LocaliseBot: Multi-view 3D object localisation with differentiable rendering for robot grasping

Sujal Vijayaraghavan[1,3], Redwan Alqasemi[2,3], Rajiv Dubey[2,3], and Sudeep Sarkar[1,3]

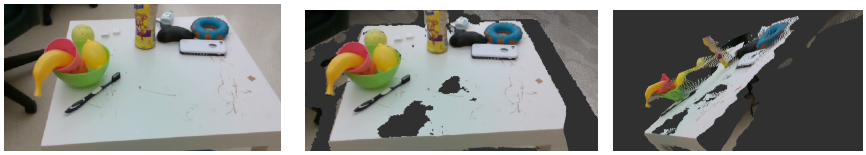
[1] Department of Computer Science
[2] Department of Mechanical Engineering
[3] University of South Florida, Tampa
{sujal,alqasemi,dubey,sarkar}@usf.edu

Abstract. Robot grasp typically follows five stages: object detection, object localisation, object pose estimation, grasp pose estimation, and grasp planning. We focus on object pose estimation. Our approach relies on three pieces of information: multiple views of the object, the camera's extrinsic parameters at those viewpoints, and 3D CAD models of objects. The first step involves a standard deep learning backbone (FCN ResNet) to estimate the object label, semantic segmentation, and a coarse estimate of the object pose with respect to the camera. Our novelty is using a refinement module that starts from the coarse pose estimate and refines it by optimisation through differentiable rendering. This is a purely vision-based approach that avoids the need for other information such as point cloud or depth images. We evaluate our object pose estimation approach on the ShapeNet dataset and show improvements over the state of the art. We also show that the estimated object pose results in 99.65% grasp accuracy with the ground truth grasp candidates on the Object Clutter Indoor Dataset (OCID) Grasp dataset, as computed using standard practice.

1 Introduction

The problem of grasp pose estimation is more challenging for gripper-based end effectors compared to suction-based ones. The former requires a good understanding of the object shape before grasping. Point cloud can provide valuable depth information, but methods that rely on depth sensor data such as point cloud information often suffer from missing or noisy data (see Figure 1).

Most grasp detection techniques operate in the pixel space. Typically, they rely on object regression followed by grasp pose regression. The estimations include detection of the object category, semantic segmentation, and for pose estimation for robot grasping, an algorithm that proposes grasp candidates. The regression lacks the crucial depth information. This results in a vast grasp candidate space to optimise. Other robot grasping techniques [6,20,14] utilise other available sensory information such as point cloud, for example.



(a) (b) (c)

Fig. 1. RGB image (a) of a scene, (b) its corresponding point cloud, and (c) and the point cloud from a different viewpoint. These images illustrate how point cloud/depth sensors can lose significant amounts of data from an original scene resulting in unstable performance of models relying on them

A 3D awareness of the objects can significantly reduce the search space for grasp pose candidates. This paper proposes utilising available knowledge of geometric shapes and properties of objects detected in the form of CAD models.

1.1 3D model fitting

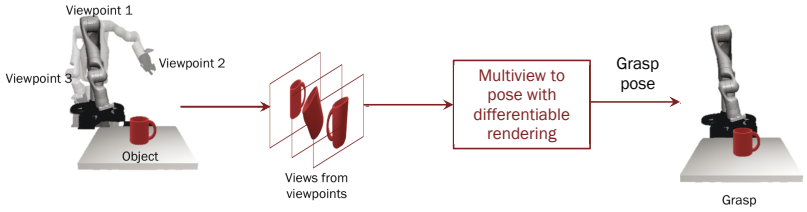
One of the primary challenges facing the task of 3D model alignment is the estimation of depth information of objects from 2D images. This is because the depth estimation problem lacks sufficient constraints from a single 2D image and consequently suffers from the *depth-scale ambiguity*, a long-time challenge tackled to date since the classical computer vision era.

