Technical Report: Supplementary Results for Temporally Consistent Object 6D Pose Estimation

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

I. Introduction

This technical report accompanies our manuscript, titled Temporally Consistent Object 6D Pose Estimation for Robot Control, which has been accepted for publication in IEEE Robotics and Automation Letters. In response to reviewer suggestions and to provide a comprehensive resource for readers, this report presents additional implementation details, ablation studies, and extended evaluations that could not be included in the journal paper due to space constraints.

The key contributions of this technical report are as follows:

- Detailed implementation methodology, including calibration procedures and covariance modeling for pose estimation.
- Extended ablation studies on object and camera pose estimation, robustness to noise, and adaptive object size thresholds.
- Detailed description of pose uncertainty estimates
- Additional quantitative and qualitative evaluations.

This document aims to serve as a supplemental resource, providing more details of the implemented methods.

II. Implementation details

A. Forward kinematics and calibration details

We use the standard calibration procedure: (i) we record a dataset of pairs containing the robot configuration and an image that captures a ChArUco board calibration target; (ii) we use these images for intrinsic calibration using the OpenCV library [1]; (iii) we estimate an initial guess for the camera pose with respect to the robot flange by using hand-eye calibration from the OpenCV library with corresponding poses of the robot flange (computed by forward kinematics) and the target (estimated by the camera); (iv) finally we perform local optimization to estimate the robot joint offsets and the refined camera pose by minimizing the reprojection errors on the ChArUco corners. We did not optimize other robot-related parameters as observed reprojection errors indicate that the current procedure provides adequate robot and hand-eye calibration. This calibration is performed offline before we start any robotics experiment. For the computation of forward kinematics, we use the Pinocchio library [**?**].

III. Additional ablation studies

A. Ablation on the accuracy of the camera and object pose estimates

We performed ablations on *HopeVideo* dataset using our constant pose motion model and recall-oriented parameters. The parameters and modeled covariances remain fixed.

First, we analyzed the effect of noise on object pose measurements. We introduced additional random noise to the translation and rotation of the measured object pose. It is important to note that CosyPose was used to obtain these measurements, so the added noise is applied on top of the inherent uncertainty provided by CosyPose. The results of the noise sensitivity analysis are presented in Tab. 1. It can be seen that both the recall and the precision decreased significantly after applying a fairly large noise with a standard deviation of 20 mm and 5 degrees. However, for smaller noise values, our approach maintains its performance.

In our second experiment, we perform sensitivity analysis for noise applied to the camera pose measurements. The results are shown in Fig. 2. Both recall and precision performance drop significantly as injected noise violates our assumption of accurate camera pose measurements that we get from a camera rigidly attached to the accurate industrial robot.

Authors are with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague. name.surname@cvut.cz

*Equal contribution.

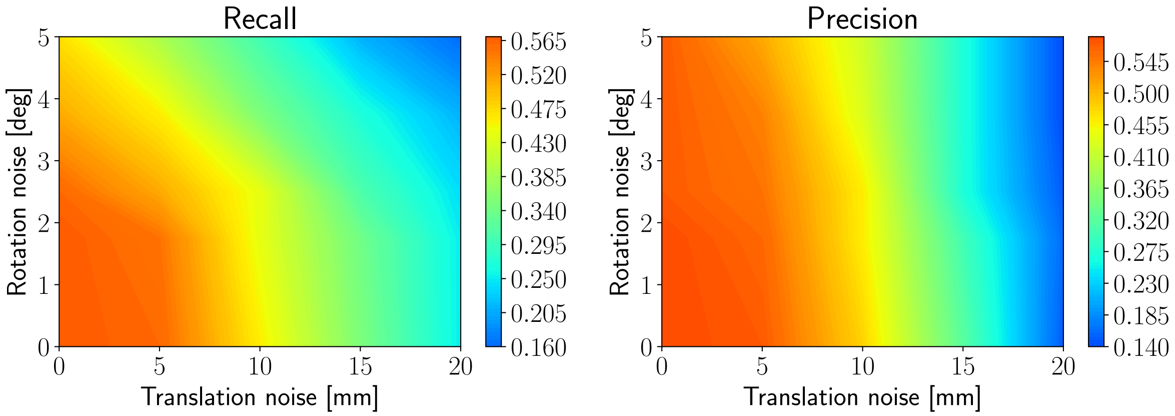


Fig. 1: **Sensitivity analysis for object pose noise.** We applied translation and rotation noise to all object pose measurements and evaluate BOP Average Recall and Average Precision of our method. Zero mean Gaussian noise was applied to CosyPose measurements with standard deviation reported on the x-axis for translation noise and on y-axis for the rotation noise.

