2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

# Color-Pair Guided Robust Zero-Shot 6D Pose Estimation and Tracking of Cluttered Objects on Edge Devices

Xingjian Yang[1] and Ashis G. Banerjee[2]

***Abstract*— Robust 6D pose estimation of novel textured objects under challenging illumination remains a significant challenge, often requiring a trade-off between accurate initial pose estimation and efficient real-time tracking. We present a unified framework explicitly designed for efficient execution on edge devices, which synergizes a robust initial estimation module with a fast motion-based tracker. The key to our approach is a shared, lighting-invariant color-pair feature representation that forms a consistent foundation for both stages. For initial estimation, this feature facilitates robust registration between the live RGB-D view and the object's 3D mesh. For tracking, the same feature logic validates temporal correspondences, enabling a lightweight model to reliably regress the object's motion. Extensive experiments on benchmark datasets demonstrate that our integrated approach is both effective and robust, providing competitive pose estimation accuracy while maintaining high-fidelity tracking even through abrupt pose changes. Code: `https://github.com/smartslab/Color-Pair-Guided-Zero-Shot-6D-Pose`**

Fig. 1. Visualization of color-pair based classification. Left to right: input image, color-pairs visualized by class, and classified surface patches based on matching. Rows compare results on synthetic (rendered) vs. real (captured) images under varying illumination to demonstrate robustness.

## I. Introduction

Estimation of an object's six-degree-of-freedom (6D) pose, which involves determining its 3D rotation and 3D translation relative to a camera, is a fundamental task in computer vision and robotics [1]. Accurate 6D pose information is crucial for a variety of applications, ranging from robotic manipulation and grasping in industrial and household environments to immersive experiences in augmented and mixed reality. The ability of an autonomous system to precisely locate and determine the orientation of objects is a key prerequisite for meaningful physical interaction. Furthermore, in dynamic scenarios, this capability must extend beyond single-frame estimation to continuous, real-time tracking, providing the temporal coherence necessary for tasks such as closed-loop robotic control.

Historically, pose estimation has focused on instance-level methods, which require costly, object-specific training and thus cannot generalize to new objects. While category-level approaches can handle unseen instances within a known class, they still fail to address entirely novel categories. This fundamental generalization gap has motivated the shift to zero-shot pose estimation, which aims to handle unseen textured objects given their 3D CAD models. Recent advances in zero-shot pose estimation are largely driven by Vision Foundation Models (VFMs) such as DINOv2 [2] and CLIP [3]. These models provide powerful descriptors that enable reliable correspondence matching between an input image and rendered templates, bridging the synthetic-to-real domain gap without task-specific fine-tuning. However, this feature matching paradigm often relies on exhaustive comparisons against a vast set of templates, resulting in significant latency. This computational cost is a critical bottleneck for real-time tracking, where systems must be resilient to drift and recover from occlusions without frequent, costly re-initialization. Further, establishing robust correspondences remains difficult for challenging objects, such as those that are textureless or symmetric. Concurrently, emerging research explores generative models, such as diffusion [4] and vision-language models (VLMs) [5], to address these limitations.

In this paper, we present a unified framework for 6D object pose estimation and tracking, specifically designed for objects with distinct surface textures under challenging real-world lighting conditions. Our approach integrates a feature representation for robust initial localization with a fast, motion-based model for real-time tracking in subsequent frames. Our main contributions are as follows:

- A novel, lighting-invariant color-pair feature descriptor that enables both robust initial pose estimation and reliable correspondence filtering for tracking (see Fig. 1)
- A robust and efficient initial pose estimation algorithm that leverages the spatial distribution of local texture features to register classified edge point clouds from a segmented RGB-D image against a 3D model
- An efficient 6D pose tracking module that leverages optical flow and depth cues with a viewpoint-invariant rotation estimator to achieve robustness against abrupt pose changes

[1]X. Yang is with the Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA. yxj1995@uw.edu
[2]A. G. Banerjee is with the Department of Industrial & Systems Engineering and the Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA. ashisb@uw.edu

## II. Related Works

Contemporary zero-shot pose estimation approaches typically leverage 3D models and are distinguished by their methodology: some train on large, diverse datasets to learn generalizable features; others build on pre-trained foundation models for training-free inference; and still others explore alternative problem formulations.

**Training-based generalization for unseen objects.** A key line of work trains on large, diverse (often synthetic) datasets to generalize to novel objects. MegaPose [6] employs an effective render-and-compare paradigm that iteratively refines a coarse pose by comparing the observed image with renderings of the CAD model. FoundationPose [7] unifies pose estimation and tracking within one framework trained on large-scale synthetic data. GenFlow [8] learns to predict optical flow between rendered and observed images and refines pose iteratively under 3D shape guidance. Other approaches include GigaPose [9], a retrieval-style orientation pipeline based on fusing local similarities.

**Training-free estimation with foundation models.** Another direction avoids task-specific training by leveraging frozen 2D/3D foundation models to establish reliable correspondences. RGB-D methods like FreeZeV2 [10] combine vision-foundation patch features with geometric priors to construct 3D descriptors for 3D–3D registration. ZeroPose [11] adopts a related correspondence-driven strategy, while SAM-6D [12] uses SAM for object masks before matching the masked cloud to rendered views. In the RGB-only setting, FoundPose [13] shows that intermediate ViT patch descriptors are sufficient for matching real images to pre-rendered CAD templates, enabling PnP/RANSAC pipelines without task-specific training. ZS6D [14] follows this correspondence or template-matching approach with frozen ViT features. Diffusion-backbone features have also been explored as competitive alternatives [4].

