Level Set-Based Camera Pose Estimation From Multiple 2D/3D Ellipse-Ellipsoid Correspondences

Matthieu Zins, Gilles Simon, Marie-Odile Berger

Level Set-Based Camera Pose Estimation From Multiple 2D/3D Ellipse-Ellipsoid Correspondences

Matthieu Zins, Gilles Simon, Marie-Odile Berger

***Abstract*— In this paper, we propose an object-based camera pose estimation from a single RGB image and a pre-built map of objects, represented with ellipsoidal models. We show that contrary to point correspondences, the definition of a cost function characterizing the projection of a 3D object onto a 2D object detection is not straightforward. We develop an ellipse-ellipse cost based on level sets sampling, demonstrate its nice properties for handling partially visible objects and compare its performance with other common metrics. Finally, we show that the use of a predictive uncertainty on the detected ellipses allows a fair weighting of the contribution of the correspondences which improves the computed pose. The code is released at `gitlab.inria.fr/tangram/level-set-based-camera-pose-estimation`.**

I. Introduction

Estimating the 6-DoF pose of a camera from an RGB image is a fundamental task in computer vision, with application in robotics, autonomous systems or Augmented Reality.

The performance of classical point-based methods depends mostly on the ability to perform correct 2D-3D matching. While these methods can achieve very good accuracy, they require heavy computations to handle outliers and large point clouds as scene models. Such point clouds generally do not contain semantic information and lack of interpretability. Also, they are limited to highly textured environments.

Recently, object-based camera pose estimation has become an attractive research direction thanks to the robustness in object detection and recognition provided by deep learning. Objects have been integrated in Simultaneous Localization and Mapping (SLAM) using different modellings, such as cuboids [1] or ellipsoids [2]. However, these methods still rely on point-based tracking or odometry measurements. SLAM++ [3] and DeepSLAM++ [4] combined an object detector with CAD models for object-level RGB-D SLAM, but necessitate depth measurements as well as precise 3D models of the objects to train their networks.

In this work, we are interested in relocalizing the camera from a single RGB image with respect to a pre-built map of objects, which is of particular interest in case of tracking failure or for re-initializing SLAM in a previously scanned environment (e.g., kidnapping problem). We place ourselves in the challenging, but more realistic, context where we do not have access to precise 3D models of the objects in the scene, and instead, we use simplified ellipsoidal models.

Close to our work, [5] also models objects with ellipsoids and proposes improved elliptic detections to compute an initial camera pose. However, this is more a rough estimate of the camera pose which is computed from only a minimal set of two or three objects and the additional objects are used in a validation step in order to select the best solution. Also, its accuracy is limited by two geometric approximations: the camera roll is null and the center of an ellipsoid projects into the center of the ellipse. In this work, we go one step further and propose a pose estimation method which optimizes ellipses alignment between the 2D object detections and the projection of their 3D ellipsoidal models, taking into account all the detected and matched objects.

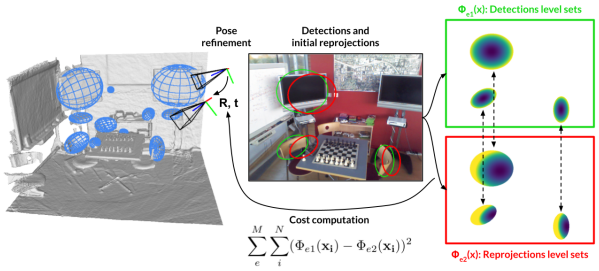

Fig. 1: Camera pose estimation by minimizing the reprojection error expressed with a level set-based metric.

One of the major challenges in estimating the camera pose from objects comes from partially visible objects. These can be occluded by other elements of the scene or partially outside the image, which, for example, happens frequently when the camera is moving. The detection of such objects are generally of poor quality and limited to the visible parts (see Figure 2). Also, contrary to keypoints which are generally in great number in an image, there are usually much less detected objects and it is therefore not conceivable to completely discard the poorly detected ones, provided that we are even able to identify them. Instead, a smart balancing of their contribution in the pose is required.

In QuadricSlam [2], Nicholson *et al*. restrict the projection of the ellipsoidal model to the on-image conic (i.e. the part of the ellipse which is inside the image, see Figure 2). This helps to deal with objects whose undetected parts are actually outside the image. However, objects that are entirely inside the image may also be poorly detected, because of occlusion or simply due to an uncommon viewoint which challenges the object detection network.

We compare different ellipse-ellipse metrics and show that our proposed cost based on level sets has nice properties

This work was supported by the MoveOn project (Inria - DFKI).
Authors are with Inria, Université de Lorraine, LORIA, CNRS.
`forename.name@inria.fr`

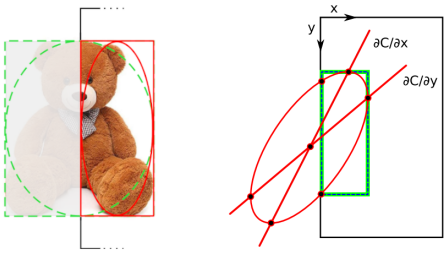

Fig. 2: Left: Incorrect detection (red) of an object partially outside the image. In reality, the detection of a partially visible object is usually even worse (see the bottom chair in Figure 7). Right: QuadricSlam bounding box [2], restricted to the on-image ellipse.

for handling partial detections. Also, inspired by the recent advances in estimating the uncertainty of a neural network prediction, we propose a weighting method that can help in determining which objects should be trusted and which should be given little importance.

