Bachelor Project

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

# Temporal Consistency for Object Pose Estimation from Images

**Vojtěch Přibáň**

**Supervisor: Ing. Vladimír Petrík, Ph.D.**
**May 2024**

# BACHELOR‘S THESIS ASSIGNMENT

## I. Personal and study details

Student's name: **Přibáň Vojtěch** Personal ID number: **507655**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

## II. Bachelor’s thesis details

Bachelor’s thesis title in English:

**Temporal Consistency for Object Pose Estimation from Images**

Bachelor’s thesis title in Czech:

**Časová konzistence odhadu polohy a orientace objektu z obrázků**

Guidelines:

1. Analyze temporal smoothness and consistency of the objects for which pose was estimated with state-of-the-art pose estimation methods, e.g. [1, 2].
2. Use a smoothing and mapping approach [3] to enforce the temporal smoothness of the predictions.
3. Compare the accuracy of the predictions (SE3 distance between the poses) with and without the temporal smoothness either on BOP datasets [4] (e.g., YCB-V [5]) or on simulated data rendered by Blender

Bibliography / sources:

[1] Labbé, Y., Manuelli, L., Mousavian, A., Tyree, S., Birchfield, S., Tremblay, J., Carpentier, J., Aubry, M., Fox, D. and Sivic, J., 2022. Megapose: 6d pose estimation of novel objects via render & compare. arXiv preprint arXiv:2212.06870.
[2] Labbé, Y., Carpentier, J., Aubry, M. and Sivic, J., 2020. Cosypose: Consistent multi-view multi-object 6d pose estimation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16 (pp. 574-591). Springer International Publishing.
[3] Dellaert, F., 2012. Factor graphs and GTSAM: A hands-on introduction. Georgia Institute of Technology, Tech. Rep, 2, p.4.
[4] Sundermeyer, M., Hodaň, T., Labbe, Y., Wang, G., Brachmann, E., Drost, B., Rother, C. and Matas, J., 2023. Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2784-2793).
[5] Xiang, Y., Schmidt, T., Narayanan, V. and Fox, D., 2017. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199.

Name and workplace of bachelor’s thesis supervisor:

**Ing. Vladimír Petrík, Ph.D. Intelligent Machine Perception CIIRC**

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **26.01.2024** Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

| Ing. Vladimír Petrík, Ph.D. | prof. Dr. Ing. Jan Kybic | prof. Mgr. Petr Páta, Ph.D. |
|---|---|---|
| Supervisor’s signature | Head of department’s signature | Dean’s signature |

## III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____ Date of assignment receipt

_____ Student's signature

# Acknowledgements

I would like to thank my supervisors Vladimír and Mederic for their involvement and patience.


# Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university thesis.

I further declare that the artificial intelligence used for grammar correction and text reformulation (ChatGPT) was used in accordance with the Guidelines for the use of Artificial Intelligence at CTU.

Prague 24. 5. 2024

. . . . . .

# Abstract

Estimating object spatial pose from an input image is one of the approaches to compute feedback signal for feedback-based robot control, for example, for model predictive control based grasping. Most of the existing pose estimation methods compute object pose in a per-frame manner, ignoring the temporal consistency. In this work, we propose using probabilistic filtering to achieve the temporal smoothness of object pose estimation and to filter out outliers that are predicted by state-of-the-art pose estimation methods. We show that enforcing temporal smoothness and filtering outliers improves the standardized pose estimation benchmarks. We experimentally validated the stability of the proposed approach for a feedback-based robot control task in which the object was tracked by the camera attached to the Franka Emika Panda robot.



**Keywords:** pose estimation; SLAM; factor graph; robotics.



**Supervisor:** Ing. Vladimír Petrík, Ph.D. Intelligent Machine Perception CIIRC



# Abstrakt

Odhad polohy objektu v prostoru z vstupního obrazu je jedním z přístupů pro výpočet zpětnovazebního signálu pro řízení robota založeného na zpětné vazbě, například pro uchopení založené na modelově prediktivním řízení. Většina stávajících metod odhadu polohy objektu počítá polohu objektu po jednotlivých snímcích a ignoruje tak časovou konzistenci. V této práci navrhujeme použití pravděpodobnostního filtrování k dosažení časové konzistence odhadu polohy objektu a k odfiltrování chybných měření, které jsou získávány nejmodernějšími metodami odhadu polohy. Demonstrujeme, že vynucení časové konzistence a odfiltrování odlehlých hodnot zlepšuje standardizované referenční testy odhadu polohy. Experimentálně jsme ověřili stabilitu navrhovaného přístupu pro úlohu řízení robota založeného na zpětné vazbě, při níž byl objekt sledován kamerou připojenou k robotu Franka Emika Panda.



**Klíčová slova:** Odhad polohy a orientace; SLAM; faktorový graf; robotika.

**Překlad názvu:** Časová konzistence odhadu polohy a orientace objektů z obrázků

# Contents

# Chapter 1

# Introduction

Estimating object spatial pose from a monocular camera has made significant progress in recent years *e.g.*, by using the render-and-compare approach [LCAS20], [LMM+22]. Our motivation is to use these pose estimations for feedback-based robot control, for example, for visual tracking or hand-over from human to robot. However, pose predictions are often inconsistent in time: some estimates are missing or outliers occur, as shown in Fig. 1.1. These inconsistencies have a significant impact on the safety and robustness of feedback robot control, as incorrect pose predictions can lead to unstable behavior. For example, incorrect pose estimation that places an object suddenly 10 cm away or incorrectly estimates that orientation suddenly changes by 180 degrees due to symmetries, can cause the controller to generate incorrectly large desired robot torques leading to dangerous motion.

To address these issues, in this thesis, we propose using probabilistic filtering, developed for Simultaneous Localization and Mapping (SLAM) applications [GKSB10], to track the motion of objects based on a stream of images captured by a camera mounted on a robot arm. The proposed approach allows us to maintain a probabilistic, temporally consistent dynamic world model containing the poses of the objects and to associate the measurements with the objects in the world iteratively. Temporally consistent poses are predicted from the world model and are safe to use in the robot control loop, as we demonstrate by feedback-based object tracking performed by the Franka Emika Panda robot. The probabilistic filtering approach allows us to address the following challenges:

- **Missing object detections** are predicted from the model (by using the motion model) to maintain temporal consistency;
- **Predicted outliers** are not associated with the existing track of the object. Instead, a new track is created for the outlier and it is predicted only if the same outliers are predicted consistently. This preserves the temporal consistency of the tracks;
- **Multiple instances** of the same objects are tracked separately in the world model so that the robot knows which instance is tracked; and
- **Discrete object symmetries** are also tracked separately to predict temporally consistent poses.


**Figure 1.1: Mustard bottle object pose estimation from images.** The plot shows the angular distance between the estimated pose and fixed reference frame. Objects are static, and therefore the distance should be constant. Red shows the per-frame estimations computed by CosyPose [LCAS20]. The filtered prediction computed by our method is shown in green. The corresponding red and green contours in the images were computed by reprojection of the object based on the estimated pose. In the ideal scene, the red and green predictions overlap as shown in the first frame. However, in more difficult scenarios (2nd and 3rd frames) the per-frame estimate is not robust and would cause instability in the control. Our approach (green) is consistent even in these challenging scenarios at the cost of delayed start of the tracking.

This thesis has the following contributions:

- we present a probabilistic filtering approach for spatial object pose tracking that is temporally consistent and therefore suitable for feedback-based robot control;
- we evaluate our approach on a real video dataset with static objects and on synthetically rendered dataset with static and dynamic objects - we outperform per-frame pose estimation on all datasets;
- we demonstrate the proposed filtering approach for a robot object tracking application with the Franka Emika Panda robot - we experimentally show that our approach leads to robust tracking in situations where per-frame estimation does not;
- we made the code open source at `www.github.com/priban42/SAM_pose`

# Chapter 2

# Related work

## 2.1 Object pose estimation.

Model-based object pose estimation is one of the core computer-vision challenge with a wide range of applications for robotics and AR/VR [Lep20, HSD+20]. The problem is most often decomposed in two stages: 2D image detection, which provides object labelled bounding boxes and masks, followed by a pose estimation for each individual detection. Deep learning-based methods are nowadays dominating standardized benchmarks for both steps [HSL+24]. A more recent challenge is to improve generalizability to objects not seen during training, both for detection [NGP+23] and pose estimation [LMM+22, NGSL24, ÖLT+23, WYKB23]. Used by some of the leading methods, the "render-and-compare" approach [LWJ+18, LCAS20, LMM+22, WYKB23] refines an initial guess by predicting object pose updates. Working with videos, this method has been shown to be competitive with the state of the art in single-view object pose tracking [SPS+22, WMRB20, DMX+21, WYKB23].

However, the single-view pose estimation problem is inherently challenging for several reasons. For RGB only methods, the geometry of pinhole projection creates a high uncertainty in the camera to object distance. Poses of object can be ill-defined due to object symmetries (*e.g.*, a bottle) or partial occlusion (*e.g.*, a cup with hidden handle), which can be taken into account during model evaluation [HMO16]. Higher uncertainty may also occur in real-world experiments, *e.g.* if the model has not been trained with enough augmentation technics [KG17]. With model-based object pose estimators performance improving rapidly, we propose a methodology that leverages off-the-shelf object pose estimators for fast and robust tracking.

## 2.2 Multiview object pose estimation.

In robotics context, it is not uncommon to have access to a multi-camera setup [PK15] or to have a camera mounted on the robot [KC+02]. This setup can be leveraged by aggregating information across views and time to create a consistent estimate of both the camera/objects poses and of the objects shapes. Commonly used representation include parametric surfaces [NMS18,

YS19, LDC+21, LD22], volume based representations [MCB+18, WSJ+20], latent codes [SWD20, LSL+21, LD22].