**Alternative representations and model-free formulations.** Beyond correspondence and render-and-compare pipelines, several works explore alternative representations or problem formulations. NOPE [15] and Zero123-6D [16] use image-to-3D view synthesis to generate pseudo-templates from a few reference images, reducing reliance on explicit CAD models. Any6D [17] first reconstructs a 3D shape, then jointly refines its scale and pose. Other representations include graph-based methods like 3DPoseLite [18] and 3D Gaussian Splatting in GS2Pose [19] for differentiable refinement. Finally, some works reframe the problem, using VLMs to locate objects by text [20] or formulating refinement as an action-decision process [21].

**6D Object Tracking.** While the above methods focus on single-frame estimation, a parallel line of work addresses 6D object tracking by modeling inter-frame motion and consistency for more efficient and stable pose updates. A classic paradigm is probabilistic filtering, with methods such as PoseRBPF [22] using a particle filter to maintain a posterior over object orientations. Others, like ROFT [23], employ Kalman filters to fuse cues such as optical flow for tracking fast-moving objects. More recently, learning-based approaches have become dominant, training networks to directly regress the relative transformation between frames (e.g., SE(3)-TrackNet [24]) or to track a sparse set of keypoints. Methods like 6-PACK [25] learn category-specific keypoints, while model-free approaches such as BundleTrack [26] track general features and use pose graph optimization for long-term consistency.

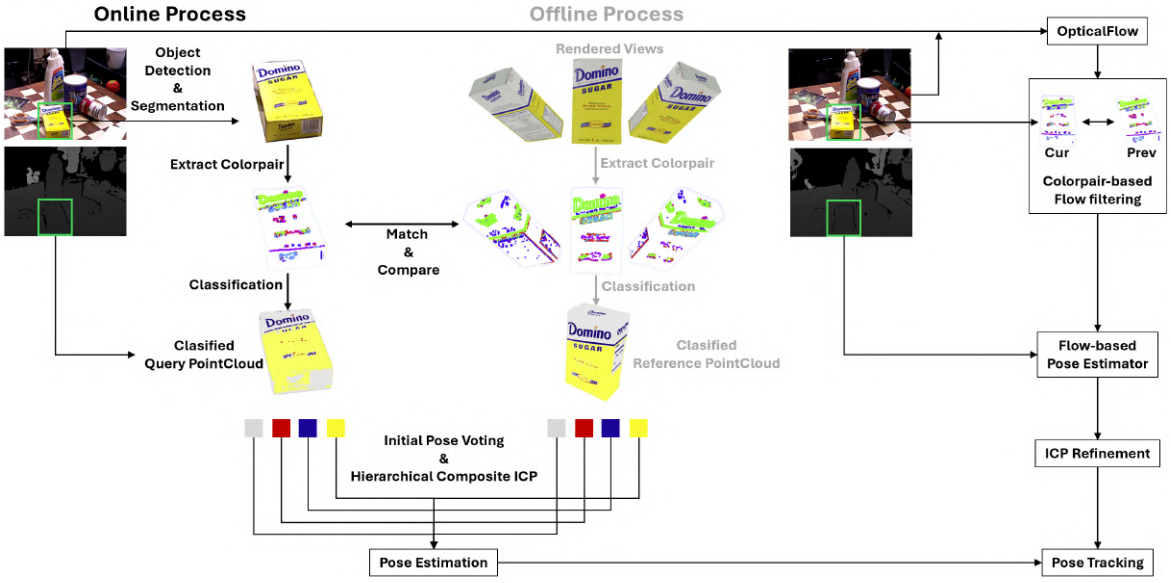


Fig. 2. Overview of the pipeline. Pose estimation: After object detection and segmentation, color-pair features are extracted from the input RGB-D frame. These are matched against features from offline rendered views to generate a classified point cloud, which is then used in a pose voting and hierarchical ICP stage to determine the pose. Pose tracking: The pose is updated frame-to-frame by feeding color-pair filtered optical flow into a motion estimator that predicts the inter-frame transformation.

## III. METHODS

Given an RGB-D image of a textured object, our goal is to determine its full 6D pose and subsequently track its motion in a video sequence. Effectively tackling this problem requires a synergistic approach that leverages illumination-invariant texture features for robust initial localization and efficient motion cues for real-time tracking. Our unified framework, depicted in Fig. 2, achieves this by integrating a novel pose estimation pipeline with a lightweight tracking module.

Our pipeline begins by localizing the target object in the input RGB-D frame. We first employ a YOLO detector to obtain a coarse bounding box, which is then refined by NanoSAM [27] to yield a precise instance mask. From the masked region, we introduce our core contribution for pose estimation: a novel feature representation based on classified color pairs. By sampling colors on both sides of texture edges, we create descriptors that are robust to illumination changes. We then generate a classified edge point cloud from the image and register it against a pre-computed, similarly classified, ground truth point cloud derived from the object's 3D mesh. This registration is performed in two steps: a Hough voting algorithm provides a coarse initial pose, which is then refined using a weighted Iterative Closest Point (ICP) algorithm that jointly optimizes correspondences across all color pair classes. Once initialized, our tracking module takes over. It computes dense optical flow between consecutive frames to establish temporal correspondences. These 2D matches, augmented with 3D information from the depth sensor, are fed into a lightweight regression model that predicts the inter-frame transformation, enabling efficient and continuous 6D pose tracking. We now describe each module of our framework in detail.

### A. ColorPair Similarity

The foundation of our unified framework rests on a key observation: the relational characteristics of color pairs across local texture edges represent an intrinsic property of an object's surface that is remarkably robust to domain shifts. This principle holds true whether the object is represented as a synthetically rendered 3D model or captured in a real-world photograph under challenging illumination. This inherent, domain-invariant consistency is the very property that allows our system to bridge the gap between the offline model and online sensor data. Our colorpair Similarity metric is therefore not merely an engineered function, but a principled method designed to quantify this stable physical phenomenon, creating a consistent foundation for both initial pose estimation and subsequent tracking.

