Mutual Hypothesis Verification for 6D Pose Estimation of Natural Objects

Kiru Park, Johann Prankl and Markus Vincze
Vision4Robotics Lab, Faculty of Electrical Engineering, TU Wien
1040 Vienna, Austria
{park,prankl,vincze}@acin.tuwien.ac.at

Abstract

Estimating the 6D pose of natural objects, such as vegetables and fruit, is a challenging problem due to the high variability of their shape. The shape variation limits the accuracy of previous pose estimation approaches because they assume that the training model and the object in the target scene have the exact same shape. To overcome this issue, we propose a novel framework that consists of a local and a global hypothesis generation pipeline with a mutual verification step. The new local descriptor is proposed to find critical parts of the natural object while the global estimator calculates object pose directly. To determine the best pose estimation result, a novel hypothesis verification step, Mutual Hypothesis Verification, is proposed. It interactively uses information from the local and the global pipelines. New hypotheses are generated by setting the initial pose using the global estimation and guiding an iterative closest point refinement using local shape correspondences. The confidence of a pose candidate is calculated by comparing with estimation results from both pipelines. We evaluate our framework with real fruit randomly piled in a box. The potential for estimating the pose of any natural object is proved by the experimental results that outperform global feature based approaches.

1. Introduction

Robotic bin picking is an essential task in the industrial environment. Industrial parts usually have an identical shape when they are manufactured using a CAD model. Robotic arms pick these parts and place them at a particular position with a particular pose [18, 9]. The robotic manipulation becomes easier if the robot knows the exact pose of the target object. Aside from this typical scenario, target objects are often natural and are randomly piled in a box, e.g., fruit and vegetables. Natural objects have individual shape variations even though they belong to the same class. The shape variation of an object makes it difficult to train a pose estimator and evaluate the estimated pose. A sufficient number of training examples need to be collected to cover most of the shape variations and 6D poses of a class. Nevertheless, it is hard to collect training data in the real environment because of the limited number of samples and the difficulty of annotating exact 6D pose information. Thus, we need to train a pose estimator using a small number of examples or synthetically generated data.

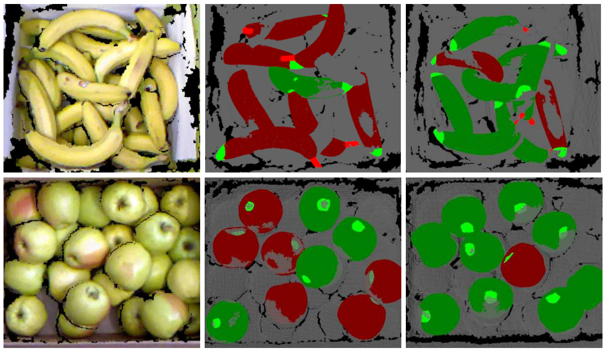

Figure 1. Pose estimation results of bananas and apples in a box. First column: reference images. Second column: results using the VFH. Third column: results using our framework. Green objects are accurate estimations. Bright green and red parts are critical parts which are used for evaluation.

The variation of shape creates an additional challenge for verifying the correctness of the estimated pose. A common technique is to compute the average distance between the corresponding points of the estimated object and the test scene [1]. However, this assumes that a perfect rigid transformation exists, which is not the case for natural objects. Therefore, it is necessary to compare positions of specific local parts of an object, e.g., whenever the stalk of an apple is detected correctly, the pose of the apple must be decided correctly. Hence, the position of detected local parts should be involved when candidates of possible poses are verified.

In this paper, we propose a novel framework for estimating the pose of natural objects regardless of highly varied shapes. We use a novel local descriptor to detect particular local shapes and a global estimator to analyze global shapes.

The robust estimation is achieved by our Mutual Hypothesis Verification (MHV) that uses mixed information derived from local and global shapes. As a summary, our paper provides the following contributions:

- A novel hypothesis verification step, Mutual Hypothesis Verification, is proposed to obtain the best estimation using combined information of the local and the global estimation pipelines, which is robust to individual shape variations of natural objects. Unlike previous work, additional hypotheses are generated by this mixed information. A novel cost function determines the best estimation by comparing pose candidates with outputs of both pipelines.

