WEAKLY SUPERVISED TRIPLET LEARNING OF CANONICAL PLANE TRANSFORMATION FOR JOINT OBJECT RECOGNITION AND POSE ESTIMATION

Kouki Ueno[1], *Go Irie*[2], *Masashi Nishiyama*[1], *Yoshio Iwai*[1]

[1] Graduate School of Sustainability Science, Tottori University, Japan
[2] NTT Corporation, Japan

ABSTRACT

We propose a method for jointly performing object recognition and pose estimation using training samples of canonical planes to reduce the effort of data label supervision. Collecting a sufficient number of training samples is important to realizing high performance. However, labeling pose parameters is time consuming. We thus train our network model using only object class labels without explicitly labeling pose parameters. To recognize objects and estimate their poses, we design a network with a spatial transformer in a contrastive learning manner such that the canonical plane of an object is always transformed to a certain pose and the features are consistent with those of the object class. Experiments show that our method has improved accuracy in object recognition and lower error in pose estimation compared with simply using triplet learning or a spatial transformer network on a publicly available dataset.

***Index Terms*—** Weak supervision, Canonical plane, Triplet learning, Spatial transformer networks

1. INTRODUCTION

There is a demand for robotic arm systems [1, 2] that are automatically able to pick and stow various objects in a warehouse. Grasping an object with a robotic arm requires two types of computer vision technique, namely object recognition, and pose estimation. In the field of the automatic warehouse, there are already many applications that adopt object recognition techniques [3, 4]. Existing methods [5, 6] that perform object recognition and pose estimation simultaneously have recently attracted attention. The existing methods perform well when many training samples with object class labels and pose parameter labels are collected. However, labeling pose parameters to manage training samples is time consuming and subjective. Weakly supervised learning methods that jointly perform object recognition and pose estimation are thus expected. Recent papers [7, 8] tackled the development of the weakly supervised learning method. Kanezaki et al. [7] assumed that object poses are categorized into several viewpoints and multiple images are available in the test process. Sundermeyer et al. [8] assumed that three-dimensional models of objects are available for training samples. We instead focus on achieving the goal without making such assumptions.

Before we consider a weakly supervised learning method, we discuss major existing methods of object recognition using triplet learning [9] and spatial transformer networks (STNs) [10]. As described in [11, 12, 13, 14], triplet learning has achieved high accuracy of object recognition in various applications. Triplet learning is used to extract invariant deep features when there is a variation in appearance due to pose changes. When training a network model, the triplet loss function decreases the distance between an anchor image (simply referred to as an anchor) and positive image (simply referred to as a positive), while increasing the distance between the anchor and negative. In contrast to triplet learning, the STN infers parameters of spatial transformation to increase the accuracy of object recognition without the use of pose parameter labels for training. After applying a spatial transformation to deep feature maps using the inferred the parameters, the STN increases the tolerance to pose changes in the recognition task.

The present paper designs a novel network model by embedding an STN module in triplet learning to jointly perform object recognition and pose estimation. We do not use pose parameter labels to train our network model. We perform weakly supervised learning for object recognition and pose estimation using only RGB images with object class labels. A straightforward idea would be to first train the network including an STN module with a standard triple learning, extract deep features for object recognition, and infer the pose parameters using the STN for pose estimation. However, such a simple combination does not work well for the following reasons.

- An STN is often implemented to alleviate the variation of pose changes through planar spatial transformations. However, we cannot simply use an STN module because the objects have three-dimensional shapes.
- The anchor of triplet learning dynamically affects the accuracy of pose estimation and object recognition. A criterion for selecting anchors that are beneficial to both has not been investigated.

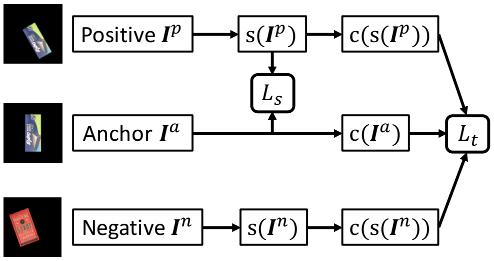

Fig. 1. Our network model using triplet learning of canonical plane transformation. Function s( ) synthesizes an image by removing pose changes using the STN. Function c( ) generates a feature map from an image using a convolutional neural network and vectorizes the feature map.