Our contributions are the following:

- We develop an optimization-based method for camera pose estimation from multiple objects observations.
- We propose to use level sets to establish a metric between ellipses, and demonstrate its nice properties for dealing with poor-quality object detections.
- We show that integrating a confidence measure in the pose refinement improves the obtained pose accuracy, especially in case of partially visible objects.

II. Related work

A. Camera pose estimation

Absolute camera pose estimation was traditionally addressed by sequentially solving two sub-problems: first a feature matching problem that seeks to establish putative 2D-3D correspondences, and then a Perspective-n-Point problem [6] that minimizes a reprojection error with respect to the camera pose. With the advances of deep learning, some parts have been replaced by learned components, such as Superpoint [7] for local features detection and description or Superglue [8] for matching. Recently, detector-free matching, which directly produces dense matches without the detection phase, have also shown promising results [9].

End-to-end learning of absolute camera pose has also gained attention with PoseNet and its evolutions [10], [11]. While these methods provide an interesting robustness to illumination changes or motion blur, they do not reach the same level of accuracy as the classical structure-based methods. Scene coordinates regression has also been explored for camera pose estimation, originally with random forests [12], and later with CNNs [13], [14]. These methods provide accurate localization, but are limited to small environments due to the fixed capacity of the network and usually require depth information for training. Most of these learning-based methods require a per-scene training and only very recent works achieve generalization across scenes [15], [16].

With the emergence of deep learning and the significant advances in object detectors such as Yolo [17] or Faster R-CNN [18], object-based localization methods have become of particular interest. Weinzaepfel *et al*. proposed to use planar objects as object-of-interest for visual localization [19]. They predict dense 2D-2D correspondences between the detected object in the query image and a reference image of this object for which 3D coordinates are known. Labbé *et al*. [20] proposed an object-level bundle adjustment to refine both the poses of cameras and objects in the context of multi-cameras and multi-objects. However, they assume to have access to the 3D models of the objects, which is generally not possible in the context of SLAM. Yang *et al*. developed a monocular object SLAM in which they represent objects with cuboids [1]. They infer initial cuboids proposals from 2D bounding boxes and refine them in a bundle-adjustment, simultaneously with the camera poses and points.

Representing objects with 3D cuboids and their observations in images with 2D bounding boxes does not allow to derive closed-form solutions to the projection equations and leads to solutions with a high combinatorics to match the 3D box corners with the 2D box edges. Li *et al*. reduced this complexity by training a viewpoint classifier [21]. Another solution is to model 2D/3D objects with ellipses/ellipsoids. Crocco and Rubino leveraged this representation in the context of 3D object localization [22] and Structure-from-Motion with objects [23], [24], using a simplified camera model. Nicholson *et al*. integrated this object representation in QuadricSLAM [2], where the problem is expressed in the form of a factor graph, combining odometry and objects measurements. Similarly, Hosseinzadeh *et al*. combined points, planes and quadrics in [25].

While the previously mentioned works start from an initial pose estimate and are more related to bundle adjustement combining points, objects, planes and odometry measurements, recent works have proposed direct solutions for object-based camera pose estimation. Gaudilliere *et al*. developed a method to estimate the camera pose from two pairs of ellipse/ellipsoid [26]. Zins *et al*. extended this work with improved elliptic detections of objects [5], replacing the original axis-aligned ellipses directly inferred from the bounding boxes provided by the object detector. In these methods, data association is solved simultaneously with the pose, within a RANSAC loop. However, only two or three objects actually contribute to the estimated pose and its accuracy may be limited by two assumptions: the camera roll is null and the center of an ellipsoid projects into the center of the projection ellipse. A refinement step, based on the maximization of Intersection-over-Union between ellipses, was proposed in [26] in order to consider all the detected objects, but without success, as it did not improve the pose accuracy and even deteriorated it in some cases.

B. Uncertainty estimation

In the last years, many works attempted to quantify the uncertainty of neural networks predictions, in particular with Bayesian neural networks [27]. In [28], Gal and Kendall

distinguished two types of uncertainties: the aleatoric uncertainty which captures the noise in the observations and the epistemic uncertainty which represents the uncertainty in the model and which could be reduced with more training data. They proposed to estimate the aleatoric uncertainty in regression tasks using an observation noise parameter which can be interpreted as a learned loss attenuation. Furthermore, they introduced Monte-Carlo Dropout as a Bayesian approximation of the model weights posterior distribution in order to estimate the epistemic uncertainty. However, several forward passes are required during inference which limits its applicability. Also, adding dropout layers to a neural network raises the questions of where should they be inserted, as well as, what dropout ratio should be used. These values are usually arbitrarily chosen, but can have a significant impact on the predicted uncertainty values. Recently, ensemble methods have been advised as the new go-to method for estimating the epistemic uncertainty [29], but again, their computational cost is high, with multiple forward passes at inference and multiple network trainings.

In this work, we follow a pragmatic approach and focus on estimating the aleatoric uncertainty in order to weight the contribution of the objects to the pose. In particular, we aim at decreasing the influence of badly detected objects.

Outline: In section III, we review different ellipse-ellipse metrics and propose to use a cost based on level sets. We then compare the metrics on a 2D ellipse registration problem in section V. Finally, our 6-DoF camera pose estimation method is explained in section VI and evaluated in section VII, as well as the chosen minimization metrics and the predictive uncertainty. The experiments of sections V and VII are also illustrated in the accompanying video.

