DOI: 10.3217/978-3-85125-524-9-16

Pose Estimation of Similar Shape Objects using Convolutional Neural Network trained by Synthetic data

Kiru Park, Johann Prankl, Michael Zillich and Markus Vincze

***Abstract*— The objective of this paper is accurate 6D pose estimation from 2.5D point clouds for object classes with a high shape variation, such as vegetables and fruit. General pose estimation methods usually focus on calculating rigid transformations between known models and the target scene, and do not explicitly consider shape variations. We employ deep convolutional neural networks (CNN), which show robust and state of the art performance for the 2D image domain. In contrast, normally the performance of pose estimation from point clouds is weak, because it is hard to prepare large enough annotated training data. To overcome this issue, we propose an autonomous generation process of synthetic 2.5D point clouds covering different shape variations of the objects. The synthetic data is used to train the deep CNN model in order to estimate the object poses. We propose a novel loss function to guide the estimator to have larger feature distances for different poses, and to directly estimate the correct object pose. We performed an evaluation using real objects, where the training was conducted with artificial CAD models downloaded from a public web resource. The results indicate that our approach is suitable for real world robotic applications.**

I. INTRODUCTION

Pose estimation of objects in color and depth images is essential for bin-picking tasks to determine grasping points for robotic grippers. Man-made objects are usually manufactured using 3D CAD models having exactly the same shapes with negligible errors. The well-constrained environment enables the robot to identify each pose by comparing features of the pre-created template and an input image [14]. However, it is not possible to provide 3D CAD models for natural objects, such as vegetables or fish, where each object has a slightly different shape. Object pose estimation with template based approaches would need a huge number of templates in order to cover each individual pose and the different shape variants. Hence, these approaches would lead to large databases and a high processing time for matching of the templates.

Recently, CNN based approaches provide reasonable results for most computer vision tasks including image classification and object detection in 2D images [13] [15]. This achievement is accomplished with a large number of training examples, e.g., [4] [7]. The 2D image datasets are usually collected from web resource and annotated by non-expert persons with tools using a user-friendly interface. For RGB-D images or 2.5D point clouds it is difficult to collect a large number of examples from public web services and it is also hard to annotate the exact poses by non-expert persons. This results in a lack of training data and causes an additional complexity to train a CNN for estimating 6D poses in the 3D space. Therefore, pre-trained CNNs are used for extracting features from color or depth images, and the extracted features are used to train linear regressors to estimate the poses [16]. Although there are several datasets which have 6D pose information for more than 15K images [9], [10], it is still not enough to train a deep CNN and none of them consider object classes with large shape variations.

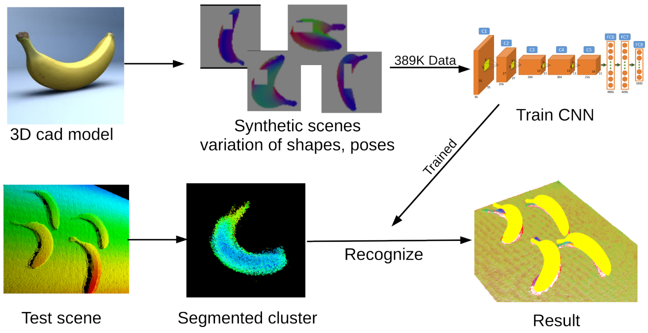


Fig. 1: Overview of the proposed framework. An artificial 3D CAD model is used to generate synthetic scenes with varied shapes and poses in order to train the deep CNN. The trained network can compute poses of each of segmented clusters.

In this paper, we propose a simple pose estimator that can be used to estimate poses of objects with shape variations, such as vegetables or fruit, using a CNN and a single depth image as input. Synthetic depth images containing various poses and shapes of a CAD model are generated to train the proposed CNN. No more template information is required after training. This simplicity is one of the advantages of the proposed model for the robust estimation of object poses with different shape variants. The experiments show that our concept is suitable for real world robotic applications.