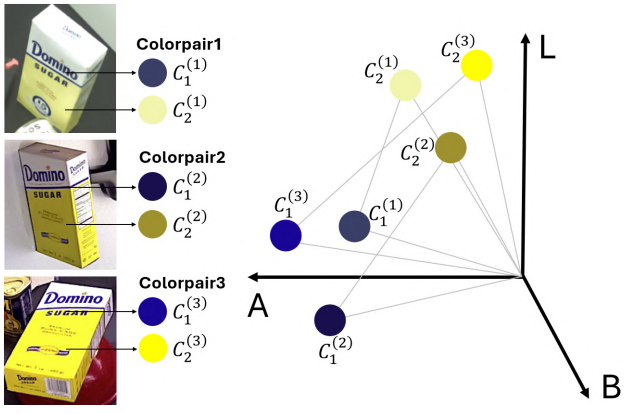


Fig. 3. Color-pair consistency under illumination changes. Left: Extracted color-pairs from the same object under different lighting conditions. Right: Visualization in the CIELAB color space, where each pair with the origin forms a triangle whose pose and edge relations remain consistent across lighting variations.

To compare two colorpairs', $i$ and $j$, we represent them as triangles, $\Delta OC_1^{(i)}C_2^{(i)}$ and $\Delta OC_1^{(j)}C_2^{(j)}$, within the LAB color space, where $O$ is the origin (black) (see Fig. 3). The similarity score is the product of three geometric comparisons:

- **Directional Alignment:** Assesses the alignment between corresponding sides $OC_1^{(i)}$ and $OC_1^{(j)}$, and between $OC_2^{(i)}$ and $OC_2^{(j)}$. The lightness (L*) channel's contribution is down-weighted to prioritize chrominance.
- **Internal Contrast:** Evaluates the directional alignment between the triangles' third sides, the vectors $C_1^{(i)}C_2^{(i)}$ and $C_1^{(j)}C_2^{(j)}$. Its lightness component is similarly down-weighted for robustness.
- **Relative Luminance:** Compares the internal luminance ratio of vertices $(C_1^{(i)}, C_2^{(i)})$ to that of $(C_1^{(j)}, C_2^{(j)})$, enforcing consistent internal lighting dynamics.

Finally, to ensure the metric is robust against the arbitrary order in which colors on either side of an edge are sampled, the comparison is performed symmetrically. An observed feature from the scene is matched against both a reference feature and its color-swapped counterpart. The maximum of the two resulting scores is taken as the definitive similarity, ensuring that the matching is invariant to the sampling order. This multi-faceted function provides a highly discriminative score for reliably identifying corresponding features, transforming a fundamental physical observation into a powerful computational tool that unifies the distinct challenges of initial estimation and continuous tracking.

### B. Edge Color-Pair Extraction

**Preprocessing** To ensure reliable feature extraction, we first suppress chromatic noise using a lightweight two-stage bilateral filter. Then a joint bilateral filter on the chromatic channels, guided by luminance, further attenuates spurious color variations, resulting in a cleaner input image for subsequent processing.

**Unified Gradient Computation** We convert the image to the CIELAB color space for robustness to illumination changes. A unified gradient is then computed where its magnitude is calculated by aggregating the energy from all three channels, while its direction is governed by the channel with the strongest local change.

**Edge Localization and Thickness Estimation** Building on the gradient map, we apply a non-maximal suppression

(NMS) procedure to distill the gradient field into crisp, one-pixel-wide contour centerlines. Our method extends NMS by concurrently measuring edge thickness. This is achieved by an iterative process where pixels within a gradient region associate with their strongest neighbor, ultimately converging to a central ridge. The number of pixels that are grouped onto each ridge point naturally defines its thickness, yielding a width map that encodes the sharpness of each contour.

**Color-Pair Sampling** The final step is to sample representative colors from both sides of each contour centerline, guided by its direction and measured width (see Fig. 4). To overcome the limitations of sampling a single point at an imprecise sub-pixel location, we first sample a profile of multiple candidates on each side of the edge at distances proportional to the local contour thickness. To robustly aggregate these candidates, we introduce a gating mechanism that operates in a statistically whitened CIELAB space. This whitening normalizes the local color distribution, leading to more stable geometric computations. Within this space, an initial color transition axis is defined, and any sampled points that are significant outliers with respect to this axis are rejected. A final representative color for each side is then computed as the median of the remaining inlier samples. This multi-stage process—combining adaptive multi-sampling with a robust gating mechanism—ensures the final color-pair accurately represents the underlying surface appearances, even across complex textures or noisy regions.

### C. Initial Pose Voting

To prepare for voting, we generate a classified scene point cloud. This is achieved by first robustly matching extracted color-pairs to a reference database. High-confidence matches are then decoupled into their constituent representative colors, which in turn are used to assign a semantic class and weight to each scene point. Our approach addresses the inherent challenge of initial pose estimation in the high-dimensional SE(3) search space by reframing the continuous alignment problem into a discrete, computationally efficient geometric hashing query. This method is designed to be robust against noise, clutter, and viewpoint variations by building a consensus from numerous locally consistent geometric correspondences.