III. Ellipse-ellipse metrics

a) Intersection-over-Union: It is a widely-used metric for comparing shapes, with applications in semantic segmentation [30], object detection [31] and tracking [32]. The Jaccard distance is often minimized (e.g., in [26]), such that

$$\Delta_{IoU}(\mathcal{E}_1, \mathcal{E}_2) = 1 - \frac{|\mathcal{E}_1 \cap \mathcal{E}_2|}{|\mathcal{E}_1 \cup \mathcal{E}_2|}. \tag{1}$$

A major weakness of IoU is that it remains constant when the two ellipses are disjoint, and thus, equally quantifies them, independently of their distance. Obviously, this behaviour is not desirable for an optimization problem. Rezatofighi *et al*. address this issue in [33], where they propose a generalized version of IoU, abbreviated GIoU, and given by

$$\Delta_{GIoU}(\mathcal{E}_1, \mathcal{E}_2) = 1 - \left(IoU - \frac{|C \setminus (\mathcal{E}_1 \cup \mathcal{E}_2)|}{|C|} \right), \tag{2}$$

where C is the smallest convex object enclosing the ellipses.

b) Bounding box corners: This metric is defined between the axis-aligned bounding boxes of the two ellipses and is computed as the quadratic error between their corners coordinates, such that

$$\Delta_{bbox}(\mathcal{E}_1, \mathcal{E}_2) = \|\mathcal{B}_1 - \mathcal{B}_2\|_2^2, \tag{3}$$

where $\mathcal{B}_i$ contains the bounds of the i-th box (min_x, min_y, max_x, max_y). The main issue of this metric is that an axis-aligned bounding box does not define a unique ellipse, and thus, an infinite number of ellipses share the same bounding box. This does not guarantee ellipses to be aligned even if their bounding boxes perfectly match. Also, boxes depend on the image axes, which induces that rotated pairs of ellipses can have different distances. QuadricSlam [2] uses an improved version of this metric, abbreviated QBbox in the next sections, to better handle objects partially outside the image (see Figure 2).

c) Algebraic error: The ellipses can also be compared using their dual-form matrices. Such a matrix represents an ellipse by the envelope of all the tangent lines to its curve. In the context of 3D ellipsoid estimation from multi-view elliptic observations, Rubino *et al*. [22] minimized the quadratic error between vectorized versions of these dual-form matrices $\mathbf{C}^*_{1|2} = \mathbf{C}^{-1}_{1|2}$, given by

$$\Delta_{vec}(\mathcal{E}_1, \mathcal{E}_2) = \|vec(\mathbf{C}^*_1) - vec(\mathbf{C}^*_2)\|_2^2, \tag{4}$$

where $vec(\cdot)$ extracts the five upper elements of a matrix.

Instead, the *Frobenius* norm of the difference between the matrices is used in [25], expressed as

$$\Delta_{fro}(\mathcal{E}_1, \mathcal{E}_2) = \sqrt{Tr((\mathbf{C}^*_1 - \mathbf{C}^*_2)(\mathbf{C}^*_1 - \mathbf{C}^*_2)^T)}, \tag{5}$$

where $Tr(.)$ is the trace.

Both of these metrics, the *vectorized* version and the *Frobenius* one, actually compare the ellipses contours. While this can provide a good accuracy, considering only the contours may suffer from a limited robustness when the two initial ellipses differ a lot. Also, the values of these metrics change if a global translation is applied on the two ellipses, which is not desirable when several pairs of ellipses, spread over the entire image, are jointly optimized. Finally, it is difficult to have a geometric interpretation of such distances based on the dual-form matrices of ellipses.

d) Distribution-based distances: Another kind of metric that can be used to compare ellipses is based on probability distributions. Indeed, an ellipse, parametrized by two axes (α, β), an orientation (θ) and a center (c_x, c_y), can be interpreted as a 2D Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ where

$$\mu = \begin{bmatrix} c_x \\ c_y \end{bmatrix}, \quad \Sigma^{-1} = R(\theta) \begin{bmatrix} \frac{1}{\alpha^2} & 0 \\ 0 & \frac{1}{\beta^2} \end{bmatrix} R(\theta)^T. \tag{6}$$

The *Wasserstein* distance, also called *Earth-moving* distance, can be used to compare such distributions. Intuitively, it represents the cost of transforming one distribution into another one. Its good convergence properties have been demonstrated by Arjovsky *et al*. in [34] for distribution learning in the context of GANs. It has also been used as a loss for training an ellipse detection network in [35]. While this distance is difficult to compute in general, a closed-form formula exists for the 2D Gaussian case, given by

$$\begin{aligned} \Delta_{\mathcal{W}_2^2}(\mathcal{N}_1, \mathcal{N}_2) = &\|\mu_1 - \mu_2\|_2^2 \\ &+ Tr(\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}})^{\frac{1}{2}}). \end{aligned} \tag{7}$$

Finally, the *Bhattacharyya* [36] distance also expresses a measure of similarity between probability distributions. In the case of two 2D Gaussian distributions, it is calculated by

$$\begin{aligned}\Delta_{\mathcal{B}}(\mathcal{N}_1,\mathcal{N}_2) =& \frac{1}{8}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) \\ &+ \frac{1}{2}\ln\left(\frac{\det\Sigma}{\sqrt{\det\Sigma_1\det\Sigma_2}}\right),\end{aligned} \tag{8}$$

where $\Sigma = \frac{\Sigma_1+\Sigma_2}{2}$.

IV. Level set-based metric

In this section we propose another metric based on level sets and implicit representation of ellipses [37], which combines the advantages of considering the contours (accuracy) and the area (better robustness).

