arXiv:1903.03953v1 [cs.CV] 10 Mar 2019

Domain Randomization for Active Pose Estimation

Xinyi Ren†, Jianlan Luo†, Eugen Solowjow‡, Juan Aparicio Ojea‡,
Abhishek Gupta†, Aviv Tamar*, Pieter Abbeel†
† UC Berkeley
‡ Siemens Corp
* Technion; work done while at UC Berkeley

***Abstract*— Accurate state estimation is a fundamental component of robotic control. In robotic manipulation tasks, as is our focus in this work, state estimation is essential for identifying the positions of objects in the scene, forming the basis of the manipulation plan. However, pose estimation typically requires expensive 3D cameras or additional instrumentation such as fiducial markers to perform accurately. Recently, Tobin et al. introduced an approach to pose estimation based on domain randomization, where a neural network is trained to predict pose directly from a 2D image of the scene. The network is trained on computer generated images with a high variation in textures and lighting, thereby generalizing to real world images. In this work, we investigate how to improve the accuracy of domain randomization based pose estimation. Our main idea is that active perception – moving the robot to get a better estimate of pose – can be trained in simulation and transferred to real using domain randomization. In our approach, the robot trains in a domain-randomized simulation how to estimate pose from a *sequence* of images. We show that our approach can significantly improve the accuracy of standard pose estimation in several scenarios: when the robot holding an object moves, when reference objects are moved in the scene, or when the camera is moved around the object.**

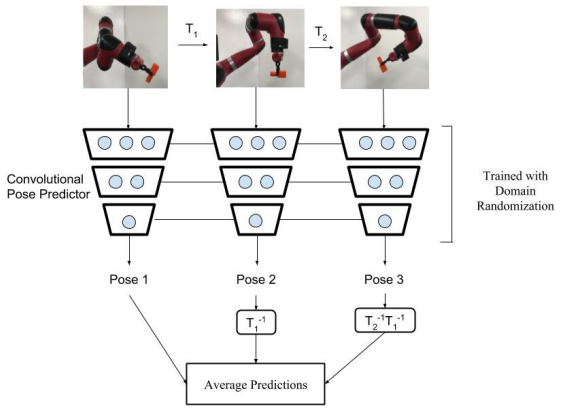


Fig. 1: Inverse Transform Domain Randomization: We show that we can improve the accuracy of real world pose prediction with multiple images of a scene with known geometric transformations between object poses in the scenes.

I. Introduction

In the past decades, robots have become dominant in industrial automation. A recent trend in manufacturing is the move toward small production volumes and high product variability [1], where reducing the manual engineering for automation becomes important. For automating many industrial tasks, such as picking, binning, or assembly, accurate pose estimation is essential. In this work, we focus on model based pose estimation from RGB cameras. This setting is relevant to many industrial applications, where a 3D model of the objects can easily be obtained, while it does not require expensive hardware such as high-precision depth cameras [2], [3], nor making modifications to the object such as adding markers [4]. Methods using markers often require significant human effort and have limited accuracy when the marker is far away or perpendicular to the image plane.

While a number of methods have been proposed for model-based pose estimation using expensive depth cameras or extensive labelled datasets in the real world [5], [6], the cost and manual effort required for these methods prevent them from being widely and easily applicable. Recently proposed methods proposing leveraging simulation as a tool for model-based pose estimation given accurate models of objects in an environment [7], [8], [9]. These methods are typically trained by leveraging known poses in simulation and training pose estimators which transfer effectively to the real world, bridging the simulation to reality gap.

In [7], it was shown that domain randomization was able to reach a 1.5 cm error on 3D pose estimation. Many robotic tasks, such as assembly or bin placing, require a much higher precision. In this work, we investigate how to improve the accuracy of pose estimation based on domain randomization such that it is suitable for high precision robotic assembly tasks.

