Viewpoint Estimation using Triplet Loss with A Novel Viewpoint-based Input Selection Strategy

Changjian Gu1,2, a, Changsheng Lu1,2, Chaochen Gu1,2, b and Xiping Guan1,2

1Department of Automation, Shanghai Jiao Tong University, Shanghai, China
2Key Laboratory of System Control and Information Processing, Ministry of China Education

Email: avincent_gu@sjtu.edu.cn; bjacygu@sjtu.edu.cn

Abstract. Viewpoint estimation is a fundamental procedure in vision-based robot tasks. A good viewpoint of the camera relative to objects can help the visual system perform better both in observation and manipulation. Recently, CNN-based algorithms, which can effectively extract discriminative features from images in challenging conditions, are utilized to handle the viewpoint estimation problem. However, most existing algorithms focus on how to leverage the extracted deep features while neglecting the spatial relationship among images that captured from various viewpoints. In this paper, we present a deep metric learning method for solving the viewpoint estimation problem. A triplet loss with a novel viewpoint-based input selection strategy is introduced, which could learn more powerful features after incorporating the spatial relationship between viewpoints. Combined with the traditional classification loss, the presented loss can further enhance the discriminative power of features. To evaluate the performance of our method, a dataset containing a large number of images generated from five different texture-less workpieces is built and the experiment results show the effectiveness of the proposed method.

1. Introduction

Generally, human beings are not necessary to calculate the precise pose of the target object relative to the eyes but roughly estimating the view of target to determine the action. In vision-based robot tasks, precise pose estimation for target is also a great challenging because of the requirement of the accurate camera and coordinate calibration. Moreover, the situations that objects perform different shapes under different views bring difficulties in pose estimation [1]. Hence, some researchers [1]-[3] believe that finding an optimal viewpoint, which is defined as a point on an object-centered view sphere, is helpful for better manipulation and observation. Viewpoint estimation has become a meaningful research subject. For example, in 3D object retrieval tasks [3]-[4], estimating the viewpoint of objects can help enhance the performance of object recognition and in active vision tasks [2], viewpoint estimation is considered as a pre-procedure of pose estimation.

Recently, in order to extract more robust and discriminative features, [5]-[7] have employed the convolutional neural networks (CNNs) to handle the viewpoint estimation problem. Viewpoint estimation is solved as a classification problem and excellent performance is obtained. In practical applications, the difference between the feature of images and that between the spatial distance of viewpoints is often incommensurate. In some cases, objects tend to have similar shapes and features observed from the front side and back side, but the distance between the two observation positions is
large. To overcome this shortcoming, [5] proposed a geometric loss to take the spatial relationship into account and improve the performance of viewpoint estimation. According to recent works, metric learning [8] can also be well used to learn the spatial relationship and seems suitable for viewpoint estimation task. In metric learning, triplet loss [9], which is proposed by Weinberger and Saul, has been introduced for training CNN in order to learn a metric or an embedding space that makes the samples from the same class closer to each other than those from different classes. The triplet loss has been implemented in training CNNs for face recognition [9] and person re-identification [10]. Similarly, if the embedding of image features in learned feature space can be adjusted on the basis of the spatial relationship of known viewpoints, the network can represent more discriminative features that take both the image self and viewpoints into account. In this paper, we propose a triplet loss function with a viewpoint-based input selection strategy and apply it training the CNN to help the network learn discriminative features. To evaluate the proposed method, some different workpieces with symmetrical and low texture properties are made and utilized to build a large-scale viewpoint estimation dataset in CAD environments. The results tested on the generated dataset demonstrate that our proposed method is feasible and efficient.

The remaining of this paper is organized as follows. The data generation procedure is introduced in Section 2 and the proposed method is described in Section 3. The implementation details and experiment results are in Section 4. Section 5 gives the conclusion.

2. Dataset Generation
In CNN-based supervised algorithms, a robust deep learning model needs to be driven by large quantities of labelled images. However, it is great expensive and time-consuming to collect and precisely label the images from different viewpoints in a viewpoint estimation task. To reduce the burden of collecting images, CAD models, which are easy to obtain in industrial scene and have high similarity with real objects, are utilized [11] to generate large-scale synthetic images instead of capturing images from real environment.