- The triplet loss function is designed to increase the accuracy of object recognition. It does not explicitly reduce the error in pose estimation.

We exploit the following strategy to overcome these issues.

- We consider how to use an STN module while maintaining a planar spatial transformation for our task. We introduce canonical planes of a three-dimensional object. Generally, an object is approximately represented by a large number of planes. We assume that an object comprises a very small number of canonical planes.
- We believe that a good anchor in triplet learning for both object recognition and pose estimation is an image that well describes the cues of an object. To select anchor images, we acquire representative canonical planes that are less affected by pose changes.
- We add a loss term to smoothly connect from the STN to triplet loss. Pose estimation using an STN is geometrically possible when the anchor and positive appear similar after applying the STN. Our method thus computes the L1 loss between the anchor and positive for a triplet loss function.

In experiments, our method improved the accuracy of object recognition from 90% to 95% compared with an existing triplet learning method while reducing the error of pose estimation from 1.1 to 0.4 compared with an existing STN method on the Rutgers APC RGB-D dataset.

2. TRIPLET LEARNING OF CANONICAL PLANE TRANSFORMATION

2.1. Overview

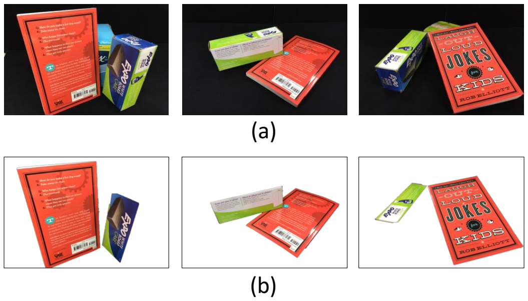

Fig. 2. Examples of observed canonical planes.

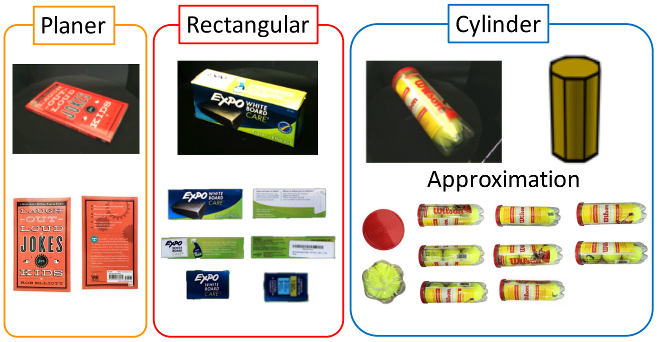

Fig. 3. Examples of three categories of object shape.

Figure 1 illustrates our network model of canonical plane transformation. In the figure, $\boldsymbol{I}^p$ represents the image of the positive for triplet learning, $\boldsymbol{I}^a$ the image of the anchor, and $\boldsymbol{I}^n$ the image of the negative. The triplet of the positive, anchor, and negative corresponds to canonical planes of objects. The positive belongs to the same object as the anchor while the negative belongs to a different object. We set the L1 loss between $\boldsymbol{I}^a$ and s($\boldsymbol{I}^p$) to evaluate the closeness of the anchor and positive when applying the STN. Our method computes the triplet loss using vectorized feature maps of the anchor, positive, and negative. Note that our method shares the same weights of c( ) and s( ) for the anchor, positive, and negative streams. In the test process, we prepare target samples of representative canonical planes. Given a query sample, our method infers relative spatial parameters of pose change from the query sample to a target sample. Our method also extracts deep features for object recognition. Canonical planes, anchors, and the L1 loss are respectively described in Sections 2.2, 2.3, and 2.4.

