A Deep Dive into "Co-op" for Novel Object Pose Estimation

Team: Yavanni Ensley, Daniel Acosta

Date: October 15, 2025

1. Summary

The problem Co-op set out to achieve is 6DoF pose estimation of arbitrary objects from a single RGB query image and a 3D model of the target. Previous methods struggle with general efficiency, accuracy (particularly with texture-less and occluded objects), as well as generalization (existing solutions required extensive retraining when a new object was desired). The solution proposed by Co-op is to use a novel two-stage framework. The first estimates the coarse pose of the object by using semi-dense correspondences; the second uses probabilistic dense flows and a render-compare strategy to refine the pose. Using this hybrid approach, Co-op is able to significantly outperform former state-of-the-art methods on all seven core datasets from the BOP challenge. It was also able to set a new benchmark record for unseen object pose estimation.

2. Introduction & Background

The ability to extract 6DoF (3D position and rotation) from simple images is a cornerstone of the robotics and augmented reality (AR) fields. Robotics uses pose estimations for grasping and manipulating objects, and AR uses them to integrate virtual objects in real-world environments. Traditionally, these methods necessitated object-specific training data, which could take hours or days to train on, far too impractical for real-world use cases.

Current strategies for pose estimation can be divided into two categories: model-free and model-based. Model-free methods attempt to estimate the pose without using a 3D model for reference; because of this, they tend to struggle with textureless and occluded objects. Furthermore, they tend to only work with limited datasets. Since Co-op is model-based, we'll dive into how they generally work. These are also split into 2 categories:

1. **Template matching**
   - Involves rendering the 3D object from numerous viewpoints to generate a large set of 2D templates. These are used to compare against the query image in order to find the template with the highest similarity. The downside is that the computational costs with large numbers of templates make the algorithms extremely slow.

2. **Feature matching**
    - The alternative to template matching, algorithms such as FoundPose and GigaPose, will first detect where the object is within the query and use it to generate a mask. From there, they can detect features shared between the masked region and the CAD model. The downside here is the dependence on a good mask.

Co-op reframes the task of pose estimation as a correspondence-finding problem between the template image and the query image. By getting away from the segmentation masks, Co-op is able to handle noisy and obstructed objects. Furthermore, the model is able to learn low-level geometric and structural information at the patch & pixel level as opposed to at the full image level.

3. Detailed Method Description

3.1 Overall Architecture: A Two-Stage Pipeline

The Co-op framework follows a two-stage pipeline designed for 6DoF pose estimation of unseen objects from a single RGB image. The first stage, the Coarse Pose Estimation, generates initial 6DoF pose estimates based on a limited number of pre-rendered templates of the desired object. The second stage, the Pose Refinement, improves upon this initial estimate via a render-and-compare process that iteratively aligns the rendered model with the provided input image.

Both stages use a unified Vision Transformer (ViT) backbone that is initialized with CroCo pre-trained weights. This allows the program to generalize object features, so that there is no need for additional object-specific training in the future.

3.2 Stage 1: Coarse Pose Estimation

The Coarse Pose Estimation stage aims to obtain a quick, accurate initial pose by estimating semi-dense correspondences between the provided image and a small set of templates. These templates are generated from 42 uniformly distributed viewpoints that cover only out-of-plane rotations. Each generated template provides RGB and depth information so that the program can use correspondence-based mapping between the 2D image and the 3D templates.

The main innovation of this stage is the hybrid representation for correspondence estimation. Instead of traditional methods, such as direct regression of coordinates, Co-op combines patch-level classification and offset regression to evaluate correspondence between the provided image and the generated templates. The input image and template are processed and split into two outputs: a tensor that predicts which template patch corresponds to each query patch, and an offset map that adjusts the exact correspondence location within that patch.

After finding these correspondences, Co-op applies a Perspective-n-Point (PnP) solver to recover the initial 6DoF pose. Co-op specifically uses EPnP with RANSAC to ensure resistance to outliers and occlusions. Each template is evaluated based on the classification confidence produced earlier, and the template with the highest value is selected for pose estimation. This allows Co-op to get coarse pose estimates quickly and accurately, all while using much fewer templates than traditional approaches.