For many practical and industrial scenarios, it may be assumed that models can be collected before starting the mapping process. SLAM++ [SMNS+13] is the first depth-based object SLAM system and formulates estimation using a probabilistic pose graph back-end. SimTrack [PK15] proposes a tightly RGB-D integrated system for robot/object pose detection and tracking. Some authors propose to tackle directly the inherent pose ambiguity of image based pose estimation. [FHD+21] explicitly trains a single view model that predicts a set of pose hypothesis that are resolved over different views using a max-mixture formulation. [MGZ+22] proposes to fuse probabilistic keypoint predictions, using the known symmetries of the object. These methods require to train a dedicated "front-end" which does not clearly shows a potential for generalization. Drawing inspiration from structure from motion pipelines, CosyPose [LCAS20] addresses the single-view pose data association problem by designing a symmetry aware RANSAC followed by bundle adjustment [FB81] and is agnostic to the single view pose estimator. We propose to address the pose the object pose ambiguities by tracking simultaneously the multiple symmetries modes of the scene objects.

## 2.3 Temporally consistent moving object estimation.

In previously mentioned SLAM-like system, the objects are assumed to be static in the environment. A natural but challenging extension to these methods is allow the multi-object live reconstruction with to dynamic objects [RA17, RBA18, XLT+19, LRR21, XDL22]. To improve the geometric consistency of the scene, Dynamic SLAM [HZMI20] is able to detect sparse landmarks moving with the same underlying rigid body motion model and include this information in a factor graph-based optimization. Motion models can also provide necessary regularization to a filter-based object pose tracker, either by penalizing large pose updates [SPS+22] or by estimating a higher order state like the object twist [IWC+16]. We propose to estimate the pose and twist of multiple objects using a factor graph formulation.

# Chapter 3

# Methodology

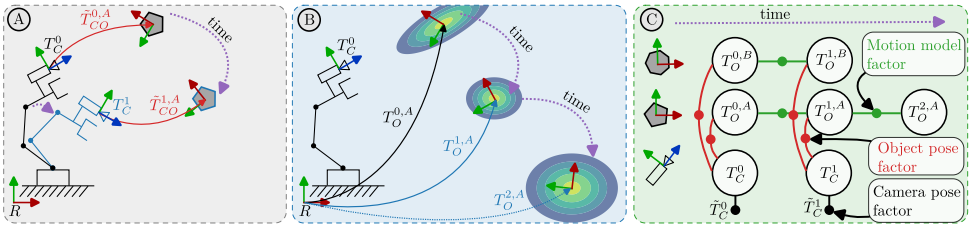


**Figure 3.1:** Our goal is to estimate the poses of objects in time with respect to the reference frame $R$ as shown in figure A. To achieve this, we use measurements in time (denoted $k$) of the camera pose $\tilde{T}_C^k$ and the object pose $\tilde{T}_{CO}^{k,A}$, where $A$ is the label of the object. Both objects and the robot are moving in time, as illustrated by the purple arrow. Our approach maintains the probabilistic world representation of the object poses as visualized in figure B, where the ellipsoids represent the position uncertainty. This uncertainty is used to filter outliers and to predict only confident poses. The map is maintained through the factor graph shown in figure C, where green factors represent the motion model, red factors represent the observations of the object pose in the camera, and black factors represent the camera pose computed by forward kinematics. Note that multiple objects could be tracked simultaneously, as shown by the two-object factor graph in the figure C. Thanks to the motion model, the poses of the objects can be extrapolated in the future (figures B and C) to resolve missing measurements.

## 3.1 Problem formulation

Our goal is to track an object $SE(3)$ pose with a moving calibrated camera rigidly attached to a robot end effector as shown in Fig. 3.1-A. To achieve that, we need to estimate the pose of $i$-th object $T_O^{k,i} \in SE(3)$ at time $k$. The poses are expressed in the common reference frame $R$. Inputs to our method are the stream of images captured by the camera and the corresponding camera poses measured by the forward kinematics of the robot. These measurements are fused in a single probabilistic estimation problem that finds a optimal trajectory of camera/objects poses, as shown in Fig. 3.1-B. The details are given next.

## 3.2 Factor graph

We formulate the tracking task as a weighted nonlinear least squares problem following the factor graph approach [DK+17]. Under the assumption of conditionally independent measurements corrupted by Gaussian noise, the optimal sequence of object and camera poses is obtained by solving:

$$\chi^* = \arg\min_{\chi} \underbrace{\sum_{k=\tau-H}^{\tau} \left\| \mathbf{r}_C^k \right\|^2_{\mathbf{\Sigma}_C}}_{\text{camera pose factors}} + \underbrace{\sum_{i=1}^{N} \sum_{k=\tau-H}^{\tau} \delta^{k,i} \left\| \mathbf{r}_O^{k,i} \right\|^2_{\mathbf{\Sigma}_O}}_{\text{object pose factors}} + \underbrace{\sum_{i=1}^{N} \sum_{k=\tau-H+1}^{\tau} \left\| \mathbf{r}_M^{k-1:k,i} \right\|^2_{\mathbf{\Sigma}_M}}_{\text{motion models factors}}, \tag{3.1}$$

where index $i$ iterates over all $N$ objects, index $k$ represents time on the fixed time horizon $H$ from the time of the last measurement $\tau$, $\mathbf{r}_X$ is the vector of residual errors weighted by covariance matrix $\mathbf{\Sigma}_X$ for $X \in \{C, O, M\}$, representing the camera $C$, the object $O$, and motion model $M$. Term $\delta^{k,i}$ is a binary "occlusion" term that accounts for a missing measurement of object $i$ in frame $k$, *e.g.*, caused by occlusion or significant motion blur. We minimize over the set of variables $\chi$, which consists of object and camera poses in time, denoted as $T_O^{k,i}$ for object $i$ at time $k$ and $T_C^k$ for camera pose at time $k$. The intuition is that

- the camera pose factors regularize the camera pose to stay close to the pose measured by robot's forwards kinematics;
- the object pose factors regularize the object pose to stay close to the measured pose w.r.t. the camera, and
- the motion model factor capture the motion of the object, *i.e.* the change of the pose and its uncertainty in time.

The weight of the individual factors is not equal; for example, the pose of the camera is measured more accurately than a pose of the object. To account for this inequality, the residuals are scaled by covariance matrices that encapsulate our confidence in the measurements. The residual errors in (3.1) depend on the optimized variables. The computation of the residuals and the corresponding covariances is described next.

## 3.3 Camera pose factor

We assume extrinsically calibrated camera and therefore the camera pose residual can be computed by comparing the $SE(3)$ distance between the estimated value and the corresponding measurement, *i.e.*, $\mathbf{r}_C^k = \mathrm{Log}((T_C^k)^{-1} \tilde{T}_C^k)$,

where symbol ˜ represents the measurement, here computed by forward kinematics, and Log is logarithm mapping from $SE(3)$ group [SDA21]. The covariance of the camera pose factor is assumed to be diagonal in the form $\mathbf{\Sigma}_C = \text{diag}(\sigma^2_{Ct}, \sigma^2_{Ct}, \sigma^2_{Ct}, \sigma^2_{Cr}, \sigma^2_{Cr}, \sigma^2_{Cr})$, where $\sigma^2_{Ct}$ represents the translational variance and $\sigma^2_{Cr}$ is the rotational variance. These variances were estimated during the camera extrinsic calibration process.

## 3.4 Object pose factor

To estimate the pose of the object from the input RGB image we use CosyPose [LCAS20]. CosyPose uses pre-trained Mask-RCNN [HGDG17] to detect objects of interest in the image together with the object identity labels. For each image, the render-and-compare strategy is used to estimate the spatial pose of the object in the camera frame based on the 3D mesh retrieved from the database based on the predicted object identity labels. The residual error for the $i$-th object in the $k$-th frame (time) is computed as $\mathbf{r}^{k,i}_O = \text{Log}((T^{k,i}_O)^{-1} T^k_C \tilde{T}^{k,i}_{CO})$, where $\tilde{T}^{k,i}_{CO}$ is the pose of the $i$-th object predicted by the CosyPose from the input image, $T^{k,i}_{CO}$ is the estimated temporally consistent pose of the object and $T^k_C$ is the estimated temporally consistent pose of the camera.

However, the set of predictions from the CosyPose is not a one-to-one mapping of our set of variables. To be able to compute the object pose residual errors, we need to resolve data association between the variables and measurements. This is done as follows. First, we select all variables that correspond to the predicted object label. From this set of variables, we choose the closest one based on the Mahalanobis distance considering the estimated covariance of the measurement. If the distance is below the manually specified threshold, denoted $\tau_{\text{outlier}}$, we associate the measurement with the variable by creating a corresponding factor in the graph. Otherwise, a new variable is created. This approach enables us to filter out outliers and track multiple instances of the same object class.