In classical computer vision, camera calibration techniques [34] estimate structure from motion. Such methods generally rely on three pieces of information: images of a scene from multiple viewpoints, camera parameters, and mapping of keypoints of an object between the multiple viewpoints. While the former two artefacts are easy to obtain, the latter is not, especially for real-time applications.

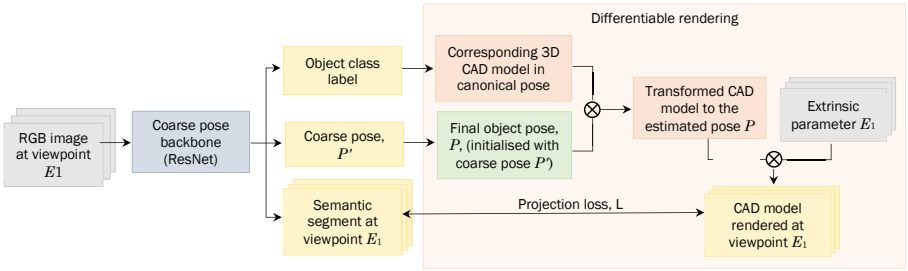
More recently, deep learning models are trained to directly estimate the 3D pose of objects in images.[18,10] Such models are trained on large datasets with the depth information annotated. They do not require any annotation at inference time. However, estimation of depth from a single viewpoint—even with a fully-trained network—is often inaccurate and unstable. Pose estiation from multiple views are now being applied to refine the initial estimates.[31]

In this work, we use the modified FCN ResNet [28] initialised with pre-trained weights and fine-tune it for coarse pose estimation. This, along with extrinsic camera parameters, are used for multiview pose refinement. Extrinsic camera parameters are often available in robot grasping problem scenarios. In order to reduce the high search space for the pose, we break down the search space into bins and, from them, select the ones with the best matching (Section 3.1). For run-time inference, multiple views in an explorative fashion are obtained (Figure 2a).

This method relies on RGB images, extrinsic camera parameters, and a knowledge base (of CAD models) and does not use point cloud or other depth



(a) Arm manipulation to capture the scene from multiple viewpoints



(b) Pipeline of the method

Fig. 2. An overview of the proposed approach which includes (a) capturing the scene from multiple viewpoints and (b) estimating the object pose from the multiple views. Multi-view pose refinement is achieved through differentiable rendering by accepting the multiple camera parameters, a 3D CAD model of the detected object, and projects them with a shared set of object pose parameters. The rendering at each viewpoint is compared with the observed image and the error is backpropagated to optimise the shared object pose parameters

sensor information, making it relevant to applications with simple RGB sensors. The overall pipeline is illustrated in Figure 2b.

2 Related work

Object pose estimation for robot grasping typically goes through three stages: object localisation, object pose estimation, and grasp pose estimation.[9] Our focus in this work is on object pose estimation and refinement. Existing methods [17,7,25,24] combine object localisation and object pose estimation into a single system (of one or more modules) and tackle the whole problem. For grasp pose estimation, some methods [40,27,43] rely on point clouds in addition to image features, whereas others use depth images.[36,4,26,5,41] In recent computer vision techniques, 3D object pose estimation is done through deep learning. Some techniques estimate object pose based on point cloud [22] or depth images.[42]

Existing methods rely on a variety of available modes of data and effectively estimate grasp pose estimation. Deep learning models [45,38,1] are trained

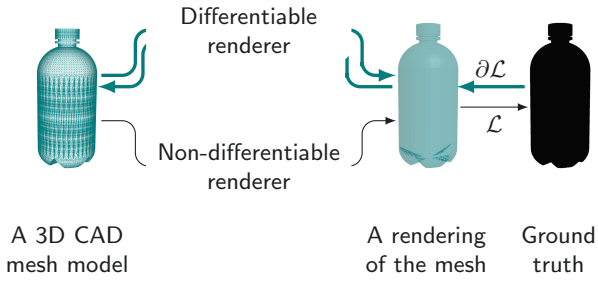


Fig. 3. Differentiable renderer preserving the forward rendering link allowing back-propagation

to implicitly estimate the object pose. An iterative inference step as a downstream task over learnt deep models is also applied to further improve estimation accuracy.[44] A recent review [9] recounts the literature based on object localisation, pose estimation, and grasp estimation.

