.

Each of the resulting scenes was then rotated about the axis normal to the ground plane by an arbitrary angle — sampled uniformly in the interval $[0, 2\pi)$ radians — and translated along the optical axis of the camera by a distance drawn from an uniform distribution on the interval $[0.9, 1.1)$ km, so that all the landmarks were placed in a visual field about 103.5 degrees wide on the horizontal image axis (in the worst case). Next, the resulting scenes were translated along the axis normal to the ground plane, to account for the height of the camera (using the same distribution employed in the generation of the heights of the landmarks). Finally, each scene was rotated about an individual axis uniformly sampled on a unit sphere,

by an angle drawn from a normal distribution with mean zero and standard deviation equal to $\frac{\pi}{10}$ radians. This final step was performed in order to account for (relatively small) inclinations of the terrain and errors in the “foveation” of the horizon line.

The imaging simulation was performed with a “noisy” pinhole camera model. On each image, a random multiplicative bias was used to model inaccuracies in the calibration of the focal length and the aspect ratio, and a random additive bias was used to model inaccuracies in the calibration of the intersection of the optical axis with the image plane. Furthermore, Gaussian noise was added to the coordinates of all image features, in order to model measurement errors in the low-level stages of vision (feature detection and 2D localization). More specifically, if the description of an arbitrary point i in 3D camera coordinates is given by $\vec{p}_i = [x_i, y_i, z_i]^T$, we compute the coordinates of the corresponding image point as:

$$[u_i, v_i] = \left[(1 + \varepsilon_m) \frac{f x_i}{z_i} + \varepsilon_a + \xi_i, (1 + \varepsilon_m) \frac{f y_i}{z_i} + \varepsilon_a + \zeta_i \right], \quad (27)$$

where f is the camera focal length; ε_m and ε_a are the multiplicative and additive biases, respectively, both unique for the whole image and drawn from normal distributions with zero mean and controlled standard deviations b_m and b_a (respectively); and ξ_i and ζ_i are Gaussian additive noise (individually generated for each feature) with zero mean and controlled standard deviation n_a . Unless stated otherwise, the experiments were performed with three different values for b_m (0.01, 0.02 and 0.04), for b_a (0.01, 0.02 and 0.04 focal lengths), and for n_a (0.001, 0.002 and 0.004 focal lengths).

For each possible combination of values of the parameters h_s , b_m , b_a and n_a such that $b_a = f b_m$, several scenes were generated in a completely independent way. For each one of them, an initial solution was computed by multiplying the angles and distances used in the generation of the original scene by a random disturbance factor with mean 1 and standard deviation 0.1 (the only exception was the axis of the final rotation, that was *added* to a disturbance vector sampled uniformly on a unit sphere, and renormalized), and then repeating the same steps taken originally. Each solution generated in this way was used to initialize five alternative pose estimation techniques: Phong–Horaud–Tao’s decoupled and coupled methods for line correspondences, Phong–Horaud–Tao’s coupled method for point correspondences, and our decoupled methods for point correspondences organized in triangles and in a ring.

Initially, we tested the accuracy and efficiency of these techniques in the general case. They were implemented in Matlab (release 4.2) and then translated to Ansi C by the Matlab compiler (with optimization options -ri). The resulting code was then compiled with gcc (optimization level -O2) and executed in a Sun Sparc Station 4, running Sun OS. For each of the 72 different combinations of the controlled parameters, we generated and tested 500 different scenes, resulting in a total of 36,000 independent executions of each method in study.

On each of these executions, the maximum number of iterations of the trust–region algorithm was fixed at 20, and per-iteration traces of four different error measures were computed. Two of these measures use the values recovered for the camera orientation and position to estimate the individual 3D positions of all the model points and compare these estimates to the respective actual 3D positions. In the other two, this process is repeated with the *true* value for the camera position, instead of the *recovered* value. The goal of these alternative measures is to evaluate the accuracy of *orientation* recovery alone. In practice this is important because accurate position recovery can be obtained directly from a GPS system. For each of these two possibilities, one of the error measures computed was the square root of the average squared positional error for all individual features, and the other was the maximum individual positional error. Here we show only the results based on the measure that uses *recovered* position and *squared* individual errors (which we call *Actual Average Squared Distance* — AASD), because the results obtained with the other three measures were qualitatively identical.