We observed that the covariance of the CosyPose prediction depends on the object size in the image space and that the uncertainty is higher in the direction of the ray that points from the camera towards the object center. This is caused by ambiguity in depth estimation, where small changes in the depth of the object may have only a small effect on the visual appearance of the object. Therefore, we define the object translation covariance model to deal with this increased uncertainty along the depth direction. In particular, we define a new coordinate frame $C'$, whose $z$-axis points towards the object. The translation covariance $\mathbf{\Sigma}_{Ot}$ of the object pose measurement is defined in the $C'$ frame and we transform it into the object frame $O$ as: $\mathbf{\Sigma}_{Ot} = R_{OC'} \mathbf{\Sigma}_{C't} R^T_{OC'}$, where $R_{OC'}$ is the rotation matrix that rotates vector from frame $C'$ to the frame $O$. The translation object covariance matrix in the $C'$ frame is defined as $\mathbf{\Sigma}_{C't} = \text{diag}(\sigma^2_{C'xy}(n_{\text{px}}), \sigma^2_{C'xy}(n_{\text{px}}), \sigma^2_{C'z}(n_{\text{px}}))$, where the individual variances depend on the number of object pixels observed in the image $n_{\text{px}}$. We visualize the covariance model in Fig. 3.2. The rotational variance in the object

frame is defined to be diagonal: $\mathbf{\Sigma}_{Or} = \mathrm{diag}(\sigma^2_{Or}(n_{\mathrm{px}}), \sigma^2_{Or}(n_{\mathrm{px}}), \sigma^2_{Or}(n_{\mathrm{px}}))$. The variances $\sigma^2_{C'xy}, \sigma^2_{C'z}$, and $\sigma^2_{Or}$ are estimated on the pose estimation dataset as shown in the experiment section. The object covariance $\mathbf{\Sigma}_O$ is composed of translational and rotational covariance assuming zero correlation between them.

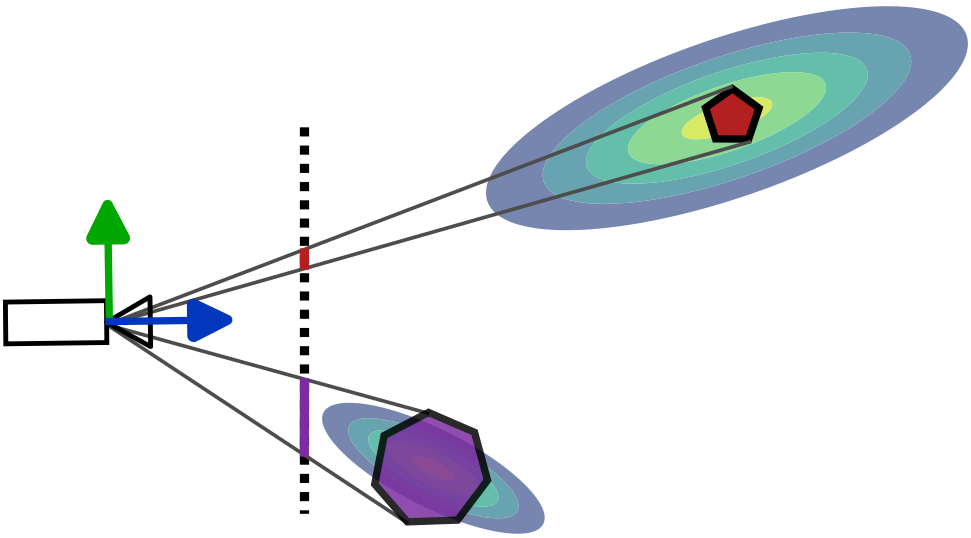

**Figure 3.2:** The visualization of the translation covariance model for the object detections. Consider two objects (red and purple) whose projection on the image plane (dotted line) is shown in red and purple, respectively. The size of the covariance ellipsoid depends on the size of the object in the image plane. The uncertainty is higher in the direction of ray that points towards the object, mitigating the fact that the depth estimation is more difficult from monocular measurements.

## 3.5 Motion model factor

Motion model predicts the motion of the object in time. We decoupled translation and rotation motion and compared two methods for motion prediction:

- constant pose, and
- constant velocity.

Our implementation of motion prediction, explained in the next chapter, allows us to predict the motion of the object based on the constant higher-order derivative assumption as well. However, we limit our experimentation to the maximum first-order derivative and leave the higher-order derivatives to future work. In this section, we also limit ourselves to the zero- and first-order derivatives for clarity of the method presentation.

The constant pose model for object $i$ residual is defined as $\mathbf{r}_M^{k-1:k,i} = \mathrm{Log}((T_O^{k-1,i})^{-1} T_O^{k,i})$ with diaganoal covariance matrix

$\mathbf{\Sigma}_M = \text{diag}(\sigma^2_{Mt}, \sigma^2_{Mt}, \sigma^2_{Mt}, \sigma^2_{Mr}, \sigma^2_{Mr}, \sigma^2_{Mr}) \cdot \Delta t$, where the translation and rotation variances $\sigma^2_{Mt}$ and $\sigma^2_{Mr}$ are chosen manually, $\Delta t$ denotes the time elapsed from the previous detection of the object $i$. With this motion model, the object pose in the world model will remain constant and its uncertainty will increase over time if no new measurements are available.

The constant velocity motion model establishes the factor on estimated derivatives of translation and rotation, from which the pose is computed via integration. Therefore, the set of variables in Eq. (3.1) is extended with the derivatives for each object and time stamp. The residual is computed as $\mathbf{r}_M^{k-1:k,i} = \begin{pmatrix} \boldsymbol{v}^{k,i} - \boldsymbol{v}^{k-1,i} & \boldsymbol{\omega}^{k,i} - \boldsymbol{\omega}^{k-1,i} \end{pmatrix}^\top$, where $\boldsymbol{v}^{k,i}$ and $\boldsymbol{\omega}^{k,i}$ represent the time derivatives of translation and rotation for the $i$-th object at time $k$. The covariance remains diagonal with constant variances for translation and rotation defined manually. With this motion model, the object pose evolves based on the estimated velocity, and the uncertainty increases over time in the absence of new measurements. The higher order derivatives (*e.g.* acceleration, jerk) for the motion model could be used in the same spirit.

## 3.6 Predictions from the world model

Defining all the factors and solving Eq. (3.1) will give us probabilistic world model of all objects and camera poses in time. We solve the optimization for each new measurement in an iterative manner. However, the probabilistic world model will also contain outliers or objects that are no longer visible in the current view. To filter them, we predict only the poses whose volume of the estimated uncertainty ellipsoid is below the manually specified thresholds $\tau_{\text{pred\_t}}, \tau_{\text{pred\_r}}$. If there are more identical labels that satisfy the above thresholds and whose translation distance is lower than 50 mm, we predict only the pose with a lower volume of covariance ellipsoid, *i.e.* the more confident track.

# Chapter 4

# Implementation

The method is implemented in Python 3 with the use of the GTSAM [DC22] Python wrapper. The aim of the implementation is to design the code suitable for near-real-time applications, such as robot manipulation tasks. To achieve this goal, the following conditions need to be met:

- The solve time of a frame should not exceed the inference time of CosyPose [LCAS20].
- The solve time should not increase indefinitely during the tracking.
- The current refined poses should be asynchronously retrievable at any time $k$ without the need to update the factor graph every time.

This chapter describes the approaches used to satisfy these conditions.

## 4.1 Limiting factor graph solve time

Finding an optimal solution of a factor graph is a task whose cost generally increases with the number of factors and variables included in the graph. The initial idea was to utilize marginalization, already implemented in the GTSAM's ISAM2 solver class. However, this approach turns out not to be applicable to our structure of the factor graph; the variables and factors were not removed as expected. To avoid this issue, we chose to explore an alternative approach to achieve near-real-time performance.

Another approach is to construct a factor graph from only the last $n$ measurements, thereby implementing a sort of sliding window. This satisfies the condition that the solve time does not increase indefinitely, as the number of factors and variables is limited by the size $n$ of the sliding window. For a sufficiently small $n$, the first condition is also met. However, an issue arises because the constructed graph has to be solved all at once for every new frame (batch of pose estimates). This significantly restricts the size of the window $n$.

Our proposed approach uses two asynchronous ISAM2 solvers, as depicted in Fig. 4.1. New pose estimates are added to both of these solvers. One of the solvers is designed to always hold more factors so that the estimated poses

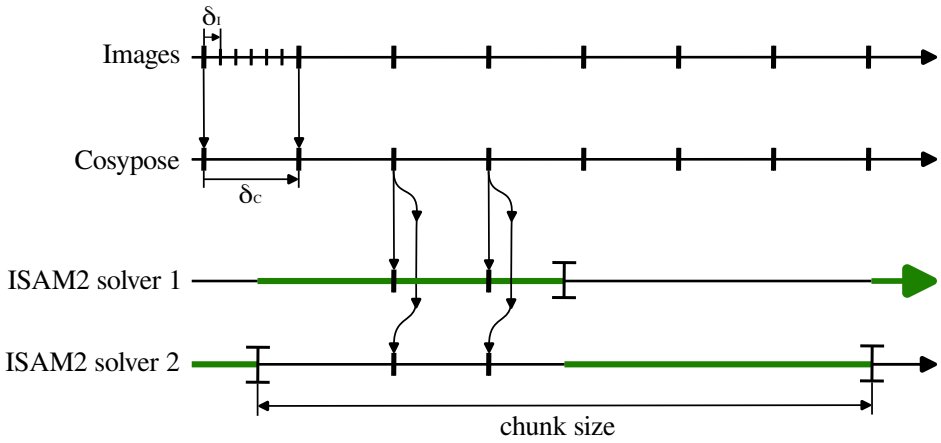


**Figure 4.1:** The first row depicts a stream of images with a typical delay between images, denoted as $\delta_I$ being 33 ms. The second row illustrates the stream of CosyPose object pose estimates, with a typical delay between estimates $\delta_C$ being 200 ms. Therefore, only some of the images are processed. The third and fourth rows illustrate that all of the pose estimates are employed in both solvers. The alternating green line indicates which solver is currently being utilized to retrieve refined pose estimates. The chunk size defines the maximum number of CosyPose estimates used in a solver before it is reinitialized.

can be retrieved from this solver. Once a certain chunk size $m$ is exceeded (where chunk size $m$ refers to the number of frames from which the detections were added to the graph), the solver instance is deleted and replaced with a new instance, with additional prior factors on the relevant variables retrieved from the deleted solver.