2.1 3D pose estimation

Numerous techniques exist for various degrees of pose estimation.[23,32] Different techniques have been applied depending on available data and modes.

More recent works rely on established detection models and built on them. For example, Mesh-RCNN [10] uses Mask-RCNN [11] as a backbone to predict a coarse voxel representation of the detected object. Further refinement on it is achieved by learning a graph neural network to infer the shape.

ShapeMask,[19] augmenting on ResNET,[12] learns to output a feature vector for the object detected. This end-to-end architecture learns to detect objects and output a vector summarising geometric features of the object.

Mask2CAD [18] learns to map the latent vector of object features generated by ShapeMask to a latent representation of 3D CAD models. In effect, the model can detect objects and retrieve an appropriate 3D CAD model. A third head is trained also to predict the object pose.

Depth estimation from a single image, however, is often unstable and less reliable. Video2CAD [31] builds on Mask2CAD to obtain a coarse pose of the object, extending the idea to multiple views in a video. Explicit constraints for localisation and scaling of the object are placed across image frames and optimised for the overall videoframes. Multiple views with sufficient distinction provide the necessary information to deduce scale and depth.

2.2 Differentiable rendering

Backpropagation is computationally possible if a traceable link of gradients from the source to the destination and vice versa can be established. With the conventional graphics rendering pipeline, this link is lost. Differentiable rendering is

a technique that preserves this link (Figure 3). The first known general-purpose differentiable rendering was proposed in 2014 [29]. Since then, numerous variants for various use cases have been developed.[16]

Differentiable rendering allows a 3D model to be iteratively rendered and optimised for its parameters. This method has been applied to estimate camera parameters and volumetric fitting. We apply it to estimate object pose.

3 Method

The proposed method consists of a trainable module appended with a fine tuning unit. The former is a single-view object localisation and pose prediction unit followed by differentiable rendering refinement unit (Section 3.1).

3.1 Refinement

We estimate the 3D pose and location of an object in the real world as a 3D homography problem with two pieces of information: images from multiple views of a scene and the camera's (extrinsic) parameters at each viewpoint. The pose and location estimation is done with differentiable rendering.

Parameters estimated The full pose estimation of an object includes nine degrees of freedom, *viz.,* its placement in the world, or the *translation vector* $\boldsymbol{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top$; its size, or the *scale vector* $\boldsymbol{s} = \begin{bmatrix} s_x & s_y & s_z \end{bmatrix}^\top$; and its orientation, or the *rotation matrix* $\boldsymbol{R}$ composed of the SO(3) rotation angles.

Problem formulation and optimisation Consider an object captured from two known viewpoints in the world, $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$. These viewpoints describe the camera's (extrinsic) parameters. Let the images so captured be denoted $\boldsymbol{I}_1 \in \mathbb{R}^2$ and $\boldsymbol{I}_2 \in \mathbb{R}^2$, respectively. As set out in the introduction to this paper in Section 1, the reason for using images from two (or more) viewpoints is to obtain sufficient constraints that are necessary for depth estimation.

Let $\boldsymbol{v} \in \mathbb{R}^3$ be a set of 3D vertices that form the 3D CAD model of the object. Now, with a differentiable rendering function $f : \mathbb{R}^3 \mapsto \mathbb{R}^2$, the CAD model $\boldsymbol{v}$ is rendered from the aforementioned two viewpoints onto two canvasses, denoted $\hat{\boldsymbol{I}}_1$ and $\hat{\boldsymbol{I}}_2$, respectively. The 3D mesh so rendered is initially positioned at some random pose $\hat{\boldsymbol{\Phi}} = \begin{bmatrix} \hat{\boldsymbol{s}}\hat{\boldsymbol{R}} | \hat{\boldsymbol{t}} \end{bmatrix}$ in the world. Our goal is to estimate the actual pose $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{s}\boldsymbol{R} | \boldsymbol{t} \end{bmatrix}$, as it is in the world coordinate system.

