Investigating Use of Keypoints for Object Pose Recognition

E. Zhixuan Zeng University of Waterloo
Yuhao Chen University of Waterloo
Alexander Wong University of Waterloo
Email: {ezzeng, yuhao.chen1, alexander.wong}@uwaterloo.ca

Abstract

Object pose detection is a task that is highly useful for a variety of object manipulation tasks such as robotic grasping and tool handling. Perspective-n-Point matching between keypoints on the objects offers a way to perform pose estimation where the keypoints also provide inherent object information, such as corner locations and object part sections, without the need to reference a separate 3D model. Existing works focus on scenes with little occlusion and limited object categories. In this study, we demonstrate the feasibility of a pose estimation network based on detecting semantically important keypoints on the MetagraspNet dataset which contains heavy occlusion and greater scene complexity. We further discuss various challenges in using semantically important keypoints as a way to perform object pose estimation. These challenges include maintaining consistent keypoint definition, as well as dealing with heavy occlusion and similar visual features.

1 Introduction

In recent years, there has been a rising interest in the use of robotic systems to handle object manipulation tasks in scenarios ranging from manufacturing to domestic settings. One important hurdle to performing automated object manipulation is estimating accurate object pose. Object pose provides important knowledge both before, during, and after a grasp action. Before a grasp, pose information can allow the robot to target different parts of the object depending on its task. During a grasp, pose information is vital for moving objects through space without collision, as well as to operate tools that are being grasped. Finally, it can be important to place an object in the correct orientation when releasing a grasp.

Object pose estimation methods can be categorised as direct pose estimation methods or Perspective-n-Point(PnP) methods. Direct pose estimation predict the 3D rotation and translation matrix of an object relative to a reference pose, such as an exact 3D model of the object in question. Examples include [1–4]. Perspective-n-Point(PnP)/RANSAC methods use an intermediate representation that is used to match up with the 3D model pose. Those representations can be categorized as either dense or sparse representations. Dense PnP methods [5–9] predicts a correspondence to a reference model for each input point. In contrast, sparse methods only predicts a limited number of correspondences or keypoints, typically in the range of 5-20. These keypoints may be defined as the corners of the 3D object bounding box (as seen in [10, 11]) or a set of points defined relative to the object surface [12–18]. These surface keypoints are most similar to pose estimation methods used in human pose estimation

Direct pose estimation and dense PnP pose estimation methods both have a heavy reliance on exact 3D reference models. Such 3D models are costly to collect and makes it difficult to expand the dataset to new objects. Sparse keypoints are more flexible, and do not require exact matches with a reference mode. The Pascal3D+ dataset [19] for example uses semantic keypoints based on a limited number of 3D models to represent the pose of a much wider variety of real world examples that do not match exactly.

Furthermore, because direct and dense PnP methods are informative only relative to a reference model, the resultant prediction contains little intrinsic information. In contrast, surface keypoints may directly inform us of part locations or surface properties. For instance, a particular keypoint can be defined as the "tip" of a object, while another may be the "end" of the handle. This direct description without the need to refer back to a reference model can decrease computation requirement during inference, removing the need to load a 3D CAD model into the system for each predicted object.

Previous work using semantically important sparse keypoints for robotic grasping are very limited in the complexity of its environment and the variety of available object classes. There is often very little occlusion, and only two or three object categories are considered.

Fig. 1: An example of keypoints detection results for drills.

Fig. 2: Example keypoint labels for various classes

In this work, we investigate the feasibility of using semantically important keypoints in object pose estimation in complex, clustered environments. We train a heatmap-based keypoint detection model on the MetagraspNet dataset [20]. Our model is evaluated based on both pose estimation performance as well as 2D keypoint similarity scores. Through experiments, we conclude that the model performance is heavily impacted by occlusion and similar nearby points.

2 Implementation Details

The keypoint detection network is implemented using mmpose [21] with a ResNet [22] backbone and a heatmap-based keypoint predictor head based off of Simple Baseline 2D [23]. A separate head is used for each prediction class. An example of keypoint detection results for the drill class can be seen in Figure 1.

The model is trained on the synthetic MetagraspNet Dataset [20]. Each object category is labelled with keypoints containing unique ids at semantically important locations on the object surface. Boxes, for example, are labeled with 8 keypoints on each corner. Some example classes are displayed in Figure 2

The 2D keypoints detected on an image can then be converted into an estimated pose through solving for the rotation and translation matrices to minimize the reprojection error from 3D-2D point correspondences using the RANSAC algorithm. Example pose results can be seen in Figure 3

2.1 Keypoints Definition

The most difficult component of using keypoints for object pose recognition is defining keypoint placement. A popular option is to use furthest point sampling, such as in [13], where a fixed number of keypoints are sampled evenly around a 3D shape. However, such a sampling method offers no semantic information.