- We propose a novel local shape descriptor to find user-defined critical parts of an object. Our novel training method enables the descriptor to be background invariant. Furthermore, we employ a CAD model to generate a synthetic template database that contains pose information of an object. This database enables a pose of an object to be directly estimated without grouping of correspondence points.

The remainder of the paper is organized as follows. In Section 2, we provide an overview of work related to 6D pose estimation of objects. The proposed framework and its details are explained in Section 3. In Section 4, we evaluate and analyze our framework using real objects and an industrial stereo sensor. Final remarks and plans for further work is discussed in Section 5.

2. Related work

In this section, we introduce an overview of the previous studies of pose estimation using 2.5D point cloud data. Tombari et al. [14] introduced a local feature descriptor which has been widely used to find correspondence between two point cloud samples. Instead of matching local shapes, the whole shape of an object has been used to estimates the pose. Rusu et al. [12] proposed a global feature that encodes geometry and viewpoint of an object from 3D point cloud data. Wohlkinger et al. [16] proposed a unique feature that uses CAD model as a training set and estimates object's pose by comparing features from a set of templates of poses with a feature from a segmented object in a target scene.

Recently, Convolutional Neural Network (CNN) based approaches achieve outstanding results in RGB and RGB-D image recognition as well as 6D pose estimation. Kehl et al. [7] employed a convolutional auto-encoder that was trained on a large collection of random local patches to have a powerful local descriptor. The training data for the auto-encoder does not require any annotation. Thus, it is easier to extract training samples from real images. However, a lot of correspondences of local features occurs when objects in the same class are in a box, which causes a high number of false positives. CNNs are also used for global pose estimation using segmented clusters. Wohlhart et al. [15] used a whole image of an object as an input to a CNN descriptor to calculate a global feature. The pose was determined by comparing features from training data with a feature from the input scene. Doumanoglou et al. [4] proposed a network to estimate object pose directly in the quaternion representation without comparing features. Recently, Park et al. [11] used Alexnet [8] trained by synthetically generated data to directly estimate the pose of bananas on a table. However, these global approaches need reasonable segmented inputs. In addition, the shape of the object in any particular pose should differ from the other view points.

Although a local or global feature performs well in a specific environment, it is better to use them together to take advantage of multiple sources of information. Aldoma et al. [1] and Fäulhammer et al. [5] employed global and local features together to generate hypotheses and verify them by a cost function, which computes the average distance of closest points from the object in the scene and the estimated model. However, the verification step is not applicable to natural objects because the shape difference between the target object and the training model is also included in the distance error. Hence, a new verification step is required to obtain the best result.

In this paper, we employ both global and local descriptors together to combine the advantages of each. In contrast to previous work [1, 5], pose hypotheses are generated by individual descriptors as well as by combining the information from both descriptors. This combined information is also used for the verification step to determine the reliability of the estimation results. We use CNNs for both descriptors, which requires a large amount of dataset for training. Wu et al. [17] and Carlucci et al. [3] showed that the synthetic images generated by CAD models could train a CNN for object classification. Thus, we use a CAD model to create synthetic examples to overcome the small number of training examples.

3. Method

The overview of the proposed framework is shown in Figure 2. The input to the framework is a segmented cluster from the target scene in 2.5D point cloud format. The input is fed into the local pipeline and the global pipeline separately for matching local features and estimating the global pose. The purpose of the proposed framework is to find the best rigid transformation matrix $T_{best} \in \mathbb{R}^{4\times 4}$ that maps points from the known model $\mathcal{M}$ to the segmented cluster of the target scene $\mathcal{S}$. Both the local and global pipelines require CAD models to generate a template database of local shapes and training examples of global shapes. CAD models are also converted to the point cloud format.