We could denote the above setup as $\hat{\boldsymbol{I}}_1 := f(\hat{\boldsymbol{\Phi}}; \boldsymbol{v}, \boldsymbol{E}_1)$ and $\hat{\boldsymbol{I}}_2 := f(\hat{\boldsymbol{\Phi}}; \boldsymbol{v}, \boldsymbol{E}_2)$. Or, more generally, for any viewpoint n,

$$\hat{\boldsymbol{I}}_n := f(\hat{\boldsymbol{\Phi}}; \boldsymbol{v}, \boldsymbol{E}_n) \tag{1}$$

This setup is illustrated in Figure 4. Note that the estimated object pose $\hat{\boldsymbol{\Phi}}$ (or the original pose $\boldsymbol{\Phi}$) in the world coordinate system is, of course, the

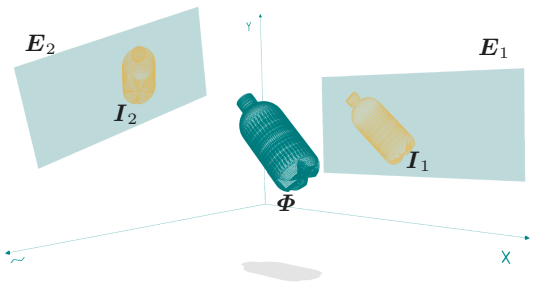


Fig. 4. 3D homography: two images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ of an object are captured from two viewpoints $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$, respectively. From this information, the object is rendered on both canvasses jointly until the object's renditions coincide with its projections in the corresponding original images. This yields us the object's pose and location in the world, $\boldsymbol{\Phi}$. This illustration shows two viewpoints. The method is generalisable to any number of viewpoints.

same for all viewpoints. The many viewpoints provide good constraints to more accurately deduce the depth of the object, t_z, thus the scale $\boldsymbol{s}$, jointly resolving the depth-scale ambiguity.

Objective The rendering $\hat{\boldsymbol{I}}_1$ is compared with its ground truth $\boldsymbol{I}_1$, and $\hat{\boldsymbol{I}}_2$ is compared with its ground truth $\boldsymbol{I}_2$. Any discrepancy is backpropagated to refine the initial estimates $\hat{\boldsymbol{\Phi}}$. This is where differentiable rendering comes in handy:

$$\frac{\mathrm{d}}{\mathrm{d}\hat{\boldsymbol{\Phi}}}\left(\hat{\boldsymbol{I}} \circ \boldsymbol{I}\right) = 0 \tag{2}$$

where "$\circ$" is an apt comparison operator defined by a loss function.

Choosing the right comparison metric that can nudge the CAD model to align with the ground truth is key to the success of this optimisation step. Hausdorff distance [3] $\mathcal{L}_{\mathrm{H}}$ acts as an objective to minimise the error between the contours of the predicted rendered poses and the target. However, a rendering beyond the canvas will result in no Hausdorff loss. To ensure that the estimated poses remain well within the canvas, the intersection over union (IoU) loss $\mathcal{L}_{\mathrm{IoU}}$ is used.

The combined loss function is given by

$$\mathcal{L}(\boldsymbol{I}, \boldsymbol{E}; \hat{\boldsymbol{\Phi}}, \boldsymbol{v}) = \lambda_1 \mathcal{L}_{\mathrm{IoU}}(\hat{\boldsymbol{I}}, \boldsymbol{I}) + \lambda_2 \mathcal{L}_{\mathrm{H}}(\hat{\boldsymbol{I}}, \boldsymbol{I}) \tag{3}$$

The coefficients λ_i control the importance of each loss term.