In this work, we aim to improve the accuracy of domain randomization based pose estimation by making the observation that robots don't have to be passive observers of a scene and can in fact interact with objects in the scene. We can perform a *known geometrical transformation* to the scene, such as moving objects in the scene, moving the arm or distractors, or changing the camera angle. Since this transformation between scenes is known and applied by the robot, all of these datapoints can be used in order to imporve the accuracy of pose predictions. We use this idea to propose a method for *active* pose estimation, which exploits the fact that being able to see an object from different angles and in different positions leads to a more accurate and robust predictions. In this way,

we find that actually leveraging consistency among multiple different images of a scene ensures a much more accurate pose estimation compared to standard ensemble methods such as domain randomization.

Using models for active pose estimation transferred from simulation, we are able to decrease the average error predicted on real camera image from 2 cm to under 0.5cm, which is sufficient to enable a variety of high precision robotic manipulation tasks which were otherwise very challenging with current methods.

II. Related Work

Our work has several connections to a number of prior works from various fields

a) 2D model based pose estimation: Model based pose estimation of rigid objects from a 2D image has been studied extensively, largely building on a predefined feature points [10], [11], [12], [13], edge detectors [14], [15], or image templates [16]. We refer to [17] for an extensive survey. Most of these algorithms rely on careful selection of the features to track, or on textured surfaces for points matching, and a careful calibration of the RGB camera. Our approach is agnostic to these factors.

b) 3D model based pose estimation: Using a depth camera, high precision pose estimation can be obtained [2], [3]. However, accurate depth cameras (e. g. a Photoneo) can be very expensive, limiting their use in many applications. Our approach only requires a 2D RGB image.

c) Fiducial markers: The use of fiducial markers has become popular in augmented reality and robotics applications [18], [4], [19]. However, in realistic industrial applications, adding fiducials to objects may be undesirable, and the accuracy of fiducial based pose detection is limited for certain poses (for example, when the fiducial is perpendicular to the image plane). Our approach does not require any external modification of the object for pose detection.

d) Pose estimation based on supervised learning: Several recent studies learned to map an image directly to pose using deep convolutional neural networks (CNNs) [5], [6]. While the CNN structure in these works is similar to ours, these works require a labeled training set for learning, which can be difficult to obtain. The domain randomization approach, in contrast, generates its own training data by rendering in simulation.

e) Active perception: The study of active perception [20], [21] concerns how a robot should take actions to better estimate parameters of its environment. To our knowledge, our work is the first to study active perception in a simulation-to-real setting.

f) Domain randomization: The gap between simulation and reality has been challenging robotics for decades. Recent work on trying to bridge this gap learns a decision making policy in simulation that works well under a wide variation in the simulation parameters, with the hope of learning a robust policy that transfers well to the real world. This idea has been explored for navigation [8] and pose estimation [7], by varying visual properties in the scene, and also for locomotion [22] and grasping [23], by varying dynamics in simulation. In this work, we consider variation in the visual domain, and combine domain randomization with active perception, to improve its accuracy in pose detection.

III. Problem Formulation and Preliminaries

We consider a model-based rigid body pose estimation problem. In our setting, we assume that we have geometrical 3D models of an object x and some reference object y. Let O_y denote a coordinate frame relative to y, and let P_x denote the 6D pose of x in the coordinate frame O_y. We are given an image of the scene I, that contains x and y, and our goal is to estimate P_x from the image.

A. Pose estimation based on Domain Randomization

Tobin et al. [7] proposed a domain randomization method for solving the pose estimation problem described above. In this method, a 3D rendering software is used to render scene images with different poses of x and y, and random textures, lighting conditions, camera orientations, and camera parameters. Let $D = \{I^1, P_x^1 ..., I_N, P_x^N\}$ denote the data set of the rendered images and matching object poses (which are known, by construction). Supervised learning is then used to train a deep neural network mapping I to P_x. Since the network is trained to work on various texture, camera, and lighting conditions, it is expected that it also works on real world images since their statistics would roughly fall under the extremely wide distribution that was trained on. By making the training distribution extremely broad in terms of components such as texture, camera, and lighting, this method is able to ensure generalization to real world test environments by reducing the covariate shift. Indeed, the method in [7] reportedly obtained an average 1.5 cm error in predicting 3D pose on real world test images.