This method has resulted in a significantly faster solve time. Additionally, it provides the added benefit of an effectively infinite window size $n$, as the information gathered in the old measurements is carried over in the prior factors when initializing the solver.

## 4.2 Asynchronous pose retrieval

The goal is to simulate the behavior illustrated in Fig. 4.2. Once an optimal solution for a factor graph consisting of multiple measurements is calculated, an estimate of an object's latest state can be retrieved. The state includes its pose, covariance, and, depending on the user's choice, linear and angular velocity, its covariance, and higher-order derivatives up to a chosen $d_{\max}$. The chosen $d_{\max}$ determines the topology of the factor graph, as shown in Fig. 4.3.

While this pose and covariance could be used as the final result, updating the graph every time an object pose is to be retrieved is impractical due to the latency introduced by the solution time and the introduction of new variables and factors to the graph, which decreases performance. Additionally, numerical instability has been observed for cases where no new measurements are made for some period of time, as shown in Fig. 4.4. Therefore, a different

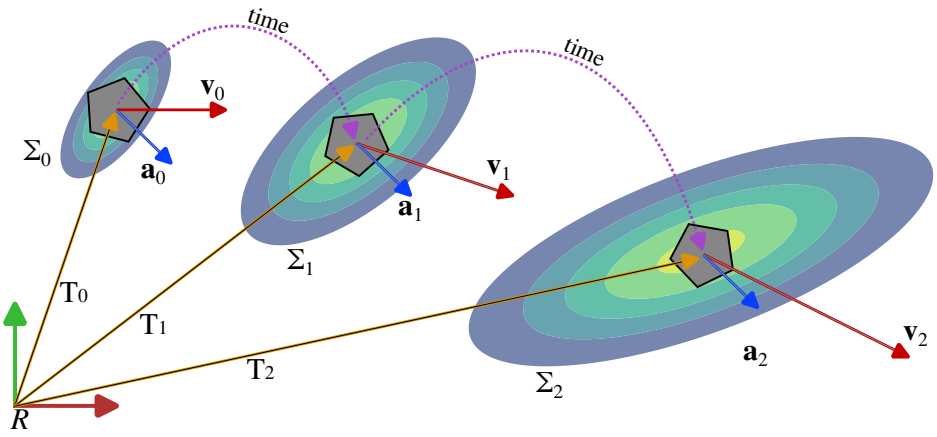


**Figure 4.2:** The figure illustrates a simplified analogy of pose extrapolation. In this scenario, the motion model assumes an initial state comprising a pose $T_0$, velocity $\mathbf{v_0}$, pose covariance $\mathbf{\Sigma}_0$ and constant acceleration $\mathbf{a_0} = \mathbf{a_1} = \mathbf{a_2}$ all defined in the global reference frame $R$. Over time, the state evolves, altering the pose and scaling the covariance matrix.

approach was chosen. Instead of relying solely on the retrieved poses, the state retrieved at time $t_0$ is integrated to retrieve the pose at time $t_0 + \Delta t$ using a closed-form formula:

$$\begin{aligned}
[\mathbf{p}_{t_0+\Delta t}, \boldsymbol{\theta}_{t_0+\Delta t}] &= [\mathbf{p}_{t_0}, \boldsymbol{\theta}_{t_0}] + \int_{t_0}^{t_0+\Delta t} [\mathbf{v}_{t_0}, \boldsymbol{\omega}_{t_0}]\, dt + \iint_{t_0}^{t_0+\Delta t} [\mathbf{a}_{t_0}, \boldsymbol{\alpha}_{t_0}]\, dt + \dots \\
&= [\mathbf{p}_{t_0}, \boldsymbol{\theta}_{t_0}] + \Delta t \cdot [\mathbf{v}_{t_0}, \boldsymbol{\omega}_{t_0}] + \frac{\Delta t^2}{2} \cdot [\mathbf{a}_{t_0}, \boldsymbol{\alpha}_{t_0}] + \dots + \frac{\Delta t^{d_{\max}}}{d_{\max}!} [\mathbf{p}_{t_0}^{(d_{\max})}, \boldsymbol{\theta}_{t_0}^{(d_{\max})}] \\
&= \sum_{i=0}^{d_{\max}} \frac{\Delta t^i}{i!} [\mathbf{p}_{t_0}^{(i)}, \boldsymbol{\theta}_{t_0}^{(i)}],
\end{aligned} \tag{4.1}$$

where $[\mathbf{p}_{t_0+\Delta t}, \boldsymbol{\theta}_{t_0+\Delta t}]$ is a vector in the tangent space of the composite Lie group $T(3) \times SO(3)$ [SDA21]. The transformation matrix is then calculated as

$$T_{t+\Delta t} = \begin{bmatrix} \mathrm{Exp}(\boldsymbol{\theta}_{t_0+\Delta t}) & \mathbf{p}_{t_0+\Delta t} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{4.2}$$

The covariance $\Sigma_{[\mathbf{p}_{t+\Delta t}, \boldsymbol{\theta}_{t+\Delta t}]}$ is calculated by performing uncertainty propagation but with an added term to account for the Brownian motion modeled by the motion model factor with covariance $\Sigma_M$ as

$$\Sigma_{[\mathbf{p}_{t+\Delta t}, \boldsymbol{\theta}_{t+\Delta t}]} = \left( \sum_{i=0}^{d_{\max}} \left( \frac{\Delta t^i}{i!} \right)^2 \Sigma_{[\mathbf{p}_t^{(i)}, \boldsymbol{\theta}_t^{(i)}]} \right) + \frac{(\Delta t)^{2d_{\max}+1}}{(d_{\max}!)^2 (2d_{\max}+1)} \Sigma_M. \tag{4.3}$$

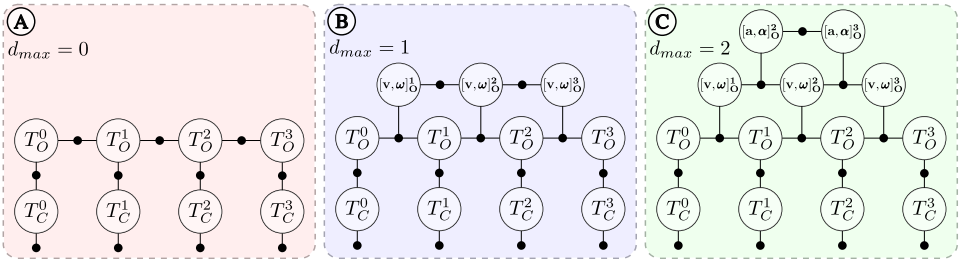


**Figure 4.3:** The topology of the factor graph is determined by the selected order of the motion model $d_{\max}$. Figure A illustrates the scenario where the motion model assumes the object pose to be constant. Figure B illustrates the case where both velocity $\mathbf{v}$ and angular velocity $\boldsymbol{\omega}$ are constant. Figure C illustrates the scenario where acceleration $\mathbf{a}$ and angular acceleration $\boldsymbol{\alpha}$ are constant. In the figure A, the estimated state of an object at time $k$ consists of $T_O^k$. In the figure B, the estimated state of an object at time $k$ consists of $T_O^k$ and $[\mathbf{v},\boldsymbol{\omega}]_{\mathbf{O}}^{\mathbf{k}}$. In the figure C, the estimated state of an object at time $k$ consists of $T_O^k$, $[\mathbf{v},\boldsymbol{\omega}]_{\mathbf{O}}^{\mathbf{k}}$ and $[\mathbf{a},\boldsymbol{\alpha}]_{\mathbf{O}}^{\mathbf{k}}$.

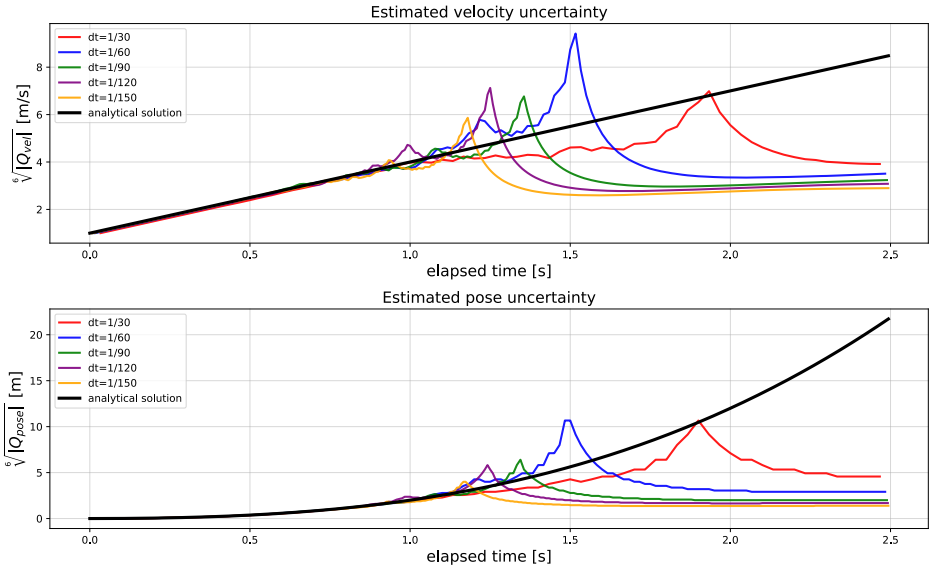


**Figure 4.4:** The figure illustrates the results of two different approaches to extrapolating pose and velocity covariance, assuming some initial $Q_{pose}$, $Q_{vel}$ and $Q_{acc}$. The colored plots depict the results of performing extrapolation by inserting new variables into the factor graph at different intervals $dt$. This approach begins to behave unpredictably after approximately one second. On the other hand, the black plots show the results of performing extrapolation using our analytical formula. This approach remains stable.