2.2. Canonical planes

Objects in a warehouse are often packed into a textured box or bag so as to prevent their deformation. We assume that the objects consist of a plurality of canonical planes. When an object is placed on a floor, the pose of the object does not continue to change. The pose settles into a static state, as shown in Figure 2(a). The object pose is thus not free to shift in all directions, and there is a bias to a specific direction. In other words, one surface is observed mainly for each object, such as in Figure 2(b). We term the surface a canonical plane. The number of canonical planes depends on the shape

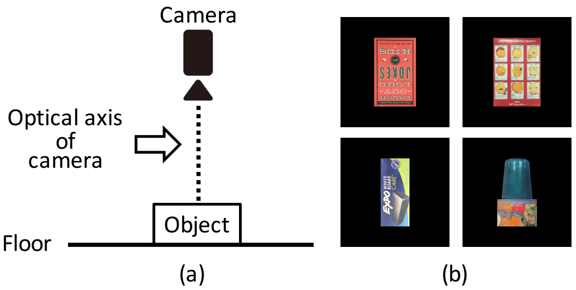

Fig. 4. Setting for acquiring anchor images.

of the object as illustrated in Figure 3. We assume that two canonical planes are observed for a planar object, six canonical planes are observed for a rectangular object, and 10 canonical planes are observed for a cylinder approximated as an octagonal prism. Using canonical planes, our method estimates the object pose through a simple STN module implemented for the planar object.

2.3. Anchor for triplet learning

To improve the accuracy of object recognition and reduce the error of pose estimation, we acquire an anchor image in which a representative canonical plane well expresses a cue of an object. We collect representative canonical planes in the setting as illustrated in Figure 4(a). We set up the camera so that its optical axis is perpendicular to the floor. We place the object parallel to the floor surface. We let the optical axis of the camera pass through the object center of gravity. Figure 4(b) shows examples of anchor images for representative canonical planes.

2.4. L1 loss for pose estimation

To jointly perform object recognition and pose estimation, we use a loss function for our network model expressed as

$$L = L_t + \lambda L_s, \tag{1}$$

where L_t is the triplet loss term for object recognition and L_s is the L1 loss term for pose estimation. To evaluate the distance between the appearances of canonical planes in images, we define L_s as

$$L_s = ||\boldsymbol{I}^a - \mathrm{s}(\boldsymbol{I}^p)||_1. \tag{2}$$

The equation returns a small value when taking similar appearances between the image of the anchor and the image of the positive after applying the STN. After learning, the STN tries to linearly transform the query image of an arbitrary object pose so as to match the anchor image, where the resulting transformation parameters inferred by the STN gives the pose parameters of the object.

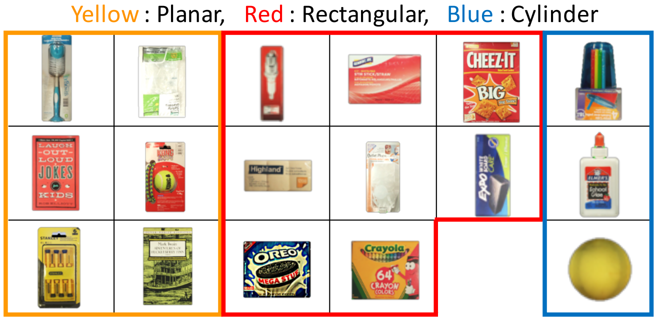

Fig. 5. Objects used in our experiments.

To evaluate the distance between vectorized feature maps in triplet learning, we define L_t as

$$\begin{aligned} L_t = \max(||\mathrm{c}(\boldsymbol{I}^a) - \mathrm{c}(\mathrm{s}(\boldsymbol{I}^p))||_2^2 - \\ ||\mathrm{c}(\boldsymbol{I}^a) - \mathrm{c}(\mathrm{s}(\boldsymbol{I}^n))||_2^2 + m, \ \ 0), \end{aligned} \tag{3}$$

where m is the margin. The equation returns a small value when the anchor and positive feature vectors of the same object are close while the anchor and negative feature vectors of different objects are not close. Object recognition can be performed through nearest-neighbor classification.