IV. Method

In this work, we propose an active perception approach based on domain randomization. To motivate our approach, we start by discussing the working hypothesis underlying domain randomization:

Working Hypothesis (Domain Randomization)**.** *There exist a set of features that can be extracted from all images in the data and are sufficient for predicting the image label (pose). These features can also be extracted from real images and are sufficient for predicting the real label.*

This working hypothesis means that if the training data is sufficiently randomized, and the neural network is expressive enough, then with enough data, the model has to discover the features which are common to all images, and base its prediction only on these features (otherwise it would suffer a higher training loss on spurious correlations that it picks up on). In that case, the network predictions are likely to transfer well to the real world.

One may question whether such features should even exist. However, for pose prediction, we know that the relative pose P_x is a purely geometrical property of the objects,

and since we assume an accurate 3D model of x, then geometrical properties (e.g., relative sizes and shapes) should be maintained in all the rendered images and also in the real images. Thus, the network has the potential to learn predictions based solely on geometrical properties of objects, abstracting away any other visual cues such as textures and lighting, and such features should transfer well to real images.

As discussed thus far, this pose estimation process is done completely passively. The robot does not interact with objects in the scene, but simply observes a single image of the scene and needs to predict the pose. In this work, we provide a key insight that we *can* in-fact interact with the scene, and apply known geometric transformations to objects in the scene. These transformations allow us to obtain a number of different images of the scene to estimate the pose of the object as the transformations are all *applied by us*. In this sense, we propose an active procedure to improve pose estimation by interacting with the scene and using multiple images to make a better prediction.

A. Active Perception based on Domain Randomization with Geometric Transformations

Recall that in the standard domain randomization problem (Section III-A), training data is in the form of image-pose pairs, $\{I, P_x\}$. Following the active perception paradigm [20], we can apply to the scene some *known* geometric transformation, with the hope that it improves our perception capabilities. For example, consider a robotic arm grasping an object, and the problem of estimating the position of the object within the robot's gripper. In this case, we can move the gripper closer to the camera to obtain a better pose estimate. Since we know the transformation applied when moving the gripper, we can potentially combine several images to obtain a better prediction. As another example, consider moving the camera to obtain a better view of the object.

Concretely, we define the **Domain Randomization with Geometric Transformations** problem (DR-GT). let $T_1, \ldots, T_k$ denote a set of k transformations that can actively be applied to the geometry of the scene *both in the real world and in simulation*. In particular, we consider rigid body transformations applied to objects in the scene and to the camera [24]. We propose to generate training data in the form of tuples $\{I, T_1(I), \ldots, T_k(I), P_x, T_1(P_x), T_k(P_x)\}$, where, slightly abusing notation, we denote by $T_i(I)$ and $T_i(P_x)$ the rendered image and pose when applying transformation T_i to the scene. The supervised learning problem we consider now is learning a mapping from $I, T_1(I), \ldots, T_k(I), T_1, \ldots, T_k$ to P_x.

B. Inverse Transform based Domain Randomization

To solve the DR-GT problem, we propose the following method, based on inverse transforms. Let T_i^{-1} denote the inverse transform of T_i.[1] Let f be the standard domain randomization mapping from I to P_x. Then, we propose to calculate

$$\begin{aligned} P_{x;0} &= f(I), \\ P_{x;1} &= T_1^{-1}(f(T_1(I))), \\ &\ldots, \\ P_{x;k} &= T_k^{-1}(f(T_k(I))). \end{aligned} \tag{1}$$

Note that for each $i \in 0, \ldots, k$, the inverse transformation in (1) means that the prediction $P_{x;i}$ is an estimate of P_x. Therefore, we can predict P_x as the sample average:

$$\hat{P_x} = \frac{1}{k+1} \sum_{i=0}^{k} P_{x;i}.$$

We term this method Inverse Transform based Domain Randomization (ITDR). We expect that as we enlarge the number of transformations k, the precision of ITDR improves. While it is seemingly naive to use the sample average as a prediction, we found that it is surprisingly effective compared to more complicated methods with models which consider several images at once as input and produce a single pose estimate directly.