For that, we represent an ellipse with an embedding function $\Phi : \mathbb{R}^2 \to \mathbb{R}$. We use the general quadratic equation of an ellipse, where its contour corresponds to the level curve of value 1, expressed as

$$\Phi(\mathbf{x}) = (\mathbf{x}-c)^T R(\theta) \begin{bmatrix} \frac{1}{\alpha^2} & 0 \\ 0 & \frac{1}{\beta^2} \end{bmatrix} R(\theta)^T(\mathbf{x}-c). \tag{9}$$

Then, the level set-based distance between two ellipses represented by their embedding functions Φ_1 and Φ_2 is defined by

$$\Delta_\Omega(\Phi_1,\Phi_2) = \int_\Omega (\Phi_1(\mathbf{x}) - \Phi_2(\mathbf{x}))^2 \mathbf{dx}. \tag{10}$$

To keep the computational cost low, the distance is calculated on a finite set of points, regularly sampled along the level curves of the first ellipse (see Figures 1 and 3), such that

$$\Delta_{lvs}(\mathcal{E}_1,\mathcal{E}_2) = \sum_{i=1}^{N} (\Phi_1(\mathbf{x_i}) - \Phi_2(\mathbf{x_i}))^2, \tag{11}$$

where $\mathbf{x_i}$ is a sample point and N the total number of points. The size of the sampling area can be adapted depending on the original distance between the two ellipses to ensure a better convergence. Also, in contrast to sampling points on a rectangular grid aligned with the image axes, our sampling preserves the invariance to rotation.

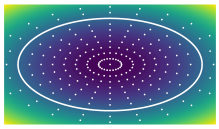

Fig. 3: Embedding function Φ and sampling points.

V. Ellipse 2D registration

We first evaluate each metric on a reduced 2D pose estimation problem, which consists of finding the rotation and translation between two ellipses $\mathcal{E}$ and $\mathcal{E}_{ref}$. $\mathcal{E}_{ref}$ is randomly generated and transformed into $\mathcal{E}$ with a rotation ($[-180°, 180°]$) and a translation ($[-60, 60]$ pixels). A detection noise (random anisotropic scaling, uniformly sampled between 0.83 and 1.2) is added to $\mathcal{E}$ to simulate an imperfect detection. The goal is to re-estimate the rigid transform by minimizing a cost between the ellipses, expressed as

$$\hat{\theta}, \hat{t} = \arg\min_{\theta,t} \Delta(\mathcal{T}(\mathcal{E},\theta,t), \mathcal{E}_{ref}), \tag{12}$$

where $\mathcal{T}(\mathcal{E},\theta,t)$ denotes rotating ellipse $\mathcal{E}$ by θ and translating it by t. While this experiment only involves a reduced problem compared to our targeted application of camera pose estimation, it highlights different behaviors of the metrics during the optimization. In Table I, we can notice that the generalized IoU as well as the algebraic distances have troubles to converge. For example, the optimization gets stuck with a flat cost using the generalized IoU when one ellipse is entirely inside the other one. On the contrary, the probabilistic metrics and the level set one perform the best. Notice that the level set metric is not subject to the flat cost problem encountered by the generalized IoU, thanks to the form of the chosen embedding function.

Metric	No noise		With noise	
	Pos. err.	Rot. err.	Pos. err.	Rot. err.
Generalized IoU	13.32	6.42	15.11	12.44
Bounding box	$1.0e^{-7}$	22.87	$1.2e^{-5}$	29.66
Algebraic Vectorized	2.49	21.69	5.05	32.30
Algebraic Frobenius	2.59	22.08	5.04	32.30
Wasserstein	$9.8e^{-4}$	$7.8e^{-4}$	$7.9e^{-4}$	$4.6e^{-4}$
Bhattacharyya	$2.0e^{-2}$	$3.9e^{-3}$	$1.7e^{-2}$	$2.5e^{-3}$
Level Set	$1.0e^{-6}$	$1.0e^{-6}$	$2.9e^{-4}$	$2.7e^{-5}$

TABLE I: Mean position (pixels) and rotation (degrees) errors obtained on the estimated transform for 10000 pairs.

VI. Camera pose refinement

The previously described ellipse-ellipse distances are the main element of the proposed object-based camera pose estimation, as our ellipsoidal models of objects are observed in the image in the form of ellipses. The projection ellipse can be computed by

$$C^* = PQ^*P^T, \tag{13}$$

where P is the camera projection matrix, Q^* is the dual-form matrix of the ellipsoid and C^* is the dual-form matrix of the ellipse.

A. Non-linear minimization problem

Similarly to the traditional structure-based methods, where the reprojection error between points is minimized, our refinement is expressed as a 6-DoF non-linear minimization problem, in which the Euclidean distance between points is replaced by a distance between ellipses, expressed as

$$\hat{R}, \hat{t} = \arg\min_{R,t} \sum_{j=1}^{N_{obj}} \Delta(\mathcal{E}_j, PQ_j^*P^T)^2, \tag{14}$$

where $P = K[R|t]$ with K being the calibration matrix of the camera and $[R|t]$ are its extrinsic parameters which are optimized. $\mathcal{E}_j$ is the j-th ellipse detected in the image, while Q_j is the given dual-form matrix of the corresponding ellipsoid and $\Delta(\cdot)$ is the chosen ellipse-ellipse metric.

