Using Diffusion Features for Template Based object Pose Estimation

Aman Chulawala, Beverley-Claire Okogwu
Department of Mechanical Engineering,
Robotics Institute,
Carnegie Mellon University
Pittsburgh, PA 15213
{achulawa, bokogwu}@andrew.cmu.edu

1 Introduction

In Computer Vision, pose estimation has become a viable and important task. With pose estimation, the *position* and *orientation* of three-dimensional objects are obtained from one or multiple images of the object[13]. The complexity here lies in the 3D object to 2D image transformation. With numerous applications, including photography, navigation, and virtual reality, this innovative process has brought about advancements utilizing Deep Neural Networks (DNNs) in breaking down the system's complexities [12].

In addition, certain adaptations have been made further to utilize the pose estimation techniques in cluttered spaces. This is an important challenge in pose estimation techniques due to the potential loss of depth information in the 3D object to 2D image transformation. The primary cause of this loss of information is the presence of obstacles and occlusions. The problem is further compounded by the variability in object appearances due to changes in viewpoint, lighting, and background.

Traditional methods to solve this problem relied heavily on feature extraction techniques, which manually identified specific points or markers on objects to estimate their pose. However, these methods often struggled in complex scenes and were not robust against these obstacles.

In our work, we utilize the LINEMOD dataset[2], Stable Diffusion [7], and the InfoNCE loss [6] during training to get a better, more accurate object pose estimation from the object templates. Our results show our approach can estimate the pose of some objects. Although a few failure attempts, we also explain the reasoning behind these failures as well.

2 Literature Review

2.1 Pose Estimation

Pose estimation is a crucial area in computer vision dedicated to identifying objects' spatial orientation and position within images. Early methodologies heavily relied on geometric approaches and manual feature identification, which were not scalable for complex scenarios [13]. With advancements in artificial intelligence, Deep Learning techniques have revolutionized this field. These models have improved the efficiency and accuracy of pose estimation tasks by learning from large datasets and identifying nuanced patterns in spatial data [4]. However, traditional methods do not do well with the occlusions. In our work, we aim to address the problem and obtain better pose estimations with and without the presence of obstructions.

2.2 Pose Estimation with Occlusion

Occlusions pose significant challenges in accurately estimating poses, as they obscure critical features of target objects. Recent research has focused on developing more resilient models that can infer complete object shapes and orientations despite partial visibility. Approaches such as context-aware neural networks and occlusion-aware pose estimation

algorithms have been instrumental in advancing this area. These techniques utilize the visible segments of objects and contextual clues from the surroundings to hypothesize the obscured parts [1, 9]. Innovations in this domain also include the integration of depth sensing and segmentation strategies to isolate and focus on the object of interest more effectively. Although these methods do fairly well in posing estimation with obstacles, they do not pay attention to the little differences/ spaces that could be missed in the estimation, leading to partial data loss. Our approach uses the diffusion-based method to recover some of this data loss and lead to a better estimation.

2.3 Diffusion-based Pose Estimation

Diffusion-based methods, which are relatively new in the pose estimation landscape, leverage the concept of progressive noise addition and reduction to synthesize diverse training examples. This process facilitates the generation of high-fidelity images under various simulated conditions, which are invaluable for training robust models. These generative models are particularly useful in creating synthetic data where real data is scarce or expensive to acquire [3]. The potential of diffusion-based models in pose estimation is promising, offering opportunities to train systems with enhanced adaptability to real-world variability in object poses and environmental conditions. However, this work's limitation lies in the dataset used and the inability to handle the occlusions accordingly. To have a more robust system, our method utilizes all three (dataset, diffusion features, and occlusion avoidance).

3 Project Scope

This project aims to improve the robustness of 3D pose estimation and better estimate objects in obstacles and occlusions in the scenes using advanced techniques and datasets.