Instead, a method similar to [16] is preferable, where keypoints are defined in areas significant to the object, such as the top, bot-

tom, and handle. Such keypoints offer important semantic information, but can be more difficult to implement for both model prediction and in defining where keypoints are placed.

During model prediction, semantically important keypoints can be less robust to occlusion due to being fewer in number. It can also be difficult to ensure that they are located in areas that are visually distinct.

During keypoint definition, it is difficult to define a generalized pattern for placing keypoints, especially for more complex objects like scissors or drills and can thus result in rather arbitrary keypoint placement.

Another challenge is symmetry. Symmetrical objects with theoretically different poses may visually be exactly identical. Human pose estimation only needs to handle bi-radial symmetry, but objects with more than a single plane of symmetry are very common. Pose estimation specific metrics, such as those in [24] often take into account symmetry in the final loss calculations, where visually identical poses are not punished, while [16] defines keypoints on the line of symmetry itself to reduce ambiguity. However, important object properties such as edges or corners often lies outside that line or plane of symmetry, and makes it difficult to limit keypoints to only along such axis. A box, for example, is intuitively defined with keypoints at the corners. However, this definition causes ambiguity for flipped or rotated poses where the particular keypoints may not necessarily match.

Our implementation defines keypoints along axis of symmetry, such as one on center-top, and another on center-bottom, but also on important features such as corners of boxes or edges of cups. Examples of keypoints on various object classes can be seen in Figure 2.

3 Experimental Results

3.1 Metrics

We evaluate the pose prediction on the average distance (ADD) [25] metric :

$$e_{\text{ADD}} = \frac{1}{m} \sum_{x \in \mathcal{M}} ||(Rx+T) - (\tilde{R}x+\tilde{T})|| \tag{1}$$

Where $\mathcal{M}$ is the set of 3D model points, m is the number of points, R and T are the ground truth rotation and translation matrices, and $\tilde{R}$ and $\tilde{T}$ are the estimated rotation and translation matrices. This can be computed in both world coordinates using only the estimated pose matrices, as well as after projecting both poses into image coordinates based on the camera's intrinsic matrix. This metric can be converted into an accuracy score by the following equation:

$$\text{Accuracy}_{\text{ADD}} = \frac{1}{n} \sum_{e_{\text{ADD}} \in \mathcal{N}} \begin{cases} 1 & \text{if } e_{\text{ADD}} < t \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Where $\mathcal{N}$ is the set of predictions, n is the number of predictions, and t is the distance threshold.

We also look at the object keypoint similarity (OKS) score for the 2D keypoints based on the coco evaluation metrics:

$$\text{OKS} = \sum_i \left[exp\left(\frac{-d_i^2}{2s^2k_i^2} \right) \right] \tag{3}$$

where d_i is the euclidean distance between detected and ground truth keypoint, s_i is object scale, and k_i is a per-keypoint constant to control falloff. Precision and recall metrics are computed with OKS at 0.5 threshold. Results can be seen in Table 1

3.2 Pose

After solving for rotation and translation matrices using RANSAC, pose estimation example results can be seen in Figure 3

Our model is trained on cereal box and drill classes. The accuracy curve for different ADD score threshold can be seen in Figure 4

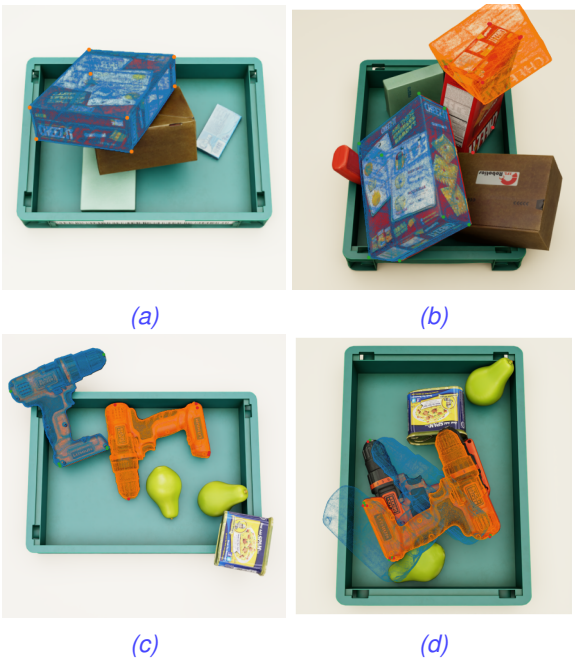


Fig. 3: Visualization of pose results.