B. Initial pose estimation and data association

While an initial camera pose can be obtained by different means (external sensor, odometry, pose prior), we rely on a slightly modified version of the method presented in [26], where the pose is solved inside a RANSAC loop which enumerates the different mapping possibilities, constrained by the objects labels. At each iteration, a minimal set of three ellipse/ellipsoid pairs are selected and the camera pose is computed with the Perspective-3-Point (P3P) algorithm on the ellipses and ellipsoids centers. With only three points, P3P provides four potential solutions, from which the best one is selected based on the overlap (IoU) between the detected ellipses and the reprojected ones. The same IoU criteria is used to select the best camera pose from all the combinations evaluated in the RANSAC. When only two objects are detected in the image, P3P is replaced by the original Perspective-2-Ellipsoid (P2E [26]), which is able to compute the camera pose from only two ellipse/ellipsoid pairs, but under the additional assumption that the camera roll is null. In both cases, the camera pose is estimated by aligning the projection rays passing through the ellipsoids centers with the back-projection rays going through the detected ellipses centers, which is only approximate. Finally, the pairs of ellipse/ellipsoid with a minimum projection overlap of 0.2 are selected as inliers and used in the optimization.

C. Uncertainty estimation

Inspired by the work of Gal [28] on aleatoric uncertainty and similarly to the work of Dong *et al*. [38] on object ellipsoid estimation, we propose a practically efficient method for estimating a predictive uncertainty on the detected ellipses. We then leverage it to weight the contribution of each object in the pose estimation. However, instead of independently modelling each regressed parameter as a univariate Gaussian distribution, we predict a global measure of uncertainty representing the geometric quality of the regressed ellipse. For that, we model the sampling-based distance (Δ) used to train the ellipse prediction network [5] as a univariate Gaussian distribution. The minimisation objective becomes

$$\mathcal{L}_{unc} = \frac{1}{2}\sigma^{-2}\Delta(\mathcal{E}_{pred}, \mathcal{E}_{gt}) + \frac{1}{2}\log\sigma^2. \tag{15}$$

In practice, we train the network to predict the log variance, $\alpha = \log\sigma^2$ to avoid numerical instability when σ is small. In the left part of the loss, σ acts as a loss attenuation, whereas the right part is a regularizer term which should avoid predicting an infinite uncertainty. This loss attenuation is self-learned and allows the network to reduce its loss even in the hardest cases, where the network is not able to predict an ellipse close to the ground truth. To predict this additional value, we simply added a fully connected layer at the end of the Multi-Layer Perceptron (MLP) part of the network. These predicted uncertainties σ_j are then directly integrated in the pose optimization, such that

$$\hat{R}, \hat{t} = \underset{R,t}{\arg\min} \sum_{j=1}^{N_{obj}} \sigma_j^{-1}\Delta(\mathcal{E}_j, PQ_j^*P^T). \tag{16}$$

VII. Experiments and Analysis

A. Implementation details

We performed all the optimizations in this work using the *Broyden-Fletcher-Goldfarb-Shanno* algorithm (*BFGS*) [39]. For the level set metric, we tested different numbers of sampling points, starting from 1600 (40 azimuths and 40 distances). We reduced this number until 24 (6 azimuths and 4 distances) without degrading the accuracy and with an noticeable computation speedup.

B. Ellipses alignment for camera pose refinement

We evaluate the estimated camera poses obtained with the different ellipse-ellipse metrics on the 7-Scenes dataset [40] (*Chess* scene). This scene is composed of 11 objects from 7 categories. We used the sequences 2, 3 and 5 for evaluation and sequences 1, 4 and 6 to build the scene model and train the object detection and ellipse prediction networks. Each sequence includes 1000 frames. We created the scene ellipsoidal model using the method from [22], with only a few manual annotations (a bounding box for each object in at least three images). We fine-tuned Faster R-CNN and followed [5] to obtain 3D-aware elliptic detections of the objects. The initial camera pose estimate and data associations are obtained in a RANSAC loop with either P2E or P3P, as explained in subsection VI-B.

Figure 4 shows the percentage of correctly localized frames over the test sequences with respect to an error threshold on the camera position. We separated the cases by the number of objects detected in the images and used in the optimization. We can first notice that our pose estimation method, including the refinement step, clearly outperforms the direct closed-form solution [26] (named "No refinement"). In terms of metrics used in the optimization, we can see that the level set one achieves the best results. The generalized IoU has also good performance in the case of four and more objects, but is clearly under-performing in images with three objects. On the contrary, QBbox performs well in the cases with three and four objects, but much less well than level set and GIoU starting from five objects. We identify the test frames which produce a significant difference between the costs in Figure 5 and analyze them more precisely in the next section. The pose optimization requires a mean time of $0.26\,\mathrm{s}$ with the level set metric, faster than the two other best performing metrics GIoU ($0.336\,\mathrm{s}$) and QBbox ($0.313\,\mathrm{s}$).

C. Costs behaviour analysis

In this section, we analyse the behaviour of the level set metric and highlight two nice properties. The bottom row of Figure 6 illustrates the case of a detected ellipse (green) significantly smaller than the projected one (red) and, comparing the first and second columns, we can notice that all the metrics favour an alignment of the centers except the generalized IoU, whose cost value remains constant, and the level set metric which results in a lower cost in case of tangency. This behaviour of the level set metric is actually linked to the relative size between the detection and