Plugging Equation (3) in Equation (2), we have $\frac{\mathrm{d}}{\mathrm{d}\hat{\boldsymbol{\Phi}}}\mathcal{L}(\boldsymbol{I}, \boldsymbol{E}; \hat{\boldsymbol{\Phi}}) = 0$, or the computationally achievable objective $\boldsymbol{\Phi}^* = \arg\min_{\hat{\boldsymbol{\Phi}}} \mathcal{L}(\boldsymbol{I}, \boldsymbol{E}; \hat{\boldsymbol{\Phi}})$. For multiview optimisation, this expression is rewritten as

$$\boldsymbol{\Phi}^* = \arg\min_{\hat{\boldsymbol{\Phi}}} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(\boldsymbol{I}_n, \boldsymbol{E}_n; \hat{\boldsymbol{\Phi}}) \tag{4}$$

where $N \geq 2$ is the number of viewpoints from which the scene has been captured. In classical techniques, a single-step optimisation strategy such as Newton's method were used.

Coarse-to-fine orientation optimisation Optimisation for all possible orientation Euler angles is a vast space of $360° \times 360° \times 360°$. To reduce this search space, it is divided into k bins along each axis, resulting in a reduced k^3 search space. This coarse level yields a few best poses, or a few best bins, n of which are further pursued to a finer detail *i.e.*, an initial k^3 searches and then $\frac{360^3}{k^3}$ for the n best bins, totalling up to $k^3 + n\left(\frac{360}{k}\right)^3$.

4 Experiments

4.1 Evaluation metrics

Pose estimation using ADD The average distortion distance (ADD) between the estimated pose and the ground truth on an object $\mathcal{M}$ is a frequently used metric to report the accuracy of pose estimation in 3D space, given by $\frac{1}{m}\sum_{\boldsymbol{x}\in\mathcal{M}} \|(\boldsymbol{R}\boldsymbol{x}+\boldsymbol{t}) - (\tilde{\boldsymbol{R}}\boldsymbol{x}+\tilde{\boldsymbol{t}})\|_2$, where $\boldsymbol{R}$ and $\boldsymbol{t}$ are the ground truth pose, and $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{t}}$ are the predicted pose. It computes the average Euclidian distance between the estimation and the ground truth. For discrete objects, this metric is the object's centroid. For symmetric objects, due to the ambiguity arising between points for certain views, the average closest point distance (ADD-S) [13], given by $\frac{1}{m}\sum_{\boldsymbol{x}_1\in\mathcal{M}} \arg\min_{\boldsymbol{x}_2\in\mathcal{M}} \|(\boldsymbol{R}\boldsymbol{x}_1+\boldsymbol{t}) - (\tilde{\boldsymbol{R}}\boldsymbol{x}_2+\tilde{\boldsymbol{t}})\|_2$, is adopted. Several seminal and recent works [39,44] evaluate their methods using ADD and ADD-S. Any grasp pose within a threshold of ADD or ADD-S is deemed correct.

Grasp pose rectangle We evaluate our method using the *grasping rectangle* metric. It was designed and published in 2011 [15] to formalise and evaluate grasp pose estimations and is used as a standard metric for evaluating grasp pose. Several state-of-the-art works [1,30,8,2] evaluate their methods using grasping rectangle. More advanced and effective metrics have been developed, a recent one of which [37] accounts for varying units and scales of features describing the object of interest.

The grasping rectangle metric is a binary decision function which, given a grasp pose, evaluates it as correct or incorrect. A proposed grasp pose is deemed correct if the proposed angle falls within a certain threshold angle ($< 30°$) from the ground truth and the proposed grasp meets a certain threshold intersection-over-union ($> 25\%$) from the grouth truth.

4.2 Dataset

OCID Grasp Object Clutter Indoor Dataset (OCID)[35] is a dataset containg 96 cluttered scenes, 89 different objects, and and over 2k point cloud annotations.

The OCID Grasp dataset[1] is created from OCID by manually annotating subsets of the latter. These annotations include an object class label and a corresponding grasp candidate. OCID Grasp consists of 1763 RGB-D images filtered from OCID, and contains over 11.4k segmented masks and over 75k grasp candidates.