We follow the previous work [12] and introduce an object-centered view sphere with a constant radius, as shown in figure 1. The point on the sphere is defined as viewpoint. Note that we only consider the upper view sphere because it is the workspace of most robot arms. In order to better describe the viewpoint, a spherical coordinate system is built on the hemisphere. In the coordinate system, 360 points are sampled along the longitude direction and 90 points are sampled along the latitude direction so that every viewpoint can be denoted by its corresponding coordinate with the polar angle ($\theta$) and azimuthal angle ($\phi$). Since a huge computational burden would be brought in if each viewpoint is treated as a separate class, we classify near neighbour viewpoints into the same class. Specifically, a viewpoint node is designated at every 45 points in the longitude direction and every 30 points in the latitude direction. Based on these viewpoint nodes, the viewpoints closer to the node are classified as the same viewpoint class so that 17 classes are obtained, and thus the viewpoint estimation problem can be simplified to classification problem. The viewpoint classes are expressed as:

$$V = \{ v_k = (\theta_i, \phi_j) | i = 1, 2, \ldots, p; \quad j = 1, 2, \ldots, q \}$$

where $v_k$ denotes the coordinate of viewpoint node and $p, q$ indicate the number of slices divided in polar and azimuthal angle, respectively.
Facilitated with the pre-defined view sphere, we program in the CAD environment to make a virtual camera automatically move on the hemisphere and render the object as images with precise viewpoint class label. However, the rendering process is carried out in the synthetic environment so that there exists the discrepancy between the rendered images and the real images. To bridge the gap, we change the light intensity and position during the rendering process and embedding different background images taken in the real environment on the rendered image. As a consequence, it is more reliable to directly use these labelled processed images, which are denoted by \( I \) indicates the image and \( V \) the label, to learn a robust classifier that can be applied in the real scene.

3. Method

A large-scale image dataset can be built using the above dataset generation procedure, which is enough for training a CNN-based classifier. When the CNN is trained on the dataset, a classification loss and triplet loss are jointly employed to encourage the network to simultaneously learn the image features and spatial relationships of different viewpoint images. In this section, triplet loss is introduced and a novel viewpoint-based input selection strategy is proposed to make the triplet loss better applied to the viewpoint estimation task.

3.1. Review on Triplet loss

Triplet loss, as its name suggested, is calculated on a triplet of samples. The goal of the triplet aims at helping the network to find an embedding space where the samples from the same class are closer than those from different classes. Specifically, the input of triplet loss consists of a pair of samples from the same class \( (s_i, s_j) \) and a pair of dissimilar ones \( (s_i, s_k) \). Generally, \( s_i \) is called as an anchor, \( s_j \) is a positive sample and \( s_k \) is a negative sample. Given such a group of input, the triplet loss can be expressed as:

\[
L_t(s_i, s_j, s_k) = \sum_{i=1}^{N} \max [m + D(s_i, s_j) - D(s_i, s_k), 0]
\]

where \( m \) denotes the margin between the same class and different classes. Here \( D \) is calculated by the squared Euclidean distance function:

\[
D(u, v) = \frac{1}{2} ||u - v||^2
\]

By applying the triplet loss function, the features of the similar samples are pulled closer in learned embedding space and those of dissimilar samples are pulled away.

3.2. Viewpoint-based input selection strategy

Recall that the criterion for defining positive and negative samples is whether the samples belonging to the same class or not, which is used for pulling similar samples and pulled dissimilar samples away in learned feature space. In viewpoint estimation problem, indeed, there exists quantitative differences between viewpoints. Therefore, we extend the application of triplet loss and propose a novel viewpoint-based triplet input selection strategy, which defines the positive and negative samples according to the
distance between viewpoints.