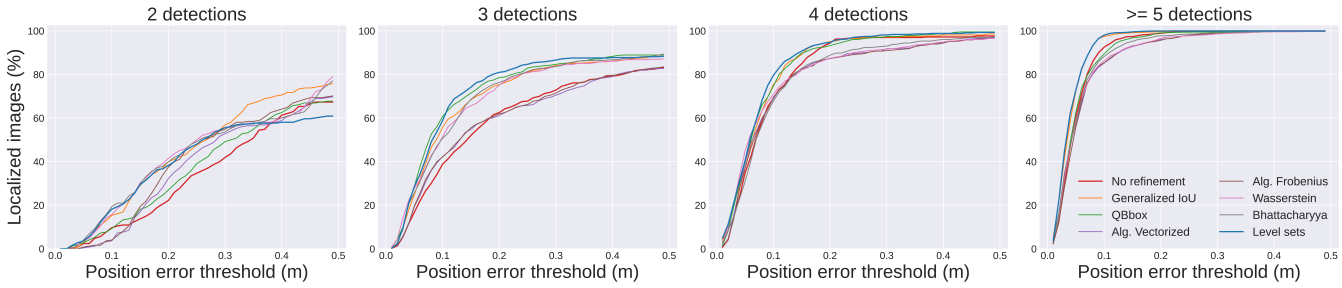

Fig. 4: Percentage of correctly localized images w.r.t. an error threshold on the position. The images are split by the number of objects used in the pose optimization. The results obtained w.r.t an error threshold on the orientation are very similar.

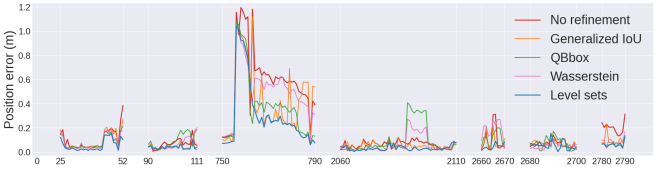

Fig. 5: Position error obtained on the test frames (horiz. axis).

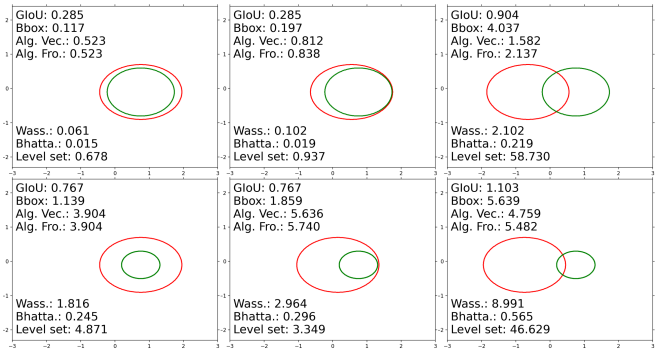

Fig. 6: Ellipse-ellipse costs between a detection ellipse (*green*) and a projection ellipse (*red*).

projection ellipses, favoring centers alignment when the sizes are similar (top row) and tangency as sizes differ (bottom row). More illustrations are available in the attached video. This behaviour proves to be particularly beneficial in the case of partially visible objects, for which aligning the centers does not make much sense and where tangency is more desirable. This is detailed in the next sub-section VII-C.1.

The third column of Figure 6 also shows that the level set cost increases much faster than the other metrics when the ellipses become distant. This induces that this particular metric highly penalizes distant ellipses, which proves to be more efficient. It will be illustrated in sub-section VII-C.2.

1) Partially visible objects: Figure 7 illustrates the case of poor-quality detection for a partially visible object (the bottom chair is slightly outside the image and seen from an uncommon viewpoint) and shows the benefits of using the level set metric.

We can first notice that the Wasserstein distance tends to align the center of the bad chair detection with the projection of its model, at the expense of degrading the alignment of the well detected objects. Actually, the Wasserstein cost

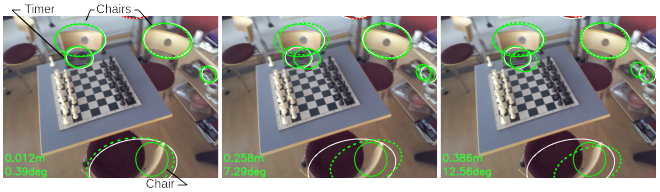

Fig. 7: Estimated pose using level set (left), Wasserstein (center) and QBbox (right). The green solid-line ellipses are the detections and the dashed ones are the objects projections obtained with the estimated camera pose. The green ellipses are used in the pose optimization, the red ones (if any) do not match any detection and the white ones are the objects projections obtained with the ground truth camera pose. The position and orientation errors are written in green.

sums a distance between centers and a difference in shapes (see Equation 7). This explains why the alignment of the bottom chair (with a large size difference) is prioritized compared the top chair and the timer (which have correct sizes). Minimizing the distance between on-image bounding boxes (QBbox) is also problematic for this frame. In fact, QBbox is efficient to deal with poor-quality detections, but only when the missing part is outside the image. In this case, however, the non-detected part of the object is still situated inside the image and the bad detection is also due to the uncommon viewpoint on the object. This induces a high cost for the bottom chair and results in a bad estimated pose. On the contrary, we showed in Figure 6 that the level set metric favours tangency when the detection is significantly smaller than the projection. This results in a much better alignment for the well detected objects and a noticeably better pose accuracy.

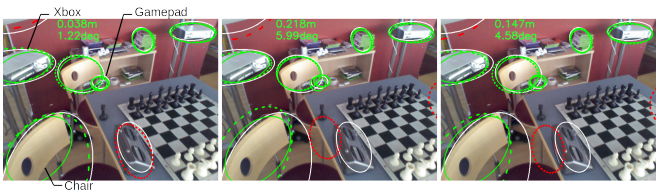

Fig. 8: Estimated pose using level set (left), GIoU (center) and QBbox (right). The color code is the same as in figure 7.