4.3 Setup

We train and experiment our method in the PyTorch environment. 3D rendering and differentiable rendering are achieved with PyTorch 3D.[33]

4.4 Results

We evaluate object pose estimation by computing the amount of overlap between the object projected with the estimated pose against the ground truth. Table 1 shows the results obtained by computing the IoU under three conditions, *viz.,* at least 25%, 50%, and 75% overlaps.

Method	**IoU** > 0.25	**IoU** > 0.5	**IoU** > 0.75
ODAM [21]	64.7/58.6/61.5	31.2/28.3/29.7	3.8/3.5/3.6
Vid2CAD [31]	56.9/55.7/56.3	34.2/**33.5**/**33.9**	**10.7**/**10.4**/**10.5**
Ours	**64.9**/**59.6**/**62.0**	**34.7**/31.6/29.2	9.6/9.3/8.4

Table 1. Precision/recall/F1 measures (tested on the ScanNet dataset)

Category	**Single frame**	**Multiple (3) frames**
bathtub	22.4	62.3
bookshelf	13.5	51.8
cabinet	26.3	48.6
chair	27.4	51.4
sofa	24.3	48.7
table	15.4	51.4
dustbin	27.9	58.7
others	23.4	47.4
global avg.	23.3	56.7

Table 2. Quantitative evaluation on the ScanNet dataset. F1 measures by category

Quantitative evaluation We apply the method on the Kinova Jaco arm to examine real-time application. Since multiple views are not readily available in the real-time setting, the robot arm makes discrete stops around an object of interest at those viewpoints.

Table 2 shows the improvement of accuracy (by F1) in segmentation by comparing against the number of frames (one versus three). Pose refinement through differentiable rendering increases with more time frames (Figure 5) and accepts a variable number of frames. We observe that after a certain threshold on the number of frames, there is very little information to be gained from other viewpoints, and the convergence rate stabilised after a few distinct frames.

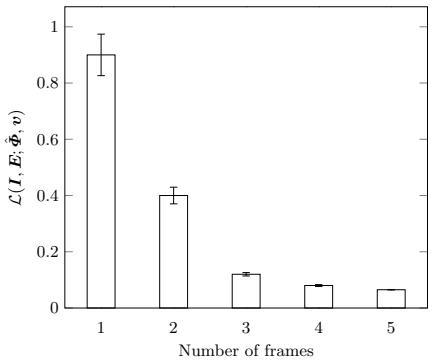


Fig. 5. Loss convergence for different number of frames. It also shows that the error range sharpens as the number of frames increases

Table 3 shows a simpler Euclidean distance metric. On a set of five distinct objects, several sets of trials were conducted with the Kinova Jaco arm. The distance is scaled to be between 0 to 1, 0 implying pinpoint accuracy and 1 being no movement of the robot arm.

Trials	Sphere	Cube	Banana	Coffeemug	bowl
50	0.12	0.12	0.17	0.14	0.16
75	0.11	0.07	0.14	0.12	0.16
100	0.09	0.03	0.12	0.09	0.18

Table 3. Euclidian distance of the predicted and reached location in world space with respect to the base of the robot. These figures are computed by normalising the distance between the robot hand and the object over the distance between the object and the robot base; 0 signifies no gap between the prediction and the ground truth, hence the best estimation; 1 means the arm has effectively not moved

4.5 Object pose estimation for grasping

To determine the cooperation of this refinement unit with grasping, we predict object pose on the OCID Grasp dataset and the grap candidates are directly selected from the annotations. Table 4 is a sanity check ensuring that with an ideal candidate proposal and grasp algorithm, the object pose estimation produces desriable results.

Dataset	**Grasp accuracy**(%)	**IoU**
OCID Grasp	99.65%	99.32%

Table 4. To verify the realism of object pose estimation, we apply the grasp pose angles directly obtained from the annotations in the OCID Grasp dataset after transforming to the camera coordinates. Accurate object pose estimation results in a successful grasp. Conversely, a successful grasp implies accurate object pose estimation.