3.3 Stage 2: Pose Refinement

The Pose Refinement stage works at a pixel level to correct any misalignments from the coarse stage. This stage uses a render-and-compare strategy: given the original coarse pose, render the object using its CAD model, and then predict the dense flow field that aligns the rendered image with the query image.

A core innovation of this stage is the introduction of Probabilistic Flow Regression, which models the correspondence field as a Laplace-distributed conditional probability rather than a single, deterministic flow. The program predicts the mean flow vector (μ) and an uncertainty scale (b), which represents the flow as a probabilistic distribution. These values allow the program to learn where the flow should move pixels, as well as how confident it is for each correspondence it generates. Co-op computes a flow confidence map (W) by combining three learned factors from this process:

1. Certainty: estimating whether a pixel in the rendered image is visible/occluded;
2. Sensitivity: identifying regions with discriminative visual texture; and
3. Flow Probability (PR): likelihood that the predicted flow lies within a small radius of the true flow.

Using this confidence-weighted representation, Co-op is able to down-weight (ignore) unreliable correspondences, leading to more accurate updates to the pose. The final pose update is done with a differentiable PnP solver (based on the Levenberg-Marquardt algorithm), which uses the confidence map as pixel-wise weights. This process is iterated several times, then ultimately refined using a Gauss-Newton step for convergence of the correct pose.

4.0 Experiments and Results

4.1 Experimental Setup

To evaluate Co-op's performance and generalization capabilities, experiments were performed on the seven core datasets of the BOP challenge, which include: LineMod Occlusion (LM-O), T-LESS, TUD-L, IC-BIN, ITODD, HomebrewedDB (HB), and YCB-Video (YCB-V). Compared to other models, Co-op was tested without any retraining, using only the models and test images provided in the challenge.

Performance was measured using Average Recall (AR), which is the average of three metrics defined in the BOP challenge: Visible Surface Discrepancy (VSD), Maximum Symmetry-Aware

Surface Distance (MSSD), and Maximum Symmetry-Aware Projection Distance (MSPD). Combined, these metrics assess the accuracy of the pose alignment produced by Co-op.

All experiments were implemented using the CroCo vision foundation model as an encoder-decoder backbone. The evaluations were performed on a single NVIDIA RTX 3090 Ti GPU, showing that Co-op can achieve high performance with practical computational efficiency.

4.2 Quantitative Results

When evaluating the model using only the coarse estimation stage, Co-op achieved a mean AR of 58.4, which outperformed the next-best method (FoundPose, with a 37.3 AR) by approximately 56.6%. Even without refinement, Co-op surpassed the full pipelines of earlier methods such as GigaPose + GenFlow, showing that the hybrid representation for semi-dense correspondence estimation was superior.

When combining the coarse and refinement stages under a single-hypothesis setting, Co-op achieved a mean AR of 64.0, outperforming all previous single-hypothesis pipelines, including GenFlow and MegaPose. This is due to the probabilistic flow refinement strategy, relying on uncertainty modeling to produce better pose corrections.

Using a five-hypothesis strategy, where multiple coarse poses are refined, and the best result is selected, Co-op further improved to a mean AR of 65.7. This is an improvement of more than 6 points over the previous best method (FoundPose, 59.6 AR), which established Co-op as the new state-of-the-art method across all BOP datasets. Despite the increase in accuracy, Co-op still only required around 4.2 seconds per image on average.

4.3 Ablation Studies

When the patch-level classification module was removed and only direct offset regression was used, the model scored much lower, from 58.4 AR to 50.2 AR.

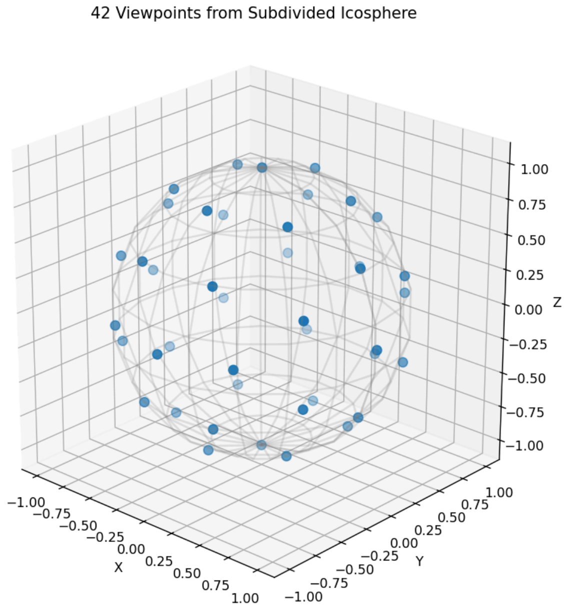
Replacing Co-op’s probabilistic flow regression with a standard deterministic flow head (like that used in GenFlow) also reduced scores, from 64.0 AR to 63.3 AR.