# Chapter 5

# Experiments

**Figure 5.1:** Example frames from HOPE-Video sequence are shown in the first row. The second row shows an overlay of the rendered poses predicted by the per-frame CosyPose [LCAS20] evaluation. It can be seen that some of the objects are not detected. Our temporally smoothed predictions are shown in the last row, mitigating the effect of missing detections.

## 5.1 Datasets

Three datasets were used for the quantitative evaluation:

- Household Objects for Pose Estimation (HOPE-Video) [TTT+22] dataset and
- two synthetically rendered datasets that we created via Blender [Com18].

Only RGB images are considered for all datasets. HOPE-Video contains 10 video sequences captured by a moving camera observing a static scene with 5-20 objects placed on a desk. The video is recorded by a robot equipped with a RealSense camera; an example of the video sequence is shown in Fig. 5.1.

The HOPE-Video dataset was chosen instead of the more commonly used YCB-V dataset for the following reasons:

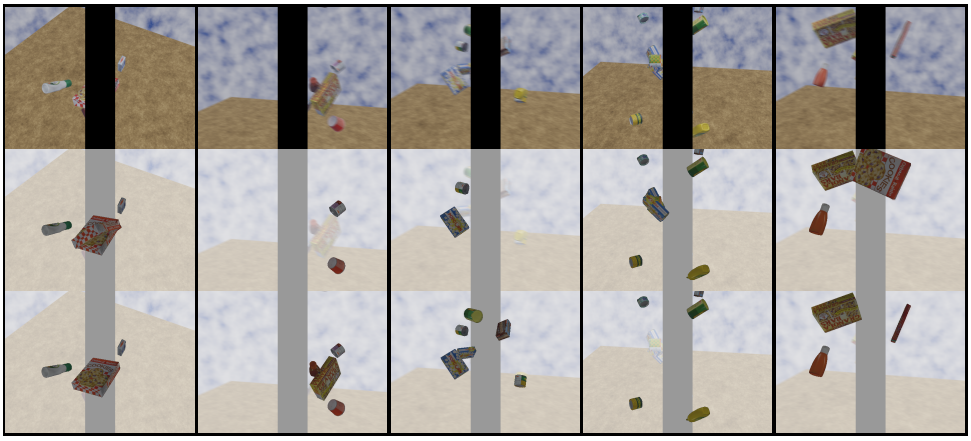

**Figure 5.2:** Example frames from *SynthYCBVDynamic* where the center of the frame is occluded by a black rectangle and some of the frames are artificially blurred. It can be seen that some of the objects are not detected by per-frame CosyPose (*e.g.*, frames 2 and 3) or that some outliers are predicted (*e.g.*, frame 5). Our temporally smoothed predictions are shown in the last row, mitigating missing detections and outliers.

- It offers faster camera movements, which are more suitable for analyzing our method's performance.
- The ground-truth is globally consistent, as shown in Fig. 5.3.

To use the dataset for the standardized BOP challenge, we developed a tool to convert the dataset into the BOP format.

The second dataset is synthetically rendered to obtain ground-truth poses for dynamically moving objects as well. The dataset contains dynamically moving rendered HOPE objects [TTT+22]. To address the real-to-sim comparison, we first rendered 10 video sequences with static objects placed on the desk in a setup similar to the HOPE-Video datset. We refer to this dataset as *SynthHOPEStatic*.

Dynamical dataset *SynthHOPEDynamic* is composed of 5-10 objects moving on randomly sampled trajectories. The trajectories are obtained by randomly sampling poses in $SE(3)$ that are connected by the Cartesian dynamical movement primitives [UNPM14] with randomly sampled weights and initial and goal velocities. The camera is also moving on a random trajectory and motion blur is applied to random frames. To simulate challenging occlusions, a uniform color box is rendered in front of the camera. In total, 10 video sequences are rendered for the *SynthHOPEDynamic* dataset. An example of the synthetic dataset is shown in Fig. 5.2.

## 5.2 Metrics

To measure performance, we calculated the average recall and average precision for the three datasets. For average recall, we rely on error metrics, which

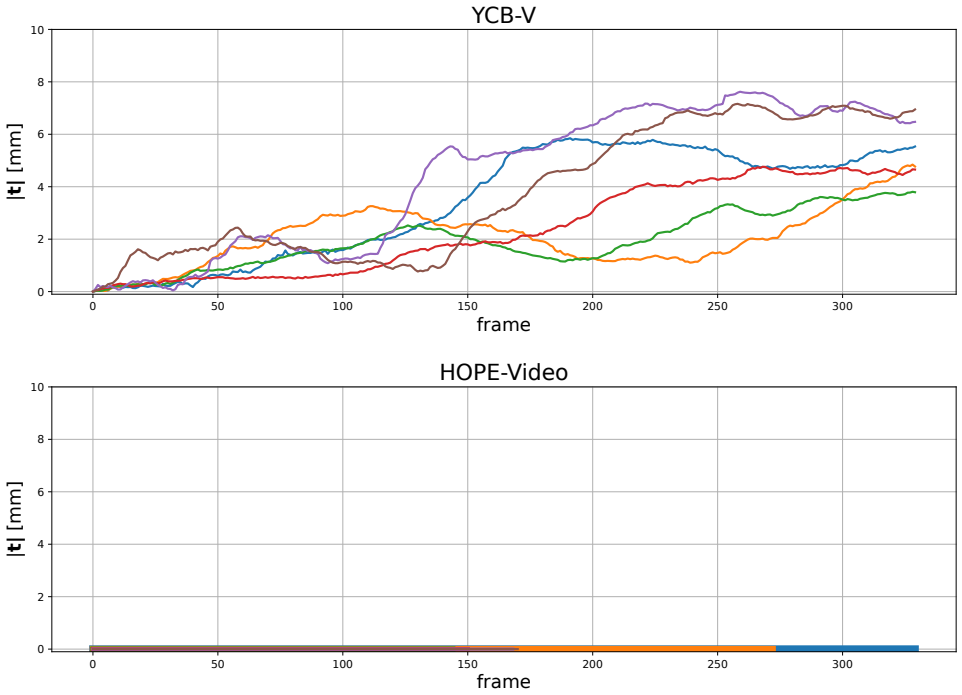


**Figure 5.3:** The figure shows the gradual drift of objects from different scenes of the YCB-V and HOPE-Video datasets computed as the translation from the objects' initial pose in the world frame. It shows that the YCB-V dataset is slowly drifting while HOPE-Video remains consistent.

are commonly used in the BOP object pose estimation challenge [HMB+18]. Recall is averaged across several thresholds and across three different metrics:

- Visible Surface Discrepancy (VSD),
- Maximum Symmetry-Aware Surface Distance (MSSD),
- Maximum Symmetry-Aware Projection Distance (MSPD).

See [HSD+20] for details on these metrics and thresholds. For precision, we used the same metric (*i.e.*, VSD, MSSD, and MSPD) and the same thresholds as used for the recall computation. Recall penalizes missing object detections and object pose estimates, while precision penalizes incorrect object detections and object pose estimates. Only objects that are at least partially visible in the image are considered in the evaluation; *i.e.*, the number of visible pixels is at least 5% of the size of the full object projection.

## 5.3 Measurement covariance estimation

We observed that translation measurement uncertainty is bigger in the direction of ray pointing towards the object of interest (*i.e.*, uncertainty $\sigma_{C'z}$) and that it depends on the size of the object in the image space, as shown in

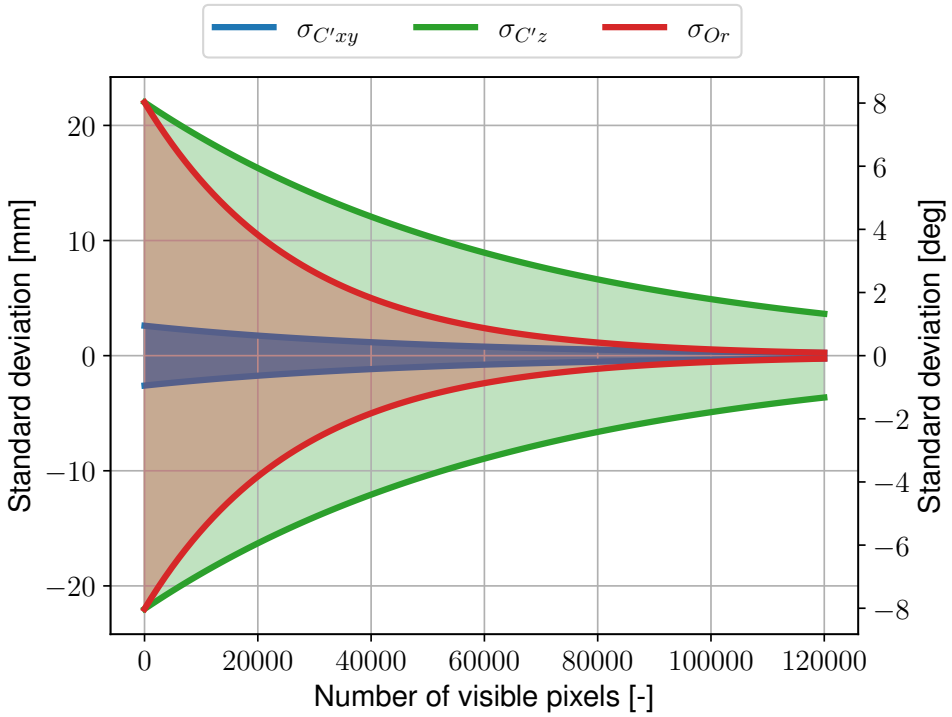


**Figure 5.4:** Estimation of the standard deviations for object measurement model based on the Hope-Video dataset.

Fig. 3.2. We use the exponential dependence on the number of visible pixels of the object in the image space in the form: $\sigma(n_{\text{px}}) = a\exp(-bn_{\text{px}})$, where $a$ and $b$ are parameters fitted separately for the translation $xy$, the translation $z$ (*i.e.*, depth) and the rotation. Translation uncertainties are estimated in the frame whose $z$ axis points toward the center of the object, while rotation uncertainties are estimated in the object frame. We used the Hope-Video dataset to estimate these uncertainties; estimated values are visualized in Fig. 5.4.