As a summary, our paper provides the following contributions:

- We propose a framework that is able to generate synthetic training images and consists of a deep CNN pose estimator for the estimation of poses of natural object classes such as vegetables and fruit.
- Pairwise training is applied to train the deep CNN with a loss function that minimizes the errors between the estimated poses and exact ground truth poses and low-level feature distances between similar poses.
- We show that our estimator successfully estimates poses of real fruit using more than two hundred test images,

All authors are with the Vision4Robotics group, Automation and Control Institute, Vienna University of Technology, Austria {park, prankl, zillich, vincze}@acin.tuwien.ac.at

which are collected with a stereo camera widely used in industrial applications.

The remainder of the paper is organized as follows. In Section II we provide an overview of related work. Our proposed approach for Deep CNN based pose estimation is introduced in Section III. In Section IV, we present experiments with our trained pose estimator with test images containing real bananas. We conclude the paper with final remarks and plans for further work in Section V.

II. RELATED WORKS

Object detection and its pose estimation is an essential task for robots and industrial applications, especially for picking and placing tasks. The exact 6D pose information of an object is required to decide about grasp points for picking and to define proper locations for placing. Therefore, pose estimation in 3D space has received a lot of attention with various approaches which dominantly include feature matching based methods and recently convolutional neural network based methods. State of the art methods are able to perform classification of objects and pose estimation at the same time [1], [18]. In the brief review below we focus on feature based approaches with a local or global descriptor and CNN based approaches.

A. Feature based approaches

Extracting features from training and test data, matching correspondence and calculating single transformation from a trained model to target scenes are typical processes of feature based approaches. Features for the 3D domain are designed to provide a generalized representation of the object shape using local attributes. One popular example is SHOT developed by Tombari et al. [17]. In [1] Aldoma et al. developed an approach which uses various features to generate possible hypotheses and select hypotheses which minimizes a cost function in order to remove false-positives. These feature based pose estimation approaches generally compute rigid transformations, which implicitly assumes that training models and target objects have the same shape. Wohlkinger et al. [19] uses CAD model to train global features to recognize real objects. This method shows robustness to shape variations, but it needs a large number of template images.

B. CNN based approaches

To employ recent convolutional neural networks, successfully used in the 2D image domain, to the 3D domain, which does not have enough training data, researchers tried to use pre-trained CNNs as a feature descriptor and trained additional classifiers for recognition and linear regression for pose estimation [16]. But [16] constrains object poses to in-plane rotation on the table, with one single degree of freedom. Generation of synthetic data is an option for training a CNN with depth images as input. [3] uses a 3D CAD models in order to train the typical CNN structure and finally gains a descriptor for a single channel depth images. This model was used for object classification tasks. Also, [3] considers object classification tasks, but this approach generates depth images from CAD models containing both, varied view points and randomly morphed shapes. CNN based 6D pose estimation is also described in [18], [5]. Both use pair-wise training to guide intermediate features to have larger distances for larger pose deviations. They design a small CNN network, which has only two convolution layers in order to train the CNN using a small number of training examples. In contrast to these approaches, we use a deep CNN which has five convolutional layers and pre-trained weights computed by a large number of 2D images. However, we refer to their pairwise training approaches to get a robust pose estimation performance.

III. METHOD

In the following paragraph we provide a detailed description of the proposed pose estimation approach, which consists of a deep CNN, generation of synthetic images and a pose refinement step for the final result, shown in Fig.1.

To be able to exploit the structure and pre-trained weights of well-established and tested CNNs taking three channels of a 2D color image as input, we transform single-channel depth images to three-channel color images. Finally, the pose estimation procedure at test time is described, including the refinement step to minimize the translational error.

A. Deep CNN for pose estimation with depth images

We employ Alexnet, which has proven results for 2D image classification tasks. The only different part is the last fully connected layer, which in our case has only four output channels for estimating the rotational transformation in quaternions, instead of a thousand channels for classification. Also, the final output is filtered by *tanh* function to provide normalized results between -1 and 1. The reason why we use a quaternion representation instead of Euler angles with three parameter is, the non-linearity and periodicity of Euler angles. For example, the numerical difference between 0 and 359 degrees is large, although the difference of the angles is small. However, the quaternion representation allows to calculate the pose difference as distance of each component of the quaternion values [12]. Most of the state of the art CNN models including Alexnet uses a 2D color image as input. State of the art for CNNs applied to depth images is to convert the depth image in the one channel to a color coded image in the three channels [6]. Among the possible color coding methods, directly matching each axis component of a surface normal to separate image channels has shown a superior performance [6]. Optionally, we use the depth value to scale the values of each pixel as described in (1) and (2).