The key intuition behind using ITDR for improved estimation is that using known transformations in an environment allows us to use a wider data distribution to make several predictions of the same pose. Since several of these transformations provide easier to model prediction problems than the original problem, it makes the accuracy of the model significantly higher in the real world.

V. Model Architecture

In order to perform accurate pose estimation directly from images, we used a convolutional neural network architecture [25]. The neural network takes a single image as input, and generates a pose as output. In our experiments we investigated predicting 3DoF pose composed of 2DoF translation and 1DoF rotation. The model architecture takes in an RGB image through 16 convolutional layers, each two convolutional layers is followed by a max-pooling operation and a ReLU nonlinearity. These convolutional layers are followed by 3 fully connected layers with decreasing hidden units and ReLU nonlinearity. This architecture is similar to the one used in [7], based on the VGG architecture [26] using convolution layers pretrained on ImageNet. The loss function for training this model is a combination of L1 regression loss for the 2DoF translation, and a cosine loss for the orientation, given by

$$L(x, \theta) = ||x - \hat{x}|| + ||cos(\theta - \hat{\theta}) - 1||,$$

where x and $\hat{x}$ are the true and predicted pose, and θ and $\hat{\theta}$ are true and predicted orientation.

For active pose estimation, we pass a number of different images through the same network and then average the predictions *after* applying a known rigid transform between them, as described in the ITDR algorithm.

[1] We restrict our approach to transformations with a well-defined inverse, such as rotations and translations.

Fig. 2: Simulation to reality transfer with active reference object tabletop movement

VI. Experiments

In this section we report our experiments on active perception using domain randomization. In our investigation, we aim to determine whether geometric transformations can give significantly better performance for model-based pose estimation in the real world. We designed our experiments to investigate whether using the robot to actively move elements of the scene yields gains in estimating the pose of objects in the scene. In particular, we investigate the performance of ITDR in the following situations: (1) Moving reference objects in the environment, (2) Moving a robot manipulator holding an object, and (3) Moving a camera held by the robot. We believe that these experiments showcase the potential of active perception used within domain randomization.

A. Moving Reference Objects

We consider estimating the pose of a peg-shaped object x resting on a table, similar to the experimental setting of [7] (see Figure 3 for more details). This is an important task for estimating where to grasp an object for further manipulation. Since the table has a fixed height, and the object is at rest, the relevant degrees of freedom are 3-dimensional – the 2D position of the object center and its orientation.[2] We predict the pose of x with respect to the green cylinder object y. For active perception, we consider moving the reference object y between a fixed set of 4 points in the corners of the table, and using ITDR to predict the pose from all images. We expect that actively moving objects in the scene will give us much more accurate pose estimation than if we had just a single image since the diversity of data to predict from is larger.

In Table I we show the results of pose estimation from a single image, and compare to using multiple images using ITDR. The improvement is on the order of 3x bringing down the estimation error to sub-centimeter ranges which enables a number of high precision applications.