5 Conclusion

In this work, we present an optimisation strategy for object pose estimation by refining coarse estimations through multi-view differentiable rendering. This approach avoids rich sensor data such as point clouds or other depth data and relies on RGB images and camera parameters at different viewpoints. This approach is comparable to or outperforms the state of the art under different conditions. We experiment this method to evaluate pose estimation depending on the grasp success rate by applying annotated grasp candidates. This method can be augmented to any well-performing segmentation model and prepended to any grasp candidate estimation algorithm.

Limitation. Since the refinement module is an online optimisation unit, pose estimation optimisation happens right at inference time. This is a limitation on time-sensitive application.

Acknowledgements. This work is part of a research grant funded by the National Science Foundation.

References

1. Ainetter, S., Fraundorfer, F.: End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 13452–13458. IEEE (2021) 3, 7
2. Asif, U., Tang, J., Harrer, S.: Graspnet: An efficient convolutional neural network for real-time grasp detection for low-powered devices. In: IJCAI. vol. 7, pp. 4875–4882 (2018) 7
3. Aspert, N., Santa-Cruz, D., Ebrahimi, T.: Mesh: Measuring errors between surfaces using the hausdorff distance. In: Proceedings. IEEE international conference on multimedia and expo. vol. 1, pp. 705–708. IEEE (2002) 6
4. Bai, F., Zhu, D., Cheng, H., Xu, P., Meng, M.Q.H.: Active semi-supervised grasp pose detection with geometric consistency. In: 2021 IEEE International Conference on Robotics and Biomimetics (ROBIO). pp. 1402–1408. IEEE (2021) 3
5. Buchholz, D., Futterlieb, M., Winkelbach, S., Wahl, F.M.: Efficient bin-picking and grasp planning based on depth data. In: 2013 IEEE International Conference on Robotics and Automation. pp. 3245–3250. IEEE (2013) 3
6. Chen, W., Jia, X., Chang, H.J., Duan, J., Shen, L., Leonardis, A.: Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1581–1590 (2021) 1
7. Chéron, G., Laptev, I., Schmid, C.: P-cnn: Pose-based cnn features for action recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 3218–3226 (2015) 3
8. Chu, F.J., Vela, P.A.: Deep grasp: Detection and localization of grasps with deep neural networks. arXiv preprint arXiv:1802.00520 (2018) 7
9. Du, G., Wang, K., Lian, S., Zhao, K.: Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. Artificial Intelligence Review **54**(3), 1677–1734 (2021) 3, 4
10. Gkioxari, G., Malik, J., Johnson, J.: Mesh R-CNN. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019) 2, 4
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) 4
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 4
13. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Asian conference on computer vision. pp. 548–562. Springer (2012) 7
14. Huang, X., Mei, G., Zhang, J., Abbas, R.: A comprehensive survey on point cloud registration. arXiv preprint arXiv:2103.02690 (2021) 1
15. Jiang, Y., Moseson, S., Saxena, A.: Efficient grasping from rgbd images: Learning using a new rectangle representation. In: 2011 IEEE International conference on robotics and automation. pp. 3304–3311. IEEE (2011) 7
16. Kato, H., Beker, D., Morariu, M., Ando, T., Matsuoka, T., Kehl, W., Gaidon, A.: Differentiable rendering: a survey (2020) 5