The core of our method is an offline database constructed

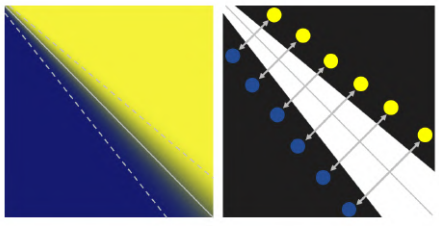

Fig. 4. Thickness-aware color-pair extraction. Left: An edge image showing a diagonal centerline whose progressively increasing width is visualized by diverging dashed lines. Right: Color-pair extraction, where the width of the central white band represents the local edge thickness that guides the color sampling.

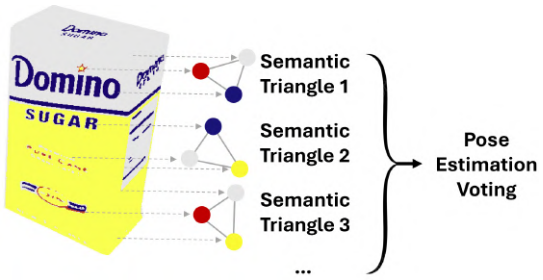


Fig. 5. Illustration of the initial pose voting scheme. A classified point cloud is shown, from which several semantic triangles (1, 2, 3, ...) are constructed by connecting vertices of different categories. These semantic triangles collectively contribute to the subsequent pose estimation voting process.

from the reference model's point cloud. We prioritize the selection of structurally significant semantic triangles, which are chosen via a quality metric that considers both geometric stability (triangle area) and global distinctiveness (the distance of the triangle's centroid from the object's center). This process populates the database with reliable geometric anchors. For each selected triangle, we compute a 7-dimensional, pose-invariant feature key, creating a hash table that maps these canonical features to their 3D vertex coordinates. This primary approach is supplemented by Point Pair Feature (PPF) based templates to maintain robustness in scenarios with fewer than three visible object classes, as our core feature relies on triangles constructed from points across three distinct classes.

During the online stage, the algorithm efficiently processes a given scene point cloud by extracting local triangles, computing their feature keys, and performing a fast lookup in the pre-built database (see Fig. 5) for an illustration). This query yields a set of candidate correspondences between scene and model triangles. For one-to-many match scenarios, where a single scene feature corresponds to multiple database entries, we employ a center-aligned equidistant sampling strategy to generate a diverse set of initial pose hypotheses using the Kabsch algorithm.

Finally, the large number of generated hypotheses, each acting as a "weak" estimator, are processed through a two-stage, coarse-to-fine Hough voting framework to reach a robust consensus. This framework ensures that a single scene feature does not disproportionately influence the voting outcome, thereby suppressing noise from ambiguous matches. By applying a non-maximum suppression strategy to the 6D pose space, we identify the most densely populated pose clusters. A final weighted averaging of the poses within these high-density clusters yields a small, consistent set of initial alignments suitable for subsequent fine-grained refinement.

### D. Pose Refinement

Following the initial coarse alignment, we perform a fine-grained pose refinement using a custom Iterative Closest Point (ICP) algorithm. Our method employs a composite optimization strategy that simultaneously leverages both the

geometry of the individual semantic parts and the overall structure of the object within each iteration.

Specifically, the refinement process is driven by two parallel sources of geometric constraints. First, we treat the point clouds of each common semantic class (e.g., pcd1, pcd2, etc.) as distinct entities. By concurrently optimizing their alignments, we implicitly enforce the fixed spatial relationships among the object components, as they all contribute to a single, unified pose update. This preserves the high-fidelity geometric details of each part. Second, to ensure global coherence, we also form an aggregated point cloud (pcd0) by combining all individual classes. This complete point cloud, representing the overall geometric structure, is included as an additional constraint in the optimization. It acts as a holistic guide, ensuring that alignment is robust against noise, occlusion, or feature-poor regions in any part.

In every ICP iteration, correspondences from both the individual class point clouds and the aggregated point cloud are established and then jointly minimized to compute a single incremental pose update. This dual source constraint mechanism allows our method to achieve both local precision and global stability. The entire process is embedded within a multi-resolution hierarchy and uses a robust kernel-based weighting for individual point pairs to guarantee convergence and precision.

### *E. Pose Tracking*

Once an initial 6D pose is established, our system begins tracking the object's motion across subsequent frames. The core task is to compute the incremental transformation between the previous and current frames, allowing for a continuous and efficient pose update.

**Feature Correspondence and 3D Lifting.** We first establish robust 2D point correspondences between consecutive frames using a pre-trained optical flow network. To ensure high quality, these correspondences are filtered by propagating the object's mask and enforcing color-pair feature consistency—the same logic applied in the pose estimation section, resulting in a sparse set of reliable matches. These 2D matches are then lifted to 3D space using the camera intrinsics and corresponding depth maps, creating two point clouds in the camera's coordinate system, one for each frame.

**Viewpoint-Invariant Feature Generation.** A critical preprocessing step in our procedural learning approach is the generation of viewpoint-invariant features (see Fig. 6). To ensure the model learns general principles of geometric motion rather than object-specific patterns, we first normalize both point clouds, $P_{cam,1}$ and $P_{cam,2}$, into a canonical reference frame. This normalization is a two-stage process. First, we center each point cloud by subtracting its centroid, $t_{est}$. Second, we compute a perspective correction matrix $R_p$ that reorients the point cloud into a standardized "look-at" pose. This matrix is constructed by defining a new basis where the z-axis ($z'$) aligns with the camera-to-centroid vector. The x-axis ($x'$) and y-axis ($y'$) are then determined via cross products with a predefined "up" vector, $u$ (typically $[0, -1, 0]^T$):