To achieve our objective, our project does the following, as will be explained in the next sections:

1. **Data Utilization**: In this work, we will utilize the LINEMOD dataset[2], best known for its difficult scenes of occluded objects and objects lacking texture information. This dataset will supply the ground truth for training and validating our pose estimation model.
2. **Model Development**: In our approach, we adapt Stable Diffusion as this method is best known for generating things out of nothing. Thus, adapting the Stable Diffusion model will generate training data from the diffusion features to help simulate scenes often misrepresented in current datasets. By doing this augmentation, we can generalize the model better.
3. **Optimizing the Loss Function**: Because we need to differentiate between near-situated features in pose estimation, we will employ the InfoNCE loss function, which will concentrate on the model's feature discrimination capabilities.
4. **Experimentation and Evaluation**: In this work, we will perform comprehensive testing to evaluate our model performance. We will do this across diverse metrics, including accuracy and estimates, with the test split of seen and unseen data. This also includes comparisons with common feature extraction methods to highlight the improvements made by our diffusion-based approach. We will also show various qualitative results, visually showing the pose estimates using the "Cat" view for various queries for success and fail cases.
5. **Results and Analysis**: As aforementioned, we will provide a detailed analysis to understand the circumstances under which the model performs well and where it fails.
6. **Conclusion**: Following our interpretation of the results, we will reiterate the problem, our method used to address the problem, and conclude how the results support our objective. We will also address the limitations from the results' analyses and also outline future work to optimize our approach further.

4 Methodology

4.1 Dataset

The LINEMOD dataset is a significant resource aimed at the development and evaluation of methods for detecting and estimating the 6 degrees-of-freedom pose of texture-less 3D objects in heavily cluttered scenes. It is particu-

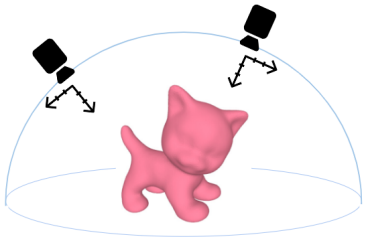


Figure 1: The templates are created by rendering the base cad model viewed from different poses. Blender API is used to create the templates for all the CAD models. This was by far the most time-consuming step, taking 8 hours to generate all the templates on an AWS instance running g5.2xlarge configuration.

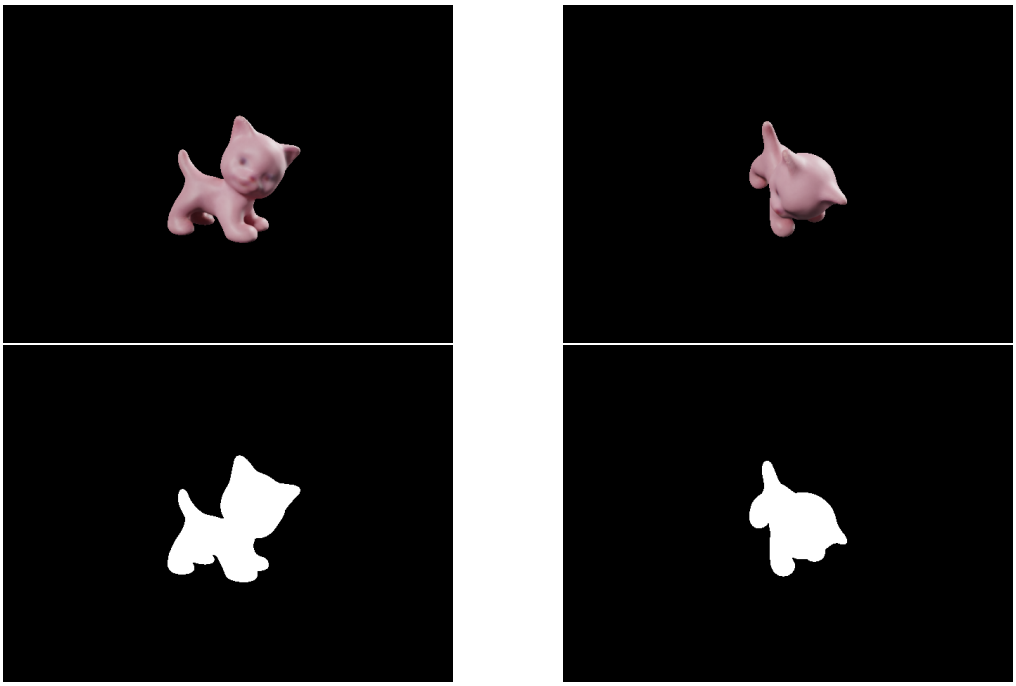


Figure 2: Examples of two templates created using the Blender sub-process thread for the Cat model of LINEMOD as viewed from two difference position on the hemispherical shell. Each template is created with a corresponding mask.