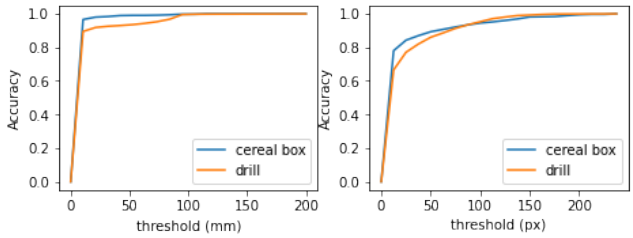


Fig. 4: ADD accuracy scores given different thresholds for world coordinates (left) and projected image coordinates (right)

4 Discussion

4.1 Occlusion

The average precision and recall scores are heavily impacted by occlusion, as shown in Table 1, .

Table 1: Average precision and recall performance for occluded and visible keypoints in drills

	AP	AR
all	83.1	87.2
occluded points	69.4	78.8
visible points	86.7	89.4

When a keypoint is occluded and not visible from the camera,, there are no longer local features for the model to identify, making it significantly more difficult to predict that keypoint's location. This issue has also been pointed out by [18]. [12] works to improve this weakness through heavy augmentation, the lack of robustness to occlusion is a key reason for the popularity of dense PnP and direct pose estimation methods compared to sparse PnP methods.

Occlusion is especially problematic with semantically important keypoints. Often, the number of defined points are fewer than those defined in automatically sampled methods, and thus the loss of one point more heavily impacts the pose prediction. An example of this can be seen in Figure 3b and 3d.

4.2 Similar keypoints

The model can be easily confused by keypoints where the local features are very similar to each other. This can happen in two different scenarios: between keypoints on the same object as well as if a similar object is within the bounding box.

An example of confusion between similar keypoints shown in Figure 5, where keypoints 0, 1, 2, and 3 are misclassified with each

(a) keypoint 0 *(b)* keypoint 1 *(c)* keypoint 2

(d) keypoint 3 *(e)* keypoint 4 *(f)* keypoint 5

Fig. 5: Visualization of heatmap for predicting a keypoint for example in Figure 1, overlayed on the original image.

other.

We demonstrate this phenomenon through calculating the percentage of keypoint observations which contain more than one peak above the detection threshold of 0.3. As seen in Table 2, 15 to 24 percent of predictions for keypoints 0-3 are confused between different possible locations. This compares to the around 5% probability for keypoints 4 and 5, which are more visually unique.

Table 2: probability that each keypoint would contain multiple detected peaks above the 0.3 confidence threshold for drills

keypoint id	probability of multiple peaks (%)
0	22.1
1	20.3
2	15.6
3	23.7
4	5.8
5	4.2

In comparison, cracker boxes have more visual graphics that aid in distinguishing between the different keypoints. An example of a cracker box can be seen in figure 6

Table 3: probability that each keypoint would contain multiple detected peaks above the 0.3 confidence threshold for cracker boxes

keypoint id	probability of multiple peaks (%)
0	7.9
1	9.6
2	6.8
3	7.2
4	11.1
5	13.2
6	16.2
7	15.8

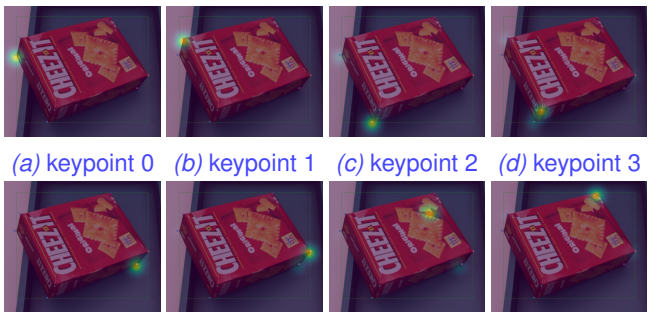
Those visual graphics result decrease the likelihood of there being multiple peaks in the heatmap prediction in comparison to keypoints 0-3 on drills, but are still less unique than keypoints 4 and 5.

Using this multiple-peaks probability, we can iteratively improve the location of various keypoints to more visually distinct locations through identifying keypoint locations to minimize this value. However, it can be difficult to do so while maintaining their semantic significance.

This confusion between keypoints is sometimes caused by symmetry. Human pose estimation deals with symmetry pre-defined pairs of key-points that can be flipped with each other (eg. left-ear, right-ear) when the image itself is flipped as a data augmentation. However, there are often more than a single axis of symmetry in objects, and it may be non-obvious which keypoints to flip.

Similar to [16], we primarily define keypoints on the line of symmetry itself to reduce ambiguity. However, important object properties such as edges or corners often lies outside that line or plane of symmetry, and makes it difficult to limit keypoints to only along such axis. A box, for example, is intuitively defined with keypoints at the corners.

