Perspective-2-Ellipsoid: Bridging the Gap Between Object Detections and 6-DoF Camera Pose

Vincent Gaudillière, Gilles Simon, Marie-Odile Berger

Perspective-2-Ellipsoid: Bridging the Gap Between Object Detections and 6-DoF Camera Pose

Vincent Gaudillière[1], Gilles Simon[1], Marie-Odile Berger[1]

***Abstract*— Recent years have seen the emergence of very effective ConvNet-based object detectors that have reconfigured the computer vision landscape. As a consequence, new approaches that propose object-based reasoning to solve traditional problems, such as camera pose estimation, have appeared. In particular, these methods have shown that modelling 3D objects by ellipsoids and 2D detections by ellipses offers a convenient manner to link 2D and 3D data. Following that promising direction, we propose here a novel object-based pose estimation algorithm that does not require any sensor but a RGB camera. Our method operates from at least two object detections, and is based on a new paradigm that enables to decrease the Degrees of Freedom (DoF) of the pose estimation problem from six to three, while two simplifying yet realistic assumptions reduce the remaining DoF to only one. Exhaustive search is performed over the unique unknown parameter to recover the full camera pose. Robust algorithms designed to deal with any number of objects as well as a refinement step are introduced. Effectiveness of the method has been assessed on the challenging T-LESS and Freiburg datasets.**

I. Introduction

Estimating the position and orientation of a camera in relation to its environment is a fundamental task in computer vision. In this problem, it is necessary to build and maintain a three-dimensional representation of the environment in which the observer operates [1]. When the scene is modeled by a 3D point cloud, the camera pose can be unambiguously recovered from four correspondences between points in the image and points in the model [2]. To achieve greater accuracy, most methods consider an arbitrary number of 2D-3D correspondences [3], [4]. However, the process efficiency is directly impacted by significant changes in viewpoints and by the lack of discrimative power of local feature descriptors in certain conditions (*e.g.* lack of texture, presence of repeated patterns).

There has recently been an explosion in the performances of automatic object detection algorithms, driven by methods based on ConvNets such as R-CNN [5], SSD [6], or YOLO [7]. This qualitative leap has led to the emergence of new approaches to solving traditional computer vision problems. Recent end-to-end methods such as poseCNN [8], SSD6D [9] and DPOD [10] have been proposed for 6D pose recovery. Such methods however need retraining when a new scene has to be considered. In order to build more flexible systems but still take advantage of progress in recognition, a new trend of research aims at considering pose computation at the level of objects. Indeed, object detection algorithms are able to recognize objects across a wide range of viewpoints and in different weather or lighting conditions. This opens the way towards more robust pose algorithms based on high-level features (objects or corners [11]) instead of traditional low-level primitives (keypoints). Li *et al.* [12], [13] proposed to use object detections to estimate relative camera poses in the case of large changes in viewpoints. However, modelling the scene by a set of 3D cuboids and the 2D detections by rectangles does not allow to derive closed-form solutions to projection equations.

Modeling object projections by ellipses allowed Crocco *et al.* to propose an analytical solution to the *Structure from Motion* (SfM) reconstruction of the scene in the form of a set of ellipsoids corresponding to objects of interest [14]. However, this method is limited to the case of orthographic projection. Perspective projection is taken into account in [15], where Rubino *et al.* proposed an analytical solution to build such a semantic 3D model from only three calibrated perspective cameras. The reconstructed model is therefore composed only of a few objects whose projections can be detected in images under a large range of viewpoints and conditions. Object detections were used in [16] to correct scale drift in monocular SLAM sequences.

In [17], Nicholson *et al.* presented a SLAM method to simultaneously build the set of 3D ellipsoids and compute the camera poses. That solution proposes to minimize a geometric reprojection error as a function of the camera's six DoF, based on initial position and orientation values provided by odometric sensors. Recently, it has been shown that the problem of camera pose estimation from ellipse-ellipsoid correspondences has at most 3 DoF, since the camera position can be inferred from its orientation, provided that at least one ellipse-ellipsoid correspondence is known [18], [19]. In particular, the possibility to compute a rough estimate of the pose from the camera orientation acquired by sensors or computed from vanishing points was demonstrated in [19]. Recovering the full camera pose from at least two objects was investigated on synthetic data in [18]. However, a prior on orientation was required and the method has proven sensitive to noise on ellipses as well as to the number of ellipses detected in the image. Following on from these works, we propose a method to recover an estimate of the full camera pose from at least two ellipse-ellipsoid correspondences that does not require prior nor sensors. Given two detected objects, the method presented in section II allows the camera orientation to be recovered as a function of only one angular parameter under two assumptions satisfied by many robotics applications. Given the possibility to derive position from orientation [18], the camera pose is the one that

[1]Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{firstname.lastname}@inria.fr

minimizes the ellipse-ellipsoid reprojection error. A robust method is then presented in section III to handle any number of objects. As shown in the experiments (section IV), this method is of particular interest when a small number of objects are visible. It thus allows localization for a large variety of viewpoints on the scene, either close-up or distant views, making this method interesting for various robotic tasks.

II. Pose Estimation from 2 Ellipse-Ellipsoid Pairs

In this section, we present the process of camera pose estimation in the minimal case of two 2D-3D correspondences. The method exploits the inherent decoupling between camera orientation and position arising from the ellipse-ellipsoid modeling paradigm, which was introduced in [18], [19], and derives an approximated analytical expression of the complete camera pose as a function of only one angular parameter.

A. Method Overview

To estimate the camera orientation, our method relies on two weak assumptions, that enable to restrict the three degrees of freedom of the orientation determination problem to only one. More specifically, our assumptions are:

1) the roll angle of the camera is zero,
2) the line defined by the two ellipsoid centers projects onto the line defined by the two ellipse centers.

This compares with [20], although Toft *et al.* make stronger assumptions than we do to reach the same number of DoF in the camera pose estimation process. Indeed, they assume that the gravity direction is known in the camera's coordinate system (*i.e.* the camera y-axis is colinear to the world z-axis), whereas we just assume coplanarity between camera's x-axis and world's horizontal plane (assumption 1). They assume that one 2D-3D point correspondence is known in the camera's coordinate system, whereas we rely on the very realistic approximation that the projection of the line connecting the centers of the ellipsoids coincide with the line connecting the centers of the ellipses (assumption 2).

The first assumption refers to the case where the x-axis $\boldsymbol{i_{cam}}$ of the camera lies on a world's horizontal plane (angle $\theta_1 = 0$). Let $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ (resp. $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$) be the center of the two ellipsoids (resp. ellipses). The second assumption implies that the vector $\boldsymbol{c} = (\boldsymbol{C_2} - \boldsymbol{C_1})/||\boldsymbol{C_2} - \boldsymbol{C_1}||$ lies on the plane passing through the camera center and the centers of the ellipses, that is $\theta_2 = 0$ (see Fig. 1). In practice, assumption 1 is nearly satisfied by numerous robotics applications. It is trivially true for autonomous driving applications. We also show in table I that θ_1 values computed on sequences acquired with a robotic arm (T-LESS) or with a handheld camera (Freiburg dataset) are small. It is also important to note that in many cases, rectification techniques based on vanishing points can be used to make assumption 1 satisfied. Due to the fact that ratios of distances are not preserved by perspective projection, the projection of $\boldsymbol{C_i}$ does not match exactly $\boldsymbol{c_i}$ and assumption 2 is not strictly verified. However the distance d between these two points is generally small. Using the camera intrinsics of the Freiburg dataset, elementary calculus show that d is smaller when the ellipsoid is farther from the camera. In addition, for a given camera/ellipsoid depth, d increases when the view line direction is close to the image plane. To give a more precise idea, when considering a sphere at a depth D from the camera, with a ratio $diameter/D = 1/10$, d ranges from 0 to 1.2 with 0.55 pixels as mean error. For an object close to the camera with $diameter/D = 1/4$, d ranges from 0 to 7 with 3.5 as mean error. This leads in practice to small values of θ_2 presented in table I and Fig. 5.

Our method proceeds in two steps. (i) Given any orientation of $\boldsymbol{i_{cam}}$ in the plane $(\boldsymbol{i_w}, \boldsymbol{j_w})$, the camera orientation is obtained by exploiting the fact that the vector $\mathbf{c}$ should lie on the plane passing through the center of the camera and the centers of the ellipses $\boldsymbol{e_1}$, $\boldsymbol{e_2}$ (presented in red in Fig. 1). In practice, two camera orientations are possible (section II-B). (ii) The position that best satisfies the ellipse - ellipsoid correspondences given each camera orientation is then computed based on the theoretical considerations presented in [18], [19].

Finally we perform a one-dimensional search of the orientation of $\boldsymbol{i_{cam}}$ and retain the one that gives rise to the best overlap between the ellipses and the projected ellipsoids.

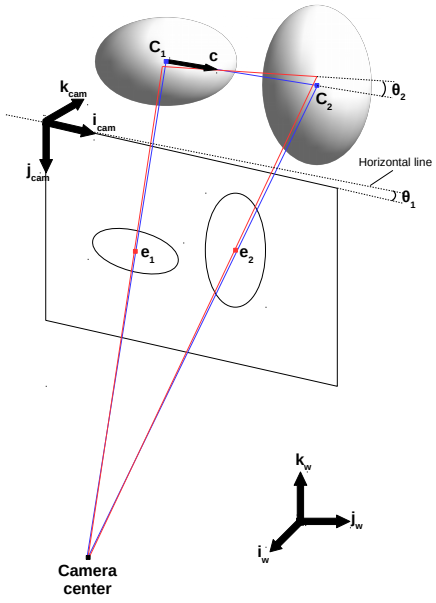


Fig. 1. Camera and scene geometry: θ_1 and θ_2 are approximated by zero.

B. Camera Orientation

We derive in this section an analytical expression of the camera orientation as a function of one angular parameter. Let us first consider three direct orthonormal bases: $\mathscr{B}_w = (\boldsymbol{i_w}, \boldsymbol{j_w}, \boldsymbol{k_w})$, referred as the *world* basis, in which the ellipsoids and the vector $\boldsymbol{c}$ are known; $\mathscr{B}_{cam} = (\boldsymbol{i_{cam}}, \boldsymbol{j_{cam}}, \boldsymbol{k_{cam}})$, referred as the *camera* basis, in which the ellipses are known; and $\mathscr{B}_p = (\boldsymbol{i_p}, \boldsymbol{j_p}, \boldsymbol{k_p})$, where $\boldsymbol{i_p}$ and $\boldsymbol{j_p}$ belong to the plane passing through the camera center and the centers of the ellipses (presented in red in Fig. 1), and

Angle	Approximation error, in °
[T-LESS] θ_1	2.20 ($\pm$0.86)
[T-LESS] θ_2 (GT ellipses)	0.29 ($\pm$0.25)
[T-LESS] θ_2 (bbox ellipses)	1.94 ($\pm$2.19)
[Freiburg] θ_1	1.33 ($\pm$1.07)
[Freiburg] θ_2 (bbox ellipses)	1.12 ($\pm$0.94)

TABLE I

MEAN ANGULAR APPROXIMATION ERRORS ($\pm$ STANDARD DEVIATION) ON TEST IMAGES: A TYPICAL SEQUENCE OF THE T-LESS DATASET (TEST_CANON/08) [21], AND ONE SUBSEQUENCE OF THE FREIBURG DATASET (FR2/DESK: 788 CAMERAS) [22].

where $\boldsymbol{k_p}$ is orthogonal to that plane. As the camera intrinsics K are known, such a basis could be built from $K^{-1}(\boldsymbol{e_2}-\boldsymbol{e_1})$ and $K^{-1}\boldsymbol{e_1}$, but any other choice is possible.

We here distinguish two cases depending on whether $\boldsymbol{c}$ and $\boldsymbol{i_{cam}}$ are colinear or not.

a) $\boldsymbol{c}$ **and** $\boldsymbol{i_{cam}}$ **are not colinear:** Let α be the angle which encodes the direction of the projection of vector $\boldsymbol{i_{cam}}$ into the horizontal plane $(\boldsymbol{i_w}, \boldsymbol{j_w})$.

We consider a fourth basis, referred as *intermediary* basis: $\mathscr{B}_{int} = (\boldsymbol{i_{cam}}, \boldsymbol{c}, \boldsymbol{i_{cam}} \times \boldsymbol{c})$, where $\times$ represents the cross product between two vectors. To consider $\mathscr{B}_{int}$ as a basis, we assume that $\boldsymbol{i_{cam}}$ and $\boldsymbol{c}$ are not colinear (the case where they are colinear is developed below in paragraph *b)*). We finally denote $\boldsymbol{v^{(b)}} = (v_x^{(b)} v_y^{(b)} v_z^{(b)})^\top$ the expression of any vector $\boldsymbol{v}$ in any basis $\mathscr{B}_b$. Therefore, the change of basis from $\mathscr{B}_{int}$ to $\mathscr{B}_w$ is related to the matrix

$$ {}^{w}P_{int} = \begin{pmatrix} cos(\theta_1)cos(\alpha) & c_x^{(w)} & \\ cos(\theta_1)sin(\alpha) & c_y^{(w)} & ... \\ sin(\theta_1) & c_z^{(w)} & \end{pmatrix} \tag{1} $$

where the last column can be easily computed as the cross product between the two first ones. The columns contain the expressions of $\mathscr{B}_{int}$ vectors into $\mathscr{B}_w$. In particular, the expression $\boldsymbol{c^{(w)}}$ of $\boldsymbol{c}$ into the world basis (second column) is known. Under assumption 1 ($\theta_1 = 0$), ${}^{w}P_{int}$ is written

$$ {}^{w}\tilde{P}_{int} = \begin{pmatrix} cos(\alpha) & c_x^{(w)} & sin(\alpha)c_z^{(w)} \\ sin(\alpha) & c_y^{(w)} & -cos(\alpha)c_z^{(w)} \\ 0 & c_z^{(w)} & cos(\alpha)c_y^{(w)} - sin(\alpha)c_x^{(w)} \end{pmatrix} $$

Similarly, the change of basis from $\mathscr{B}_{int}$ to $\mathscr{B}_{cam}$ is related to the matrix ${}^{cam}P_{int}$ given in (2) (see top of next page), where β is an unknown angle that encodes the direction of the projection of vector $\boldsymbol{c}$ into the plane $(\boldsymbol{i_p}, \boldsymbol{j_p})$. Here again, columns contain the expressions of $\mathscr{B}_{int}$ vectors into $\mathscr{B}_{cam}$. Under assumption 2 ($\theta_2 = 0$), ${}^{cam}P_{int}$ becomes ${}^{cam}\tilde{P}_{int}$, whose expression is given in (3) (see top of next page).

The camera orientation is then represented by the matrix

$$ {}^{w}R_{cam} = {}^{w}P_{int}\,{}^{cam}P_{int}^{-1} $$

and our goal is to compute the approximated orientation

$$ {}^{w}\tilde{R}_{cam} = {}^{w}\tilde{P}_{int}\,{}^{cam}\tilde{P}_{int}^{-1} \tag{4} $$

Let's demonstrate that ${}^{w}\tilde{R}_{cam}$ depends only on α (1 DoF). Indeed, since $\mathscr{B}_w$ is an orthonormal basis, the angle γ between $\boldsymbol{i_{cam}}$ and $\boldsymbol{c}$ satisfies

$$ cos(\gamma) = \begin{pmatrix} cos(\alpha) \\ sin(\alpha) \\ 0 \end{pmatrix} \cdot \begin{pmatrix} c_x^{(w)} \\ c_y^{(w)} \\ c_z^{(w)} \end{pmatrix} \tag{5} $$

$$ = cos(\alpha)c_x^{(w)} + sin(\alpha)c_y^{(w)} \tag{6} $$

Since the dot product between vectors does not depend on the orthonormal basis in which vectors are expressed, γ also satisfies

$$ cos(\gamma) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} cos(\beta)i_{p,x}^{(cam)} + sin(\beta)j_{p,x}^{(cam)} \\ cos(\beta)i_{p,y}^{(cam)} + sin(\beta)j_{p,y}^{(cam)} \\ cos(\beta)i_{p,z}^{(cam)} + sin(\beta)j_{p,z}^{(cam)} \end{pmatrix} $$
$$ = cos(\beta)i_{p,x}^{(cam)} + sin(\beta)j_{p,x}^{(cam)} $$

Thus

$$ cos(\gamma) = \sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}(cos(\beta)cos(\delta) + sin(\beta)sin(\delta)) $$

where δ is defined such that

$$ \begin{cases} cos(\delta) = \frac{i_{p,x}^{(cam)}}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}} \\ sin(\delta) = \frac{j_{p,x}^{(cam)}}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}} \end{cases} $$

using (6) we finally obtain

$$ cos(\beta - \delta) = \frac{cos(\gamma)}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}} $$
$$ = \frac{cos(\alpha)c_x^{(w)} + sin(\alpha)c_y^{(w)}}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}} $$

Finally, it remains only two possibilities for β as a function of α assuming that α is known:

$$ \beta = \delta \pm \arccos\left(\frac{cos(\alpha)c_x^{(w)} + sin(\alpha)c_y^{(w)}}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}} \right) \tag{7} $$

b) $\boldsymbol{c}$ **and** $\boldsymbol{i_{cam}}$ **are colinear:** If $\boldsymbol{i_{cam}}$ and $\boldsymbol{c}$ are colinear, the camera orientation estimation method presented above cannot be applied. However, the colinearity means that $\boldsymbol{c}$ is horizontal ($c_z^{(w)} = 0$), and that $\boldsymbol{i_{cam}} = \pm\boldsymbol{c}$:

$$ \boldsymbol{i_{cam}^{(w)}} = \begin{pmatrix} c_x^{(w)} \\ c_y^{(w)} \\ 0 \end{pmatrix}, \text{ or } \boldsymbol{i_{cam}^{(w)}} = \begin{pmatrix} -c_x^{(w)} \\ -c_y^{(w)} \\ 0 \end{pmatrix} $$

Moreover, it also causes that the vectors $(\boldsymbol{k_w}, \boldsymbol{k_w} \times \boldsymbol{c})$ define a plane that contains $\boldsymbol{j_{cam}}$. In other words, there is

$$ {}^{cam}P_{int} = \begin{pmatrix} 1 & cos(\theta_2)cos(\beta)i_{p,x}^{(cam)} + cos(\theta_2)sin(\beta)j_{p,x}^{(cam)} + sin(\theta_2)k_{p,x}^{(cam)} & 0 \\ 0 & cos(\theta_2)cos(\beta)i_{p,y}^{(cam)} + cos(\theta_2)sin(\beta)j_{p,y}^{(cam)} + sin(\theta_2)k_{p,y}^{(cam)} & -cos(\theta_2)cos(\beta)i_{p,z}^{(cam)} - cos(\theta_2)sin(\beta)j_{p,z}^{(cam)} - sin(\theta_2)k_{p,z}^{(cam)} \\ 0 & cos(\theta_2)cos(\beta)i_{p,z}^{(cam)} + cos(\theta_2)sin(\beta)j_{p,z}^{(cam)} + sin(\theta_2)k_{p,z}^{(cam)} & cos(\theta_2)cos(\beta)i_{p,y}^{(cam)} + cos(\theta_2)sin(\beta)j_{p,y}^{(cam)} + sin(\theta_2)k_{p,y}^{(cam)} \end{pmatrix} \quad (2) $$

$$ {}^{cam}\tilde{P}_{int} = \begin{pmatrix} 1 & cos(\beta)i_{p,x}^{(cam)} + sin(\beta)j_{p,x}^{(cam)} & 0 \\ 0 & cos(\beta)i_{p,y}^{(cam)} + sin(\beta)j_{p,y}^{(cam)} & -(cos(\beta)i_{p,z}^{(cam)} + sin(\beta)j_{p,z}^{(cam)}) \\ 0 & cos(\beta)i_{p,z}^{(cam)} + sin(\beta)j_{p,z}^{(cam)} & cos(\beta)i_{p,y}^{(cam)} + sin(\beta)j_{p,y}^{(cam)} \end{pmatrix} \quad (3) $$

an angle α' such that $\boldsymbol{j_{cam}} = cos(\alpha')\boldsymbol{k_w} + sin(\alpha')(\boldsymbol{k_w} \times \boldsymbol{c})$

$$ \boldsymbol{j}_{\boldsymbol{cam}}^{(\boldsymbol{w})} = cos(\alpha') \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + sin(\alpha') \begin{pmatrix} -c_y^{(w)} \\ c_x^{(w)} \\ 0 \end{pmatrix} = \begin{pmatrix} -sin(\alpha')c_y^{(w)} \\ sin(\alpha')c_x^{(w)} \\ cos(\alpha') \end{pmatrix} $$

Thus the camera orientation matrix can be directly written as a function of α':

$$ {}^{w}\tilde{R}_{cam} = \begin{pmatrix} c_x^{(w)} & -sin(\alpha')c_y^{(w)} & cos(\alpha')c_y^{(w)} \\ c_y^{(w)} & sin(\alpha')c_x^{(w)} & -cos(\alpha')c_x^{(w)} \\ 0 & cos(\alpha') & sin(\alpha') \end{pmatrix} $$

or

$$ {}^{w}\tilde{R}_{cam} = \begin{pmatrix} -c_x^{(w)} & -sin(\alpha')c_y^{(w)} & -cos(\alpha')c_y^{(w)} \\ -c_y^{(w)} & sin(\alpha')c_x^{(w)} & cos(\alpha')c_x^{(w)} \\ 0 & cos(\alpha') & -sin(\alpha') \end{pmatrix} \quad (8) $$

where the columns are the expressions of the camera basis vectors into the world basis. The last column is derived as the cross product between the two first ones, using the fact that $\boldsymbol{c}$ is normalized (${c_x^{(w)}}^2 + {c_y^{(w)}}^2 = 1$).

C. Camera Position

Previous works ([18], [19]) have demonstrated that the camera position can be derived from its orientation, as soon as one ellipse-ellipsoid pair is known. The main insights of the references are presented below.

In what follows, the *backprojection cone* refers to the cone generated by the lines passing through the camera center and any point on the projected ellipse. Let us denote $A^{(w)} \in \mathbb{R}^{3\times3}$ the quadratic form of an ellipsoid expressed in $\mathscr{B}_w$, and $B'^{(cam)} \in \mathbb{R}^{3\times3}$ the quadratic form of the backprojection cone associated to the corresponding ellipse expressed in $\mathscr{B}_{cam}$.

$$ B'^{(w)} = {}^{w}R_{cam} B'^{(cam)} {}^{w}R_{cam}^{\top} $$

It has been proven that the couple of matrices $\{A^{(w)}, B'^{(w)}\}$ has two distinct generalized eigenvalues (multiplicities 1 and 2). Denoting $\boldsymbol{\Delta}^{(w)}$ the vector connecting the center of the ellipsoid to the camera center expressed in $\mathscr{B}_w$, and $\boldsymbol{\delta}^{(w)}$ a generalized eigenvector of norm 1 associated to the eigenvalue of multiplicity 1 (let's say σ), $\boldsymbol{\Delta}^{(w)}$ is given by the formula:

$$ \boldsymbol{\Delta}^{(w)} = k\boldsymbol{\delta}^{(w)} \quad (9) $$

where k satisfies the matrix equation (10). The sign of k is obtained by applying the chirality constraint, which ensures that the objects lie in front of the camera.

$$ k^2(A_i^{(w)}\boldsymbol{\delta}_i^{(w)}{\boldsymbol{\delta}_i^{(w)}}^{\top}A_i^{(w)} - {\boldsymbol{\delta}_i^{(w)}}^{\top}A_i^{(w)}\boldsymbol{\delta}_i^{(w)}A_i^{(w)}) = \sigma_i B_i'^{(w)} - A_i^{(w)} \quad (10) $$

In theory, vectors $\boldsymbol{\Delta}^{(w)}$ associated to each ellipsoids define the same camera center. In practice, the camera center is computed as the centroid of corresponding noisy positions.

D. Pose Computation Algorithm

The orientation can be computed with methods a) or b) described in section II-B depending on whether $\boldsymbol{c}$ and $\boldsymbol{i_{cam}}$ are colinear or not. If $\boldsymbol{c}$ is not horizontal, then method a) applies. If not, we compute the two possible solutions given by a) and b) and keep the one which gives the best overlap in the Jaccard sense.

Whether a) or b) method is considered, ${}^{w}\tilde{R}_{cam}$ has only one degree of freedom. We thus perform an exhaustive search over potential α or α' values using uniform discretization of $[0°; 360°]$ interval into N values.

In the case where $\boldsymbol{c}$ and $\boldsymbol{i_{cam}}$ are not colinear, we compute for each discretized value of α the two possible β values using (7), and derive the two possible camera orientations using (4). In total, we compute $2N$ camera orientations. If $\boldsymbol{c}$ is horizontal, solution b) is computed as well. During this second search over discretized values of (α'), we assume that $\boldsymbol{c}$ and $\boldsymbol{i_{cam}}$ are colinear, and obtain the orientations from (8). In total, $4N$ camera orientations are computed.

Then, for each potential camera orientation, we derive the camera position using the method described in Section II-C, and evaluate the correctness of the full camera pose by measuring the Jaccard distances between detected and reprojected ellipses. More specifically, considering A and B two image regions delimited by ellipses, the Jaccard distance $J(A, B)$ is defined as:

$$ J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} $$

where $|A \cap B|$ is the area of intersecting ellipses, and $|A \cup B|$ the area of their union. Finally, the selected pose is the one that minimizes the Jaccard distance averaged over the two ellipse-ellipsoid pairs.

III. Robust Pose Estimation and Refinement

In order to deal with more than two object detections, we have designed a RANSAC-based algorithm to obtain the best possible initial pose, followed by a refinement step to improve the estimation accuracy.

A. RANSAC P2E Procedure for Pose Estimation

The main idea of *RANSAC P2E* is to consider successively every possible pair of detected objects (let's say $N_{2Dpairs}$). Given the mapping between 2D objects detected in the image and 3D objects from the model, $N_{2Dpairs}$ poses can be computed using the algorithm presented in II. The consensus is then computed for each pose. A correspondence is considered as an inlier if the Jaccard distance between the reprojected ellipsoid and the 2D ellipse is smaller than a certain threshold (0.5 in the experiments). As usual, the best pose is the one that maximizes the number of inliers. If several poses have had this maximal size, the retained pose is the one that minimizes the mean Jaccard distance of the inlier set. An exhaustive search among the pairs of correspondences is possible as the number of objects in an even large scene remains relatively small (maximum dozens of objects). In practice, the number of 2D-3D objects correspondences which are examined depends on the number of 3D objects that belong to the same class. Indeed, since only object classes are detected, a label, e.g. *chair*, may match each particular 3D chair instance of the scene model. Suppose for example that N_1 objects labeled as *chair* are detected in the image and suppose that there are N_2 instances of chairs in the scene model. Then $N_1 \times N_2$ possible correspondences between 2D and 3D objects are generated.

B. Pose Refinement

Once a first camera pose estimate has been computed, one can apply a refinement step which consists in optimizing an ellipsoid reprojection error as a function of the standard camera pose parameters. Here again, our ellipse-ellipsoid modeling paradigm enables to reduce the number of parameters of the objective function from 6 to 3. Advantages and limits of such a method are discussed in Section IV-A.2. If a CAD model of the scene is available, iterative minimization of the distance between the projection of the models and image features can also be used to refine our pose estimation.

IV. Experiments and Evaluation

A. T-LESS Dataset Experiments

The T-LESS Dataset [21] is composed of twelve scenes with around 500 cameras per scene. Each scene exhibits a few texture-less symmetrical objects, that are 10 to 30 cm long and laid close to each other. The cameras are roughly located on a semi-sphere of radius 75 cm around the centroid of the objects. Available depth information was ignored in our experiments. In the following, we report experimental results on the representative test_canon/08 sequence, that includes 504 images and 6 objects. During experiments, each object received a unique label, resulting in an unambiguous mapping between 2D detections and 3D model instances.

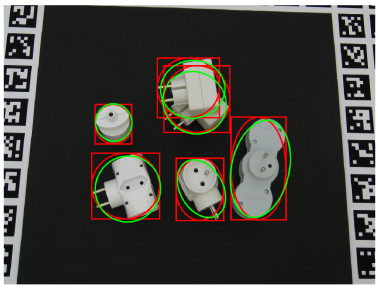


Fig. 2. Ground truth ellipses obtained by projecting the ellipsoids with the ground truth camera matrix are in green, whereas bbox ellipses are in red.

Ellipses	Nb. of objects N_o	Ori. error (°)	Loc. error (cm)
GT	2	3.37 (± 31.62)	3.99 (± 23.65)
	3	2.71 (± 0.96)	3.03 (± 1.44)
	4	2.51 (± 0.91)	2.77 (± 1.38)
	5	2.50 (± 0.90)	2.83 (± 1.37)
	6	2.46 (± 0.89)	2.76 (± 1.36)
bbox	2	9.99 (± 65.07)	12.23 (± 43.06)
	3	4.41 (± 7.77)	6.14 (± 9.33)
	4	3.78 (± 2.56)	5.03 (± 3.18)
	5	3.36 (± 2.18)	4.48 (± 2.67)
	6	3.15 (± 1.96)	4.09 (± 2.42)

TABLE II

T-LESS: Median (± standard deviation) errors of our RANSAC-like pose estimation method.

In the following results, we either consider as detections the *ground truth ellipses* (GT), that is to say ellipses that are obtained by reprojecting the ellipsoids with ground truth camera matrices, or *bounding box ellipses* (bbox), *i.e.* the ones that are fitted into the bounding boxes of the 2D objects. The difference between these two types of ellipses is illustrated in Fig. 2.

1) RANSAC P2E:

To evaluate how the pose accuracy depends on the number of objects detected in the image, N_o objects were randomly picked in each image of the sequence ($2 \leq N_o \leq 6$). The influence of the bias induced by considering bbox ellipses with principal axes oriented along the x and y directions was also examined. Results are presented in Table II for GT and bbox ellipses. Averages and standard deviations of the error are computed over the sequence.

Symmetry or quasi-symmetry in the set of ellipsoids may lead to several candidate poses with similar reprojection errors, possibly misleading our best pose selection method. This often occurs when only 2 objects are considered. A third object can generally disambiguate the solution selection. Despite this, our method achieves an acceptable level of pose accuracy, taking into account the inherent error induced by our simplifying assumptions (see Table I for comparison), as well as the potential bias on detected ellipses (bbox). An interesting feature is the fact that the performance increases with the number of objects in the scene.

2) Object-based pose refinement:

The obtained initial poses (referred as *RANSAC P2E* in Fig. 3) were then refined by optimizing three different types of reprojection errors. The first one is the geometric reprojection

error introduced in [17] (mean quadratic distance between bounding boxes vertices of detected and reprojected ellipses), the second one is the algebraic error derived from [14], [15] (algebraic distance between vectors formed by the 5 parameters characterizing dual ellipses in homogeneous coordinates: see Equation (11) and [15] for the notations), and the third one is the Jaccard distance (see Section II-D).

$$\text{Algebraic error: } \sum_i ||\beta_i C_i^* - P Q_i^* P^\top||^2 \qquad (11)$$

The three types of error were minimized as a function of the six camera pose parameters (referred as *RANSAC P2E + opt geom6*, *algebr6*, and *Jaccard6*) or only of the three orientation parameters, in which case the camera position was derived from its orientation as explained in II-C (*geom3*, *algebr3*, and *Jaccard3*). The results are presented in Fig. 3. When the ellipses are perfectly detected (GT ellipses column), and for most optimized errors, the refinement step enables significant correction of errors induced by the two initial simplifying assumptions. Note that empty bars in the graphs represent zero localization errors.

In practical settings, only bbox ellipses are available. Reported results (left column) bring up the fact that ellipse-based pose refinement does not automatically improve the pose accuracy as it was expected. More precisely, when the number of objects is too small (< 5 objects), optimization procedure will in average degrade the method performance due to a noise overfitting effect. In contrast, it will take advantage of more objects to extract a sufficient degree of generality from the data, allowing the optimized pose to be more accurate than the initial one. Finally, in practical settings with few objects in the image and rough detections, a pose refinement step based on local features should be preferred to ellipse-based ones. For instance, Fig. 3 (last column) shows an example of the result obtained after iterative minimization of the reprojection error between the contours of a CAD model of the scene and the contours of the image obtained by Canny filtering. The initial estimate (Fig. 3, top-right) was obtained by using our RANSAC P2E pose computation method.

3) Effect of the roll angle value assumption:
To assess the robustness of our method with respect to the error introduced by our first assumption ($\theta_1 = 0$), larger errors were artificially generated by introducing in equation (1) θ_1 values further away from the real ones (-5° and -10° as assumed values whereas real values range from 0° to 3.5°). Mean ($\pm$ standard deviation) errors on θ_1 were measured on estimated cameras (referred as *initial error*) and on refined cameras (*final err.*) over the sequence. Bbox ellipses were considered in these experiments. The results presented in Table III show that although bad initial assumptions on θ_1 values lead to larger errors, the refinement step is in average able to significantly reduce this error, especially when the number of object increases.

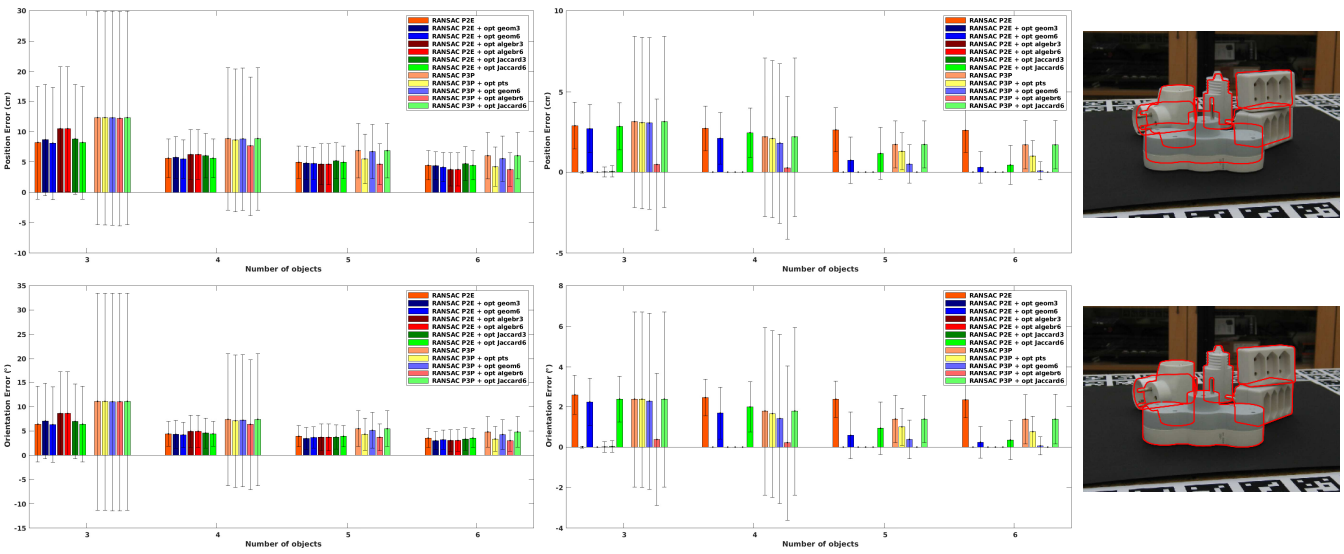
4) Comparison with PnP:
We have compared our ellipse-ellipsoid based approach to a point-based approach in which the objects are assimilated to their centroids (ellipsoids centers in 3D, bounding boxes centers in 2D). Then, a classic RANSAC P3P algorithm was used to recover the camera pose, followed by a 6-DoF optimization of the point-based reprojection error (referred as *RANSAC P3P + opt pts* in Fig. 3). It is important noting that our method requires only 2 objects to recover the pose, whereas the point-based approach requires at least 4 points, or 3 points with additional information. Indeed, with only 3 points, P3P induces 4 exact solutions and one cannot disambiguate between them without a fourth correspondence or additional information. In our experiments, the retained solution was the one that gives rise to the smallest ellipsoid reprojection error (in the sense of Jaccard distance). Pose errors obtained with P2E in the case of 2 objects ($9.50(\pm 23.65)$cm and $10.91(\pm 31.62)°$ with GT ellipses, $34.86(\pm 43.06)$cm and $44.33(\pm 65.07)°$ with bbox ellipses) are not presented in the figure to make it clearer. When bbox ellipses are considered (left column), our initial pose estimation method is in average more accurate than the point-based one, and the gap in accuracy tends to be lower when the number of objects increases. When GT ellipses are considered, the opposite effect is observed, since augmenting the number of correspondences does not significantly improve the accuracy of *RANSAC P2E*. This is due to the fact that in this case the pose is very constrained by center correspondences and that assumption 1 contributes to slightly bias the estimation. Whatever the refinement method used, it is important noting that the ellipse-ellipsoid modeling allows for a higher confidence into the results, since the standard deviation of the pose error (represented by vertical error bars) is significantly lower in our case.

Assumed value	0°	-5°	-10°
Initial error	2.20 ($\pm$ 0.86)	7.20 ($\pm$ 0.86)	12.20 ($\pm$ 0.86)
Final err. (3 obj.)	2.05 ($\pm$ 1.12)	5.59 ($\pm$ 2.46)	9.02 ($\pm$ 4.31)
Final err. (4 obj.)	1.88 ($\pm$ 1.11)	5.02 ($\pm$ 2.71)	8.08 ($\pm$ 4.52)
Final err. (5 obj.)	1.83 ($\pm$ 1.09)	4.63 ($\pm$ 2.93)	6.79 ($\pm$ 4.90)
Final err. (6 obj.)	1.70 ($\pm$ 1.13)	4.11 ($\pm$ 2.96)	5.55 ($\pm$ 4.84)

TABLE III

T-LESS: MEAN ($\pm$ STANDARD DEVIATION) INITIAL AND FINAL ERRORS ON θ_1 VALUES (IN °) DEPENDING ON THE INITIAL ASSUMPTION.

5) Comparison with learning-based algorithms: We also intended to compare our approach with learning-based methods such as poseCNN [8], SSD6D [9] or DPOD [23]. However, results are only available on datasets composed of one object whereas our method requires at least two objects. Some authors provided experiments on the OccludedLINEMOD dataset that contains several objects [8], but these objects are moved from one image to the next preventing us from building a 3D model. This does not allow to conduct any fair comparison. We nevertheless compare our method to CorNet [11], which aims at computing pose from recognized generic 3D corners without specific scene retraining. Results are available on object 20 from scene 08 (T-LESS). For corNet, no solution with an IOU larger than .8 is available whereas we obtain a succes rate which



bbox ellipses GT ellipses

Fig. 3. Refinement issues on the T-LESS dataset. Mean (with standard deviation) position and orientation errors before and after refinement with bbox ellipses (left) and GT ellipses (middle). Right: Pose refinement based on contour registration. Top: Perspective projection of a CAD model of the scene based on RANSAC P2E. Bottom: Pose refinement by iterative minimization of the reprojection error of the model edges.

goes from 66% (pose from 2 objects) to 98.6% (6 objects). Only 34% of success rate is obtained by CorNet with a lower requirement of 0.4, whereas ours ranges from 95.0% to 100%. Considering the 3D metric ADD, our pose from 2 objects is as accurate as CorNet, whereas our accuracy is much higher with more objects.

B. Freiburg Dataset experiments

The Freiburg Dataset [22] provides large and realistic environments that exhibit several objects of interest, making this dataset suitable for assessing the efficiency of object-based camera pose estimation methods. In our experiments, we consider a subset of 788 cameras from the *Freiburg2/desk* sequence. These images have been selected such that at least three objects are detected by YOLO [7] in each of them. Ellipsoidal models of objects were first built off-line from a dozen of images picked among the 2965 images of the sequence, using the method described in [15]. By contrast with T-LESS experiments (Section IV-A), the 2D-3D data associations are not known. However, YOLO labels were transferred to 3D ellipsoids during model building and, at test time, we use our extended RANSAC-like procedure presented in Section III-A to associate 2D and 3D data as well as to estimate the camera pose. Results are presented in Fig. 4, in comparison with the point-based approach already described in Section IV-A.4.

Considering the RANSAC P3P, a small distance threshold leads to discard most images (less than 4 inliers), but gives accurate results on 75% of images when the method succeeds, whereas a large threshold enables to compute a pose for a large proportion of images, but at the price of lower accuracy. On the contrary, our parameter-free method was able to process all images and provides the most accurate results: 4.76° (± 3.40°) in average in orientation, and 12.26cm (± 8.19cm) in average in position, over the 788 images. The lower level of performance here in comparison with the T-LESS experiments comes from the fact that the bounding boxes detected by YOLO often suffer from important noise and/or occlusions.

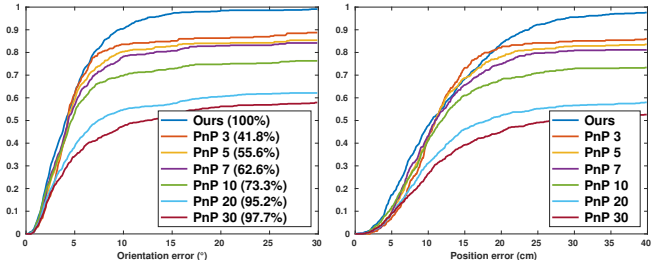


Fig. 4. Freiburg: Cumulative density functions of orientation and position errors in comparison with PnP. *PnP X* refers to the threshold X (in pixels) used to discriminate between inliers and outliers. For each method, percentages indicate the proportion of images with successful pose computation.

Figure 5 shows several typical situations with which our method can be confronted. At the bottom of each case are given information about the RANSAC input data and the localization error. Case 1 is an easy case in that a large number of objects were detected and correctly classified. Our method obviously obtains a good accuracy in such situations. In case 2, only 3 objects have been detected. This corresponds to the minimum number of objects needed to be robust to one classification error and to be able to disambiguate between multiple matching hypotheses. Object labeling is correct in case 2, but two cups and two bowls are instantiated in the model, which is well addressed here. Case 3 is more challenging: four objects have been detected but two of them appear several times in the model (the cup and the book), three have a shape far from that of an ellipsoid (the

(1) C:20, N_{in}:5 (6.4cm, 4.1°) error on $\Theta_2 = 1.0°$ (2) C:21, N_{in}:3 (15.4cm, 3.6°) error on $\Theta_2 = 2.3°$ (3) C:29, N_{in}:4 (21.1cm, 4.4°) error on $\Theta_2 = 1.7°$ (4) C:2, N_{in}:2 (15.4cm, 7.6°) error on $\Theta_2 = 6.1°$ (5) C:3, N_{in}:2 (263.0cm, 151.3°) error on $\Theta_2 = 45.3°$

Fig. 5. Example of typical situations with which our method can be confronted. First row: detection boxes obtained by YOLO. Second row: projected 3D model after RANSAC P2E. Ellipsoids classified as inliers are drawn in green, others in black. Last rows: number of 2D-3D matching hypotheses C, number of inliers N_{in} and localization errors (translation, rotation) for each case, then errors made on Θ_2 angles. The scene model consists of $N_o = 16$ ellipsoids in all these experiments.

book, the Teddy bear, and especially the plant), and finally two are partially outside the image boundaries. Despite this, the pose accuracy remains reasonable, thanks to the fairly high number of detections. Case 4 is even more difficult since only two objects were detected. Moreover, their shape is far from that of an ellipsoid and they partially fall out of the image. The box corresponding to the plant is also particularly disproportionate. Although pose accuracy suffers slightly (see the orientation error), it is not aberrant. What helped here is the fact that these two objects have been correctly classified and appear only once in the model. Case 5, however, makes our method fail: in this truncated view of the scene, only two objects have been detected, including a false positive (the corner of the desk is detected as a book). Among the three books in the model, one is arbitrarily chosen, which results in an aberrant pose.

V. Conclusion

In this paper, we have presented a novel object-based pose estimation method relying on two weak simplifying assumptions. Pose estimation is thus turned into a 1-DoF problem, solved using exhaustive search over the unknown parameter. A factor limiting the accuracy of our method is the ellipse detection process, that makes our method based on very coarse ellipses. This procedure will be reconsidered in our future work. Moreover, we have shown that our method is capable of processing scenes with few objects. A strategy will be developed to jointly take advantage of this capability and the benefits of the PnP approach with more objects.

References

[1] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE TVCG*, 2016.

[2] 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 *CVPR*, 2011.

[3] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[4] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Ep*n*p: An accurate $O(n)$ solution to the p*n*p problem," *Int. Journal of Computer Vision*, 2009.

[5] R. B. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *ECCV*, 2016.

[7] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," http://arxiv.org/abs/1804.02767, 2018.

[8] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *Robotics: Science and Systems*, 2018.

[9] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *ICCV*, 2017.

[10] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6d pose object detector and refiner," in *ICCV*, 2019.

[11] G. Pitteri, S. Ilic, and V. Lepetit, "CorNet: Generic 3D Corners for 6D Pose Estimation of New Objects without Retraining," in *ICCV Workshop on Recovering 6D Object Pose*, 2019.

[12] J. Li, D. Meger, and G. Dudek, "Context-coherent scenes of objects for camera pose estimation," in *IROS*, 2017.

[13] J. Li, Z. Xu, D. Meger, and G. Dudek, "Semantic scene models for visual localization under large viewpoint changes," in *CRV*, 2018.

[14] M. Crocco, C. Rubino, and A. Del Bue, "Structure from motion with objects," in *CVPR*, 2016.

[15] C. Rubino, M. Crocco, and A. D. Bue, "3d object localisation from multi-view image detections," *IEEE TPAMI*, 2018.

[16] D. P. Frost, O. Kähler, and D. W. Murray, "Object-aware bundle adjustment for correcting monocular scale drift," in *ICRA*, 2016.

[17] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE RA-L*, 2019.

[18] V. Gaudillière, G. Simon, and M.-O. Berger, "Camera Pose Estimation with Semantic 3D Model," in *IROS*, 2019.

[19] ——, "Camera Relocalization with Ellipsoidal Abstraction of Objects," in *ISMAR*, 2019.

[20] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl, "Semantic match consistency for long-term visual localization," in *ECCV*, 2018.

[21] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects," *WACV*, 2017.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IROS*, 2012.

[23] S. Zakharov, I. Shugurov, and S. Ilic, "Dpod: 6d pose object detector and refiner," in *ICCV*, 2019.