Specifically, for a single triplet loss calculation, three samples \( S_1 = \{ I_{a}, V_a \}, S_2 = \{ I_{a}, V_a \} \) and \( S_3 = \{ I_{a}, V_a \} \) are randomly selected from the dataset. In these three points, two of them may be from the same viewpoint class but all three points cannot belong to the same class. On one hand, if two of the samples belong to the same class, one is taken as anchor \( (S_a) \), the other as positive sample \( (S_p) \) and the remaining third one as the negative sample \( (S_n) \). On the other hand, if the three samples belong to three different classes, the viewpoint distance between each two of them is calculated. Then the pair of samples with the largest distance and that with smallest distance is selected. These two pairs must have one common sample, which is marked as \( S_a \). The sample with smaller distance relative to the anchor is taken as \( S_p \) and that with bigger distance as \( S_n \). In order to better quantify the difference between viewpoints, the two-dimension coordinates \( (\theta, \psi) \) of viewpoint class is uniformly transformed into a rectangular coordinate system:

\[
V' = (x, y, z) = (\cos(\theta)\cos(\psi), \cos(\theta)\sin(\psi), \sin(\theta))
\]

Then the viewpoint Euclidean distance between two samples is calculated. In figure 2, the selection strategy is more intuitively expressed.

![Image](image_url)

**Figure 2.** The illustration of the input selection strategy

The triplet loss function is thus modified from equation (2) and has the following form:

\[
L_t(S_a, S_p, S_n) = \sum_{i=1}^{N} \max \left[ m + D(f(I_{a}), f(I_{a})), 0 \right] - D(f(I_{a}), f(I_{a})), 0 \right]
\]

\[
(5)
\]

where \( f \) is the mapping of the network and \( f(\cdot) \) represents the output vector from the last full-connected layer.

Equation (5) reveals that the triplet loss is zero if the distance between negative sample and anchor is bigger than the summation of the margin and the distance between the positive sample and anchor. Otherwise, the loss is non-zero, which encourages the network to learn a better embedding space during the training process.

Combined with a cross-entropy loss term for classification, the loss function implemented in our case is expressed as:

\[
L_{\text{total}}(S_a, S_p, S_n) = \frac{1}{3} (L_{\text{ce}}(S_a) + L_{\text{ce}}(S_p) + L_{\text{ce}}(S_n)) + \lambda L_t(S_a, S_p, S_n)
\]

\[
(6)
\]

where \( Y_S \) and \( \hat{Y}_S \) denote the prediction and the ground-truth for sample \( S \), \( N \) is the number of class and \( \lambda \) is influence factor of the triplet loss term for the whole loss function, which is chosen as 1 in this paper.
Figure 3. The implementation framework in the training process. Details of VGG16 is described in [13].

The overall framework is shown in figure 3. Minimizing the $L_{ce}$ can help the network learn features from image itself and minimizing the $L_t$ can achieve two properties: 1) enlarging the distance between features from farther viewpoint classes and 2) reducing the distance between those from closer or similar classes. In other words, by minimizing the whole loss function, the network is encouraged to learn the features from the image itself and simultaneously learn the spatial relationship between viewpoint classes. In fact, triplet loss can also be regarded as a regularization item from the spatial level so that the features learned by the network are more representative.

4. Experiment

In this section, we will present the details of the dataset preparation and the experiment implementation, and then show the experimental results and corresponding analysis.

4.1. Data preparation

The viewpoint estimation dataset consists of five different workpieces and their corresponding images are captured from various viewpoints. All the chosen workpieces have the characteristics of low texture, uniform color, and symmetry, which are very common in industrial scenes. We follow the procedure introduced in Section 2 to generate the workpiece image dataset. In fact, we utilize the 3DsMax and the Maxscript to program the movement of the virtual camera and the rendering process. For each viewpoint class of one workpiece, the virtual camera first moves to the pre-defined viewpoint node to render images and then moves to 30 viewpoints randomly selected near this node to render more images. In the rendering process at each viewpoint, the workpiece rotates around its own central axis to cover more conditions. Every rendered image is automatically annotated with precise viewpoint class label based on the position of the virtual camera. This process is repeated 17 times because we define 17 viewpoint classes in Section 2 so that one workpiece totally contains about 15000 images. When the images are rendered, the intensity and position of lighting are randomly changed and some background images captured from the real environment are embedded on the rendered images to bridge the gap between synthetic domain and real domain. In addition, the whole dataset is split into a training dataset for learning and a testing dataset for evaluation at a ratio of 7:3. Each dataset for every workpiece is denoted from $WP_1$ to $WP_5$. Some images from the generated dataset ($WP_i$) are shown in figure 4.
4.2. Comparision
In order to demonstrate the advantage of our proposed method, the geometric loss, which is implemented in [5] [12], is chosen for comparison. The formula of geometric loss can be written as follow:

\[ L_{geo}(I) = - \sum_{i} \sum_{v} e^{-d(v, \hat{v})/\sigma} \ln P(V) \]  

where \( P(V) \) is the predicted probability of viewpoint class \( V \) for input \( I \) and \( \sigma \) is a tuneable parameter.

It is obvious that geometric loss has also considered the distance between the viewpoints. However, the main difference lies in that the geometric loss is calculated based on the prediction and the ground truth, while the triplet loss is calculated by the feature vectors outputted by the network.

4.3. Implementation details
As described in Section 3, we chose a CNN as the model to solve the viewpoint estimation problem, where 16-layer VGG-Net [13] is chosen as the CNN-based model. Before performing experiments, the CNN-based model is initialized with pre-trained weights on Image-Net. Three different loss function are implemented for comparison: 1) simple cross-entropy loss \( L_{ce} \); 2) cross-entropy loss and geometric loss \( (L_{ce} + L_{geo}) \); 3) cross-entropy loss and triplet loss \( (L_{ce} + L_{tr}) \).

In every training step, a batch of training images, which are all resized to \( (300 \times 300) \) pixels and randomly cropped as \( (227 \times 227) \) pixels, is fed into the network. Then the model is continuously optimized using Stochastic Gradient Descent (SGD) algorithm with a batch size of 32 examples, learning rate of 0.001, momentum of 0.9 based on the implemented loss function. In a single experiment, all training images are trained by 10 epochs so that the loss can reach convergence. All experiments are conducted through a deep learning toolkit: Pytorch [14] on our computer with CPU: Intel i7-5820k and accelerated by GPU: Nvidia GTX Titan.

4.4. Experiments result

4.4.1. Quantification. After conducting the experiments, we evaluate the model on the testing dataset and adopt 'accuracy' as the evaluation metric to quantify the performance:

\[ \text{Accuracy} = \frac{\sum_{i} \Phi(\hat{V}_i, V_i)}{N} \times 100\% \]  

where \( N \) denotes the number of images in the testing dataset. \( \Phi \) is defined as an indicator that \( \Phi(a, b) = 1 \) if \( a = b \) else \( \Phi(a, b) = 0 \).

The evaluation results on testing dataset of \( WP_1 \) to \( WP_5 \) are shown in table 1.
We notice that our proposed method outperforms those applying simple classification loss or combining the geometric loss and classification loss. For the viewpoint estimation task, our method outperforms the other two methods by 8.2% and 2.1% accuracy, respectively. The improvement demonstrates that triplet loss can effectively encourage the network to learn more discriminative features.

4.4.2. Visualization. To further demonstrate the effectiveness of our method, the learned embedding space of WP₁ is plotted in figure 5 using Tensorboard [15], which is a toolkit that often used in deep learning research to visualize the learned feature representation with the aid of some dimensionality reduction algorithms so as to explore the distribution of learned features. Here we choose t-SNE [16] as the dimensionality reduction algorithm to reduce the feature vector outputted by the last fully-connected layers from 4096 dimensions to 2 dimensions.

![Figure 5. Visualization of learned features tested on one of the datasets (WP₁). The data points are 2D-features reduced from high dimensional features using t-SNE. Features from class (0,0) are marked with the red boxes and those from (0,180) are with the blue boxes. Best viewed in colour.](image)

The three graphs show that only minimizing classification loss can cluster the features of similar images but minimizing geometric loss or triplet loss can distinguish features according to the viewpoint distance, making the features of similar images that belong to different class more discriminative. For example, the images from (0,0) and (0,180) show similar image features while the two viewpoint class has a relatively large dissimilarity. Features of the two classes learned only with classification loss are very close because the loss just focuses on the images self. In contrary, Lgeo and Lt can effectively make the features farther away based on the viewpoint distance. Moreover, we see that feature embedding learned by our method exhibits tighter clustering and less misclassification data points, which demonstrates that the method can encourage the network to be more representative and enhance the performance in viewpoint estimation task.