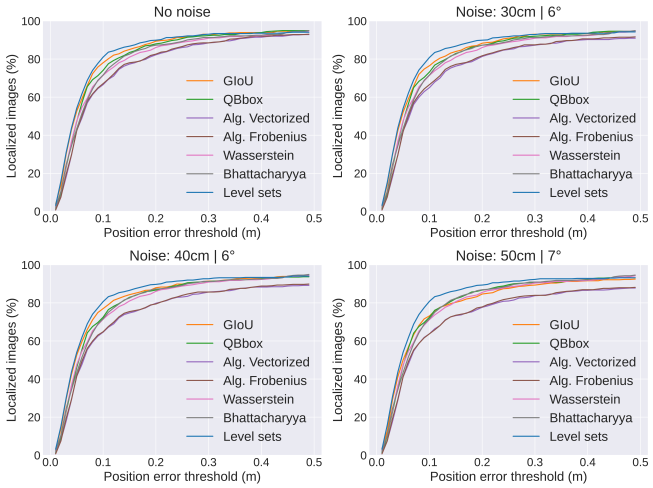

Fig. 9: Percentage of correctly localized images w.r.t. a position error threshold for an increasing level of noise added to the initial pose estimate.

2) Near vs. far objects contributions: Figure 8 compares the results obtained with level set, GIoU and QBBox. Although this frame contains partially visible objects (the bottom chair and the left Xbox), these are not problematic for the selected metrics. Indeed, GIoU does not enforce centers alignment for a smaller detection ellipse (the cost is flat) and QBbox is able to correctly handle the poorly detected objects because the missing parts are situated outside the image. The challenge, here, is more related to adjusting the contribution between objects, especially between the near and far ones. Far objects are more subject to misalignment between their detection and projection ellipses when the pose estimate is erroneous. Consequently, they constitute good anchors for the pose estimation and a strong misalignment should be highly penalized.

Absence of normalization, such as with QBbox, naturally puts more importance on the large objects in the image (i.e., near and/or large objects in the scene). For its part, the generalized IoU is naturally normalized, and thus, tends to equalize the objects contributions. Also, as shown in the third column in Figure 6, this cost increases very slowly when the ellipses move away from each other. On the contrary, the level set metric is also naturally normalized by the chosen embedding function (the ellipse contour always corresponds to the level curve of value 1), but produces a high cost for distant ellipses (see Figure 6). This metric therefore takes more advantage of the far objects. Indeed, we can observe a better alignment between detection and projection ellipses for the gamepad and a better pose accuracy using the level set metric in Figure 8.

D. Convergence analysis

We randomly simulated noisy initial camera poses to evaluate the convergence basins of the different metrics. The results are available in Figure 9 for different levels of noise on the camera position and orientation. We can notice that the level set metric is particularly robust. Wasserstein and Bhattacharrya are only slightly affected by noise, but they globally perform less well. On the contrary, QBbox, GIou and the algebraic distances are more strongly affected by a noisy initial camera pose.

E. Uncertainty-driven pose estimation

We first show the reliability of the predicted uncertainty on crop images of the objects. We added an increasing level of occlusion to the image to simulate a low-quality ellipse prediction. As expected, the estimated uncertainty increases with the occlusion ratio (see Figure 10). We then evaluate the benefits offered by the uncertainty guidance in the camera pose refinement. Globally, the weighting proposed in equation 16 improves the pose accuracy (see Figure 11). Some particularly interesting cases are shown in Figure 12. We can first notice that the estimated pose (without uncertainty) is better using the level set metric compared to Wasserstein. This can be explained by the right chair which is slightly outside the image and whose detection is smaller than expected. As explained in section VII-C.1, the level set metric handles this case better than the Wasserstein distance which pushes more towards centers alignment. Additionally weighting the objects costs by confidence still further improves the estimated pose for both metrics. We can observe that projections of the right chair is more loosely constrained by the detection, which also allows the more confident objects to reach a better alignment (the Xbox and the left chair). Figure 11 shows the global improvement of using uncertainty on the entire test sequence. In particular, we can notice that the benefit obtained for the Wasserstein distance is slightly larger than for the level set or QBbox metrics, probably because these two metrics naturally deal better with partially visible objects.

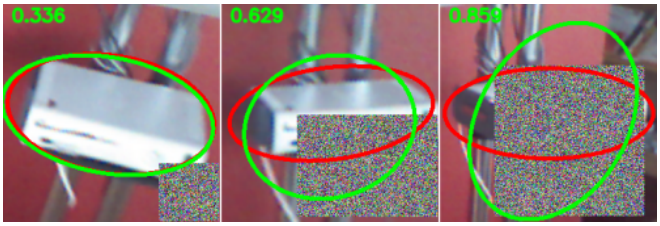

Fig. 10: Predicted (green) and ground truth (red) ellipses for an increasing level of occlusion. The predicted uncertainties are written in the top-left corners.

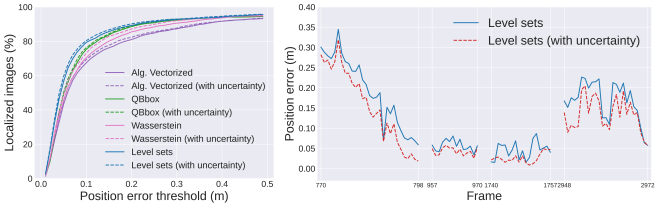

Fig. 11: Comparison between the standard version and the uncertainty-driven pose optimization.