Initializing the model from randomized weights rather than the CroCo pre-trained parameters caused the model to score significantly lower, from 58.4 AR to 46.4 AR for coarse estimation and from 64.0 AR to 61.2 AR for refinement.

5. Hands-On Findings & Replication

Unfortunately, the team behind Co-op has not published its source code or the models used, so replicating their findings is not possible. However, we were able to implement the initial template generation as well as a non-functional skeleton implementation of the pose-refinement.

Regarding dataset preprocessing, we generated the aforementioned 42 viewpoints along the isosphere and plotted them:



While at first glance, 42 points may seem like an inadequate number, however, in our testing, most random viewpoints we chose were quite similar to at least one template. This leads us to believe that the number of iterations required by the pose refinement algorithm would generally be small. From these points, we used the **Pytorch3D** library to generate an RGB and a depth image:

At this point, the initial setup was completed, and without access to the models, resources (their models took several days across multiple A100 GPUs to train), or classical knowledge to reimplement the remainder of the pipeline, we created a skeleton implementation of the pose refinement.

```
# Generate a random viewpoint along the sphere. In reality,
# This would be an unknown.
random_viewpoint = torch.randn(1, 3)[0]
random_viewpoint = random_viewpoint / random_viewpoint.norm()


# Generate the 42 standard viewpoints on an icosphere
viewpoints = generate_icosphere_viewpoints()
viewpoints = torch.from_numpy(viewpoints).to(device=device).float()
    * camera_dist


# Find the viewpoint with the minimum Euclidean
# distance to the ground truth
distances = torch.linalg.norm(viewpoints - random_viewpoint, dim=1)
nearest_idx = torch.argmin(distances)
coarse_camera_position = viewpoints[nearest_idx].unsqueeze(0)
```

This yields us the coarse pose that was theoretically found in the first stage. We can now enter the refinement loop:

```
for i in range(num_iterations):
    print(f"--- Iteration {i+1}/{num_iterations} ---")
    cameras = FoVPerspectiveCameras(device=device, R=current_R, T=current_T, fov=60)

    # Render our current pose for render and compare
    rendered_images = renderer(mesh, cameras=cameras)
    z_buffer = renderer.rasterizer(mesh, cameras=cameras).zbuf

    # This serves as a placeholder; as such, the algorithm cannot work correctly
    # The team behind Co-op used proper machine learning/deep learning to calculate
these values
    predicted_flow, confidence_weights =
mock_flow_and_confidence_estimation(rendered_rgb, query_image)

    points_3d = unproject_depth_to_3d(rendered_depth, cameras)
    points_2d = get_2d_points_from_flow(predicted_flow)

    proj_transform = cameras.get_projection_transform().get_matrix()
    K = proj_transform[0, :3, :3]

    R_update, T_update = weighted_pnp_solver(
        points_3d,
        points_2d,
        camera_intrinsics_matrix=K,
        weights=confidence_weights
    )

    new_R = torch.bmm(R_update, current_R)
    new_T = torch.bmm(T_update.unsqueeze(1), current_R).squeeze(1) + current_T

    # Here you would update the pose;
    # the values here will be wrong due to our mocking
    # current_R, current_T = new_R, new_T
```

6. Conclusion

Our assigned paper presented Co-op, an extremely effective and robust method for 6DoF pose estimation of untrained/unseen objects. It uses a hybrid coarse estimation stage with a probabilistic refinement stage in order to achieve state-of-the-art results on the seven core BOP challenge datasets. Its novel contributions are as follows:

1. **Detector-free, hybrid-representation**
    - It combines patch-level classification and offset regression in order to estimate a coarse pose for any general object.
2. **Probabilistic Flow for Refinement**
    - Co-op models flow estimation with probabilities lets the model handle occlusions and textureless surfaces robustly.