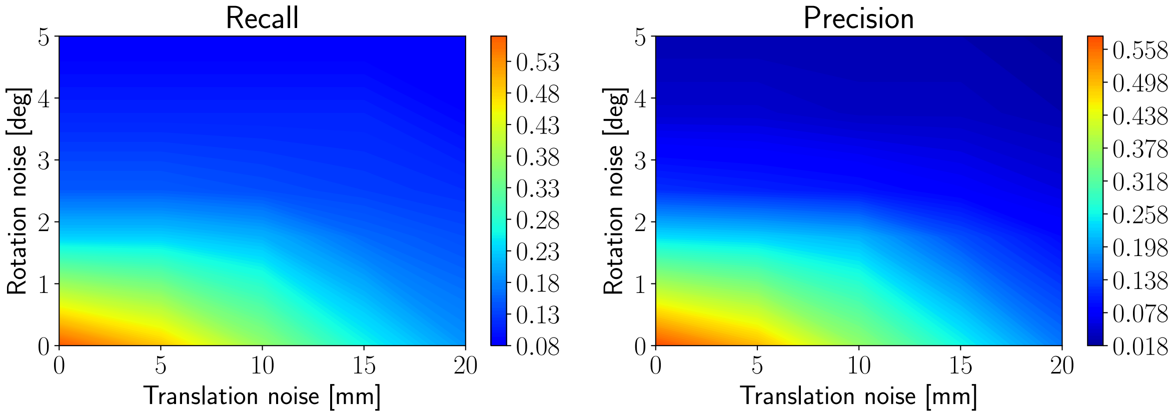


Fig. 2: **Sensitivity analysis for camera pose noise.** We applied translation and rotation noise to all camera pose measurements and evaluate BOP Average Recall and Average Precision of our method. Zero mean Gaussian noise was applied to camera pose measurements with standard deviation reported on the x-axis for translation noise and on y-axis for the rotation noise.

B. Ablation of object size threshold

We conduct an ablation study to compare setting identical-label object distance threshold of 50 mm to adaptive threshold computation. To compute the adaptive threshold, we compute the size of the object using the bounding sphere, *i.e.*, the smallest sphere that fully encloses the object. The radiuses are shown in Fig. 3 with datasets statistics shown in Tab. I. Given the significant variance in radiuses, we have implemented the adaptive threshold as you proposed, using the object radius as the value for thresholding. We compare the adaptive threshold with fixed value of 50 mm and observe an increase in precision as shown in Tab. II.

TABLE I: **Datasets statistics of object sizes.** Statistics on bounding sphere radius for YCBV and HOPE object models. All values are in millimeters.

Dataset	min	max	mean	std
HOPE [2]	38	126	68	25
YCBV [3]	45	135	89	27

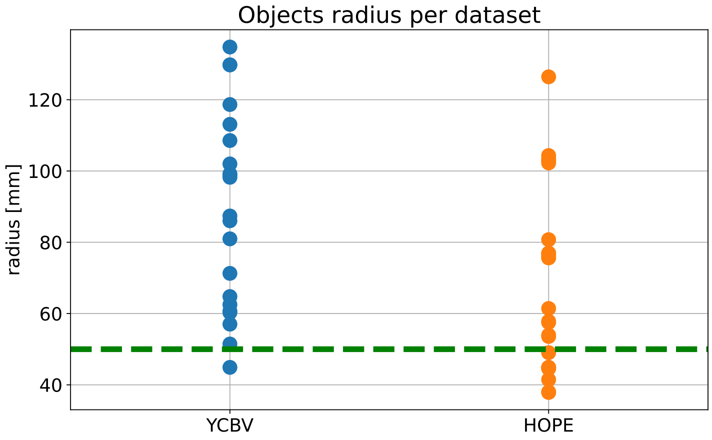


Fig. 3: Radius of bounding sphere for YCBV and HOPE object models. Green dashed line shows the fixed value of 50 mm that we used for prediction thresholding in the initial submission.

TABLE II: **Ablation of adaptive prediction threshold.** We compare the adaptive prediction threshold, which value is set to the radius of object's bounding sphere, with the fixed prediction threshold set to 50 mm. We evaluate results on the HOPEVideo dataset. We use a constant pose motion model and recall-oriented parameters.

	Fixed threshold	Adaptive threshold
Recall	0.57	0.57
Precision	0.58	0.61

C. *Cost function ablation*

Our hypothesis for this ablation is that our data association method is designed to prevent the bias introduced by outliers. If a measurement is an outlier, *i.e.*, far from the tracked variable, it is not associated with that variable. As a result, all associated measurements remain close to the tracked variable, eliminating the need for an explicit robust cost function. Data association is crucial because we do not know the number of objects beforehand; they are dynamically created by our data association mechanism online. To validate this hypothesis, we have implemented the Huber loss [4] for object pose measurement and ablate various values for the Huber loss parameter δ. The results, shown in Tab. III, indicate that a robust cost function is not necessary in the presence of our data association mechanism.

TABLE III: **Ablation of robust cost function.** Huber loss with various thresholding values is compared to L2 norm that we used in optimization. We evaluate results on the *HOPEVideo* dataset. We use a constant pose motion model and recall-oriented parameters.

	Huber Loss				L2 Norm (Ours)
	$\delta = 10^{-3}$	$\delta = 10^{-2}$	$\delta = 5 \cdot 10^{-2}$	$\delta = 10^{-1}$	
Recall	0.5708	0.5706	0.5708	0.5708	0.5725
Precision	0.5846	0.5857	0.5857	0.5846	0.5835

IV. Covariance model for object pose measurements

To estimate the covariance, we use the *HOPEVideo* dataset on which we estimated object poses using CosyPose. Let us denote CosyPose pose estimation in the camera frame as $\tilde{R}_C\,, \tilde{\boldsymbol{t}}_C$ (rotation and translation) and ground-truth object pose in the camera frame as R_C, t_C. We compute the translation error in the camera frame as

$$\boldsymbol{e}_C = \tilde{\boldsymbol{t}}_{\boldsymbol{C}} - \boldsymbol{t}_C\,, \tag{1}$$

and the angular error in the object frame as

$$\theta_O = \|x\| \tag{2}$$

where $\log$ is the SO(3) log function. We filter outliers from these measurements by removing samples with less than 50 visible pixels, with rotation error greater than the 70th percentile (*i.e.*, remove 30% of the worst data) and with translation error greater than the 95th percentile. The rotation error percentile is higher because object symmetries are ignored in our error computation. We plot the translation errors in the first row of Fig. 4.

However, we observe that the variance of the x and y axes is negatively affected by errors in depth estimation. To address this issue, we introduced a rotated camera frame, denoted C'. This frame has the z axis oriented towards the center of the object, for which the pose is estimated, as illustrated in Fig. 3 in the submitted manuscript. To compute the error in the frame C', we first find the rotation matrix that rotates the vectors from the frame C to the frame C', denoted $R_{C'C}$. This is defined uniquely as the rotation that rotates $\boldsymbol{t}_c$ to vector $\boldsymbol{u} = (0, 0, 1)^\top$:

$$\boldsymbol{v} = \overline{\boldsymbol{t}_c} \times \boldsymbol{u}, \tag{3}$$

$$\phi = \arccos(\overline{\boldsymbol{t}_c} \cdot \boldsymbol{u}), \tag{4}$$

$$R_{C'C} = \exp(\phi \boldsymbol{v}), \tag{5}$$

$$\tag{6}$$

where overline indicates normalized vector, symbol ”·” indicates dot product, and $\exp$ is SE3 exponential. Expressing the CosyPose estimation as well as the ground truth pose in frame C' and computing the translation error in C' give us:

$$\tilde{\boldsymbol{t}}_{C'} = R_{C'C} \tilde{\boldsymbol{t}}_C, \tag{7}$$

$$\boldsymbol{t}_{C'} = R_{C'C} \boldsymbol{t}_C, \tag{8}$$

$$\boldsymbol{e}_{C'} = \tilde{\boldsymbol{t}}_{C'} - \boldsymbol{t}_{C'}\,. \tag{9}$$

We plot the translation errors expressed in C' frame in the second row of Fig. 4. Please note that the variance in the x and y axes was reduced.

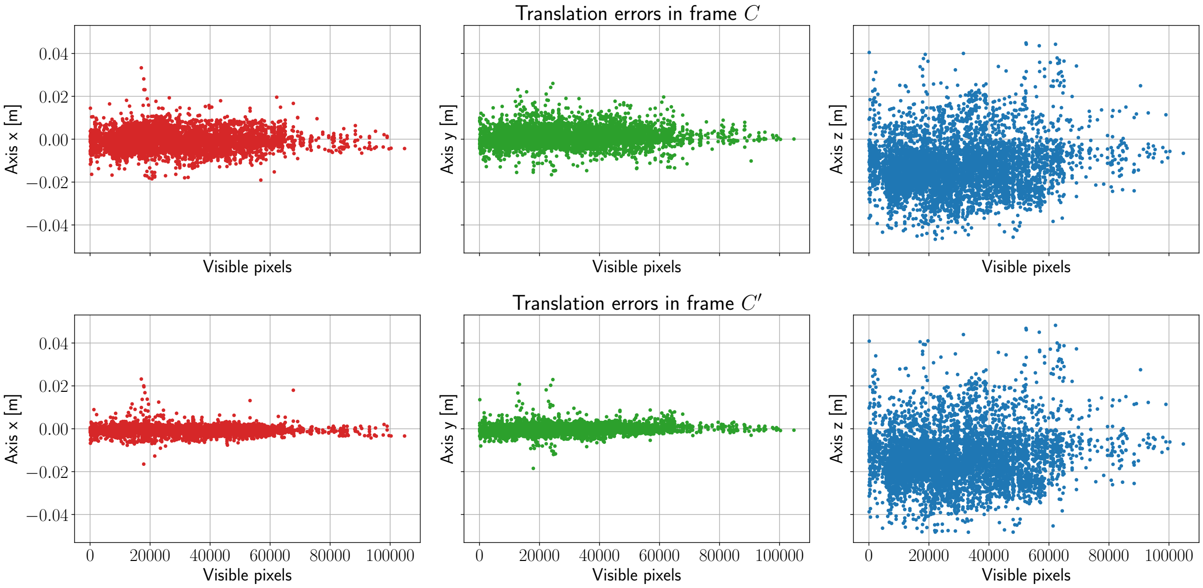


Fig. 4: **Translation errors for object pose estimation with CosyPose.** The first row shows the errors in camera frame C separately for x, y, and z axes (columns). The second row shows the errors in rotated camera frame C'. Please note that the variance of errors in x and y axes is lower in the rotated camera frame (bottom).

Fitting covariance models. The variance in Fig. 4 shows the dependency on object visibility in pixels. We model this dependency with an exponential function of the form: $\sigma(n_{\mathrm{px}}) = a \exp(-b n_{\mathrm{px}})$ where σ is the standard deviation and n_{px} is the number of visible pixels in the image. To find unknown parameters a and b, we split the data into 100 chunks linearly based on the number of visible pixels. For each chunk, we compute n_{px} as the average of the samples in the chunk, and we estimate the standard deviation in the chunk as

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} e^2}\,, \tag{10}$$

where n is the number of samples and e is the computed error (*e.g.*, error in z-axis in the frame C'). We use curve fitting to estimate parameters a and b on the chunk values.

As errors in the x and y axes have similar values, we estimate a single curve for the standard deviation on these axes and another curve for the standard deviation on the z axis. We do this estimation for errors in both frames C and C' and visualize it in Fig. 5. The results show that the fitted model predicts lower covariance in the frame C' for x and y axes.

As a baseline, we also estimate the isotropic covariance model. We distinguish two variants, (i) the isotropic translation standard deviation that does depend on object visibility in the image and (ii) the constant isotropic translation model. Both are visualized in Fig. 6.

The uncertainty of the rotation is estimated in the object frame; we distinguish a constant and size-dependent variants, as shown in Fig. 7. The size-dependent variant is coupled with size-dependent variants of translation uncertainty, while the constant variant is used only for the isotropic constant covariance model. The standard deviations are combined into covariance models as described in Sec. III paragraph *The object pose measurement factor.* in the submitted manuscript.

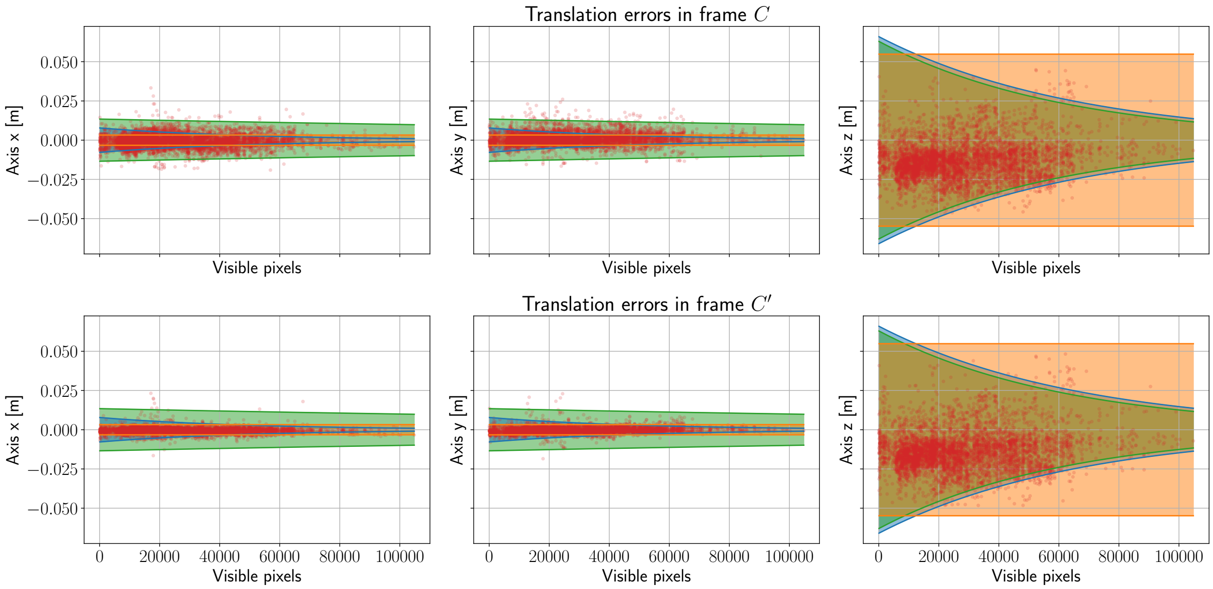


Fig. 5: **Standard deviation estimation.** The error samples that we use for estimation are shown in red in camera frame C (top row) and rotated frame C' (bottom row). We plot standard deviation with 3 σ. We fit standard deviation in C frame (shaded area in green) and in C' frame (shaded area in blue). Both models are plotted in both rows for better visual comparison. It can be seen that standard deviation model in the frame C' is smaller. The standard deviation is also decreasing for more visible objects as the estimation is easier if object covers larger area in the image. In addition, we also plot visibility independent standard deviations estimated for frame C' (orange shaded area).

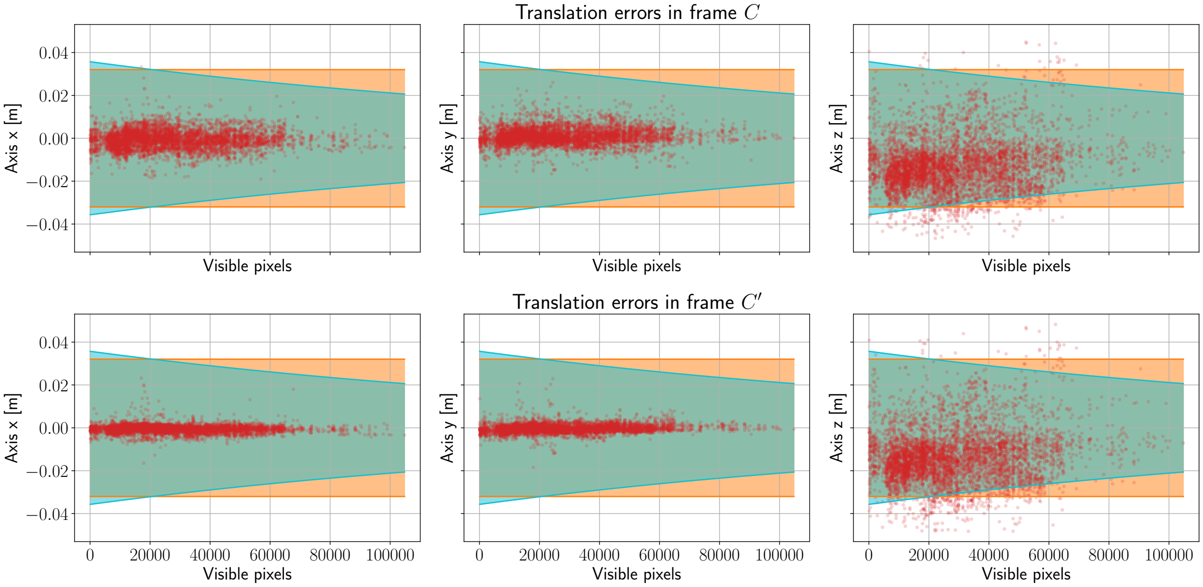


Fig. 6: **Estimation of the Standard deviation for isotropic models.** A single standard deviation parameter is fit for all axes. We plot standard deviation with 3 σ. We distinguish two models: (i) dependent on the object visibility (orange) (ii) constant (cyan).

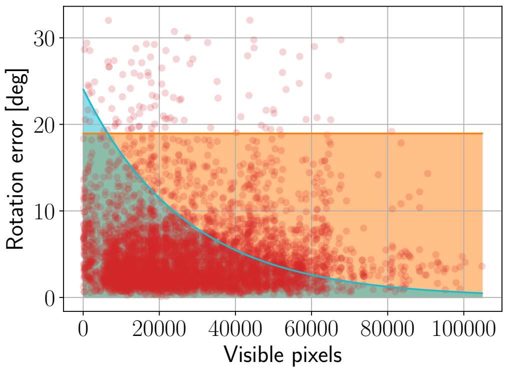


Fig. 7: **Standard deviation estimation for rotation uncertainty.** We plot standard deviation with 3 σ. We distinguish two models: (i) dependent on the object visibility (orange) (ii) constant (cyan).

Ablation of covariance models. We compare the different covariance models on the *HOPEVideo* dataset. We use a constant pose motion model and recall-oriented parameters for the ablation. The results are shown in Tab. IV. Note that the first row of the table corresponds to the covariance model that we proposed to use in the manuscript, it is shown in blue in Fig. 5 for translation and in cyan in Fig. 7 for rotation. The second row corresponds to the visibility independent model shown with orange color in Fig. 5 and Fig. 7 for translation and rotation, respectively. The third row corresponds to the green color for translation in Fig. 5. The remaining two rows represent isotropic translation models, shown in Fig. 6 and Fig. 7 by cyan for the fourth row and orange for the fifth row of the table. 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.

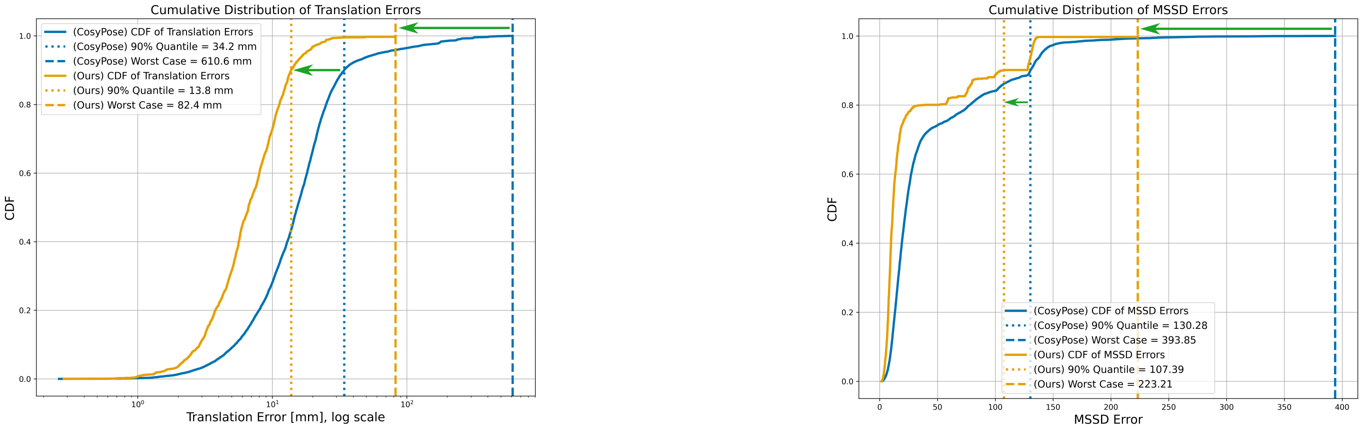
TABLE IV: **Ablation of covariance models.** We compare various covariance models for object pose measurements on the *HOPEVideo* dataset using the constant pose motion model and recall-oriented parameters. Average Recall and Average Precision are computed by considering all frames of the video and all objects that are visible in the image with at least 5% of the object size. We compared various aspects of the covariance models: (i) isotropic vs. decoupled on the x, y and z axes, (ii) visibility dependent vs. constant; and (iii) for the decoupled variant we compared the estimated covariance for the rotated frame C' and the camera frame C. Note that the isotropic model is frame independent. 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

V. Additional evaluation and analysis

A. Translation error analysis

We conducted a quantitative analysis of outliers using the *HOPEVideo* dataset, comparing the translation and MSSD errors from CosyPose with those from our method. The results are presented in Fig. 8, which clearly illustrates the advantages of our approach in reducing pose estimation errors.



(a) Quantitative analysis of Translation Errors.

(b) Quantitative analysis of MSSD Errors.

Fig. 8: **Quantitative analysis of errors for pose estimation.** These figures present the 90th percentile and the worst-case of translation and MSSD errors for both CosyPose and our method with precision-oriented parameters. The translation error plot is in a logarithmic scale. The results illustrate the effectiveness of our approach in reducing pose estimation errors (highlighted with green arrows), particularly in comparison to CosyPose, demonstrating a significant improvement in the reliability of pose estimation.

B. Linear velocity model

Our implementation allows for higher-order derivatives for the motion model, *e.g.* constant acceleration, constant jerk. The linear velocity motion model would correspond to constant acceleration, and we show the results for a dataset with dynamically moving objects in Tab. V. The result shows a comparable recall performance and a small decrease in precision performance compared to the constant velocity motion model. However, the choice of motion model depends on the target application - in our real-world experiments with human-controlled objects, the constant-velocity motion model was sufficient as we do not have a strong prior on object motion.

TABLE V: **Ablation of motion models.** Evaluation of constant velocity and constant acceleration motion models on *SynthHOPEDynamic* dataset for recal-oriented parameters.

	Constant velocity motion model	Constant acceleration motion model
Recall	0.50	0.51
Precision	0.69	0.65

References

[1] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[2] 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.

[3] 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.

[4] P. J. Huber, “Robust estimation of a location parameter,” in *Breakthroughs in statistics: Methodology and distribution*. Springer, 1992, pp. 492–518.