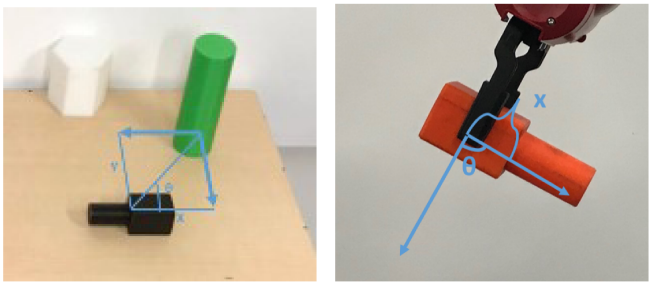
We also investigate how to choose the best set of transformations for the reference object such that the pose estimation error is minimized. It is preferable for us to choose as few points as possible while ensuring accurate pose estimation. We choose pairs amongst the four points shown in Figure 5 to evaluate whether particular transformations are more effective than others. In Table I we report the error of each pair of possible reference object positions on real data. We see that the real world error when moving object y to the diagonal corners is significantly lower than if we moved the object to corners which share an edge. This is likely caused because it gives a more significant difference in the relative poses.

This experiment helps us understand the effect of moving objects in the scene actively in enabling better pose estimation. We see that simply moving reference objects in the scene and using the known geometric transformations allows us to estimate pose more accurately for real-world peg insertion tasks.

[2]Note that Tobin et al. [7] only predicted the 2D position, while here we also predict orientation.

Average error	x [cm]	y [cm]	θ [radians]
Using only one image	1.57	1.10	0.065
Reference Object on Diagonal	0.64	0.456	0.025
Reference Object in Parallel	1.17	0.47	0.038
Reference Object on Four Corners	0.62	0.46	0.037

TABLE I: Table showing the mean prediction error for pose prediction for the moving reference object scenario described in Section VI-A. This table shows that performance of pose prediction transferred form simulation to reality can be significantly improved by using the robot actively to modify the scene being considered. Prediction error is low when the reference object is placed at all four corners or on diagonal corners rather than along an edge of the table. This indicates that multiple images with known transformations do help reduce pose prediction error, and the choice of which transforms to use has a significant effect on the prediction error.



(a) Relative pose measurements with respect to green reference object (Section VI-A)

(b) Relative pose measurements when the object is grasped in the gripper (Section VI-B)

Fig. 3: Experiment setups: (a) estimate pose relative to movable reference object. (b) estimate pose relative to robot gripper.

B. Moving Robot Manipulator Holding an Object

From the results above, we see that moving reference objects in the scene to get a wider variety of relative poses significantly helps with pose estimation. Alternatively, we can consider a scenario where an object has already been

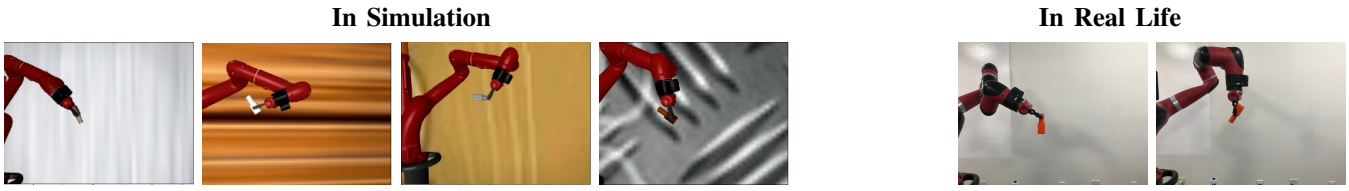


Fig. 4: Simulation to reality transfer with active gripper motion. Left: simulated images with domain randomization. Right: real images. Active perception here is based on moving the robot gripper.

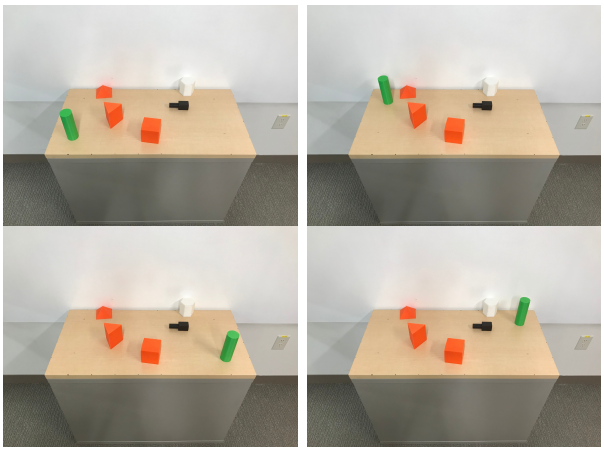


Fig. 5: Different transform applied to the reference object as described in Section VI-A. The green cylinder is the reference object and we are estimating the pose of the black peg. As seen from these figures we can transform the position of the green cylinder to four different positions and use multiple images to improve pose estimation