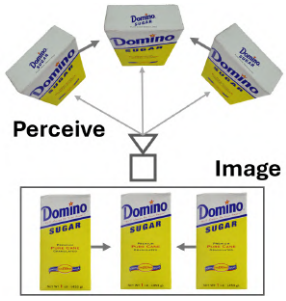


Fig. 6. Illustration of Perspective Normalization. This process converts a 3D point cloud into a canonical coordinate frame to create viewpoint-invariant features. By aligning varied initial views (top) to a consistent appearance (bottom), this step decouples the object's translation from its rotation, simplifying the learning task.

$$z' = \frac{t_{est}}{\|t_{est}\|} \quad x' = \frac{u \times z'}{\|u \times z'\|} \quad y' = z' \times x' \tag{1}$$

The full transformation applied to the original point cloud $P_{cam}$ is:

$$P_{corrected} = R_p^T \cdot (P_{cam} - t_{est}) \tag{2}$$

After transforming both point clouds into this shared canonical space, we construct a 15-dimensional feature vector for each point correspondence, concatenating the normalized 3D coordinates, 2D projections, and the 3D/2D displacement flows between frames. This normalized feature set decouples the problem: by making the input independent of the object's absolute pose, the network can focus solely on regressing the relative rotation between the two canonicalized point clouds. This decoupling is key to the model's generalization across arbitrary geometries.

**Relative Pose Estimation via Procedural Learning.** The viewpoint-invariant feature vectors serve as input to our estimation model. To ensure the model generalizes to arbitrary object geometries and complex motions, we depart from traditional training on finite datasets. Instead, we introduce a novel training paradigm based on real-time procedural data generation. During training, we programmatically generate a virtually infinite stream of complex, randomized 3D shapes, each with unique topological features. For each unique shape, we simulate a realistic tracking event by rendering it from two distinct viewpoints, yielding paired, noisy point clouds with precise ground truth for the relative transformation. This methodology forces the model to learn the underlying geometric principles of motion from first principles, rather than memorizing features of specific objects.

This procedurally generated data is used to train our rotation estimator, an Attention-DGCNN that enhances the standard DGCNN [28] architecture by replacing its global max-pooling layer with a self-attention mechanism, which is designed to regress the relative rotation from the viewpoint-invariant features. The predicted relative rotation then serves as a strong prior for the subsequent estimation of the relative translation. Together, they form an incremental transformation that updates the object's pose. To mitigate drift from

accumulated prediction errors, we refine the updated pose using a fixed small number of point-to-plane ICP iterations. This process grounds the pose by re-aligning the object's model with the current depth map, and this refined pose is then carried forward to the next tracking cycle.

## IV. Experiments

### A. Datasets and Setup

We evaluate our 6D pose estimation method on 12 test videos in the YCB-Video (YCB-V) dataset that are provided in the BOP format. YCB-V contains 21 objects in cluttered scenes with varying illumination. As our method relies on texture cues, we restrict our evaluation to fifteen texture-rich objects (see Table I). Our approach performs zero-shot pose estimation and assumes a preceding object detection stage. To simulate this input, we use the ground-truth bounding box perturbed by a random positional offset of up to 10% of its height and width. This bounding box then serves as a prompt for the NanoSAM [27] model to generate the object's segmentation mask. The evaluation metrics are computed using the dataset's official camera intrinsics and ground-truth poses. To compare against state-of-the-art methods, we filter their publicly available results from the BOP Challenge website for our selected object subset.

We evaluate our 6D pose tracking performance on six labeled synthetic sequences from the Fast-YCB dataset [23], which is designed for moderate-to-fast object motions. We follow the official ROFT protocol for evaluation. All trackers are initialized using the DOPE pose from the first frame. Our tracker, which employs NeuFlow v2 [29] for optical flow estimation, processes the main 30 Hz RGB-D stream. To simulate a resource-constrained scenario with delayed perception, the protocol stipulates that perception inputs, such as segmentation masks from Mask R-CNN, are provided at a delayed and non-synchronized 5 Hz frequency. Drift is mitigated by validating the flow-warped mask against these periodic inputs, re-triggering the initialization module if inconsistency is detected. The evaluated sequences feature the following objects: cracker box, sugar box, tomato soup can, mustard bottle, gelatin box, and potted meat can.

### B. Evaluation and Analysis

We evaluate 6D pose accuracy using ADD and ADD-S metrics. Both compute the mean distance between model points transformed by the predicted and ground-truth poses—ADD uses correspondences, while ADD-S uses nearest neighbors. A pose is correct if the error is less than 10% of the object diameter.

**Pose Estimation.** We compare our method against several state-of-the-art methods including FoundationPose [7], FoundPose [13], FreeZeV2 [10], GenFlow [8], and GigaPose [9] on the YCB-Video dataset. As shown in Table I, our method achieves an average ADD-S of 78.6 and an average ADD of 60.9. Although methods such as FoundationPose (92.3 ADD-S) or FreeZeV2 (86.3 ADD-S) achieve higher accuracy scores, our approach remains competitive with top-performing methods, indicating a favorable balance between accuracy and robustness.

In terms of efficiency, these competing methods generally rely on powerful GPUs (e.g., A100 or V100) and employ a two-stage pipeline. The initial zero-shot segmentation step, often performed by models like CNOS, introduces a variable overhead: it requires approximately 0.2-0.4 seconds per frame with an efficient FastSAM [32] model, which increases to over 1.5 seconds using the original SAM [33]. This overhead contributes to total reported processing times that often span multiple seconds (e.g., FoundationPose at 9.9s; GenFlow at 16.7s; FreeZeV2 at 14.3s). In contrast, our method assumes input from a fast object detector (e.g., YOLO) and is designed for on-device deployment. It achieves 7 FPS on a Jetson AGX Orin while processing five objects simultaneously, demonstrating its suitability for embedded, real-time applications with further optimization.