$$I_D = 1.0 - \frac{P_z - min_z + \delta}{max_z - min_z + 2\delta} \tag{1}$$

$$P_{data} = I_D[N_x\, N_y\, N_z] \tag{2}$$

where P_{data} describes a single data point represented in the three channels. I_D is the scaled depth value and the remaining three values N_x, N_y and N_z are the individual axis of the surface normal. The depth value I_D is normalized using the maximum and the minimum value of the point cloud with a margin δ to avoid zero values. Furthermore, the normalized depth value is subtracted from one to get higher values for closer points. Finally, every point is projected to a pixel in the 2D image.

B. Generation of synthetic training data

To train a CNN a large number of training examples is required, which cover each possible viewpoint of the object. We developed a fully autonomous data generation framework, which is able to cover all possible poses and shape variations. A 3D CAD model, e.g. from a public web resource or a reconstructed 3D scanned model, can be used as a reference model for this framework. The first step is to convert the CAD model to a point cloud format and to transform the reference coordinate system to the centroid of the model. After that, rotations for each axis are defined with 5 degree increments, which results in about 373K possible poses. In addition to the pose transformation, the shape transformation, i.e., scaling and shear is also defined for each pose. Scale and shear factors for each axis is randomly selected between a specified range in order to cover possible variations of the object. The reference model is transformed with the defined transformation matrix. Then it is placed to a location with a proper distance – usually found in the pose estimation scenario – to the camera. Self-occluded points are removed using a standard ray tracing of a camera view. Additionally, a randomly placed 2D rectangle is used to remove small parts of the object, in order to simulate partial occlusions and segmentation errors. Finally, the remaining points are used to render a depth image and non-object points or background points are filled with mean values (e.g. $P_{data} = [0.5\ 0.5\ 0.5]$ in case the normalized values are within $[0..1]$). The finally generated image is stored including the pose transformation using quaternions, i.e. in the same format the deep CNN provides.

C. Pairwise training for robust pose estimation

As proposed in [18], [5] our network is trained with input pairs to minimize feature distances of similar poses and maximize feature distances of different poses. The pose difference of a training pair is defined as the Euclidean distance between each quaternion component. Hence, a pair of training examples with a pose distance less than ρ_s is regarded as as positive pair and if the distance is larger than ρ_d it is regarded as a negative example (cf. 3).

$$\omega = \begin{cases} 1, & \text{if } ||q_{anchor} - q_{pair}||_2 < \rho_s\,, \\ 0, & \text{if } ||q_{anchor} - q_{pair}||_2 > \rho_d\,. \end{cases} \tag{3}$$

ω is given to the loss function to determine whether the current pair of images is positive or not, as described in (6). q_{anchor}, q_{pair} denote four-dimensional vectors of each pose transformation serialized from quaternion representation.

The whole input batch for each iteration is filled with positive and negative pairs. As described in Fig. 2, a data pair is fed into the CNNs with the same weights and computed separately. To calculate the loss in each iteration, we use the output of the seventh fully connected layer with 4096 dimensions and the last fully connected layer with 4 dimensions, which is furthermore used to predict the rotation information in quaternion.

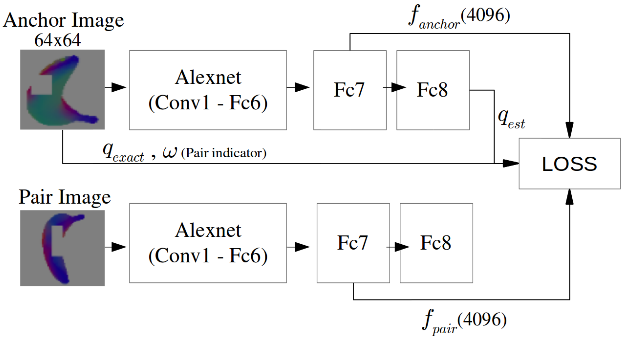


Fig. 2: Streamlines for pairwise training using shared weights for the CNNs. Output from both streamlines, i.e. the 7th layers and the last layers are used to compute the loss for the annotated training pairs.