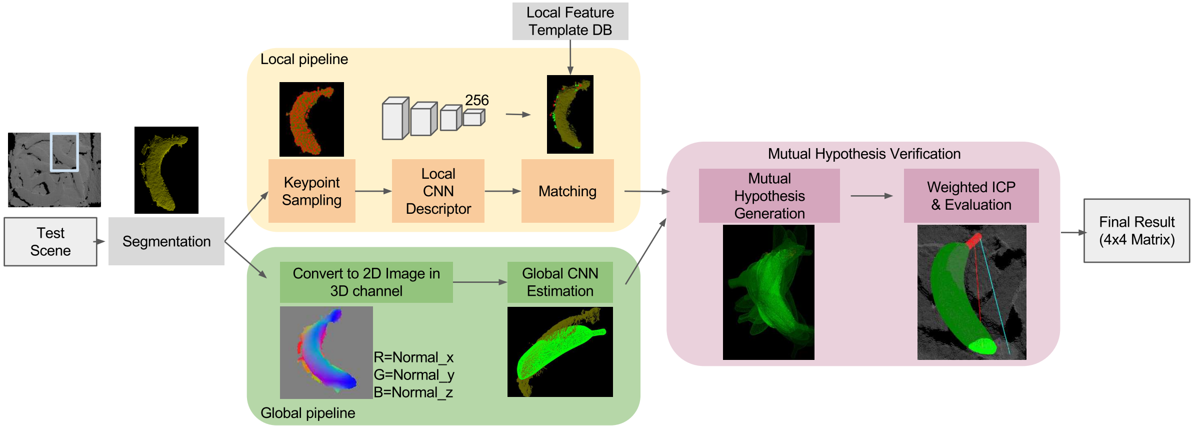

Figure 2. An overview of the Mutual Hypothesis Verification framework.

3.1. CNN based local descriptor

A CNN based local descriptor is trained by uniformly sampling local patches of randomly piled objects in a box. The local descriptor should be invariant to the background of the target object to avoid extracting features from the highly cluttered surrounding. We can ignore occluded objects behind the target objects in this scenario because they are not reachable by the robot arm. Hence, we propose a novel method to train a local descriptor that is robust to various backgrounds using a Convolutional Auto-Encoder framework (CAE). The CAE framework has been used for semantic segmentation of 2D images [2] as well as dimensionality reduction of RGB-D images [7]. Hence, we employ CAE to extract the feature vector of local shapes while reducing the effect of backgrounds. The overview is shown in Figure 3. The segmentation information of objects is given when the data is collected as introduced in Section 4.1. To keep the 3D information of local patches as much as possible, we use a 3D voxel grid as an input. Previous studies showed that the 3D voxel grid filled with the Truncated Signed Distance Function (TSDF) contains enough information for a local descriptor [19, 13]. Thus, the input of the encoder is a 3D voxel grid array of a local patch with TSDF values. The output of the decoder is also the 3D voxel grid filled with TSDF values. When training the network, we assign the input with the local patch extracted from the real dataset without segmentation information and the target output with only points included in the dominant segment. Therefore, we guide the output of the encoder to have closer values regardless of background. Euclidean loss between the target output V_{tgt} and the computed output from the network V_{est} is given by,

$$L_{local} = \frac{1}{2N}\sum_{n=1}^{N}||V_{tgt} - V_{est}||_{2.}^{2} \tag{1}$$

3.2. Generation of Local Templates and Matching

The primary purpose of the local descriptor is to find critical shapes of an object that can be defined by a user. The important part of the target object differs among domains, e.g., the detection of the stalk of an apple is essential if the stalk should face upward. Hence, we use an annotation tool to indicate the significant part of the target object manually using a CAD model of the target class. The annotated parts are used to generate templates of local patches $\mathcal{L}$. The annotated CAD model is placed in the particular distance from the virtual camera with uniformly sampled poses. Local patches of the parts are saved in a database if the parts are still visible after removing the self-occluded parts. Each template $\mathcal{L}^i$ stores the feature vector $\mathcal{L}^i_f$ that is encoded by the local descriptor, the part number $\mathcal{L}^i_p$, the rotation matrix $\mathcal{L}^i_r \in \mathbb{R}^{3\times3}$ and the relative position of the object's center from the center of the local patch $\mathcal{L}^i_t \in \mathbb{R}^{3\times1}$. Even if the user selects only one part as a critical shape, this set of templates enables a locally matched shape to define the pose of an object without additional grouping of other local points.