This distinction highlight a different design focus: while not surpassing the top methods in raw accuracy, our approach provides a pragmatic alternative, delivering competitive performance at a substantially lower computational cost. This makes it particularly well-suited for robotic applications where low latency and onboard deployment are critical.

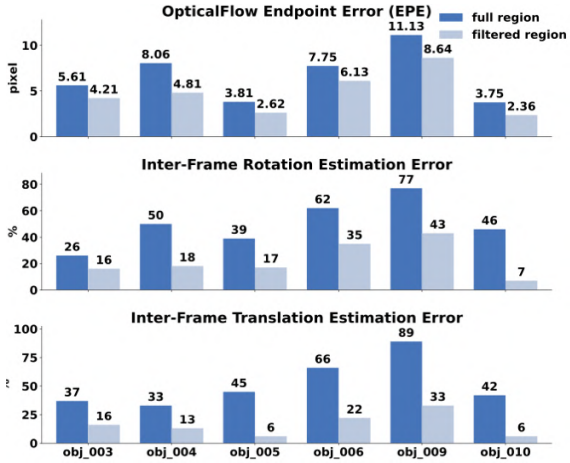


Fig. 7. Quantitative comparison of Translation Error, Rotation Error, and Endpoint Error (EPE) across different objects. Translation Error denotes the magnitude of translational residuals normalized by the true translational motion. Rotation Error is the residual rotation angle normalized by the true rotational motion. EPE represents the average 2D endpoint error of predicted scene flow. Results compare pose estimated from the flow of points sampled on the full object surface versus pose estimated from the flow of points sampled on color-pair–filtered surface regions.

**Pose Tracking.** As detailed in Table II, our proposed method demonstrates a substantial improvement in performance over all baseline approaches on the Fast-YCB dataset, which features objects undergoing rapid motion. Specifically, our approach achieves an average ADD score of 93.13. In addition to its superior accuracy, our method is highly efficient, capable of processing at over 20 FPS on the NVIDIA Jetson Orin platform, underscoring its suitability for real-world robotic applications.

To further evaluate the robustness of our tracker, we

conducted an experiment where the input is downsampled to every 5th frame, effectively simulating scenarios with abrupt pose changes. Remarkably, even under this challenging condition, our method achieves an average ADD score of 84.66. This result not only demonstrates strong robustness to sudden displacements, but also surpasses the performance of all competing methods that operate on the full, dense video sequence. These findings collectively validate that our approach offers a highly accurate, efficient, and robust solution for 6D object pose tracking.

**Ablation Study.** We conduct an ablation study to validate our core contribution: using a color-pair similarity metric to filter correspondences for robust pose estimation. To isolate the quality of the resulting matches, we estimate the relative pose between frame pairs sampled at a five-frame interval using a raw estimator without any subsequent ICP refinement. This setup stresses the ability to find geometrically consistent correspondences under significant motion. We compare the pose accuracy derived from two inputs: (1) all available optical flow correspondences, and (2) only those correspondences deemed consistent by our color-pair metric. As shown in Fig. 7, while the optical flow's endpoint error (EPE) is only modestly reduced, the impact on the downstream pose calculation is substantial. We observe a pronounced reduction in both rotation error and translation error, with both metrics normalized by the magnitude of the ground-truth inter-frame motion. This result demonstrates that our color-pair metric is effective at rejecting ambiguous temporal matches, thereby improving the geometric integrity of the correspondences used for pose estimation.

### C. Proof-of-concept Demonstration

Our TensorRT-optimized pipeline runs on a LoCoBot platform equipped with an Intel RealSense D435 camera. The pose estimation module processes images on an on-board NVIDIA Jetson AGX Xavier. Fig. 8 shows the experimental setup and a qualitative result, where the object's textured mesh is rendered at its estimated pose, demonstrating robustness to a slight texture mismatch between the physical object and its reference mesh file.

Fig. 8. Real-World Experiment. Left: experimental setup. Right (top to bottom): input scene, color-pair classification, and the final rendered pose estimate.

## V. Discussion

For pose estimation, our results are competitive but do not surpass the highest-performing methods. We attribute this gap partly to our method's reliance on depth data for both voting and ICP refinement. Real-world depth maps can suffer from noise and distortions, whereas 2D positional cues from RGB images are often more precise. Future work could therefore focus on integrating these 2D cues during refinement to improve accuracy. Regarding pose tracking, our method demonstrates strong performance on the Fast-YCB dataset, a result attributed to the synergy of our robust filtering and refinement strategy and the dataset's clean depth data. This combination allows the tracker to operate reliably without frequent re-correction. However, the noticeable drop in accuracy when tracking at a 5-frame interval (which occasionally requires re-initialization) highlights that abrupt pose changes remain a challenge. Therefore, future work could extend the system's reliance on a single previous frame to incorporate multiple keyframes, establishing a more stable temporal baseline and improving resilience to rapid motions or feature scarcity. Finally, our framework's robustness depends on the presence of clear surface textures. On sparsely textured and/or low-light surfaces with poorly discernible details, reduced feature matching consistency can impact both estimation and tracking precision.