17. Kumra, S., Joshi, S., Sahin, F.: Antipodal robotic grasping using generative residual convolutional neural network. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 9626–9633. IEEE (2020) 3
18. Kuo, W., Angelova, A., Lin, T.Y., Dai, A.: Mask2CAD: 3D shape prediction by learning to segment and retrieve. In: European Conference on Computer Vision. pp. 260–277. Springer (2020) 2, 4
19. Kuo, W., Angelova, A., Malik, J., Lin, T.Y.: Shapemask: Learning to segment novel objects by refining shape priors. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019) 4
20. Le, T.T., Le, T.S., Chen, Y.R., Vidal, J., Lin, C.Y.: 6d pose estimation with combined deep learning and 3d vision techniques for a fast and accurate object grasping. Robotics and Autonomous Systems **141**, 103775 (2021) 1
21. Li, K., DeTone, D., Chen, Y.F.S., Vo, M., Reid, I., Rezatofighi, H., Sweeney, C., Straub, J., Newcombe, R.: Odam: Object detection, association, and mapping using posed rgb video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5998–6008 (2021) 8
22. Li, X., Wang, H., Yi, L., Guibas, L.J., Abbott, A.L., Song, S.: Category-level articulated object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3706–3715 (2020) 3
23. Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: DeepIM: deep iterative matching for 6D pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018) 4
24. Li, Z., Ji, X.: Pose-guided auto-encoder and feature-based refinement for 6-dof object pose regression. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 8397–8403. IEEE (2020) 3
25. Li, Z., Wang, G., Ji, X.: 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. pp. 7678–7687 (2019) 3
26. Litvak, Y., Biess, A., Bar-Hillel, A.: Learning pose estimation for high-precision robotic assembly using simulated depth images. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 3521–3527. IEEE (2019) 3
27. Liu, H., Cao, C.: Grasp pose detection based on point cloud shape simplification. In: IOP Conference Series: Materials Science and Engineering. vol. 717, p. 012007. IOP Publishing (2020) 3
28. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015) 2
29. Loper, M.M., Black, M.J.: OpenDR: An approximate differentiable renderer. In: European Conference on Computer Vision. pp. 154–169. Springer (2014) 5
30. Luo, Z., Tang, B., Jiang, S., Pang, M., Xiang, K.: Grasp detection based on faster region cnn. In: 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM). pp. 323–328. IEEE (2020) 7
31. Maninis, K.K., Popov, S., Niesser, M., Ferrari, V.: Vid2CAD: CAD model alignment using multi-view constraints from videos. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022) 2, 4, 8
32. Pitteri, G., Bugeau, A., Ilic, S., Lepetit, V.: 3D object detection and pose estimation of unseen objects in color images with local surface embeddings. In: Proceedings of the Asian Conference on Computer Vision (ACCV) (November 2020) 4
33. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv:2007.08501 (2020) 8

34. Song, L., Wu, W., Guo, J., Li, X.: Survey on camera calibration technique. In: 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics. vol. 2, pp. 389–392 (2013). https://doi.org/10.1109/IHMSC.2013.2402
35. Suchi, M., Patten, T., Fischinger, D., Vincze, M.: Easylabel: a semi-automatic pixel-wise object annotation tool for creating robotic rgb-d datasets. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 6678–6684. IEEE (2019) 7
36. Supancic, J.S., Rogez, G., Yang, Y., Shotton, J., Ramanan, D.: Depth-based hand pose estimation: data, methods, and challenges. In: Proceedings of the IEEE international conference on computer vision. pp. 1868–1876 (2015) 3
37. Tan, T., Alqasemi, R., Dubey, R., Sarkar, S.: Formulation and validation of an intuitive quality measure for antipodal grasp pose evaluation. IEEE Robotics and Automation Letters **6**(4), 6907–6914 (2021) 7
38. Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6d object pose prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 292–301 (2018) 3
39. Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., Birchfield, S.: Deep object pose estimation for semantic robotic grasping of household objects. arXiv preprint arXiv:1809.10790 (2018) 7
40. Vohra, M., Prakash, R., Behera, L.: Real-time grasp pose estimation for novel objects in densely cluttered environment. In: 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). pp. 1–6. IEEE (2019) 3
41. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: Densefusion: 6d object pose estimation by iterative dense fusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3343–3352 (2019) 3
42. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2642–2651 (2019) 3
43. Wei, W., Li, D., Wang, P., Li, Y., Li, W., Luo, Y., Zhong, J.: Dvgg: Deep variational grasp generation for dextrous manipulation. IEEE Robotics and Automation Letters (2022) 3
44. Wu, Y., Fu, Y., Wang, S.: Deep instance segmentation and 6d object pose estimation in cluttered scenes for robotic autonomous grasping. Industrial Robot: the international journal of robotics research and application (2020) 4, 7
45. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199 (2017) 3