5. Conclusion
In this work, the triplet loss function that allows for better tackling the viewpoint estimation problem has been implemented. We focus on extending the application of triplet loss and propose a novel viewpoint-based input selection strategy of triplet loss. Besides, a complete workpiece image dataset

**Table 1. Accuracy on five workpieces using three loss functions.**

|       | WP₁ | WP₂ | WP₃ | WP₄ | WP₅ | Average |
|-------|-----|-----|-----|-----|-----|---------|
| Lcea | 86.0% | 84.8% | 89.9% | 79.4% | 88.5% | 85.7% |
| Lgeo + Lcea | 92.5% | 91.8% | 93.1% | 88.2% | 93.8% | 91.8% |
| Lₜ + Lcea (ours) | 94.5% | 93.4% | 96.7% | 89.8% | 95.2% | 93.9% |

We notice that our proposed method outperforms those applying simple classification loss or combining the geometric loss and classification loss. For the viewpoint estimation task, our method outperforms the other two methods by 8.2% and 2.1% accuracy, respectively. The improvement demonstrates that triplet loss can effectively encourage the network to learn more discriminative features.
using different workpieces and CAD environment for viewpoint estimation is built. Compared to the classification loss and geometric loss, our method can learn more discriminative and structural features, which demonstrates that the method is effective and feasible in viewpoint estimation task and our work may be of importance for vision-based robot tasks in industrial applications.

Acknowledgement
This work is supported by National Natural Science Foundation of China under the grants 61503243 and U1405251, and in part by the National Instrument Development Special Program of China (2013YQ3065101)

References
[1] Browatzki B, Tikhanoff V, Metta G, and Bülthoff H H 2012 Active object recognition on a humanoid robot Proc. Int. Conf. on Robotics and Automation vol 44 (Saint Paul: IEEE) pp 2021–28
[2] Chen S, Li Y and Kwok N M 2011 Active vision in robotic systems: A survey of recent developments International Journal of Robotics Research 33 1343–77
[3] He X, Zhou Y, Zhou Z, Bai S, and Bai X 2018 Triplet-center loss for multi-view 3d object retrieval Proc. Conf. on Computer Vision and Pattern Recognition (Salt Lake City)
[4] Su H, Maji S, Kalogerakis E, and Learned-Miller E 2015 Multi-view Convolutional Neural Networks for 3D Shape Recognition Proc. Int. Conf. on Computer Vision (Santiago: IEEE) pp 945–53
[5] Su H, Qi C. R., Li Y, and Guibas L J 2016 Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views Proc. Int. Conf. on Computer Vision (Santiago: IEEE) pp 2686–94
[6] Wohlkinger W, Aldoma A, Rusu R B and Vincze M 2012 3DNet: Large-scale object class recognition from CAD models Proc. Int. Conf. on Robotics and Automation (Saint Paul: IEEE) pp 5384–91
[7] Qi C R, Su H, Nießner M, Dai A, Yan M and Guibas L J 2016 Volumetric and Multi-view CNNs for Object Classification on 3D Data. Proc. Conf. on Computer Vision and Pattern Recognition (Las Vegas: IEEE) pp 5648–56
[8] Wang J, Zhou F, Wen S, Liu X and Lin Y 2017 Deep metric learning with angular loss Proc. Int. Conf. on Computer Vision (Venice: IEEE) pp 2612–20.
[9] Schroff F, Kalenichenko D and Philbin J 2015 Facenet: a unified embedding for face recognition and clustering Proc. Conf. on Computer Vision and Pattern Recognition (Boston: IEEE) pp 815–23.
[10] Hermans A, Beyer L and Leibe B 2017 In defense of the triplet loss for person re-identification
[11] Wang Y, Li S, Jia M and Liang W 2016 Viewpoint Estimation for Objects with Convolutional Neural Network Trained on Synthetic Images Pacific Rim Conference on Multimedia (Xi’an: Springer) pp 169–79
[12] Zhang L J, Gu C C, Wu K J, