Temporally Consistent Object 6D Pose Estimation for Robot Control

Kateryna Zorina[1*], Vojtech Priban[1*], Mederic Fourmy[1], Josef Sivic[1] and Vladimir Petrik[1]

***Abstract*—Single-view RGB object pose estimators have reached a level of precision and efficiency that makes them good candidates for vision-based robot control. However, off-the-shelf methods lack temporal consistency and robustness that are mandatory for a stable feedback control. In this work, we develop a factor graph approach to enforce temporal consistency of the object pose estimates. In particular, the proposed approach: (i) incorporates object motion models, (ii) explicitly estimates the object pose measurement uncertainty, and (iii) integrates the above two components in an online optimization-based estimator. We demonstrate that with appropriate outlier rejection and smoothing using the proposed factor graph approach, we can significantly improve the results on standardized pose estimation benchmarks. We experimentally validate the stability of the proposed approach for a feedback-based robot control task in which the object is tracked by the camera attached to a torque controlled manipulator. *Index Terms*—Visual Tracking, Computer Vision for Automation**

I. INTRODUCTION

SINGLE view object pose estimation from an RGB camera has made significant progress in recent years [1] *e.g.*, by using the render-and-compare approach [2], [3]. Our motivation is to use object pose estimates for feedback-based robot control, for example, for visual tracking or an object hand-over from a human to a robot. However, pose predictions are often inconsistent in time: some estimates are missing or outliers occur, as shown in Fig. 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 estimates may place an object suddenly 10 cm away or incorrectly estimate that orientation suddenly changes by 180 degrees due to symmetries, and cause the controller to generate incorrectly large desired robot torques leading to dangerous motion. Object trackers [4], [5] provide consistent poses but may fail if objects are occluded or out of view.

To address these issues, in this paper, we build on advances in Simultaneous Localization and Mapping (SLAM) [6] and develop a probabilistic smoothing approach 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 consisting of object poses. Temporally consistent poses are predicted from the world model and are safe to use in the robot control loop. The probabilistic smoothing approach allows us to address the following challenges: (i) **Missing object detections** are predicted by the model (via a motion model) to maintain temporal consistency; (ii) **Outlier rejection** is implemented to maintain temporal consistency. (iii) **Multiple instances** of the same object are tracked separately in the world model so that the robot knows which instance is tracked; and (iv) **Discrete object symmetries** are tracked separately to predict temporally consistent poses.

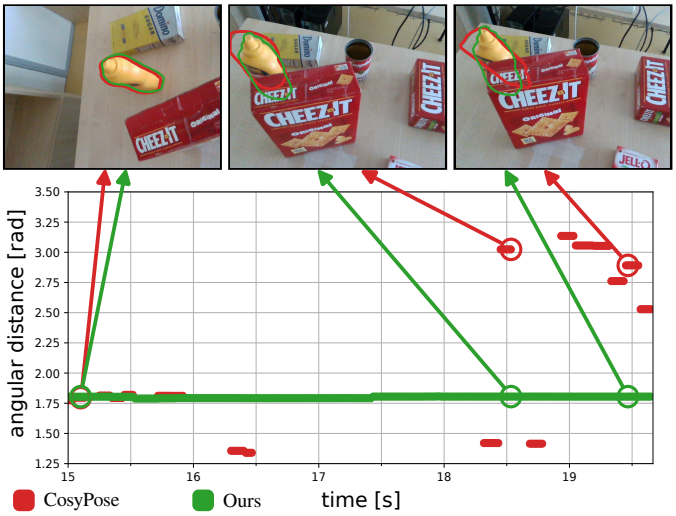

Fig. 1: **Mustard bottle object pose estimates from images.** The plot (bottom) shows the angular distance between the estimated pose and a fixed reference frame. The shown objects are static, and therefore the distance should be constant. The red dots show the per-frame estimates computed by an object pose estimator CosyPose [2]. Filtered predictions computed by our method are shown in green. The corresponding red and green contours in the images (top) were computed by reprojecting the object model using the estimated pose. In the first shown frame (left), both predictions are correct and overlap. However, in more difficult scenarios (middle and right) the per-frame estimates (red) are incorrect and would cause instability in the control. Our approach (green) is correct even in these challenging partially occluded scenarios.

Manuscript received: July, 16, 2024; Revised October, 5, 2024; Accepted October, 31, 2024.

This paper was recommended for publication by Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by AGIMUS, euROBIN, and FRONTIER projects, funded by the European Union under GA no. 101070165, 101070596, and 101097822.

[1]Kateryna Zorina, Vojtech Priban, Mederic Fourmy, Josef Sivic and Vladimir Petrik are with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague. name.surname@cvut.cz

*Equal contribution.

Digital Object Identifier (DOI): see top of this page.

In summary, this paper has the following contributions: (i) we present a probabilistic smoothing approach for temporally consistent object pose tracking suitable for feedback-based robot control; (ii) we evaluate our approach on a standard real video dataset with static objects and on synthetically rendered dataset with static and dynamic objects - we achieve superior performance on all evaluated datasets; (iii) we demonstrate the proposed smoothing approach in a robot object tracking application with a Franka Emika Panda manipulator - we experimentally show that our approach leads to robust tracking in situations where per-frame estimation fails. Our code is open-source available at https://github.com/priban42/temporal_pose.

II. RELATED WORK

Object pose estimation. Model-based object pose estimation is one of the core computer-vision challenges with a wide range of applications for robotics and AR/VR [7], [8]. The problem is most often decomposed into two stages: 2D image detection, which provides object-labeled bounding boxes and masks, followed by a pose estimation for each individual detection. Learning methods currently dominate the standardized benchmarks for both steps [1]. A more recent challenge addresses generalizability to objects unseen during training, both for detection [9] and pose estimation [3], [10], [11], [12]. Used by some of the leading methods, the "render-and-compare" approach [13], [2], [3], [12] 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 single-view object pose tracking [4], [14], [15], [12].

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 objects can be ill-defined due to object symmetries (*e.g.* a bottle) or partial occlusion (*e.g.* a cup with a hidden handle). Higher uncertainty may also occur in real-world experiments, *e.g.* if the model is trained with insufficient data augmentation [16]. With model-based object pose estimators performance improving rapidly, we propose a method that uses off-the-shelf object pose estimators for fast and robust object tracking.

Multiview object pose estimation. In robotics, it is common to have a multi-camera setup [5] or a camera mounted on the robot [17]. This setup can be leveraged by aggregating information across views and time to create a consistent estimate of both the camera/object poses and of the object shapes. Commonly used representations include parametric surfaces [18], [19], [20], [21], volume based representations [22], [23] or latent codes [24], [25], [21].

In many practical industrial scenarios, it may be reasonably assumed that object models are available before starting the tracking process. SLAM++ [26] is the first depth-based object SLAM system and formulates the estimation using a probabilistic pose graph back-end. SimTrack [5] proposes a tightly integrated RGB-D system for robot/object pose detection and tracking. Others directly tackle the inherent pose ambiguity of image-based pose estimation, *e.g.*, [27] explicitly trains a single view model that predicts a set of pose hypothesis that are resolved over different views using a max-mixture formulation. The work [28] fuses 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 [2] addresses the single-view pose data association problem by designing a symmetry-aware RANSAC [29] followed by bundle adjustment [30] and is agnostic to the single view pose estimator. We propose to address the object pose ambiguities by tracking simultaneously the multiple symmetry modes of the objects in the scene.

Temporally consistent moving object estimation. In previously mentioned SLAM-like systems, the objects are assumed to be static in the environment. A natural but challenging extension to these methods is to allow multi-object live scene reconstruction with dynamic objects [31], [32], [33], [34], [35]. To improve the geometric consistency of the scene, Dynamic SLAM [36] 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 regularization to filter-based object pose trackers, either by penalizing large pose updates [4] or by estimating a higher-order state like the object twist [37]. We propose to estimate the pose and twist of multiple objects using a factor graph.

III. TEMPORALLY CONSISTENT POSE ESTIMATION

Problem formulation. Our goal is to track the $SE(3)$ pose of an object with a moving calibrated camera rigidly attached to the robot end effector, as shown in Fig. 2-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 into a single probabilistic estimation problem that finds the optimal trajectory of the camera and the object poses, as shown in Fig. 2-B. The main technical challenges are: (i) Achieving fast joint optimization of object poses over time while considering the uncertainties of the object pose measurements together with object motion models, which we address using a factor graph approach; (ii) Appropriate modelling of the measurement uncertainty of object pose, for which we develop a model that captures the difficulty of depth estimation from the RGB image; and (iii) Outlier rejection and data association, which we address by comparing the incoming measurements with the estimated pose distributions of the tracked objects. The details follow.

Factor graph. We formulate the temporally consistent pose estimation task as a weighted nonlinear least squares problem following the factor graph approach [38]. Under the assumption of conditionally independent measurements corrupted by Gaussian noise, the optimal sequence of object and camera poses is obtained by solving:

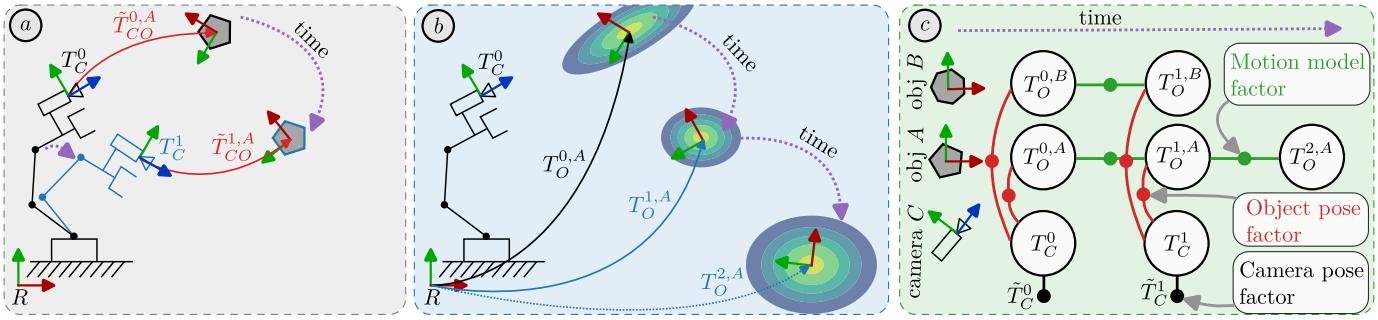

Fig. 2: **Overview.** Our goal is to estimate the poses of objects in time with respect to the reference frame R as shown in figure a. To achieve this, we use measurements at a time step k of the camera pose $\tilde{T}_C^k$ and the object pose $\tilde{T}_{CO}^{k,A}$, where A is the label of the object. Both objects and the robot are moving in time, as illustrated by purple arrows. Our approach maintains the probabilistic world representation of the object poses as visualized in figure b, where the ellipsoids represent the poses uncertainty. This uncertainty is used to filter outliers and predict only confident poses. The map is maintained through the factor graph shown in figure c, where the green factors represent the motion model, the red factors represent the observations of the object pose in the camera, and the 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 to the future (figures b and c) to resolve missing measurements due to, *e.g.*, sudden occlusion.

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

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 τ, $\mathbf{r}_X$ 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 an occlusion or a significant motion blur. We minimize the above cost over the set of variables χ, which consists of object and camera poses over time 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 (i) the camera pose factors regularize the camera pose to stay close to the pose measured by robot's forwards kinematics; (ii) the object pose factors regularize the object pose to stay close to the measured pose w.r.t. the camera, and (iii) the motion model factor captures the motion of the object, *i.e.* the change of the pose and its uncertainty over time.

To account for this inequality, the residuals are scaled by covariance matrices that represent our confidence in the measurements. The residuals are scaled by covariance matrices that represent our confidence in the measurements. The computation of the residuals and the corresponding covariances is described next.

The camera pose measurement factor. We perform hand-eye calibration using the OpenCV library [39]. Therefore the camera pose residual can be computed by comparing the $SE(3)$ distance between the estimated value and the corresponding measurement, *i.e.*, $\mathbf{r}_C^k = \text{Log}((T_C^k)^{-1}\tilde{T}_C^k)$, where symbol˜represents the measurement, here computed by forward kinematics, and Log is the logarithm mapping from $SE(3)$ group [40]. The covariance of the camera pose factor is assumed to be diagonal in the form $\mathbf{\Sigma}_C = \text{diag}(\sigma_{Ct}^2, \sigma_{Ct}^2, \sigma_{Ct}^2, \sigma_{Cr}^2, \sigma_{Cr}^2, \sigma_{Cr}^2)$, where σ_{Ct}^2 represents the translational variance and σ_{Cr}^2 is the rotational variance.

The object pose measurement factor. To estimate the pose of the object from the input RGB image we use CosyPose [2]. CosyPose uses Mask-RCNN [41] to detect known objects bounding boxes, masks, and labels in the image. 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 $\mathbf{r}_O^{k,i} = \text{Log}((T_O^{k,i})^{-1} T_C^k \tilde{T}_{CO}^{k,i})$, where $\tilde{T}_{CO}^{k,i}$ is the pose of the i-th object predicted by the CosyPose from the input frame k, $T_{CO}^{k,i}$ is the estimated temporally consistent pose of the object in frame k and T_C^k is the estimated temporally consistent pose of the camera at time k.

To compute the object pose residual, we need to resolve *data association* between the variables (*i.e.* object poses $T_O^{k,i}$) and the CosyPose measurements (*i.e.* $\tilde{T}_{CO}^{k,i}$). 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 translation and rotation distances are below the manually specified thresholds, denoted $\tau_{\text{outlier\_t}}$ and $\tau_{\text{outlier\_r}}$, we associate the measurement with the variable by creating a corresponding factor in the graph. Otherwise, a new variable is created. This approach creates a robust cost function, enables us to filter out outliers, and track multiple instances of the same object class or various discrete symmetries of the same object.

We observe that the covariance of the CosyPose prediction

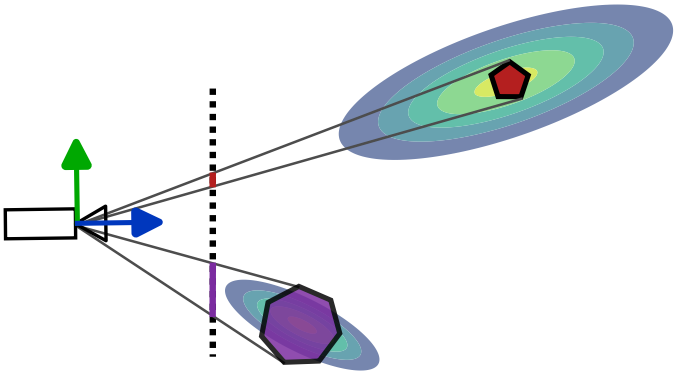

Fig. 3: **Measurement covariance model.** Visualization of the translation covariance model for the object pose estimations. Consider two objects (red and purple) whose projection on the image plane (dotted line) is shown in red and purple, respectively. The size of the covariance ellipsoid depends on 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.

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, as shown in Fig. 3. This is caused by ambiguity in depth estimation, where large 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 position corresponds to the original camera frame and the 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: $\mathbf{\Sigma}_{Ot} = R_{OC'}\mathbf{\Sigma}_{C't}R_{OC'}^T$, 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_{px}. We visualize the covariance model in Fig. 3. The rotational variance in the object frame is defined to be diagonal: $\mathbf{\Sigma}_{Or} = \text{diag}(\sigma^2_{Or}(n_{\text{px}}), \sigma^2_{Or}(n_{\text{px}}), \sigma^2_{Or}(n_{\text{px}}))$. Models for the variances $\sigma^2_{C'xy}, \sigma^2_{C'z}$, and σ^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 covariances assuming zero correlation between them.

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: (i) constant pose, and (ii) constant velocity. The constant pose model for the residual of object i is defined as $\mathbf{r}_M^{k-1:k,i} = \text{Log}((T_O^{k-1,i})^{-1}T_O^{k,i})$ 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 σ^2_{Mt} and σ^2_{Mr} are chosen manually, Δt 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.

Fig. 4: **Qualitative results on HOPE-Video.** Comparison between our method and CosyPose [2] on HOPE-Video sequence (the first row). It can be seen that some of the objects are not detected by CosyPose (the second row). Our temporally smoothed predictions are shown in the last row, largely mitigating the issue of missing detections.

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. (1) is extended with the derivatives for each object and time stamp. The residual is computed as $\mathbf{r}_M^{k-1:k,i} = \left(\boldsymbol{v}^{k,i} - \boldsymbol{v}^{k-1,i}, \quad \boldsymbol{\omega}^{k,i} - \boldsymbol{\omega}^{k-1,i}\right)^\top$, where $\boldsymbol{v}^{k,i}$ and $\boldsymbol{\omega}^{k,i}$ represent the time derivatives of translation and rotation, respectively, 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.

Predictions from the world model. Defining all the factors and solving Eq. (1) gives us a probabilistic world model of all objects and camera poses in time. We solve the optimization for each new measurement in an iterative manner. Incoming pose measurements are assigned to either existing or new variables in a factor graph. Outliers are included as potential valid measurements, but we only predict variables whose estimated uncertainty ellipsoid volume falls below 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 the radius of the object's bounding sphere, we predict only the pose with the lower volume of the covariance ellipsoid, *i.e.* the most confident track. This filtering allow us to track multiple discrete symmetries of the same object and to select only the most confident hypothesis for the robot control.

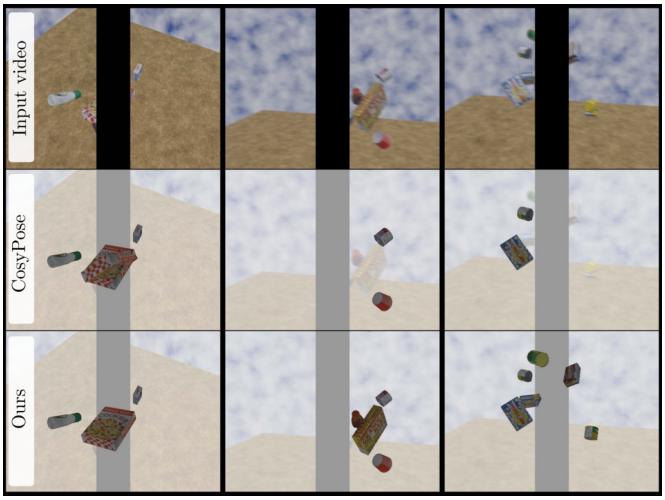

Fig. 5: **Qualitative results on SynthHOPEDynamic.** Comparison between our method and CosyPose [2] on *SynthHOPEDynamic* sequence shown in the first row. The center of the frame is occluded by a black rectangle, and some of the frames are artificially blurred in the input video. It can be seen that some of the objects are not detected by per-frame CosyPose shown in the second row (*e.g.*, frames 2 and 3) or that some outliers are detected (*e.g.*, rotated cookie box in frame 5). Our temporally smoothed predictions (the last row) largely mitigate missing detections and outliers.

IV. EXPERIMENTS

Datasets. Four datasets were used for the quantitative evaluation: (i) Household Objects for Pose Estimation (HOPE-Video) [42] dataset, (ii) YCB-V [43], [44] and (iii) two synthetically rendered datasets created via Blender [45]. 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. 4. YCB-V test set contains 12 video sequences captured by a moving camera observing a static scene with a subset of 21 of the YCB [43] objects.

The remaining datasets are synthetically rendered using HOPE objects [42]. 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 [46] 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. In total, we have three static datasets depicting stationary objects and one dynamic dataset with moving objects.

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 are commonly used in the BOP object pose estimation challenge [47]. Recall is averaged across several thresholds and across three different metrics: (i) Visible Surface Discrepancy (VSD), (ii) Maximum Symmetry-Aware Surface Distance (MSSD), and (iii) Maximum Symmetry-Aware Projection Distance (MSPD). See [8] 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.

Measurement covariance estimation. We empirically observe that translation measurement uncertainty is bigger in the direction of ray pointing towards the object of interest (*i.e.*, standard deviation $\sigma_{C'z}$) and that it depends on the size of the object in the image space, as shown in Fig. 3. We propose to model the dependence of the standard deviation on the number of visible pixels of the object as an exponential function of 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 coordinate frame whose z-axis points toward the center of the object, while rotation uncertainties are estimated in the object coordinate frame. We used the Hope-Video dataset to estimate these uncertainties.

Ablation study. Several thresholds need to be tuned for the proposed filtering method. We manually set horizon H to 30 frames corresponding to 1s in our setup, the outlier prediction thresholds $\tau_{\text{outlier\_t}}$ and $\tau_{\text{outlier\_r}}$ are set to 100 mm and 10°. The prediction thresholds $\tau_{\text{pred\_t}}$, $\tau_{\text{pred\_r}}$, and the variances of the motion models 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. 6; we use it to select two sets of thresholds for each motion model. These sets of thresholds correspond to recall- and precision-oriented parameters as shown in Fig. 6.

Covariance models. In addition to threshold selection, we

TABLE I: **Comparison of different covariance models.** Average Recall and Average Precision are computed by considering all frames of the video and all objects that are visible in the image with at least 5% of the object size. The highest values are shown in bold.

Decoupled	Visibility dependent	frame C'	recall	precision
✓	✓	✓	**0.571**	**0.609**
✓	×	✓	0.570	0.608
✓	✓	×	0.531	0.574
×	✓	N/A	0.483	0.549
×	×	N/A	0.498	0.542

TABLE II: 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" baseline refers to our method modified to use only the last 3 frames. We present recall and precision averaged across VSD, MSSD and MSPD metrics. The best results for recall and precision are shown in bold.

Method	HOPE-Video		YCB-V		SynthHOPEStatic		SynthHOPEDynamic	
	recall	precision	recall	precision	recall	precision	recall	precision
CosyPose [2]	0.39	0.57	0.81	0.72	0.53	0.69	0.44	0.66
Short-horizon	0.40	0.59	0.81	0.72	0.54	0.75	0.40	0.71
Ours (const. pose, recall-oriented)	**0.57**	0.61	**0.82**	0.77	**0.74**	0.85	*not applicable*	
Ours (const. pose, precision-oriented)	0.43	**0.65**	0.81	**0.80**	0.63	**0.89**	*not applicable*	
Ours (const. vel., recall-oriented)	0.47	0.60	0.80	0.79	0.58	0.79	**0.45**	0.76
Ours (const. vel., precision-oriented)	0.43	0.64	0.81	0.77	0.53	0.83	0.41	**0.79**

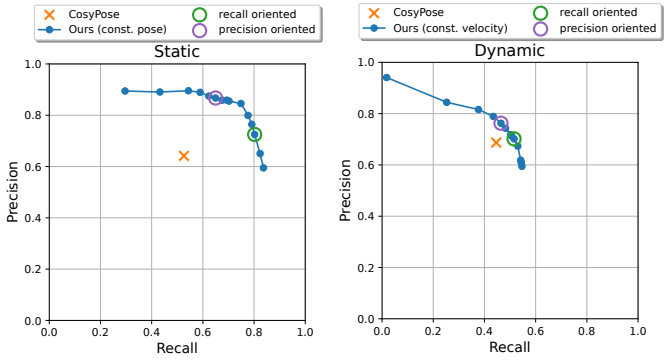

Fig. 6: **Ablation study** for the constant pose motion model (left) evaluated on three scenes of the static synthetic dataset and for the constant velocity motion model (right) evaluated on three scenes of the dynamic synthetic dataset. The precision-recall trade-off is controlled by hyperparameters of our model. Recall oriented parameters are selected such that recall is maximal and precision is at least at CosyPose level. The precision oriented parameters are chosen analogously.