In the test phase, key-points are uniformly sampled from the input segment and encoded using the local CNN descriptor. The Kd-tree is used to find ten nearest neighbor templates from the database. To determine whether each nearest template is similar or not, we train a small decision network that performs better than simply using an l_2 distance of local features [6]. A CAD model is used to train the decision network. The positive pairs of similar local shapes are generated from the center of the same point with different random noise and random scaling of the object size in the same pose. The negative samples are extracted randomly from points that do not belong to the user defined parts. The decision network has an input as a concatenated feature vector of a template and a local patch. The input is followed by two fully connected layers with 1024 out-

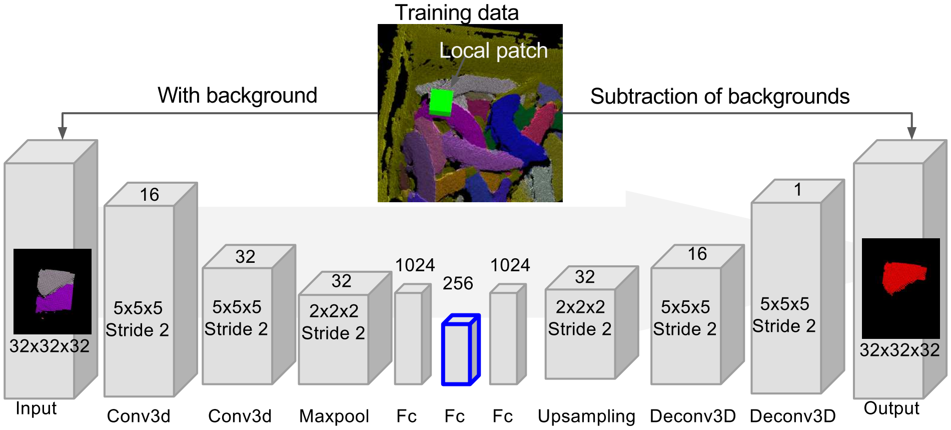

Figure 3. The convolutional auto-encoder architecture of proposed local descriptor. The output of the fully connected layer in the middle, which has 256 dimensions, is used as a feature vector to match with local templates.

puts and two outputs. The last two outputs are regarded as a probability of similarity and dissimilarity of the pair. If the local shape is turned out to be similar with the template $\mathcal{L}^i$, the CAD model $\mathcal{M}$ is initially positioned using the rotation $\mathcal{L}^i_r$ and the relative position of the centroid $\mathcal{L}^i_t$. After removing the self-occluded part from $\mathcal{M}$, we calculate the average distance of the closest points between the aligned model and the target scene $\mathcal{S}$. The new local hypothesis $\mathcal{P}^{new} = \mathcal{L}^i$ is stored in the local hypotheses pool $\mathcal{P}$ if the mean distance is less than a threshold θ_l.

3.3. Pipeline using Global Shapes

The role of the global pipeline is to estimate the pose directly by analyzing the whole shape of the segmented cluster. Park et al. [11] used 2D images in three channels by mapping each component, x,y and z, of surface normals per each pixel instead of using pure depth images in one channel. They used Alexnet with initial weights trained by the Imagenet dataset and achieved acceptable recognition rate with few bananas placed freely on a table. We simplify their method and reduce the complexity of training by removing pair-wise training and the dependency on depth intensity for each pixel. The output of the CNN is a quaternion representation of the rotation transformation that has four values, and the loss function of the network is a simple Euclidean loss between ground truth pose q_{gt} and the estimated pose q_{est} as described by