## 5.4 Ablation study

Several thresholds need to be tuned for the proposed filtering method. We manually set the outlier prediction thresholds $\tau_{\text{outlier\_t}}$ and $\tau_{\text{outlier\_r}}$ to 100 mm and 10°. The prediction thresholds $\tau_{\text{pred\_t}}$ and $\tau_{\text{pred\_r}}$ were chosen based on the ablation study in which we evaluated the precision-recall curve for various values of these hyperparameters. Subsets containing three scenes from our synthetic datasets were used to select thresholds for the constant pose model (subset of *SynthHOPEStatic*) and for the constant velocity model (subset of *SynthHOPEDynamic*). The result of the ablation is shown in Fig. 5.5. From the ablation, we selected two sets of thresholds for each motion model. These sets of thresholds correspond to recall-oriented parameters and precision-oriented parameters as visualized in Fig. 5.5.

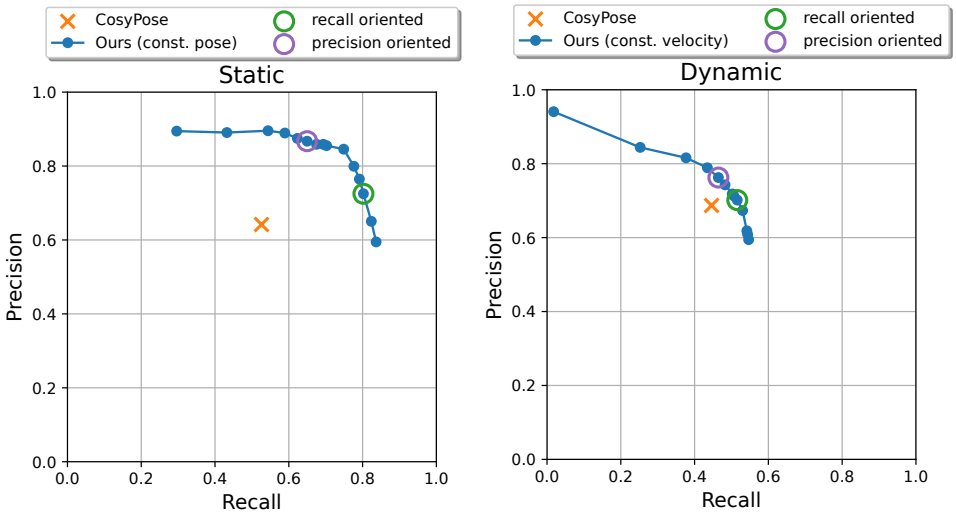


**Figure 5.5:** Ablation study for constant pose model evaluated on three scenes of the static synthetic dataset (left) and for constant velocity model evaluated on three scenes of the dynamic synthetic dataset (right). The precision-recall trade-off is controlled by hyperparameters of our model. Recall oriented parameters are selected such that recall is maximal and precision is at least at the CosyPose value. Similarly for the precision oriented parameters.

## 5.5 Quantitative evaluation

We evaluated the performance of our method on the three datasets mentioned above. The results are summarized in Tab. 5.1. Two baselines are considered: (i) per-frame CosyPose [LCAS20] and (ii) short-horizon filtering, in which only the last three frames were used for our method.

For static object datasets (*i.e. HOPE-Video* and *SynthHOPEStatic*) both constant pose and constant velocity motion models are evaluated. It can be seen that our methods outperformed the baselines in recall (if recall-oriented) while achieving comparable precision. Similarly, for a precision-oriented variant, we outperform the baselines in precision while achieving a comparable recall. The precision-recall trade-off can be controlled by hyperparameters. The constant pose motion model achieved better performance than the constant velocity motion model as it has a stronger prior about the motion of the objects.

For the dynamic object dataset, we evaluated the constant velocity motion model. We outperform the baselines in the same spirit as for the static object datasets.

| Dataset | Method | Recall | | | | Precision | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | VSD | MSSD | MSPD | AVG. | VSD | MSSD | MSPD | AVG. |
| HOPE-Video | CosyPose [LCAS20] | 0.40 | 0.36 | 0.42 | 0.39 | 0.59 | 0.52 | 0.61 | 0.57 |
| | Short horizon SAM | 0.41 | 0.36 | 0.42 | 0.40 | 0.61 | 0.53 | 0.62 | 0.59 |
| | Ours (const. pose, recall-oriented) | 0.57 | 0.58 | 0.57 | **0.57** | 0.59 | 0.59 | 0.58 | 0.58 |
| | Ours (const. pose, precision-oriented) | 0.44 | 0.44 | 0.42 | 0.43 | 0.67 | 0.66 | 0.63 | **0.65** |
| | Ours (const. vel., recall-oriented) | 0.47 | 0.42 | 0.49 | 0.46 | 0.60 | 0.54 | 0.63 | 0.59 |
| | Ours (const. vel., precision-oriented) | 0.44 | 0.40 | 0.46 | 0.43 | 0.64 | 0.58 | 0.67 | 0.63 |
| SynthHOPEStatic | CosyPose [LCAS20] | 0.58 | 0.51 | 0.52 | 0.53 | 0.74 | 0.65 | 0.66 | 0.69 |
| | Short horizon SAM | 0.58 | 0.51 | 0.51 | 0.54 | 0.81 | 0.71 | 0.72 | 0.75 |
| | Ours (const. pose, recall-oriented) | 0.83 | 0.81 | 0.80 | **0.81** | 0.81 | 0.78 | 0.77 | 0.79 |
| | Ours (const. pose, precision-oriented) | 0.70 | 0.68 | 0.67 | 0.68 | 0.91 | 0.89 | 0.87 | **0.89** |
| | Ours (const. vel., recall-oriented) | 0.66 | 0.60 | 0.61 | 0.62 | 0.81 | 0.73 | 0.74 | 0.76 |
| | Ours (const. vel., precision-oriented) | 0.61 | 0.56 | 0.57 | 0.58 | 0.86 | 0.79 | 0.80 | 0.82 |
| SynthHOPEDynamic | CosyPose [LCAS20] | 0.44 | 0.39 | 0.51 | 0.44 | 0.65 | 0.58 | 0.77 | 0.66 |
| | Short horizon SAM | 0.39 | 0.35 | 0.47 | 0.40 | 0.69 | 0.62 | 0.82 | 0.71 |
| | Ours (const. vel., recall-oriented) | 0.49 | 0.45 | 0.56 | **0.50** | 0.68 | 0.62 | 0.77 | 0.69 |
| | Ours (const. vel., precision-oriented) | 0.44 | 0.41 | 0.50 | 0.45 | 0.73 | 0.68 | 0.84 | **0.75** |

**Table 5.1:** BOP Average Recall and Average Precision evaluated on three video datasets by considering all frames of the video and all objects that are visible in the image at least 5% of the object size. "Recall-oriented" and "precision-oriented" refer to different configurations aimed at maximizing average recall or precision while ensuring that the average of the other metric is at least as good as CosyPose. Terms "const. pose" and "const. velocity" denote different motion models. The "Short-horizon SAM" baseline refers to our method modified to use only the last 3 frames. The best results for AVG. recall and precision are shown in bold.

## 5.6 Qualitative robotic experiment

To validate the stability of the proposed filtering method, we performed several robotics experiments. For all experiments, we used a Franka Emika Panda robot equipped with a RealSense D435 camera attached to its end-effector (eye-in-hand configuration). The camera mount was calibrated. The camera produces a 60 Hz RGB video stream with a resolution of 640x480 pixels. We conducted the following robotic experiments to demonstrate the advantages of the method:

- **Static scene** objects pose estimation in which the robot is guided by the human hand and, while moving, it estimates the poses of objects that are statically placed in front of the robot;
- **Dynamic scene** objects pose estimation, where robot remains static and estimates the poses of objects that are moved by human; and
- **Dynamic object tracking** where robot maintains constant pose with respect to the tracked object.

In the first two experiments, the robot is not controlled on the basis of the predicted poses and our method can be applied directly. Please, see the supplementary video for recording of the experiments.

For the dynamic object tracking experiment, a feedback-based controller was designed that uses the proposed probabilistic dynamic world. The controller architecture is visualized in Fig. 5.6. With this control architecture and using the proposed filtering method, we achieve stable tracking in a challenging scenario in which the object was hidden behind the occluder, as shown in Fig. 5.7. The analysis of the image stream for the tracking experiment is shown in Fig. 5.8. The full experiment is shown in the supplementary video.