conducted an ablation of different variants of covariance models to evaluate their effect on performance. The models were evaluated on the *HOPEVideo* dataset using the constant pose motion model and recall-oriented parameters. The results are shown in Tab. I. We ablate various aspects of the model: isotropic vs. decoupled along the x, y, and z axes, constant vs. visibility dependent, and expressed in camera frame vs. in rotated frame C'. The results show that our proposed covariance model has the best performance, indicating that the rotated camera frame C' is important for accurate modeling of the covariance. In contrast, the dependency on the object's visibility in the image shows only a minor improvement.

Quantitative evaluation. We evaluated the performance of our method on the four datasets mentioned above. The results are summarized in Tab. II. Two baselines are considered: (i) per-frame CosyPose [2] 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*, *YCB-V* and *SynthHOPEStatic*) both constant pose and constant velocity motion models are evaluated. It can be seen that our methods outperformed the baselines in recall (the recall-oriented variant) while achieving comparable precision. Similarly, for the precision-oriented variant, we outperform the baselines in precision while achieving a comparable recall. The precision-recall trade-off can be controlled by the 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 a similar manner.

Our approach outperforms both the per-frame CosyPose and the short-horizon smoothing baselines. Our results show that longer horizon and smoothing can be used to control recall-precision trade-off and therefore it can be tuned for robustness that is required for feedback robot control.

Comparison with state-of-the-art. To further validate our approach, we extend the evaluation to the *YCB-V* dataset, utilizing the ADD-S and ADD(-S) metrics, which are commonly used to assess object pose estimation accuracy. In Tab. III, we present a comparison of our method against several state-of-the-art approaches on this dataset. Our method consistently outperforms the reported techniques.

Sensitivity to pose estimation backbone. In Tab. IV, we compare our method's performance using different pose estimation backbones. Specifically, we evaluate CosyPose and MegaPose as the underlying pose estimation backbones and then compare them with our approach. The results show that our method improves both recall and precision metrics, regardless of the chosen backbone. This demonstrates that our approach is not tightly coupled to a specific pose estimation model and can be effectively integrated with various state-of-the-art methods without significant performance degradation.

TABLE III: **Comparison to state-of-the-art methods.**

Method	ADD-S	ADD(-S)
CosyPose [2]	0.9	0.84
Merrill et al. [28]	0.9	0.85
Xu et al. [48]	-	0.83
Di et el. [49]	0.91	0.84
Ours	**0.94**	**0.9**

TABLE IV: **Sensitivity to backbone.** We compare our approach using different backbones on the *HOPEVideo* dataset.

Method	recall	precision
CosyPose	0.39	0.57
Ours (CosyPose backbone)	0.57	0.61
MegaPose	0.37	0.54
Ours (MegaPose backbone)	0.54	0.61

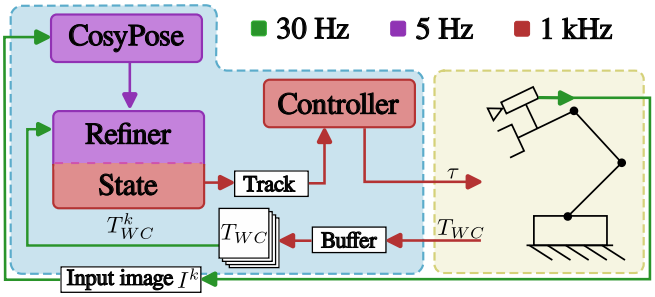

Fig. 7: **The robot control architecture** used for the tracking experiment. First, an image I^k is used with CosyPose to generate object pose estimates. These estimates are then fed into the proposed Refiner along with the camera pose T^k_{WC} whose timestamp corresponds to the time stamp of the input 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 Refiner produces an estimate of the State, *i.e.*, the probabilistic world model. Note that although the world model is updated at 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 Refiner, 30 Hz for the camera, and 1 kHz for the state extrapolation and robot controller.

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 calibrated RealSense D435 camera attached to its end-effector. 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: (i) **Static scene** objects pose estimation in which the robot is guided by a human hand and, while moving, it estimates the poses of objects that are statically placed in front of the robot; (ii) **Dynamic scene** object pose estimation, where the robot remains static and estimates the poses of objects that are moved by a human; and (iii) **Dynamic object tracking** where the robot maintains constant pose with respect to a target 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 the recording of the experiments.

For the dynamic object tracking experiment, we implement the Cartesian impedance control [50] using the estimated target object pose as reference. The controller architecture is visualized in Fig. 7. With this control architecture and using the proposed filtering method, we achieve stable tracking in a challenging scenario in which the object is hidden behind an occluder, as shown in Fig. 8. The analysis of the image stream for the tracking experiment is shown in Fig. 9. The full experiment is shown in the supplementary video.

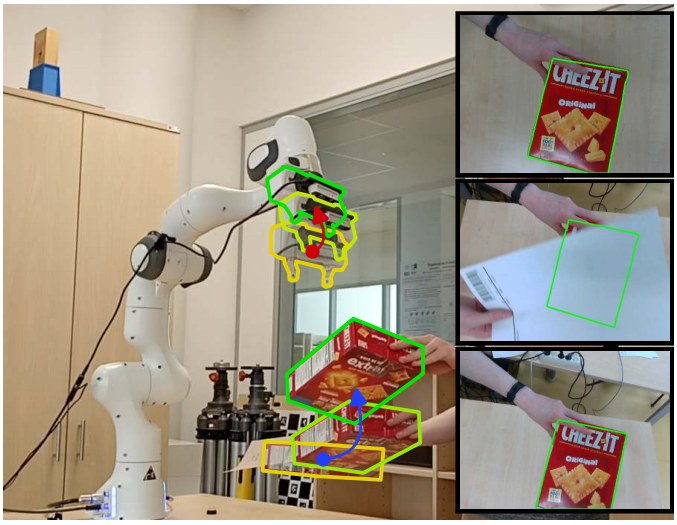

Fig. 8: **Robot tracking experiment.** The illustration depicts a selected sequence of images recorded during an experiment where the robot attempts to maintains a constant relative end-effector transformation with respect to the Cheez-it box from the YCB [43] dataset. During the tracking process, the object is occluded by a sheet of paper, demonstrating the temporal consistency and stability of the refined pose estimates.

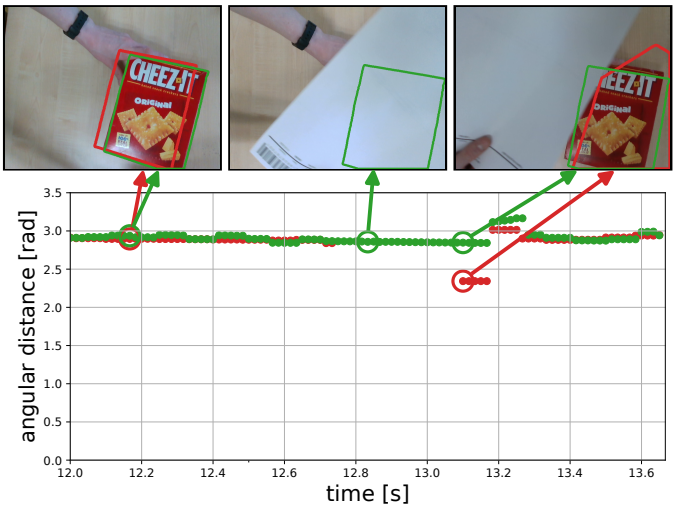

Fig. 9: **Analysis of robot tracking experiment.** The evolution of the object angular distance for the robot tracking experiment. If the object is not occluded, CosyPose and our method predicts the object pose accurately (first frame). However, when object is completely occluded the per-frame evaluation cannot evaluate the pose of the object (second frame). Finally, if the object is partially visible, CosyPose predicts wrong orientation while the proposed estimator remains stable (third frame).

V. CONCLUSION

Accurate and temporally consistent object pose estimation is crucial for robot interaction with both static and dynamically moving objects. This work demonstrates that it is beneficial to consider the full stream of images rather than the per-

frame estimates to achieve robust temporally smooth predictions. The proposed algorithm for probabilistic filtering has been validated both quantitatively on three benchmarks and qualitatively by tracking experiments involving a Panda robot, showing improved results while running in real time.

Limitations. This work addressed object symmetries as separate tracks in the factor graph. Although it works for discrete symmetries (*e.g.* a box), continuous symmetries (*e.g.* a cylinder) would create many low-confidence tracks that would be difficult to use for robot control. This may be addressed by assuming known symmetries and modifying the object pose factor, which is left for future work.

REFERENCES

[1] T. Hodan, M. Sundermeyer, Y. Labbe, V. N. Nguyen, G. Wang, E. Brachmann, B. Drost, V. Lepetit, C. Rother, and J. Matas, "Bop challenge 2023 on detection, segmentation and pose estimation of seen and unseen rigid objects," *arXiv:2403.09799*, 2024.

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

[3] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic, "Megapose: 6d pose estimation of novel objects via render&compare," in *CoRL*, 2022.

[4] M. Stoiber, M. Pfanne, K. H. Strobl, R. Triebel, and A. Albu-Schäffer, "Srt3d: A sparse region-based 3d object tracking approach for the real world," *IJCV*, 2022.

[5] K. Pauwels and D. Kragic, "Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking," in *IROS*, 2015.

[6] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE ITSM*, 2010.

[7] V. Lepetit, "Recent advances in 3d object and hand pose estimation," *arXiv:2006.05927*, 2020.

[8] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "Bop challenge 2020 on 6d object localization," in *ECCV Workshops*, 2020.

[9] V. N. Nguyen, T. Groueix, G. Ponimatkin, V. Lepetit, and T. Hodan, "Cnos: A strong baseline for cad-based novel object segmentation," in *ICCV*, 2023.

[10] V. N. Nguyen, T. Groueix, M. Salzmann, and V. Lepetit, "Gigapose: Fast and robust novel object pose estimation via one correspondence," in *CVPR*, 2024.

[11] E. P. Örnek, Y. Labbé, B. Tekin, L. Ma, C. Keskin, C. Forster, and T. Hodan, "Foundpose: Unseen object pose estimation with foundation features," *arXiv:2311.18809*, 2023.

[12] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," *arXiv:2312.08344*, 2023.

[13] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *ECCV*, 2018.

[14] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains," in *IROS*, 2020.

[15] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking," *TRO*, 2021.

[16] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, 2017.

[17] D. Kragic, H. I. Christensen *et al.*, "Survey on visual servoing for manipulation," 2002.

[18] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *RAL*, 2018.

[19] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *TRO*, 2019.

[20] K. Li, D. DeTone, Y. F. S. Chen, M. Vo, I. Reid, H. Rezatofighi, C. Sweeney, J. Straub, and R. Newcombe, "Odam: Object detection, association, and mapping using posed rgb video," in *ICCV*, 2021.

[21] T. Laidlow and A. J. Davison, "Simultaneous localisation and mapping with quadric surfaces," in *3DV*, 2022.

[22] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *3DV*, 2018.

[23] K. Wada, E. Sucar, S. James, D. Lenton, and A. J. Davison, "Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion," in *CVPR*, 2020.

[24] E. Sucar, K. Wada, and A. Davison, "Nodeslam: Neural object descriptors for multi-view shape reconstruction," in *3DV*, 2020.

[25] Z. Landgraf, R. Scona, T. Laidlow, S. James, S. Leutenegger, and A. J. Davison, "Simstack: A generative shape and instance model for unordered object stacks," in *ICCV*, 2021.

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

[27] J. Fu, Q. Huang, K. Doherty, Y. Wang, and J. J. Leonard, "A multi-hypothesis approach to pose ambiguity in object-based slam," in *IROS*, 2021.

[28] N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang, "Symmetry and uncertainty-aware object slam for 6dof object pose estimation," in *CVPR*, 2022.

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.

[30] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International Workshop on Vision Algorithms*, 2000.

[31] M. Rünz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," in *ICRA*, 2017.

[32] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *ISMAR*, 2018.

[33] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in *ICRA*, 2019.

[34] K. Li, H. Rezatofighi, and I. Reid, "Moltr: Multiple object localization, tracking and reconstruction from monocular rgb videos," *RAL*, 2021.

[35] B. Xu, A. J. Davison, and S. Leutenegger, "Learning to complete object shapes for object-level mapping in dynamic scenes," in *IROS*, 2022.

[36] M. Henein, J. Zhang, R. Mahony, and V. Ila, "Dynamic slam: The need for speed," in *ICRA*, 2020.

[37] J. Issac, M. Wüthrich, C. G. Cifuentes, J. Bohg, S. Trimpe, and S. Schaal, "Depth-based object tracking using a robust gaussian filter," in *ICRA*, 2016.

[38] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, 2017.

[39] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[40] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv:1812.01537*, 2021.

[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[42] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield, "6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark," in *IROS*, 2022.

[43] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *ICAR*, 2015.

[44] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv:1711.00199*, 2017.

[45] *Blender - a 3D modelling and rendering package*, Blender Foundation, 2018. [Online]. Available: http://www.blender.org

[46] A. Ude, B. Nemec, T. Petrić, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," in *ICRA*, 2014.

[47] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis *et al.*, "Bop: Benchmark for 6d object pose estimation," in *ECCV*, 2018.

[48] Y. Xu, K.-Y. Lin, G. Zhang, X. Wang, and H. Li, "Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization," in *CVPR*, 2022.

[49] Y. Di, F. Manhardt, G. Wang, X. Ji, N. Navab, and F. Tombari, "So-pose: Exploiting self-occlusion for direct 6d pose estimation," in *ICCV*, 2021.

[50] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Journal on Robotics and Automation*, 1987.