grasped, but its position within the gripper is not known accurately. This would be a typical case when the pose estimation before grasping is not perfect. It is also important in robotic reinforcement learning experiments [27], where during learning, interaction with other objects in the environment can move an object that is grasped within the gripper.

Fig. 6: Different transform applied to the gripper with an object gripper in it. We want to estimate the exact relative position of the peg with respect to the gripper. As seen from these figures we can move the gripper to many different positions, with known transformations. We can use the additional viewpoints to improve the accuracy of pose estimation

As in the previous section, the object x is peg-shaped, while the reference object y in this case is the robot gripper. We estimate a 2-dimensional pose: the distance of the center of the object from the gripper, and its orientation within the gripper. The measured quantities are depicted in Fig 3b. These are challenging to estimate with extreme precision but are extremely important for the tasks we consider.

For active perception, in this case we move the robot gripper between a set of 5 fixed positions, and use ITDR to estimate the pose from all images. The different movements of the gripper show the camera different elements of the object itself, which is likely to help with better pose estimation since the model can latch on to different parts of the object. We find that this strategy indeed helps with pose estimation in the real world. We are able to identify the offset and the angle of the object grasped within the gripper significantly more accurately. ITDR performs significantly better than the baseline of simply using a single image and a model trained with domain randomization. As we can see from Table II, the x position error is improved by around 20% and the angle accuracy is improved by around $3\times$, from 0.129 to 0.047. Additionally, we find that using fewer gripper locations leads to worse performance. This suggests that using the multiple images does indeed improve performance and scales with using more images for estimation.

Average error	x [cm]	θ [radians]
one image	0.30	0.129
five images	0.26	0.047
two images	0.27	0.086

TABLE II: Table showing the mean prediction error for pose prediction for moving gripper scenario described in Section VI-B. We see that using multiple images with known geometric transformations is able to significantly reduce the angle error and provide some improvements in the estimation of the offset of the gripped object as well.

Note that in this setting, the object does not change pose with respect to the gripper, therefore the inverse transformations in ITDR are just the identity.

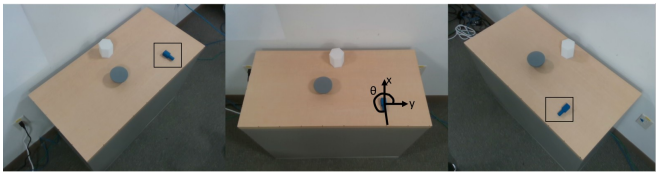


Fig. 7: Different camera angles of the same setting with target object boxed up. We want to estimate the position and orientation of the target object relative table corner. We can use multiple observing angles to explore the geometric properties of the target object and achieve a better pose estimation.

C. Moving a Robot-Held Camera

In this experiment, we demonstrate that actively moving the camera can improve the pose prediction performance. Figure 7 depicts our experimental setup.

Average error	x [cm]	y [cm]	θ [radians]
Using only one image	1.97	0.12	0.11
Using three images	1.50	0.08	0.08

TABLE III: Table showing the mean prediction error for pose prediction for the moving camera scenario described in Section VI-C. We see that using multiple images taken from different point of view is able to reduce both the coordinates and angle error.