As a measure of efficiency, we used the *elapsed* times of orientation recovery, since in this particular application, the camera position only needs to be estimated at a much coarser temporal scale. Then, each

measure considered (that is, errors on each iteration and the total execution time) had its average and standard deviation computed for the whole set of 36,000 executions.

The evolution of the global error as a function of the iteration number is displayed in Fig. 2. In this and all the other error plots presented here, the solid and dotted lines at the top represent the results obtained with Phong–Horaud–Tao’s decoupled and coupled solutions for line correspondences, respectively, the dashed and dash-dotted lines correspond to our decoupled solutions for point correspondences organized in triangles and in a ring, respectively, and the solid line at the bottom represents the results obtained with Phong–Horaud–Tao’s coupled solution for point correspondences.

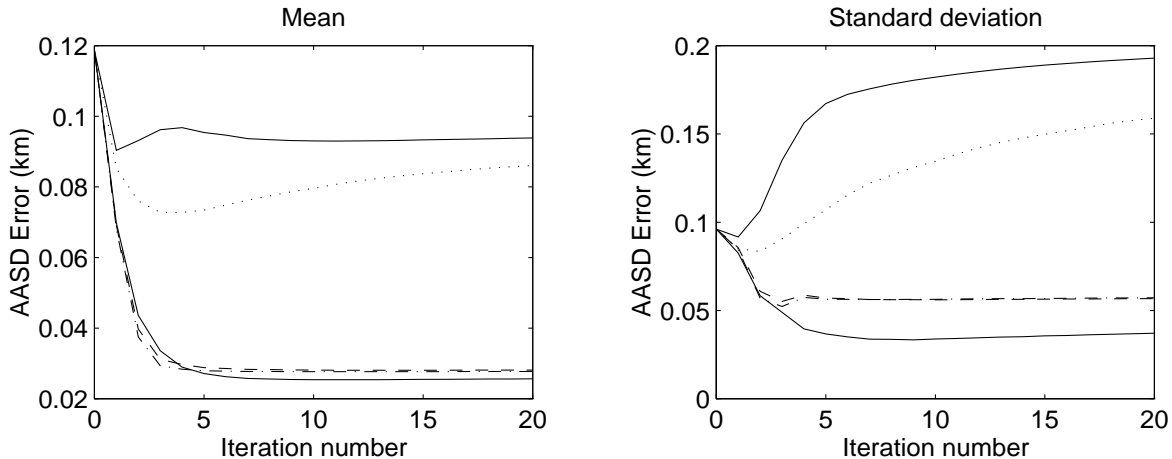


Figure 2: Convergence of a 3D-space error measure (AASD error) with respect to the number of iterations of the trust-region optimization. Top solid, dotted, dashed, dash-dotted and bottom solid lines represent the results for decoupled recovery from lines, coupled recovery from lines, decoupled recovery from triangles, decoupled recovery from a ring of points and coupled recovery from points, respectively.

It can be observed that the three solutions that make use of point correspondences have very similar convergence properties. Phong–Horaud–Tao’s solution has a smaller final standard deviation, indicating that our algorithms may have certain convergence problems in very particular problem instances. However, the convergence of the *average* AASD error with our algorithms is slightly faster. For instance, if the maximum number of iterations had to be limited to two, in order to cope with real-time constraints, our solutions would actually yield more accurate answers on average, with minimal differences in terms of standard deviation. Furthermore, in all three cases, the final values achieved for this error are very similar: 28.1 m for our decoupled version with triangles, 27.7 m for our decoupled version with the ring of features, and 25.7 m for Phong–Horaud–Tao’s coupled solution.

On the other hand, the two solutions based on line correspondences have clear convergence problems after a few iterations, even in terms of the average errors. The final AASD error for the decoupled and coupled solutions are 93.9 m and 86.1 m, respectively. This is a strong empirical confirmation of our theoretical claim that point correspondences are inherently more powerful than line correspondences.

The elapsed times are displayed in Fig. 3. The central aspect of this chart, which we want to stress here, is the fact that the coupled solutions are considerably slower than their decoupled counterparts. If we compare the average times plus standard deviations (since these techniques are intended to be used in real-time conditions), the coupled solution for line correspondences takes about 74% longer than the decoupled one. With point correspondences, Phong–Horaud–Tao’s original solution takes about 61% longer than the our fastest decoupled version (the one based on grouping the features in triangles).

Another important fact is that our solution with features organized in a ring takes approximately 21% longer than the one that uses the triangles, in spite of both having virtually the same accuracy. So, as

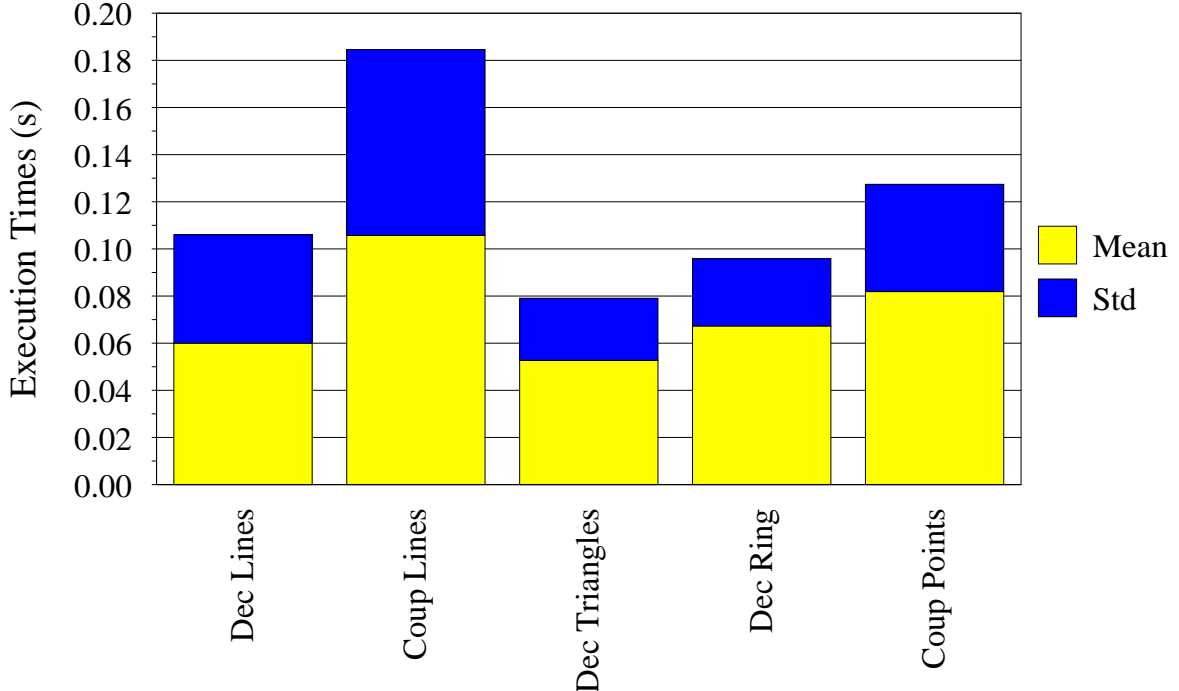


Figure 3: Execution times for orientation recovery.

Theorem 1 predicted, the triangle organization allows one to reduce the overall execution time without any major impact in terms of the quality of the final solutions. Finally, a quite unexpected result was the fact that our fastest solution was even faster than the original decoupled solution based on line correspondences. This happens because the extra complexity of our scheme is only reflected in the computation of the coefficients for the error function, which is performed just once and not for every single iteration. So, the faster convergence of our technique can easily outweigh this additional initial overhead.

We also performed some experiments to compare the sensitivity of the different algorithms with respect to changes in scene conditions such as the elevation variance in the visible terrain, the number of visible landmarks, the levels of bias introduced by inaccuracies in the camera calibration process, and the level of noise in the 2D localization of the image features. In all these experiments, the number of iterations of the trust-region algorithm was reduced to 5, because as shown in Fig. 2, only minor additional improvements are achieved after that point (if any at all). Also, rather than computing per-iteration traces of the different error measures, we recorded only the *best* solutions found throughout the optimization process. But on the other hand, on each experiment we performed independent executions and computed global statistics with a series of different settings of the scene condition whose impact on the pose recovery was being evaluated.

The sensitivity measurements with respect to the variation of h_s , the standard deviation of the normal distribution of the terrain height, are plotted in Fig. 4 (10 values between 1.25 and 640 m, with 3,600 scenes per value). It can be seen that the accuracy of the algorithms based on point correspondences is roughly invariant with respect to the value of h_s . The algorithms based uniquely on line correspondences, however, are quite sensitive to this parameter, because of the singularity when the scene is perfectly planar. In the case of scenes with features uniformly distributed in a spherical region, for instance, this difference should vanish. But the important fact is that the reduction in the accuracy gap between line-based and point-based techniques happens gradually, rather than in a sharp fashion. So, even in very generic scenarios with big height variations some modest gains can be obtained if a point-based algorithm is used.

To further stress the stability of our algorithm, we present some additional results that show how smoothly its accuracy changes when the experimental setup is modified. The variation of the AASD error with respect

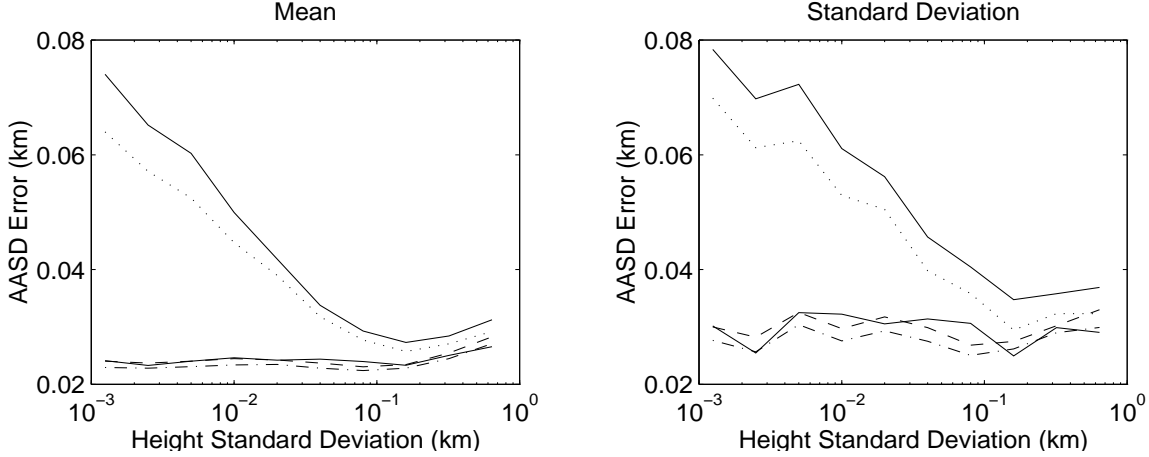


Figure 4: Error sensitivity with respect to the standard deviation of the terrain’s height distribution. Top solid, dotted, dashed, dash–dotted and bottom solid lines represent the results for decoupled recovery from lines, coupled recovery from lines, decoupled recovery from triangles, decoupled recovery from a ring of points and coupled recovery from points, respectively.

to the number of visible features in the scene is displayed in Fig. 5. There is a slight tendency of reduction in the error when the number of features is increased, which is quite natural, because the measurement errors for individual features are averaged out when many of them are used. The important fact is that for all the 10 different values of the number of landmarks tested (3,600 different scenes each), the greater accuracy of the methods based on point correspondences is unquestionable.

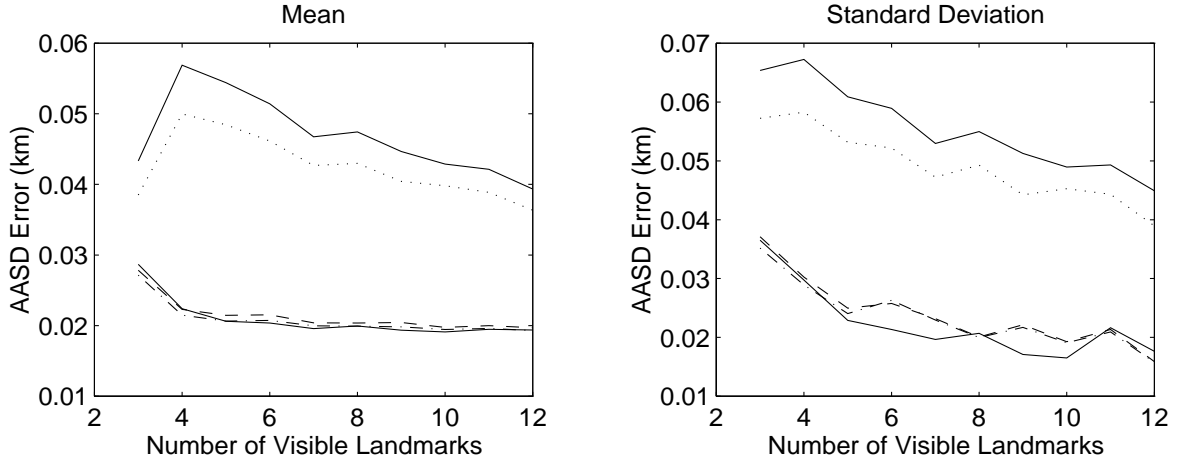


Figure 5: Error sensitivity with respect to the number of visible landmarks in the scene. Top solid, dotted, dashed, dash–dotted and bottom solid lines represent the results for decoupled recovery from lines, coupled recovery from lines, decoupled recovery from triangles, decoupled recovery from a ring of points and coupled recovery from points, respectively.

Finally, the results obtained for the variations of the biases b_m b_a and of the additive noise n_a are qualitatively very similar. So, here we show only the former, in Fig. 6 (10 values between 1.25×10^{-3} and 0.64, with 3,600 scenes per value). It can be seen that when the bias (noise) is increased to relatively high levels, the average accuracies of all different techniques tend to collapse into a single function, as the image signal is gradually overpowered. But on the other hand, when the bias (noise) is reduced beyond a certain critical level, the accuracies of the different types of techniques converge to very distinct “intrinsic”

error levels and the superiority of the versions based on point correspondences becomes clear.

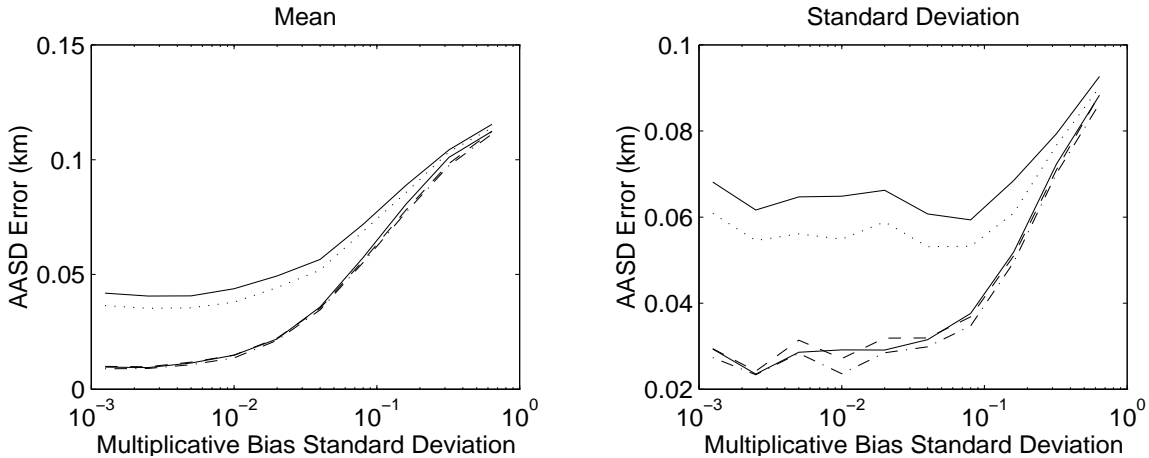


Figure 6: Error sensitivity with respect to the standard deviation of the multiplicative calibration bias. Top solid, dotted, dashed, dash-dotted and bottom solid lines represent the results for decoupled recovery from lines, coupled recovery from lines, decoupled recovery from triangles, decoupled recovery from a ring of points and coupled recovery from points, respectively.

6 Conclusion

In the past ten years or so, several different solutions have been proposed to the classical problem of model-based pose recovery. However, not much work has been devoted to comparing the relative strengths and weaknesses of these solutions in real-life applications. In particular, some important recent advances towards a definite answer to this problem [16; 7; 14; 4] rely on alignment constraints derived uniquely from line correspondences.

In the present work we show that line constraints are essentially weaker than constraints derived from point correspondences in terms of expressive power, and we characterize exactly the conditions under which this gap arises. So, from a theoretical point of view, the work presented here is important as a warning for the users of methods based on line correspondences that these techniques may face certain singularities in real-life applications.

From a more practical point of view, one of the main reasons why line correspondences are so popular is the well-known fact that they allow the complete separation between orientation and position recovery. This trick cuts the dimensionality of the problem in half and usually yields much more efficient techniques. In the current paper we show how this separation can be achieved in techniques that preserve the full power of point correspondences. In particular, we propose one such technique that performs independent orientation recovery from a set of point correspondences, by replacing pairs of model points with virtual model edges and then using the fact that these edges have intersections in 3D space to derive tighter alignment constraints than those that would be possible with generic line correspondences alone.

Yuan’s algorithm [18] is, to the best of our knowledge, the only alternative solution for independent orientation recovery from point correspondences available in the literature — Liu *et al* [9] suggest a trivial way of creating generic line correspondences from point correspondences, but this obviously does not add any strength to line-based algorithms. The key idea of Yuan’s method is to preprocess a $(n - 1) \times 3$ matrix that encodes the known structure of the n three-dimensional model points in order to obtain a minimal structural representation with only r rows, where r is the *rank* of the original structure matrix. This minimal structural representation is then used to derive a parametric description of the nine elements that compose the 3×3

matrix representing the unknown pose. Finally, the free parameters in this description are determined through a numerical optimization that guarantees the orthonormality of the resulting 3×3 pose matrix.

The major drawback of Yuan's approach is that the exact format of the parametric descriptions for the pose matrix's elements depends both on the number of feature points (n) and on the rank of the structure matrix (r). For instance, if the right correspondences for exactly four model points in general position are known, the numerical optimization in Yuan's method is carried out in a four-dimensional space. However, if only three point correspondences in general position are available (and thus the target is a 2D object), a six-dimensional optimization is needed. There is no middle ground between planar and non-planar scenes. The determination of the type of structure available is carried out prior to the optimization phase and the numerical behavior in each possible case is completely distinct. Thus, if one decides to be strict about ruling scenes as planar, in quasi-planar scenes the proximity of a singularity may generate numerical instability. But on the other hand, if a more liberal planarity criterion is adopted, then the use of routines designed for planar scenes in instances that are actually not planar may yield inaccurate results.

In our, technique, on the contrary, a transition between these two types of instances happens in a continuous, gradual way, since there is no singularity in the planar case. Hence we claim that, to the best of our knowledge, this paper introduces the first method for independent orientation recovery that fully exploits the superiority of point correspondences over line correspondences. Furthermore, our technique is much simpler and more intuitive than Yuan's method, because the number of parameters used in the numerical optimization is fixed (four) and the meaning of each of these parameters is very clear, since they directly encode the rotational components of the unknown pose.

We formulate our technique as an extension of Phong-Horaud-Tao's solution for line correspondences, in order to make use of the robustness of their trust-region optimization algorithm and the efficiency of their dual-quaternion pose representation. But actually, the alignment constraints that we derive are not tied in any particular way to these two particular aspects of Phong-Horaud-Tao's algorithm and one might perfectly well use our geometrical formulation with other optimization techniques such as a simple first-order iterative gradient method, and other pose representation schemes such as Euler angles and a three-element translation vector.

Finally, we suggested a very specific example of a real-life application that may benefit from our technique, namely outdoors visual navigation of a mobile robot. Extensive evaluation with synthetic data showed that, for this particular application, our proposed algorithm is significantly more accurate and even a bit faster than Phong-Horaud-Tao's line-based decoupled solution, and on the other hand, it is significantly faster than Phong-Horaud-Tao's coupled solution for point correspondences, while retaining roughly the same accuracy in the average case.

References

- [1] S. L. Altmann. *Rotations, Quaternions and Double Groups*. Clarendon Press, 1986.
- [2] H. Araujo, R. L. Carceroni, and C. M. Brown. A fully projective formulation to improve the accuracy of Lowe's pose-estimation algorithm. *Comp. Vis. Image Und.*, 70(2):227–238, 1998.
- [3] F. Cozman and E. Krotkov. Position estimation from outdoor visual landmarks for teleoperation of lunar rovers. In *Proc. IEEE Workshop Applications Comp. Vis.*, pages 156–161, 1996.
- [4] D. F. DeMenthon and L. S. Davis. Model-based object pose in 25 lines of code. *Int. J. Comp. Vis.*, 15:123–141, 1995.
- [5] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.

- [6] D. B. Gennery. Visual tracking of known three-dimensional objects. *Int. J. Comp. Vis.*, 7(3):243–270, 1992.
- [7] R. Horaud, S. Christy, F. Dornaika, and B. Lamiroy. Object pose: The link between weak perspective, paraperspective and full perspective. *Int. J. Comp. Vis.*, 22(2):173–189, 1997.
- [8] R. Horaud, B. Conio, O. Le Boulleux, and B. Lacolle. An analytic solution for the perspective 4-point problem. *CVGIP*, 47:33–44, 1989.
- [9] Y. Liu, S. Huang T, and O. D. Faugeras. Determination of camera location from 2-D to 3-D line and point correspondences. *IEEE Trans. PAMI*, 12(1):28–37, 1990.
- [10] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artif. Intell.*, 31(3):355–395, 1987.
- [11] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. PAMI*, 13(5):441–450, 1991.
- [12] N. Navab and O. Faugeras. Monocular pose determination from lines: Critical sets and maximum number of solutions. In *Proc. IEEE Conf. CVPR*, pages 254–260, 1993.
- [13] D. Oberkampf, D. F. DeMenthon, and L. S. Davis. Iterative pose estimation using coplanar feature points. *Comp. Vis. Image Und.*, 63(3):495–511, 1996.
- [14] T. Q. Phong, R. Horaud, and P. D. Tao. Object pose from 2-D to 3-D point and line correspondences. *Int. J. Comp. Vis.*, 15:225–243, 1995.
- [15] S. Sullivan and J. Ponce. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Trans. PAMI*, 20(10):1091–1097, 1998.
- [16] T. N. Tan, G. D. Sullivan, and K. D. Baker. Model-based localisation and recognition of road vehicles. *Int. J. Comp. Vis.*, 27(1):5–25, 1998.
- [17] M. W. Walker and L. Shao. Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Und.*, 54(3):358–367, 1991.
- [18] J. S.-C. Yuan. A general photogrammetric method for determining object position and orientation. *IEEE Trans. Rob. Aut.*, 5(2):129–142, 1989.