The second scenario which causes confusion between similar keypoints is the presence of nearby objects of the same class. A great example of this can be seen in Figure 3d. The bounding box



(a) keypoint 0 *(b)* keypoint 1 *(c)* keypoint 2 *(d)* keypoint 3

(e) keypoint 4 *(f)* keypoint 5 *(g)* keypoint 6 *(h)* keypoint 7

Fig. 6: Visualization of heatmap predictions for a cracker box.

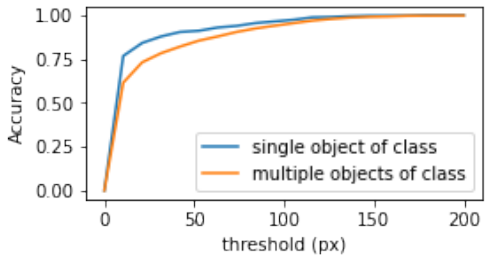


Fig. 7: ADD accuracy scores given different thresholds in projected image coordinates for images with multiple objects of the same class compared to images with only a single object of a given class

for the bottom drill (blue model, green keypoints) overlaps heavily with the top drill. When predicting the keypoints, only keypoint 4 (see Figure 5e for location) was correctly predicted on the bottom drill. The other points were all placed on the top drill. This suggests that the model is only focusing on the local features in the near vicinity of the keypoint location rather than the overall object. A comparison of ADD accuracy scores for images containing multiple objects of a given class vs those with a single object of a given class can be seen in Figure 7, with the former case performing notably worse.

5 Conclusion

Through training a heatmap-based pose detection model on the MetagraspNet dataset, we demonstrate that semantically important keypoints can be an effective way to estimate the pose of objects. However, the complexity of the available object classes results in keypoint definitions that do not have a consistent pattern, leading to ambiguities on where to place them on new objects.

Furthermore, the complex scenes with high level of occlusion, sometimes with similar objects on top of each other, present in the MetagraspNet dataset reveal challenges that were not well observed in previous implementations. In particular, the model struggles heavily with occluded keypoints as well as with the presence of points with similar visual features nearby. Those similar local features may be present on the same object, or on other objects close-by. We demonstrate this issue through observing that the probability of multiple peaks in the heat-map prediction varies based for different keypoints based on the local visual uniqueness of that point (as shown in Tables 2 and 3). We further demonstrate this problem through the decrease in ADD performance when multiple objects of the same class are present in the same object (as shown in Figure 7).

Future work may use the multi-peak probability to identify poor keypoints and iteratively improve the keypoint definition to ensure easy recognizably. In addition, we may also explore alternative ways to represent object poses, such as with categorical keypoints (as opposed to keypoints with unique ids), as well as volumetric primitives, so that the representation is less arbitrarily defined.

References

[1] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in

cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[2] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1521–1529.

[3] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.

[4] N. Pereira and L. A. Alexandre, "Maskedfusion: Mask-based 6d object pose estimation," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020, pp. 71–78.

[5] O. Hosseini Jafari, S. K. Mustikovela, K. Pertsch, E. Brachmann, and C. Rother, "ipose: instance-aware 6d pose estimation of partly occluded objects," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 477–492.

[6] Z. Li, G. Wang, and X. Ji, "Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7678–7687.

[7] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7668–7677.

[8] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 611–16 621.

[9] Y. Di, F. Manhardt, G. Wang, X. Ji, N. Navab, and F. Tombari, "So-pose: Exploiting self-occlusion for direct 6d pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 396–12 405.

[10] M. Rad and V. Lepetit, "Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3828–3836.

[11] A. Grabner, P. M. Roth, and V. Lepetit, "3d pose estimation and 3d model retrieval for objects in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[12] B. Chen, T.-J. Chin, and M. Klimavicius, "Occlusion-robust object pose estimation with holistic representation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2929–2939.

[13] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 632–11 641.

[14] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "Ffb6d: A full flow bidirectional fusion network for 6d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3003–3013.

[15] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3d object pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–134.

[16] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," in *The International Symposium of Robotics Research*. Springer, 2019, pp. 132–157.

[17] M. Robson and M. Sridharan, "A keypoint-based object representation for generating task-specific grasps," in *2022 IEEE International Conference on Automation Science and Engineering*, 2022.

[18] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-dof object pose from semantic keypoints," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2011–2018.

[19] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[20] Y. Chen, M. Gilles, E. Z. Zeng, and A. Wong, "Metagraspnet: A large-scale benchmark dataset for vision-driven robotic grasping via physics-based metaverse synthesis," in *2022 IEEE International Conference on Automation Science and Engineering*, 2022.

[21] M. Contributors, "Openmmlab pose estimation toolbox and benchmark," https://github.com/open-mmlab/mmpose, 2020.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[24] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "Bop challenge 2020 on 6d object localization," in *European Conference on Computer Vision*. Springer, 2020, pp. 577–594.

[25] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.