$$L_{global} = \frac{1}{2N}\sum_{n=1}^{N} ||q_{gt} - q_{est}||^2_{2}. \tag{2}$$

We generate synthetic training images using the same CAD model that is employed in the local pipeline. The shape of the CAD model is morphed by randomly defined scaling and shearing factors. The morphed model is placed at a certain distance from the virtual camera in a pose defined uniformly (e.g., every five degrees for each axis). Occluded parts are removed using z-buffering. Random noise is applied to every training image to simulate sensor noise, occlusion, and segmentation errors. A randomly placed rectangle removes points inside it to simulate partial occlusion and segmentation errors. In addition, we apply additional noise on boundary regions by randomly removing points or adding Gaussian noise on depth measurements. If the shape of the target object is invariant to particular axis, the axis is ignored while training the estimator.

In test time, a depth image is converted to a 2D image in three channels using the values of surface normal, the same way as in training. Then, the rotation of the object is estimated directly. The CAD model is roughly placed at the centroid of the segmented cluster and rotated to the estimated pose, then self-occluded parts are removed. The new centroid of the CAD model is obtained to match with the centroid of the segmented cluster again. Finally, the initial pose $\mathcal{G}_T$ of the global hypothesis $\mathcal{G}$ is defined as

$$\mathcal{G}_T = \begin{bmatrix} \mathcal{G}_r & 2C - \mathcal{G}_c \\ O & 1 \end{bmatrix}, \tag{3}$$

where $\mathcal{G}_r \in \mathbb{R}^{3\times3}$ denotes the rotational matrix converted from the quaternion pose output of the global estimator network, $C \in \mathbb{R}^{3\times1}$ is the centroid of the segmented target object, and $\mathcal{G}_c$ denotes the centroid of the CAD model after removing the self-occlusion points. $O \in \mathbb{R}^{1\times3}$ is a matrix which has zero for all entries.

3.4. Mutual Hypothesis Verification

The purpose of the MHV step is to verify the best result given the set of hypotheses from the local pipeline $\mathcal{P} = \{\mathcal{P}^1, \ldots, \mathcal{P}^i, \ldots, \mathcal{P}^n\}$ and the global pipeline $\mathcal{G}$.

Joining of similar local hypotheses is performed to remove redundant hypotheses generated from the same local parts and to find correspondence points generated from different local parts to make local hypotheses stronger. In the first iteration, local hypotheses that have the same part number are combined. $\mathcal{P}^i$ and $\mathcal{P}^j$ are combined if the distance between centers of local patches is less than ρ_s and the pose difference of local hypotheses is less than θ_s as described by

$$\begin{aligned} ||\mathcal{P}^i_t - \mathcal{P}^j_t||_2 < \rho_s \wedge D_{max}(\mathcal{P}^i_r, \mathcal{P}^j_r) < \theta_s, \\ \text{where}, \mathcal{P}^i_p = \mathcal{P}^j_p. \end{aligned} \tag{4}$$

The function $D_{max}(A, B)$ obtains the maximum angular difference between rotational matrix A and $B \in \mathbb{R}^{3\times3}$. Combined hypotheses generate a new final hypothesis $\mathcal{F}$ and more hypothesis are merged based on their average center $\mathcal{F}_t$ and average rotational pose $\mathcal{F}_r$. When the hypothesis $\mathcal{P}^i$ does not find any similar hypotheses, a new final hypothesis is created. Thus, all of the local hypotheses should belong to at least one final hypothesis.

In the second iteration, every final hypothesis is compared with others to create a new final hypothesis if part numbers are not a subset of the other final hypothesis. In this case, centers of local parts must have a sufficient distance to each other and have a small angular difference as described by

$$\begin{aligned} ||\mathcal{F}^i_t - \mathcal{F}^j_t||_2 > \rho_d \wedge D_{max}(\mathcal{F}^i_r, \mathcal{F}^j_r) < \theta_d, \\ \text{where}, \mathcal{F}^i_p \notin \mathcal{F}^j_p \wedge \mathcal{F}^j_p \notin \mathcal{F}^i_p. \end{aligned} \tag{5}$$

As a result, a new hypothesis $\mathcal{F}^{new}$ based on multiple parts is created with overall rotational pose $\mathcal{F}^{new}_r$ and a set of center points for each part $\mathcal{F}^{new}_t = \{\mathcal{F}^{new}_{t,1}, \cdots, \mathcal{F}^{new}_{t,p}, \cdots, \mathcal{F}^{new}_{t,n}\}$, where $\mathcal{F}^{new}_{t,p}$ denotes the center point of the part number p in the new hypothesis. In addition, $\mathcal{F}^{new}_p$ is a set of local part numbers that were included in the hypothesis.