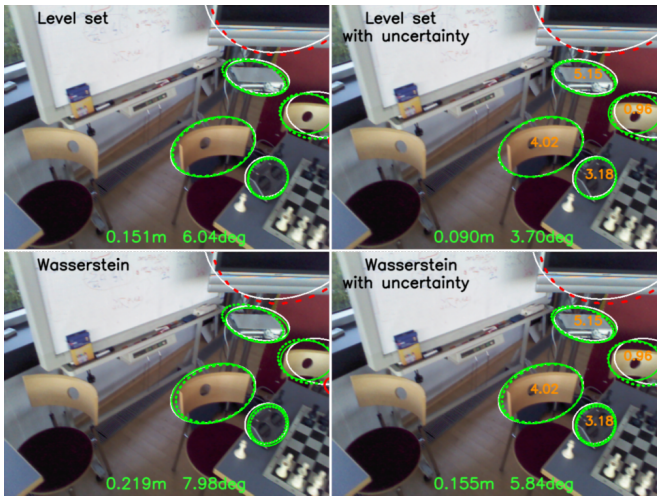

Fig. 12: Comparison between the standard (left) and the uncertainty-driven estimation (right). The level set metric is used in the top row and the Wasserstein distance in the bottom row. The color code is the same as in Figure 7. Additionally, the objects weighting are shown in orange.

VIII. Conclusion

We proposed a camera pose estimation method based on objects, which aims at aligning their detections in the image with the projection of their ellipsoidal 3D models. We showed that establishing a cost between ellipses is not as straightforward as it is for points and that a metric based on level sets has good convergence properties. In particular, it helps to deal with the challenging case of partially visible objects. Finally, we proposed a practical method for predicting uncertainty in a deep neural network, which reflects the quality of the predicted objects ellipses and showed how the weighting of each object contribution can help to improve the accuracy of the refined pose.

References

[1] S. Yang and S. A. Scherer, "Cubeslam: Monocular 3-d object SLAM," *IEEE Trans. Robotics*, 2019.

[2] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented SLAM," *IEEE Robotics Autom. Lett.*, 2019.

[3] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: simultaneous localisation and mapping at the level of objects," in *CVPR*, 2013.

[4] L. Hu, W. Xu, K. Huang, and L. Kneip, "Deep-slam++: Object-level RGBD SLAM based on class-specific deep shape priors," *CoRR*, 2019.

[5] M. Zins, G. Simon, and M. Berger, "3d-aware ellipse prediction for object-based camera pose estimation," in *3DV*, 2020.

[6] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o(n) solution to the pnp problem," *International Journal of Computer Vision*, 2009.

[7] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *CVPR Workshops*, 2018.

[8] P. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *CVPR*, 2020.

[9] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *CVPR*, 2021.

[10] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *ICCV*, 2015.

[11] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *ICCV*, 2017.

[12] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. W. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *CVPR*, 2013.

[13] E. Brachmann and C. Rother, "Learning less is more - 6d camera localization via 3d surface regression," in *CVPR*, 2018.

[14] M. Bui, S. Albarqouni, S. Ilic, and N. Navab, "Scene coordinate and correspondence learning for image-based localization," in *BMVC*, 2018.

[15] P. Sarlin, *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *CVPR*, 2021.

[16] L. Yang, Z. Bai, C. Tang, H. Li, Y. Furukawa, and P. Tan, "Sanet: Scene agnostic network for camera localization," in *ICCV*, 2019.

[17] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, 2020.

[18] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[19] P. Weinzaepfel, G. Csurka, Y. Cabon, and M. Humenberger, "Visual localization by learning objects-of-interest dense match regression," in *CVPR*, 2019.

[20] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *ECCV*, 2020.

[21] P. Li, T. Qin, and S. Shen, "Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving," in *ECCV*, 2018.

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

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

[24] P. Gay, V. Bansal, C. Rubino, and A. D. Bue, "Probabilistic structure from motion with objects (psfmo)," in *ICCV*, 2017.

[25] M. Hosseinzadeh, Y. Latif, T. Pham, N. Sünderhauf, and I. D. Reid, "Structure aware SLAM using quadrics and planes," in *ACCV*, 2018.

[26] V. Gaudillière, G. Simon, and M. Berger, "Perspective-2-ellipsoid: Bridging the gap between object detections and 6-dof camera pose," *IEEE Robotics Automation Letters*, 2020.

[27] M. Abdar, *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, 2021.

[28] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Annual Conference on Neural Information Processing Systems*, 2017.

[29] F. Gustafsson, M. Danelljan, and T. Schön, "Evaluating scalable bayesian deep learning methods for robust computer vision," in *CVPR*, 2020.

[30] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[31] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *ECCV*, ser. Lecture Notes in Computer Science, 2014.

[32] M. Kristan, *et al.*, "The ninth visual object tracking VOT2021 challenge results," in *ICCV Workshops*, 2021.

[33] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *CVPR*, 2019.

[34] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *ICML*, 2017.

[35] S. Pan, S. Fan, S. W. K. Wong, J. V. Zidek, and H. Rhodin, "Ellipse detection and localization with applications to knots in sawn lumber images," in *WACV*, 2021.

[36] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, 1943.

[37] B. Rosenhahn, T. Brox, D. Cremers, and H. Seidel, "A comparison of shape matching methods for contour based pose estimation," in *Combinatorial Image Analysis, 11th International Workshop*, 2006.

[38] W. Dong and V. Isler, "Ellipse regression with predicted uncertainties for accurate multi-view 3d object estimation," *CoRR*, 2021.

[39] M. Avriel, *Nonlinear Programming: Analysis and Methods*. Dover Publications, 2003.

[40] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time RGB-D camera relocalization," in *IEEE ISMAR*, 2013.