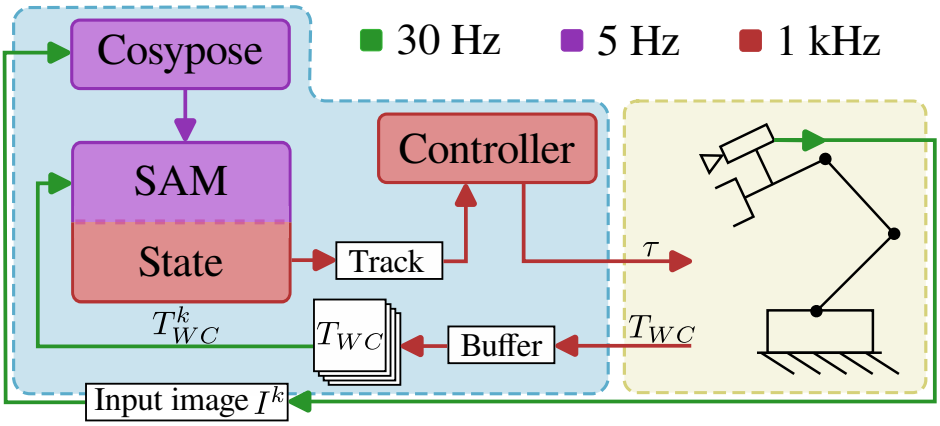


**Figure 5.6:** The robot control architecture used for the tracking experiment. First, an image $I^k$ is used with CosyPose to generate object pose estimates. These estimates are then fed into the proposed SAM refiner along with the camera pose $T^k_{WC}$ whose timestamp corresponds to the time stamp of the input image $I^k$ used in CosyPose. This synchronization is achieved by buffering the poses $T_{WC}$ and subsequently selecting the one with the closest timestamp. The SAM refiner produces an estimate of the State, *i.e.*, the probabilistic world model. Note that although the world model is updated at the CosyPose frequency, the State is computed at the robot control frequency using the motion model. Finally, a track selected by the user is used as input for the robot Controller, which computes the motor torques $\boldsymbol{\tau}$ required to move the robot into the desired pose. The typical processing frequencies of individual modules are 5 Hz for Cosypose and SAM refiner, 30 Hz for the camera, and 1 kHz for the state extrapolation and robot controller.

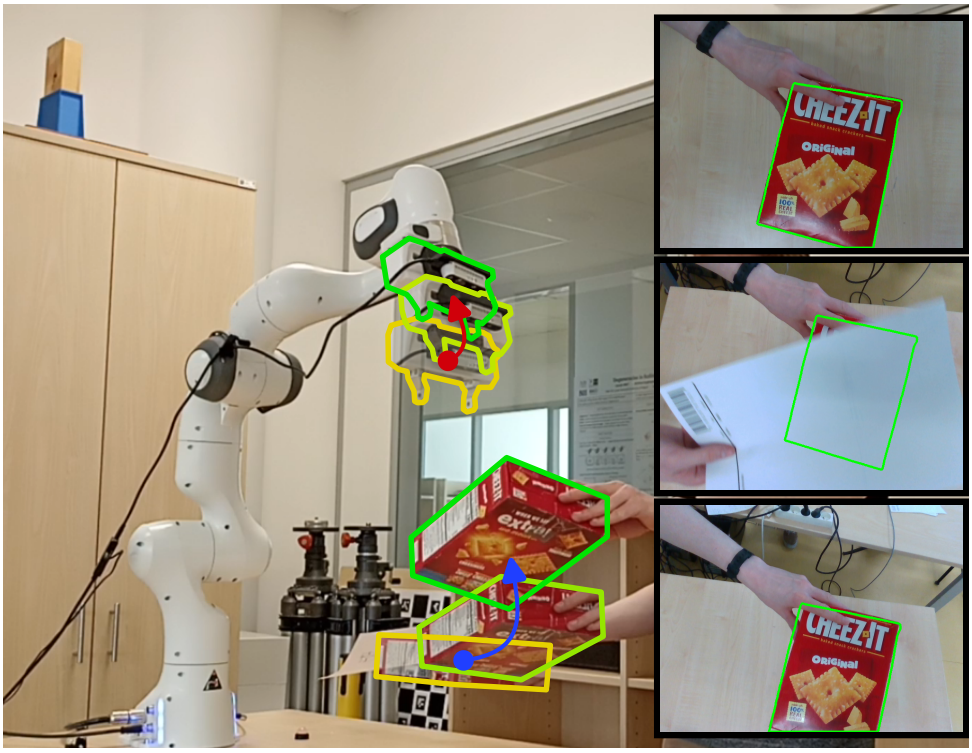


**Figure 5.7:** The illustration depicts a selected sequence of images recorded during an experiment where the robot attempts to maintain a constant relative end effector transformation with respect to the Cheez-it box from the YCB [CSW+15] dataset. During the tracking process, the object is occluded by a sheet of paper, demonstrating the temporal consistency and stability of the refined pose estimates.

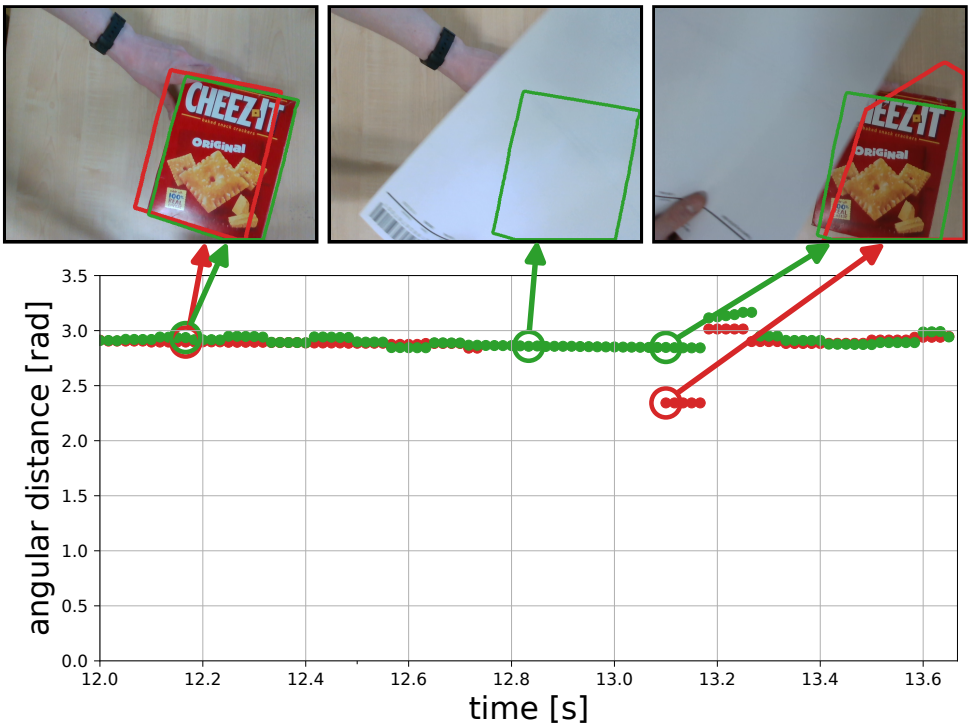


**Figure 5.8:** The evolution of the object angular distance for the robot tracking experiment. If object is not occluded, the CosyPose and our method predicts the object pose accurately (first frame). However, when object is completely occluded the per-frame evaluation cannot evalute the pose of the object (second frame). Finally, if the object is partially visible , the CosyPose predicts wrong orientation while the proposed filtering remains stable (third frame).

# Chapter 6

# Limitations

## 6.1 Discrete and continuous symmetries

This work does not address continuous and discrete symmetries. Continuous symmetries are evident in solids of revolution, such as the YCB bowl shown in Fig. 6.1, where the object exhibits symmetry around the z-axis. Discrete symmetries are observed in objects like the YCB wood block, which possesses 8 rotational discrete symmetries. This poses issues since any of these symmetries are equally likely to be detected, leading to the creation of multiple tracks that actually correspond to a single object.

One approach to addressing continuous symmetries could involve modifying the estimated covariance matrix to be infinite in the rotational part along the axis of symmetry. This adjustment would serve as a means to approximate the actual probability distribution of the estimated pose.

Addressing discrete symmetries could be achieved by treating every measurement of a discretely symmetric object as multiple different measurements according to the symmetries' transformations. This approach serves as a means to approximate the real probability distribution of the object's pose given the measurement.

## 6.2 Track merging

The current implementation does not address the scenario where two or more tracks of the same object instance are created, and their states converge to the same configuration. In such cases, the matching process might randomly assign new measurements to one of those tracks, depending on the choice of parameters used. While this occurrence is not common with the current configuration, it is a potential issue.

One way to address this issue could be to implement some form of track merging, where tracks with similar states are either removed or combined. Another approach could involve allowing a single measurement to be associated with multiple tracks.

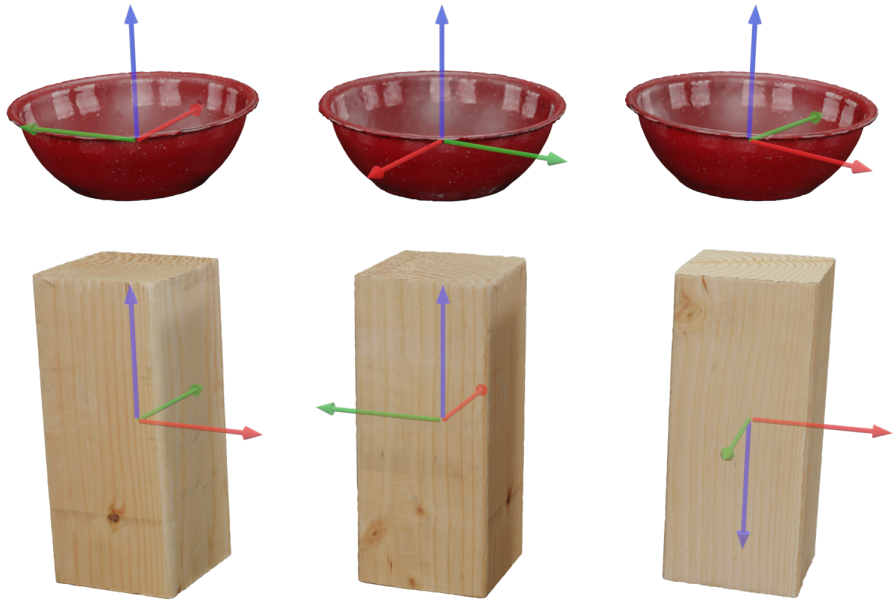

**Figure 6.1:** The figure displays two different objects from the YCB dataset in various orientations. The red bowl exhibits continuous symmetry around the Z-axis, while the wooden block possesses 8 discrete rotational symmetries.