Joining of the global hypothesis is performed after generating a set of final hypotheses from local hypotheses. Every final hypothesis $\mathcal{F}^i$ from the local hypotheses generates a new hypothesis $\mathcal{F}^{new}$ with the global estimation $\mathcal{G}$. The new hypotheses have the same initial rotational pose and translation of $\mathcal{G}$ while storing a set of centroids of each part $\mathcal{F}^i_t$ in the hypotheses $\mathcal{F}^i$. Each centroid of local parts in $\mathcal{F}^i_t$ is used to guide the ICP step to locate the corresponding part of the CAD model to the scene. As a result of this step, the number of final hypotheses is doubled. Finally, the pure global hypothesis $\mathcal{G}$ is also added to the final set of hypotheses $\mathcal{F}$ without any correspondence from local hypotheses.

Refining pose and calculating cost is performed with the set of final hypotheses $\mathcal{F} = \{\mathcal{F}^1, \cdots, \mathcal{F}^n\}$. Each final hypothesis is refined by the ICP algorithm. If the final hypothesis includes local correspondences of local parts, correspondences are weighted by a factor of α. This means distance errors between these correspondences are stronger by the factor of α than errors between general closest points when the ICP tries to minimize the average distance error. The transformation matrix T resulting from the ICP step is updated to the final hypothesis $\mathcal{F}^i_T = T$. Then, the cost value is computed as follows,

$$\mathcal{C}^i = w_e E^i + w_g D(\mathcal{G}_r, \mathcal{F}^i_T) + w_l D(\mathcal{F}^i_r, \mathcal{F}^i_T) + \frac{w_m}{\mathcal{F}^i_N + 1}. \tag{6}$$

E^i is the scene fitness value calculated by computing the average distance of closest points from the transformed model to the target scene. The function $D(A, B)$ calculates average rotational difference between the rotation matrix $A \in \mathbb{R}^{3\times3}$ and transformation matrix $B \in \mathbb{R}^{4\times4}$. The maximum value of the function D is bounded by 15°. Therefore, the second term applies a penalty if the final pose differs from the globally estimated pose while the third term applies a penalty if the final pose differs from the locally estimated pose. $\mathcal{F}^i_N$ denotes the number of local hypotheses that is included in the hypothesis. Thus, the last term counts how many local correspondences are supporting the hypothesis $\mathcal{F}^i$. w_e, w_g, w_l and w_m weight each term. The weights are set to balance how much the pose depends on the global or local shapes. Hence, the best result is obtained easily by finding the hypothesis that has the minimum cost value.

4. Evaluation

The evaluation is performed with real bananas and apples. Test images are captured with an Ensenso N35, an industrial stereo sensor that provides only depth information with a resolution of 640×512 pixels. The sensor is fixed at $0.9m$ above the ground plane of the target box to measure all objects in the box. The framework is implemented on a computer that has an Intel i7-6700K and a NVIDIA GTX1080, which is used for training both the local and the global CNN descriptors. The local descriptor is trained using all samples from both bananas and apples. Thus, the same local descriptor is used regardless of the target object class. A CAD model is employed for each experiment. CAD models are taken from a public CAD database and scaled to the real size of the object. The CAD model is used to generate the template database of the local pipeline and training images for the global pipeline. The size of input images for the global pose estimator is set to 64×64 pixels. The size of a cube for the local shape patch is fixed as $4cm^3$. For all experiments, parameters are set as $\theta_l = 0.005m, \rho_s = 0.025m, \theta_s = 15°, \rho_d = 0.03m, \theta_d = 30°$ and $\alpha = 10$.