larly designed for scenarios where objects lack distinctive textures and are surrounded by complex backgrounds with occlusions.

The dataset utilizes the LINEMOD approach, which integrates depth and color information to create templates representing different views of an object. These templates are learned from 3D models and are essential for object detection. The LINEMOD dataset consists of 13 registered video sequences, each containing over 1100 frames, featuring 13 different texture-less household objects with distinctive color, shape, and size characteristics. CAD models for all objects are provided.

This dataset provides a valuable benchmark for advancing the field of 6D object pose estimation, especially in scenarios with limited texture information. These characteristics satisfied our requirements and we used the LINEMOD (LM) dataset for our model.

We follow the approach outlined in template-pose [5] which is used to crop and split the data into three non-overlapping sections, and reserve 10% of the dataset for testing purposes. To create the templates for the objects, we follow the algorithm outlined by Wohlhart et al [11]. This results in 301 templates generated using Blender per object when viewed from a hemispherical shell created around the target object. The templates and input images to the model

are cropped at the center of the objects with the ground truth pose. This pre-prossessing pipeline is adapted from the Diffusion Features pipeline created by Wang et al [10].

4.2 Implementation

4.2.1 Feature Extraction

Image features are critical when it comes to pose estimation using template matching. Diffusion features are discriminative in nature and easy to work with. We made use of the Stable Diffusion (SD) model [8] for our purposes. When an image is provided as an input to the SD model, it generates a set of features from a UNet.

Given a set of these raw diffusion features from the n layers in the SD model, we need an architecture to fuse them into a single feature F. Proposed aggregation architectures in the literature make use of an extractor ϕ_{ext} and an aggregator ϕ_{agg}. The extractor is employed to align the raw diffusion features to a single dimension while the aggregator simply performs an aggregation operation like Element-wise Addition. The overall process can be represented with the following equation.

$$F = \phi_{\text{aggr}}\left(\phi^1_{\text{ext}}(f_1), \phi^2_{\text{ext}}(f_2), \ldots, \phi^n_{\text{ext}}(f_n)\right)$$

Development of an aggregation architecture was not within the scope of our work and we used a simple off-shelf aggregation architecture proposed in [10] (Architecture (a) - Vanilla aggregation network). The process involves feeding images (no noise) of size (512 x 512) into the SD model and aggregate features from its UNet into output features with a dimension of (32 x 32).

4.2.2 Training Pipeline

To tailor our method for object pose estimation, we begin by freezing the weights of the SD model, ensuring that the learned features remain unchanged during the training of our aggregation networks. This step is crucial as it allows us to focus solely on the training of the aggregation networks under the supervision of pose estimation tasks. We adopt a strategy similar to template-pose, where for each real image in the dataset, we generate one positive pair template and (M - 1) negative pair templates. The positive pair consists of the real image and a template that correctly represents its pose, while the negative pairs are composed of the real image and templates with incorrect poses. The training objective is to fine-tune our model to enhance the similarity of representations in positive pairs while concurrently diminishing the similarity in negative pairs. This is achieved through the implementation of the InfoNCE loss function, which effectively discriminates between positive and negative samples by maximizing and minimizing their agreement, respectively.

The **InfoNCE** loss is a contrasting loss function used primarily in self-supervised learning. It stands for Noise-Contrastive Estimation and is designed to maximize the mutual information between a pair of variables. The InfoNCE loss function discriminates between each positive pair and its associated negative pairs by optimizing the negative log probability of correctly classifying the positive sample. By optimizing this loss, the model learns to bring the

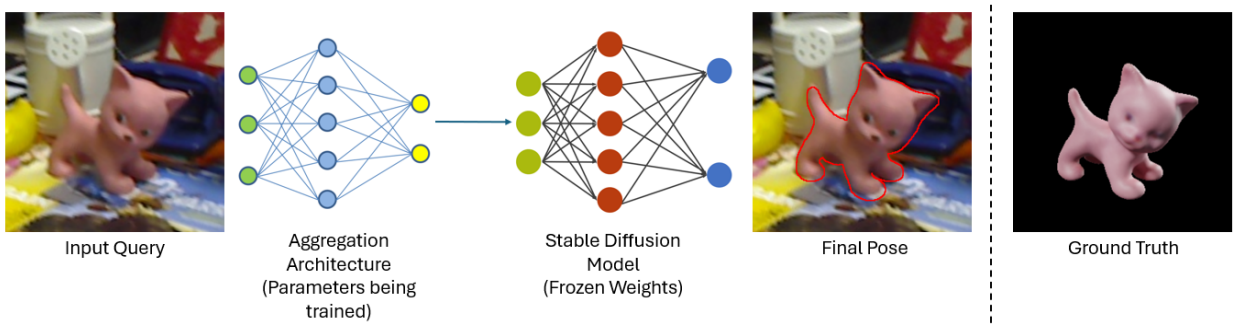


Figure 3: Overall Training Pipeline

representations of positive pairs closer together while pushing away the representations of negative pairs, effectively learning a representation space where similar samples are near each other and dissimilar ones are far apart.

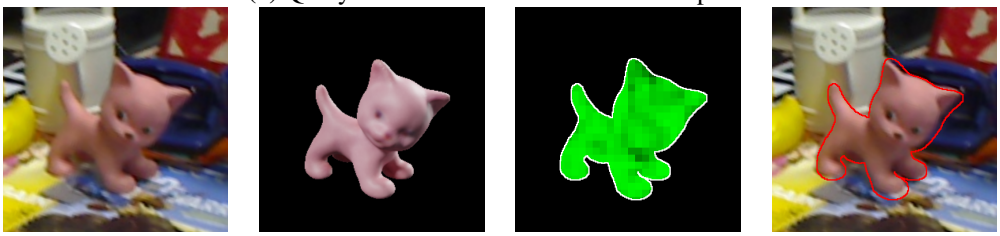
4.2.3 Testing Pipeline

During the testing phase, our model's performance is evaluated based on its ability to accurately estimate the pose of objects in new images. We commence this process by identifying the template that exhibits the highest similarity to the input image. This template, which has been previously annotated with the correct identity and pose information, is then used as a reference to infer the pose of the object in the input image. To determine the degree of similarity, we employ the same measurement technique as template-pose. Initially, we compute the cosine similarity between the feature representations of the input image and the templates. Subsequently, we apply a predefined threshold and a template mask to filter out irrelevant features. The final step involves averaging the values of the remaining features post-filtering to obtain a singular similarity score, which serves as the basis for pose estimation.

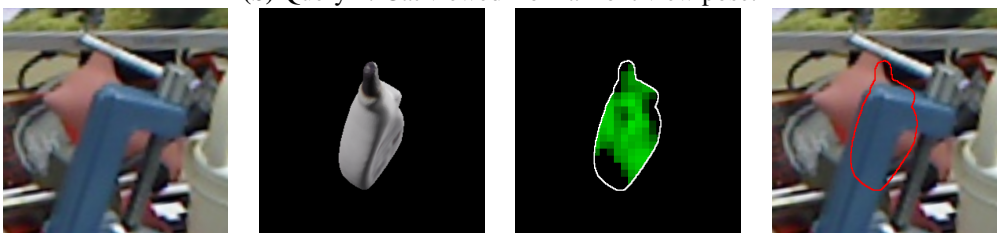
4.3 Evaluation Metric

In the evaluation of pose estimation for the LM dataset, two key metrics are employed: pose error and accuracy. The pose error is quantified by the geodesic distance between the predicted rotation ($\hat{R}$) and the ground truth rotation (R). This distance is calculated using the formula:

(a) Query 1: Cat viewed from an elevated pose.