3. EXPERIMENTS

3.1. Dataset

To evaluate the accuracy of object recognition and the error of pose estimation, we used three-dimensional mesh models of 17 objects included in the Rutgers APC RGB-D dataset [15]. Figure 5 shows examples of the objects. We generated positive samples of triplet learning by applying an affine transformation to the same object as in the anchor, and negative samples by applying an affine transformation to different objects. We randomly set the parameters of affine transformation in the range of translation of $[-50, 50]$ pixels, range of rotation of $[-60, 60]$ degrees, and range of scale of $[0.8, 1.2]$. We used anchor images as target samples in the test process. We randomly generated query samples in the same manner as the generation of training samples. We used 10,000 training triplet samples, 1000 query samples, and 90 target samples. The size of each sample was 100×100 pixels. We repeated the procedure nine times.

In triplet learning, we computed 128-dimensional feature vectors using two convolutional layers, one max pooling layer, and one global max pooling layer. In the STN, we used a regressor having two convolutional layers, two max pooling layers, and two fully connected layers.

We used the correct-match rate to evaluate the accuracy of object recognition. We used the Frobenius norm between the inferred affine matrix and labeled affine matrix to evaluate the error in pose estimation.

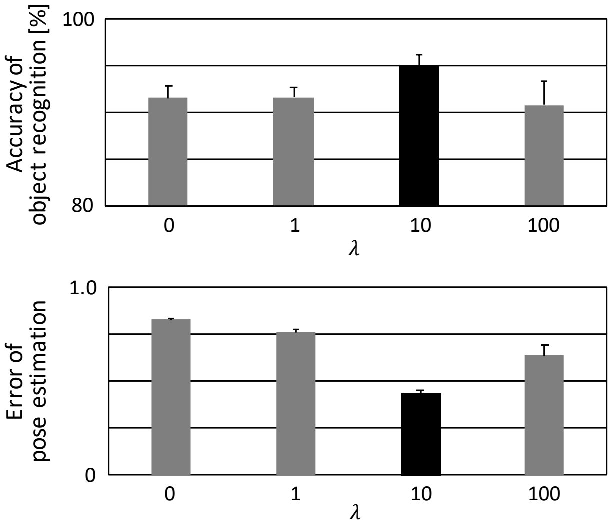

Fig. 6. Accuracy of object recognition and the error in pose estimation for a varying parameter λ.

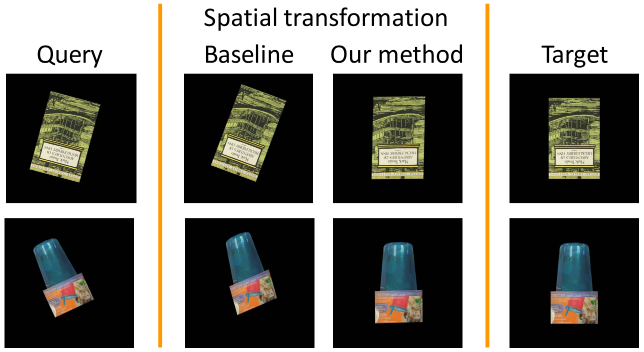

Fig. 7. Examples of query samples after removing pose changes.

3.2. Basic performance

We evaluated the performance of our method while changing the parameter λ in Equation (1). We set λ =0, 1, 10, 100. Note that $\lambda = 0$ refers to the baseline method where the STN is simply set in triplet learning without the use of L_s.

Figure 6 shows the average matching rate of object recognition and the average Frobenius norm of pose estimation. We see that the performances for $\lambda = 1, 10$ were superior to the performance for $\lambda = 0$. We confirmed that L_s for our method works well in terms of increasing the accuracy of object recognition and reducing the error in pose estimation. We obtained high performance when $\lambda = 10$.

Figure 7 shows examples of the target samples, and the query samples after passing through the STN module. We see that the samples modified using the baseline method have almost the same appearances as the query images. Meanwhile, we see that the images obtained using our method are close to the target samples. We believe that L_s plays an important role in correctly estimating relative spatial parameters and removing the effects of pose changes.