4.1. Collection of Real Dataset

The ground truth segmentation of our box dataset is annotated automatically. We put all object in a box at the be-

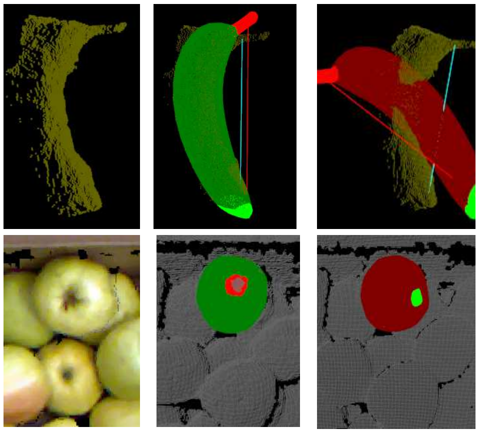

Figure 4. Examples of accurate and false estimation. First column: the reference image. Second column: accurate estimation. Third column: false estimation.

ginning, and we pick an object from the box carefully to avoid moving other objects for every step until picking all objects. We compute the difference between every consecutive scene in reverse order of the recording sequence to identify newly added cluster for every sequence and annotate the segment as a single object. It is clear that the position of the object is the same for further sequences. Hence, the segment of the object is automatically annotated in further scenes unless the object is invisible. Poses of objects are manually annotated by hand. As we mentioned before, poses of natural objects are difficult to annotate clearly because of their different shapes. Instead of defining a transformation matrix of each object, we annotate areas of local shapes of objects that are important to evaluate the estimation result.

4.2. Experiment with Bananas

We define the stalk and the opposite edge of the banana as critical parts. We collect five sets of sequences with 30 bananas in the box for each. Hence, approximately 150 test images are used for evaluation, and 2300+ objects are contained in the dataset regardless of their visibility. The segmentation is not the scope of this paper. Therefore, we use ground truth segmentation as an input of our framework. The target segments are selected if more than 95% points of the object are visible in the current scene. The centroids of the two critical parts define a 3D vector which represents the pose of a banana. The angular difference of vectors that are defined by the estimated pose and the ground pose is regarded as a rotational error. The average distance of closest points between the estimation and the segmented scene is used to check the translational error and additional pose difference. In this experiment, we define that the estimation is accurate if the rotational error is less than $15°$, distances between corresponding critical parts are less than $0.03m$ and the average distance error is less than $0.005m$. We compare our proposed method with global features ESF [16] and VFH [12]. Pose templates for global feature matching are generated by the same method when we generate the local patch templates in Section 3.2. The global estimation is stronger for bananas because of the variability of shapes from different view points. Hence, the weights of the cost functions are $w_e = 1.0, w_g = 0.5, w_l = 0.1$ and $w_m = 0.1$, which weight more on the global estimation and the scene fitness.

Method	S1	S2	S3	S4	S5	AVG
VFH [12]	0.33	0.29	0.24	0.29	0.20	0.26
ESF [16]	0.16	0.13	0.15	0.22	0.17	0.17
Ours Global	0.63	0.71	0.47	0.72	0.74	0.66
VFH+Local	0.68	0.71	0.53	0.60	0.68	0.61
Ours Local	0.66	0.81	0.60	0.63	0.66	0.67
Ours	**0.81**	**0.94**	**0.63**	**0.79**	**0.79**	**0.79**

Table 1. Estimation accuracy of the banana dataset.

Method	S1	S2	S3	S4	S5	AVG
VFH [12]	0.43	0.60	0.44	0.38	0.30	0.43
ESF [16]	0.46	0.65	0.60	0.52	0.25	0.50
Ours Global	0.40	0.53	0.50	0.57	0.28	0.45
VFH+Local	**0.52**	0.81	**0.77**	0.78	**0.49**	0.67
Ours Local	0.51	**0.82**	**0.77**	**0.80**	**0.49**	0.67
Ours	**0.52**	0.81	**0.77**	**0.80**	**0.49**	**0.68**

Table 2. Estimation accuracy of the apple dataset.