## 6.3 Camera pose retrieval

The current implementation assumes the camera pose $T_{WC}$ to be known, which is easily achievable with the calibrated robot used in the qualitative experiments. However, this approach might be impractical for other applications. Therefore, a different method of estimating the camera pose $T_{WC}$ could be more suitable, such as visual odometry, IMU motion integration, or a combination of both. This could potentially open up opportunities for use in virtual and augmented reality.

# Chapter 7

# Conclusion

Accurate and temporally consistent object pose estimation is crucial for many applications in robotics and augmented reality. Current methods for single-view object pose estimation from an RGB camera often struggle with challenging scenarios caused by image noise, blur, or occlusion. We implemented a method that performs probabilistic filtering of object estimates across multiple measurements to improve the performance of an single-view object pose estimation method.

For this purpose, we used probabilistic filtering developed for Simultaneous Localization and Mapping (SLAM) applications. We formulated the task using the factor graph approach and included a motion model that generalizes the pose estimation task to dynamically moving objects. Our implementation achieves near-real-time performance.

The proposed algorithm has been validated through both a quantitative study on a benchmark and qualitative experiments. The quantitative experiments were performed on the HOPE-Video dataset as well as our two custom-generated datasets. The qualitative experiments involved a physical robot and objects from the YCB dataset. Both the quantitative and qualitative experiments have shown improvement over the baseline.

# Bibliography


[Com18] Blender Online Community, *Blender - a 3d modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[CSW+15] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar, *The ycb object and model set: Towards common benchmarks for manipulation research*, 2015 International Conference on Advanced Robotics (ICAR), 2015, pp. 510–517.

[DC22] Frank Dellaert and GTSAM Contributors, *borglab/gtsam*, May 2022.

[DK+17] Frank Dellaert, Michael Kaess, et al., *Factor graphs for robot perception*, Foundations and Trends® in Robotics **6** (2017), no. 1-2, 1–139.

[DMX+21] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox, *Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking*, IEEE Transactions on Robotics **37** (2021), no. 5, 1328–1342.

[FB81] Martin A Fischler and Robert C Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM **24** (1981), no. 6, 381–395.

[FHD+21] Jiahui Fu, Qiangqiang Huang, Kevin Doherty, Yue Wang, and John J Leonard, *A multi-hypothesis approach to pose ambiguity in object-based slam*, 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2021, pp. 7639–7646.

[GKSB10] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard, *A tutorial on graph-based slam*, IEEE Intelligent Transportation Systems Magazine **2** (2010), no. 4, 31–43.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, *Mask r-cnn*, Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[HMB+18] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al., *Bop: Benchmark for 6d object pose estimation*, Proceedings of the European conference on computer vision (ECCV), 2018, pp. 19–34.

[HMO16] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek, *On evaluation of 6d object pose estimation*, Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14, Springer, 2016, pp. 606–619.

[HSD+20] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas, *Bop challenge 2020 on 6d object localization*, Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, Springer, 2020, pp. 577–594.

[HSL+24] Tomas Hodan, Martin Sundermeyer, Yann Labbe, Van Nguyen Nguyen, Gu Wang, Eric Brachmann, Bertram Drost, Vincent Lepetit, Carsten Rother, and Jiri Matas, *Bop challenge 2023 on detection, segmentation and pose estimation of seen and unseen rigid objects*, arXiv preprint arXiv:2403.09799 (2024).

[HZMI20] Mina Henein, Jun Zhang, Robert Mahony, and Viorela Ila, *Dynamic slam: The need for speed*, 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 2123–2129.

[IWC+16] Jan Issac, Manuel Wüthrich, Cristina Garcia Cifuentes, Jeannette Bohg, Sebastian Trimpe, and Stefan Schaal, *Depth-based object tracking using a robust gaussian filter*, 2016 IEEE international conference on robotics and automation (ICRA), IEEE, 2016, pp. 608–615.

[KC+02] Danica Kragic, Henrik I Christensen, et al., *Survey on visual servoing for manipulation*, Computational Vision and Active Perception Laboratory, Fiskartorpsv **15** (2002), 2002.

[KG17] Alex Kendall and Yarin Gal, *What uncertainties do we need in bayesian deep learning for computer vision?*, Advances in neural information processing systems **30** (2017).

[LCAS20] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic, *Cosypose: Consistent multi-view multi-object 6d pose estimation*, Springer, 2020, pp. 574–591.

[LD22] Tristan Laidlow and Andrew J Davison, *Simultaneous localisation and mapping with quadric surfaces*, 2022 International Conference on 3D Vision (3DV), IEEE, 2022, pp. 1–9.

[LDC+21] Kejie Li, Daniel DeTone, Yu Fan Steven Chen, Minh Vo, Ian Reid, Hamid Rezatofighi, Chris Sweeney, Julian Straub, and Richard Newcombe, *Odam: Object detection, association, and mapping using posed rgb video*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5998–6008.

[Lep20] Vincent Lepetit, *Recent advances in 3d object and hand pose estimation*, arXiv preprint arXiv:2006.05927 (2020).

[LMM+22] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic, *Megapose: 6d pose estimation of novel objects via render & compare*, arXiv preprint arXiv:2212.06870 (2022).

[LRR21] Kejie Li, Hamid Rezatofighi, and Ian Reid, *Moltr: Multiple object localization, tracking and reconstruction from monocular rgb videos*, IEEE Robotics and Automation Letters **6** (2021), no. 2, 3341–3348.

[LSL+21] Zoe Landgraf, Raluca Scona, Tristan Laidlow, Stephen James, Stefan Leutenegger, and Andrew J Davison, *Simstack: A generative shape and instance model for unordered object stacks*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13012–13022.

[LWJ+18] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox, *Deepim: Deep iterative matching for 6d pose estimation*, Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 683–698.

[MCB+18] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger, *Fusion++: Volumetric object-level slam*, 2018 international conference on 3D vision (3DV), IEEE, 2018, pp. 32–41.

[MGZ+22] Nathaniel Merrill, Yuliang Guo, Xingxing Zuo, Xinyu Huang, Stefan Leutenegger, Xi Peng, Liu Ren, and Guoquan Huang, *Symmetry and uncertainty-aware object slam for 6dof object pose estimation*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14901–14910.

[NGP+23] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan, *Cnos: A strong baseline for cad-based novel object segmentation*, Proceedings of the IEEE/CVF

International Conference on Computer Vision, 2023, pp. 2134–2140.

[NGSL24] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit, *Gigapose: Fast and robust novel object pose estimation via one correspondence*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.

[NMS18] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf, *Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam*, IEEE Robotics and Automation Letters **4** (2018), no. 1, 1–8.

[ÖLT+23] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan, *Foundpose: Unseen object pose estimation with foundation features*, arXiv preprint arXiv:2311.18809 (2023).

[PK15] Karl Pauwels and Danica Kragic, *Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking*, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 1300–1307.

[RA17] Martin Rünz and Lourdes Agapito, *Co-fusion: Real-time segmentation, tracking and fusion of multiple objects*, 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 4471–4478.

[RBA18] Martin Runz, Maud Buffier, and Lourdes Agapito, *Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects*, 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, 2018, pp. 10–20.

[SDA21] Joan Solà, Jeremie Deray, and Dinesh Atchuthan, *A micro lie theory for state estimation in robotics*, 2021.

[SMNS+13] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison, *Slam++: Simultaneous localisation and mapping at the level of objects*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 1352–1359.

[SPS+22] Manuel Stoiber, Martin Pfanne, Klaus H Strobl, Rudolph Triebel, and Alin Albu-Schäffer, *Srt3d: A sparse region-based 3d object tracking approach for the real world*, International Journal of Computer Vision **130** (2022), no. 4, 1008–1030.

[SWD20] Edgar Sucar, Kentaro Wada, and Andrew Davison, *Nodeslam: Neural object descriptors for multi-view shape reconstruction*, 2020 International Conference on 3D Vision (3DV), IEEE, 2020, pp. 949–958.

[TTT+22] Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield, *6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark*, International Conference on Intelligent Robots and Systems (IROS), 2022.

[UNPM14] Aleš Ude, Bojan Nemec, Tadej Petrić, and Jun Morimoto, *Orientation in cartesian space dynamic movement primitives*, 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 2997–3004.

[WMRB20] Bowen Wen, Chaitanya Mitash, Baozhang Ren, and Kostas E Bekris, *se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains*, 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 10367–10373.

[WSJ+20] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison, *Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 14540–14549.

[WYKB23] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield, *Foundationpose: Unified 6d pose estimation and tracking of novel objects*, arXiv preprint arXiv:2312.08344 (2023).

[XDL22] Binbin Xu, Andrew J Davison, and Stefan Leutenegger, *Learning to complete object shapes for object-level mapping in dynamic scenes*, 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2022, pp. 2257–2264.

[XLT+19] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger, *Mid-fusion: Octree-based object-level multi-instance dynamic slam*, 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 5231–5237.

[YS19] Shichao Yang and Sebastian Scherer, *Cubeslam: Monocular 3-d object slam*, IEEE Transactions on Robotics **35** (2019), no. 4, 925–938.