The loss function L for training can be separated into two part as described in (4).

$$L = l_r + l_f \tag{4}$$

For N batch images per each iteration, l_r represents a regression error between the annotated pose and the estimated pose which is defined as Euclidean distance (cf. 5), while l_f of 6 represents contrastive loss to guide features to have a smaller distance for similar poses and a larger distance for different poses.

$$l_r = \frac{1}{2N} \sum_{n=1}^{N} ||q_{est} - q_{exact}||_2^2 \tag{5}$$

$$l_f = \frac{1}{2N} \sum_{n=1}^{N} (\omega) d^2 + 2(1-\omega) max(1-d, 0)^2 \tag{6}$$

$d = ||f_{achor} - f_{pair}||_2$ denotes the Euclidean distance between features computed from the seventh fully connected layer. ω, the parameter to classify training pairs as positive or negative examples, with similar or different poses is set in the data generation process. This contrastive loss has generally been used to train Siamese networks, which compare pairs of images [8]. In each iteration weights of the CNNs are updated to minimize the loss function using a stochastic gradient descent (SGD) solver. For this l_r is used to update all weights of the CNN, while l_f effects all weights except those of the last fully connected layer.

D. Estimation procedure

In contrast to the training, for pose estimation only a single stream line with one deep CNN is used. The last fully connected layer directly predicts the pose represented in quaternion. Given a depth image or a point cloud we classify

TABLE I: Pose estimation results with the proposed CNN

	Proposed CNN with ICP	Proposed CNN without ICP	ICP from Random Pose
Precision	**0.956**	0.822	0.265
Time (ms)	140±32	129±32	155±33

segmented objects. For the sake of simplicity in this paper we use a simple dominant plane segmentation and a nearest neighbour clustering of 3D points. The pre-processing to provide the input to the CNN is identical as for training (cf. III-B). The trained CNN directly estimates the rotation for the input segment. The corresponding tentative translation is computed from the centroid of the reference model and the segmented point cloud. Finally, a pose refinement step is performed. Basically, the translational error is dominantly caused by the difference between centroids of the reference model and the test image. This is because the centroid of the reference model is derived by the whole object shape, while the test image lack of occluded parts. To minimize this error, self-occluded parts of the reference model are removed after initial alignment, and the centroid of the reference model is recalculated. As a final step, we apply an iterative closest point (ICP) algorithm.

IV. EXPERIMENTS

We perform experiments to prove our concept with real bananas. An artificial 3D CAD model of a banana is selected and converted into a point cloud, further used to generate training images and store the ground truth pose. Scaling and shear transformations are randomly varied from 0.8 to 1.2 for each of three directions of views generated every 5 degree along each axis. The margin δ to calculate the depth to color conversion is set to 0.5. The CNN is implemented with the Caffe framework [11]. We set the initial weights using the pre-trained network, trained with Imagenet data [4]. To decide about positive and negative examples for pairs training examples, we set the threshold $\rho_s = 0.2$ for positive and to $\rho_d = 1.0$ for negative examples. Positive and negative pairs are randomly selected during the first epoch of cycles. The set of pairs is then fixed for further iterations to reduce training time. Every input image is re-sized to 64x64 pixel, while keeping the ratio between heights and widths of the rendered view. Test images are captured with an Ensenso N35, an industrial stereo sensor that provides only depth information with a resolution of 640x512. We assume robust segmentation results for the test scenes. Therefore, we placed the bananas on the table with enough distance to each other, in order to robustly extract segments, after detecting the dominant plane. We prepare five test scenes consisting of multiple bananas and approximately 278 scenes containing single banana per image using four different bananas. Estimated poses are evaluated manually. The criterion for the evaluation is based on the graspablity of the detected object, i.e. if the estimated pose is accurate enough to successfully grasp the object it is counted as positive. All experiments are performed with an Intel i7-6700K and a NVIDIA GTX1080 train the CNN.