The experiment results are summarized in Table 1, total 708 bananas are visible and used for the evaluation. The result shows our proposed framework outperforms other global feature based methods. MHV shows that combining the global and local pipelines improves accuracy beyond their individual performances. Interestingly, the combination of VFH features with our proposed local pipeline is definitely better than using VFH only. This proves that the local pipeline guides local estimations properly as expected.

4.3. Experiment with Apples

We collect five sets of sequences with 25 apples in a box for each. Hence 125 test images and 1600+ objects are contained. Like bananas, the stalk and the opposite concave part of the apple are defined as critical parts. A thin stem at the stalk of the apple is barely captured by the sensor while the surrounding concave shape is well observed. In addition, in contrast to the banana, the stalk concave and the opposite concave cannot be seen at the same time. Also they cannot be easily distinguishable by only using depth information. Hence, we regard those concave shapes as identical parts when we evaluate estimated poses. Therefore, we define that the estimation is accurate if the distances between the closest concave part of the estimated model and the ground truth are less than $0.03m$ and the scene fitness error is less than $0.005m$. Apples have similar global shapes regardless of its pose. Thus, the estimation from the global pipeline is not reliable, as well as the scene fitness is also similar even when the estimation is wrong. Hence, the weights for scene fitness w_e and global poses w_g are set close to zero while setting a higher weight for the number of local matches w_m. Thus, the weights of the cost function are set to $w_e = 0.1, w_g = 0.01, w_l = 0.01$ and $w_m = 2.0$.

Table 2 shows the result of the experiment. Total 1057 apples visible more than 95% are estimated. The result shows that the methods using proposed local descriptor and verification step outperform the other methods regardless of the type of the global estimator. The accuracy is higher than 0.75 for test set 2,3 and 4. However, the test set 1 and 5 show lower accuracy. This is because they contain many apples that do not have any observed concave part. Hence, the local descriptor tries to find the closest concave shape without actually observed points, which causes false estimations. We expect that these false estimations should be removed by detecting convex shapes, which are not defined as a critical part in this experiment.

5. Conclusion

We introduced the concept of estimating poses of natural objects using local and global shapes together. The experimental results show that our proposed framework estimates poses of natural objects robustly regardless of high shape variations. The newly designed local descriptor showed particular superior results for estimating the pose of apples. The local descriptor helps to decide the pose of an object even when their global shapes are similar in different poses. Hence, the framework can be applied to any other vegetable and fruit by setting appropriate weights for the cost function. For further work, we will investigate segmentation methods to extract clusters from the real environment without any prior knowledge. Finally, the output of the framework can be used as an initial pose of a non-rigid registration step [10] to identify which points belong to critical parts or proper regions for grasping.

Acknowledgment

The research leading to these results has received funding from OMRON Corporation.

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*, 38(7):1383–1396, 2016.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[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] 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.

[5] T. Fäulhammer, M. Zillich, J. Prankl, and M. Vincze. A multi-modal rgb-d object recognizer. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 733–738. IEEE, 2016.

[6] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015.

[7] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. *arXiv preprint arXiv:1607.06038*, 2016.

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

[9] 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*, pages 5623–5630. IEEE, 2016.

[10] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.

[11] K. Park, J. Prankl, M. Zillich, and M. Vincze. Pose estimation of similar shape objects using convolutional neural network trained by synthetic data. In *Proceedings of the OAGM-ARW Joint Workshop 2017*, pages 87–91, 2017.

[12] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2155–2162. IEEE, 2010.

[13] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.

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

[15] 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*, pages 3109–3118, 2015.

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

[17] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[18] J. Xu, S. Pu, G. Zeng, and H. Zha. 3d pose estimation for bin-picking task using convex hull. In *Mechatronics and Automation (ICMA), 2012 International Conference on*, pages 1381–1385. IEEE, 2012.

[19] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.