3.3. Comparisons with existing methods

We compared the performance of our method with that of an existing triplet learning method [9]. We used a triplet network by removing the STN s( ) and L1 loss term L_s from our network. We only evaluated the accuracy of object recognition because the triplet network was unable to estimate object poses. The average accuracy achieved using the triplet network is 90.2 ± 1.5 whereas the accuracy of our method is 95.1 ± 1.1. We confirmed that our network combining triplet learning with an STN using the L1 loss is superior to the simple triplet network.

We next compared the performance of our method with that of an existing STN method [10]. We combined a recognition module with the STN module used in our network. The recognition module has two convolutional layers, two max pooling layers, and two fully connected layers. We used cross-entropy loss in the existing method. We evaluated the accuracy of object recognition and the error in pose estimation. The average accuracy and error when using the existing STN method are 92.2 ± 4.4 and 1.09 ± 0.44 whereas the accuracy and error when using our method are 95.1 ± 1.1 and 0.44 ± 0.02. We confirmed that our method embedding an STN module in triplet learning is superior to the existing STN method.

3.4. Comparison with the random selection of anchors

As described in Section 2.3, our method selects anchor images by acquiring representative canonical planes. We here compare the performance of our method with that of the random selection of anchors. The average accuracy and error when using the random selection are 89.1 ± 1.1 and 0.82 ± 0.02 whereas the average accuracy and error when using our method are 95.1 ± 1.1 and 0.44 ± 0.02. We confirmed that the acquisition of representative canonical planes is effective in terms of improving the performance of object recognition and pose estimation.

4. CONCLUSIONS

We proposed a weakly supervised learning method to jointly recognize objects and estimate their poses through the triplet learning of canonical plane transformation. We demonstrated that our method outperforms the single use of triplet learning or the STN even though we did not provide pose parameter labels.

In future work, we will further evaluate our method on datasets of objects having various shapes. We will also explore the use of homography transformation instead of affine transformation to treat pose changes that are more complex.

5. REFERENCES

[1] P. Baker and Z. Halim, “An exploration of warehouse automation implementations: cost, service and flexibility issues,” *Supply Chain Management: An International Journal*, vol. 12, no. 2, pp. 129–138, 2007.

[2] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, “Analysis and observations from the first amazon picking challenge,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2018.

[3] W. Liu, D. Anguelov, and D. Erhan, “Ssd: Single shot multibox detector,” in *Proceedings of European Conference on Computer Vision*, 2016, pp. 21–37.

[4] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6517–6525.

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

[6] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T. Kim, “Pose guided rgbd feature learning for 3d object pose estimation,” in *Proceedings of IEEE International Conference on Computer Vision*, 2017, pp. 3876–3884.

[7] A. Kanezaki, Y. Matsushita, and Y. Nishida, “Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[8] M. Sundermeyer, Z. C. Marton, M. Durner, M. Brucker, and R. Triebel, “Implicit 3d orientation learning for 6d object detection from rgb images,” in *Proceedings of European Conference on Computer Vision*, 2018, pp. 712–729.

[9] J. Wang, Y. Song, T. Leung, C. Rosenberg, Wang, J. Philbin, B. Chen, and Y. Wu, “Learning fine-grained image similarity with deep ranking,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[10] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015, vol. 2, pp. 2017–2025.

[11] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, “Person re-identification by multi-channel parts-based cnn with improved triplet loss function,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1335–1344.

[12] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad: Cnn architecture for weakly supervised place recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1437–1451.

[13] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, “End-to-end learning of deep visual representations for image retrieval,” *International Journal of Computer Vision*, vol. 124, no. 2, pp. 237–254, 2017.

[14] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[15] C. Rennie, R. Shome, K. E. Bekris, and A. F. D. Souza, “A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place,” *IEEE Robotics and Automation Letters*, vol. 1, pp. 1179–1185, 2016.