(b) Query 2: Cat viewed from a front view pose.



(c) Query 3: Occluded cat view seen from the back.

Figure 4: Results from the testing pipeline. Poses were classified for query images from all 13 models present in the LINEMOD dataset. In each query row, the first image is the cropped input image from the dataset. The second image is the closest template from the cat templates. The third image is the alignment score between the template and the query image. The last image is the pose estimate result presented as an outline around the area of interest. Query (a) and Query (b) are positive results from the model. Query (c) is an example of where the model fails. The occlusions result in an incorrect estimate that the target is a glue bottle and this results in an incorrect pose estimate.

$$d(R^T, \hat{R}) = \frac{\arccos\left(\frac{\mathrm{tr}(R^T\hat{R})-1}{2}\right)}{\pi}$$

Here, tr represents the trace operation on a matrix. This metric effectively captures the angular discrepancy between the predicted and actual orientations, providing a robust measure of the model's performance in estimating 3D poses.

Accuracy, particularly in the context of unseen objects, is defined as the percentage of test images for which the pose error is below a specified threshold (λ), and the predicted object class ($\hat{c}$) is correct. The accuracy metric, denoted as (Acc.), is expressed as:

$$\text{Acc.} = \begin{cases} 1 & \text{if } d(R^T, \hat{R}) < \lambda \text{ and } \hat{c} = c \\ 0 & \text{otherwise} \end{cases}$$

The threshold (λ) is typically set to 15, following the template-pose methodology, which is referred to as the Acc15 metric. This stringent criterion ensures that the model is evaluated based on its ability to accurately predict both the pose and the class of objects, even when the object class is not known during testing.

5 Results

We trained our model at a learning rate of 1e-3 for 20 epochs. We are mainly training the aggregation architectures and not the weights of the Stable Diffusion model. Testing happens on the 10% of the LINEMOD dataset which is never seen during training. We see results from all the 13 CAD objects. In Figure 4 we see three different samples of the Cat model. Query (a) and (b) are of images where we have a clear view of the cat model. The trained model performs well with such images. However the model fails on high occlusion images like Query (c). The accuracy of such high occlusion images is low, resulting in an incorrect class assignment (glue bottle in this case). The result pose estimate is garbage due to the depende of the pose estimate on the correct classification of the object as we discussed in Evaluation Metric section.

Metric	Value
seen_err	3.06641
seen_acc	0.99809
unseen_err	5.10877
unseen_acc	0.98432
unseen_occ_err	9.05187
acc	0.80289
unseen_occ_acc	0.81657

Table 1: Testing results from the images in the test split of the dataset.

6 Conclusions

We have presented our method, which uses diffusion features for template-based object pose estimation. We have also performed our experiments and shown both qualitative and quantitative results that support how this diffusion-template-based method achieves better pose estimation. In addition, we have also outlined the success and failure cases using the current method.

6.1 Limitations and Future Work

Our results outlined some failure cases, particularly when the occlusion is very large, covering most of the object view. In this case, we have also shown that the template is mismatched to be aligned more with the occlusion than with the object. To address the failure, we hope to find better ways to directly pay attention to the object irrespective of how much the objects are visible in the scene.

References

[1] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 623–630. IEEE, 2010.

[2] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[5] Van Nguyen Nguyen, Yinlin Hu, Yang Xiao, Mathieu Salzmann, and Vincent Lepetit. Templates for 3d object pose estimation revisited: Generalization to new objects and robustness to occlusions. 2022.

[6] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *ArXiv*, abs/2112.10752, 2022.

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. 2021.

[9] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019.

[10] Tianfu Wang, Guosheng Hu, and Hongguang Wang. Object pose estimation via the aggregation of diffusion features. 2024.

[11] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. *arXiv*, 2015.

[12] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[13] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.