As in Section VI-A, we estimate the pose of a peg-shaped object x resting on a table among various distractor objects, and a fixed reference object y. To obtain more contrasting result, the peg is 0.7^3 times smaller than the one use in VI-A. Here, we mounted our camera to the robot's end effector, and we actively change the viewpoint by moving the robot arm to various positions. In particular, we chose a set of three fixed positions in which we can view the object from. Note that similarly to Section VI-B, the object does not change pose with respect to the reference, therefore the inverse transformations in ITDR are just the identity.

In Table III, we show the results for our method. We observe a significant improvement in pose prediction for the x, y coordinates the object. To better understand these results, in Figure 8 we plot the prediction error as a function of the orientation of the object. We observe that for some orientations, estimating the pose from a single viewpoint is very difficult, which is attributed to the asymmetric shape of the peg – when placed such that it is perpendicular to the camera plane, most of the object is occluded, making it difficult to estimate a correct orientation. In test cases, adding additional viewpoints significantly improves the results.

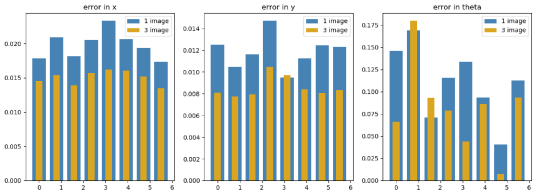


Fig. 8: Distribution of the error in horizontal direction, vertical direction, and θ with respect to the orientation of the object(θ). Using one image, we see a high prediction error when the peg is orientated away from the camera (θ around 3.14) due to the asymmetric shape of the peg. When using three images, the prediction for this previous difficult object orientation is significantly improved.

VII. Discussion and Future Work

In this work, we explored the use of active perception within the domain randomization paradigm. We have shown that active perception strategies which are able to interact with objects in the scene with known geometric transformations can significantly improve the performance of pose estimation compared to passive perception approaches. In particular, we have reduced the 1.5cm pose estimation error in domain randomization state-of-the-art to less than 0.6cm, which can lead to new robotic capabilities in downstream tasks such as tight fitting assembly problems.

In future work, we intend to explore additional methods for improving the performance of domain randomization based pose estimation, for example, in a semi-supervised setting where unlabeled images from the real domain are available to the learning algorithm.

Acknowledgements

This work was supported in part by Siemens Corporation.

References

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[2] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3d pose estimation from a single depth image," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 731–738.

[3] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3d sensor." in *ICRA*. Citeseer, 2012, pp. 1724–1731.

[4] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3400–3407.

[5] 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.

[6] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 292–301.

[7] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 23–30.

[8] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," *arXiv preprint arXiv:1611.04201*, 2016.

[9] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," *CoRR*, vol. abs/1709.07857, 2017. [Online]. Available: http://arxiv.org/abs/1709.07857

[10] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International journal of computer vision*, vol. 15, no. 1-2, pp. 123–141, 1995.

[11] I. Skrypnyk and D. G. Lowe, "Scene modelling, recognition and tracking with invariant image features," in *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2004, pp. 110–119.

[12] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1284–1306, 2011.

[13] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 615–628, July 2006.

[14] C. Harris, *Tracking with Rigid Objects*. MIT Press, 1992.

[15] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 932–946, 2002.

[16] F. Jurie and M. Dhome, "Hyperplane approximation for template matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 996–1000, 2002.

[17] V. Lepetit, P. Fua *et al.*, "Monocular model-based 3d tracking of rigid objects: A survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005.

[18] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*. IEEE, 1999, pp. 85–94.

[19] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello, "Rune-tag: A high accuracy fiducial marker with strong occlusion resilience," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 113–120.

[20] R. Bajcsy, "Active perception," 1988.

[21] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.
[22] I. Mordatch, K. Lowrey, and E. Todorov, "Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 5307–5314.
[23] J. Tobin, W. Zaremba, and P. Abbeel, "Domain randomization and generative models for robotic grasping," *arXiv preprint arXiv:1710.06425*, 2017.
[24] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.
[25] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.
[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[27] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015.