TABLE I

Pose estimation evaluation on the YCB dataset (ADD-S / ADD).

| Object | FoundationPose [7] | | FreezeV2 [10] | | GenFlow [8] | | GigaPose [9] | | MegaPose [6] | | SAM6D [12] | | ZeroPose [11] | | Ours | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADD | ADD-S | ADD | ADD-S | ADD | ADD-S | ADD | ADD-S | ADD | ADD-S | ADD | ADD-S | ADD | ADD-S | ADD | ADD-S |
| 002_master_chef_can | 54.1 | 84.0 | 55.8 | 82.6 | 52.9 | 81.2 | 54.1 | 81.3 | 53.5 | 76.5 | 43.0 | 75.5 | 23.9 | 78.0 | 63.0 | 77.0 |
| 003_cracker_box | 87.8 | 91.3 | 88.1 | 91.7 | 77.0 | 82.3 | 86.8 | 90.5 | 70.4 | 74.2 | 75.9 | 79.4 | 46.5 | 75.7 | 73.3 | 87.1 |
| 004_sugar_box | 81.9 | 87.8 | 90.0 | 93.8 | 78.9 | 84.6 | 83.5 | 90.0 | 72.2 | 79.1 | 88.8 | 93.1 | 72.6 | 84.8 | 73.9 | 80.3 |
| 005_tomato_soup_can | 66.5 | 80.1 | 71.2 | 86.1 | 69.6 | 84.7 | 70.5 | 85.0 | 69.6 | 83.7 | 70.6 | 85.5 | 16.3 | 83.0 | 72.2 | 82.4 |
| 006_mustard_bottle | 87.3 | 91.6 | 91.3 | 94.0 | 77.5 | 90.3 | 79.3 | 91.2 | 76.9 | 91.2 | 90.8 | 94.1 | 62.1 | 91.5 | 77.3 | 85.3 |
| 007_tuna_fish_can | 64.7 | 83.4 | 61.6 | 81.6 | 66.8 | 83.2 | 67.4 | 83.2 | 67.1 | 83.6 | 55.7 | 75.6 | 6.3 | 75.7 | 62.3 | 77.7 |
| 008_pudding_box | 78.3 | 87.0 | 89.1 | 92.6 | 10.8 | 22.0 | 33.8 | 43.5 | 30.9 | 46.0 | 84.1 | 89.0 | 29.7 | 59.1 | 49.3 | 74.7 |
| 009_gelatin_box | 81.2 | 90.2 | 90.3 | 93.0 | 77.3 | 88.6 | 77.3 | 88.6 | 77.4 | 88.3 | 87.1 | 92.0 | 76.9 | 85.2 | 78.7 | 86.7 |
| 010_potted_meat_can | 59.7 | 79.8 | 60.9 | 80.1 | 51.8 | 70.4 | 61.5 | 74.9 | 57.4 | 69.4 | 46.3 | 72.2 | 28.4 | 71.8 | 60.4 | 73.8 |
| 021_bleach_cleanser | 73.5 | 80.8 | 86.7 | 91.3 | 71.6 | 87.5 | 72.8 | 88.0 | 76.4 | 88.6 | 87.8 | 92.9 | 56.0 | 83.3 | 79.0 | 84.3 |
| 035_power_drill | 85.7 | 91.3 | 88.8 | 92.6 | 82.3 | 88.1 | 84.6 | 90.5 | 73.8 | 80.6 | 85.3 | 88.4 | 81.4 | 88.7 | 71.3 | 82.3 |
| 037_scissors | 79.6 | 85.5 | 85.2 | 92.2 | 66.9 | 81.9 | 77.3 | 89.3 | 17.5 | 34.2 | 70.4 | 81.9 | 52.8 | 83.9 | 73.3 | 78.7 |
| 040_large_marker | 48.1 | 88.8 | 59.2 | 90.6 | 24.5 | 89.3 | 38.3 | 89.2 | 34.3 | 90.8 | 31.2 | 87.2 | 27.2 | 86.2 | 34.0 | 82.7 |
| 051_large_clamp | 13.7 | 78.0 | 3.6 | 77.1 | 24.5 | 63.5 | 20.7 | 64.8 | 8.7 | 58.2 | 19.2 | 62.2 | 25.8 | 57.0 | 24.7 | 69.3 |
| 052_extra_large_clamp | 19.2 | 67.8 | 24.9 | 55.9 | 21.4 | 36.4 | 25.4 | 50.5 | 8.7 | 21.6 | 9.4 | 39.9 | 13.4 | 42.7 | 20.6 | 56.7 |
| Average | 65.4 | 86.6 | 69.8 | 86.3 | 56.9 | 75.6 | 62.2 | 80.0 | 53.0 | 71.1 | 63.0 | 80.6 | 41.3 | 76.4 | 60.9 | 78.6 |

TABLE II

POSE TRACKING EVALUATION ON THE FAST YCB DATASET.

| method | DOPE [30] | ROFT [23] | PoseRBPF [22] | SE(3)-TrackNet [24] | Cross-Domain Fusion [31] | Ours | Ours (every 5 frame) |
|---|---|---|---|---|---|---|---|
| 003_cracker_box | 54.92 | 78.50 | 68.94 | 63.02 | 65.48 | 91.79 | 81.88 |
| 004_sugar_box | 60.01 | 81.15 | 82.78 | 73.70 | 70.88 | 96.12 | 85.76 |
| 005_tomato_soup_can | 64.14 | 79.00 | 75.93 | 80.82 | 77.77 | 90.50 | 75.73 |
| 006_mustard_bottle | 57.20 | 73.10 | 82.92 | 74.83 | 69.64 | 95.41 | 79.29 |
| 009_gelatin_box | 60.01 | 74.26 | 11.32 | 69.00 | 66.65 | 91.21 | 72.43 |
| 010_potted_meat_can | 57.03 | 73.87 | 87.29 | 71.30 | 75.06 | 93.73 | 85.43 |
| Average | 58.83 | 76.59 | 68.10 | 72.06 | 70.91 | 93.13 | 80.09 |

## VI. CONCLUSIONS

In this paper, we present a unified framework for zero-shot 6D object pose estimation and tracking, designed for robust performance on edge devices. Our method introduces a novel, lighting-invariant edge color-pair feature for accurate initial pose estimation, which is then seamlessly integrated with a high-speed, optical flow-based tracker. Experiments demonstrate that our approach achieves competitive accuracy on standard benchmarks while operating in real-time on resource-constrained hardware, highlighting its practical value for real-world robotic applications.