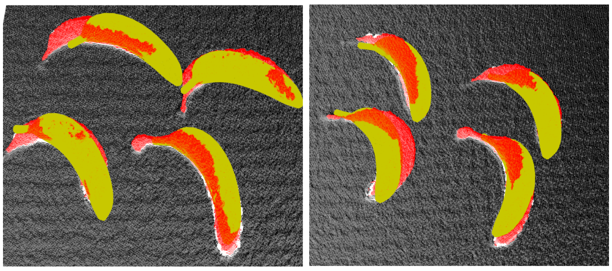
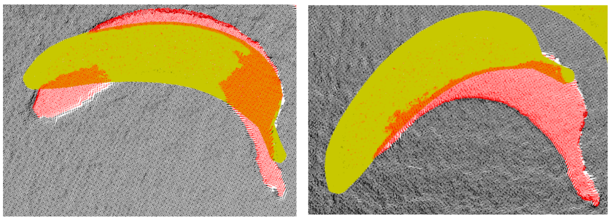


Fig. 3: Visualization of the estimated poses of multiple bananas. Red: real bananas in the test scene, yellow: estimation results



(a) Initial alignment before ICP (b) final alignment after 10 iteration

Fig. 4: Example of a bad alignment after ICP. This example is converged to match with an edge part of the banana

A. Results for bananas

Fig. 3 briefly shows the results for the test scenes containing multiple bananas. As shown in Table 1, the overall accuracy after pose refinement is about 95.6% and the computational time for each segment is about 0.14 second for each object, which is highly acceptable for robot grasping tasks.

B. Side effect of refinement steps using ICP

ICP generally improves the results. However, it sometimes causes worse alignment as shown in Fig. 4. This is because of the shape difference between the reference model and target scenes. The general ICP, which we use assumes a rigid transformation between the reference model and target model. Hence, depending on the inlier threshold ICP converges to partially fit to the scene, while the remaining point cloud does not contribute.

V. CONCLUSIONS

In this paper, we proved the concept of estimating poses of objects with a high shape variance using a deep CNN estimator. Furthermore, the proposed framework is able to use any kind of artificial or real scanned 3D model in order to generate enough data for training the deep CNN. This on going research will further be improved with the following ideas:

- The general rigid transformation ICP is not enough to refine the pose because the shape difference between the reference model and the individual objects. We refer to

non-rigid ICP [2] as an option to further improve the pose estimation.
- The preparation of an extensive annotated dataset will lead to an objective evaluation of our approach with various parameters and settings and a comparison to state of the art methods.
- Here, we assumed a correct segmentation result. In future we need to investigate optimal segmentation methods for real world experiments.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement No. 600623, STRANDS and industrial funding from OMRON Corporation in Japan

REFERENCES

[1] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypothesis verification framework for 3d object recognition in clutter," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1383–1396, 2016.

[2] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid icp algorithms for surface registration," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[3] F. M. Carlucci, P. Russo, and B. Caputo, "A deep representation for depth images from synthetic data," *arXiv preprint arXiv:1609.09713*, 2016.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[5] A. Doumanoglou, V. Balntas, R. Kouskouridas, and T.-K. Kim, "Siamese regression networks with efficient mid-level feature extraction for 3d object pose estimation," *arXiv preprint arXiv:1607.02257*, 2016.

[6] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 681–687.

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[8] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. IEEE, 2006, pp. 1735–1742.

[9] 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.

[10] T. Hodan, P. Haluza, S. Obdrzalek, J. Matas, M. Lourakis, and X. Zabulis, "T-less: An rgb-d dataset for 6d pose estimation of texture-less objects," *arXiv preprint arXiv:1701.05498*, 2017.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[12] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] E. Muñoz, Y. Konishi, V. Murino, and A. Del Bue, "Fast 6d pose estimation for texture-less objects from a single rgb image," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5623–5630.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[16] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1329–1335.

[17] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European Conference on Computer Vision*. Springer, 2010, pp. 356–369.

[18] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3d pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3109–3118.

[19] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *2011 IEEE International Conference on Robotics and Biomimetics*, Dec 2011, pp. 2987–2992.