## ACKNOWLEDGMENT

We acknowledge the use of ChatGPT in text editing.


## REFERENCES


[1] T. Hodan *et al.*, "BOP challenge 2023 on detection segmentation and pose estimation of seen and unseen rigid objects," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 5610–5619.

[2] M. Oquab *et al.*, "DINOv2: Learning robust visual features without supervision," *arXiv:2304.07193*, 2023.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[4] B. Von Gimborn, P. Ausserlechner, M. Vincze, and S. Thalhammer, "Diffusion features for zero-shot 6DoF object pose estimation," *arXiv:2411.16668*, 2024.

[5] J. Corsetti, D. Boscaini, C. Oh, A. Cavallaro, and F. Poiesi, "Open-vocabulary object 6D pose estimation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 18 071–18 080.

[6] Y. Labbé *et al.*, "MegaPose: 6D pose estimation of novel objects via render & compare," *arXiv:2212.06870*, 2022.

[7] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "FoundationPose: Unified 6D pose estimation and tracking of novel objects," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 17 868–17 879.

[8] S. Moon, H. Son, D. Hur, and S. Kim, "GenFlow: Generalizable recurrent flow for 6D pose refinement of novel objects," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 10 039–10 049.

[9] V. N. Nguyen, T. Groueix, M. Salzmann, and V. Lepetit, "GigaPose: Fast and robust novel object pose estimation via one correspondence," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 9903–9913.

[10] A. Caraffa, D. Boscaini, and F. Poiesi, "Accurate and efficient zero-shot 6D pose estimation with frozen foundation models," *arXiv:2506.09784*, 2025.

[11] J. Chen, Z. Zhou, M. Sun, R. Zhao, L. Wu, T. Bao, and Z. He, "ZeroPose: CAD-prompted zero-shot object 6D pose estimation in cluttered scenes," *IEEE Trans. Circuits Syst. Video Technol.*, 2024.

[12] J. Lin, L. Liu, D. Lu, and K. Jia, "SAM-6D: Segment anything model meets zero-shot 6D object pose estimation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 27 906–27 916.

[13] E. P. Örnek, Y. Labbé, B. Tekin, L. Ma, C. Keskin, C. Forster, and T. Hodan, "FoundPose: Unseen object pose estimation with foundation features," in *Eur. Conf. Comput. Vis.*, 2024, pp. 163–182.

[14] P. Ausserlechner, D. Haberger, S. Thalhammer, J.-B. Weibel, and M. Vincze, "ZS6D: Zero-shot 6D object pose estimation using vision transformers," in *IEEE Int. Conf. Robot. Autom.*, 2024, pp. 463–469.

[15] V. N. Nguyen, T. Groueix, G. Ponimatkin, Y. Hu, R. Marlet, M. Salzmann, and V. Lepetit, "NOPE: Novel object pose estimation from a single image," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 17 923–17 932.

[16] F. Di Felice, A. Remus, S. Gasperini, B. Busam, L. Ott, F. Tombari, R. Siegwart, and C. A. Avizzano, "Zero123-6D: Zero-shot novel view synthesis for RGB category-level 6D pose estimation," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 14 204–14 211.

[17] T. Lee, B. Wen, M. Kang, G. Kang, I. S. Kweon, and K.-J. Yoon, "Any6D: Model-free 6D Pose Estimation of Novel Objects," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2025, pp. 11 633–11 643.

[18] M. Dani, K. Narain, and R. Hebbalaguppe, "3DPoseLite: a compact 3D pose estimation using node embeddings," in *IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1878–1887.

[19] J. Mei, J. Li, and C. Meng, "GS2Pose: Two-stage 6D object pose estimation guided by Gaussian splatting," *arXiv:2411.03807*, 2024.

[20] T. Pulli, S. Thalhammer, S. Schwaiger, and M. Vincze, "From Words to Poses: Enhancing Novel Object Pose Estimation with Vision Language Models," *arXiv:2409.05413*, 2024.

[21] B. Busam, H. J. Jung, and N. Navab, "I like to move it: 6D pose estimation as an action decision process," *arXiv:2009.12678*, 2020.

[22] 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," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1328–1342, 2021.

[23] N. A. Piga, Y. Onyshchuk, G. Pasquale, U. Pattacini, and L. Natale, "ROFT: Real-time optical flow-aided 6D object pose and velocity tracking," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 159–166, 2021.

[24] 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 *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 10 367–10 373.

[25] C. Wang *et al.*, "6-PACK: Category-level 6D pose tracker with anchor-based keypoints," in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 10 059–10 066.

[26] B. Wen and K. Bekris, "BundleTrack: 6D pose tracking for novel objects without instance or category-level 3D models," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2021, pp. 8067–8074.

[27] NVIDIA, "NanoSAM: A distilled Segment Anything (SAM) variant for real-time operation on NVIDIA Jetson Platforms," https://github.com/NVIDIA-AI-IOT/nanosam, 2024.

[28] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[29] Z. Zhang, A. Gupta, H. Jiang, and H. Singh, "NeuFlow v2: High-efficiency optical flow estimation on edge devices," *arXiv:2408.10161*, 2024.

[30] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv:1809.10790*, 2018.

[31] J. Wang *et al.*, "Cross-domain fusion and embedded refinement-based 6D object pose tracking on textureless objects," *J. Intell. Manuf.*, vol. 36, no. 3, pp. 1563–1577, 2025.

[32] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv:2306.12156*, 2023.

[33] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.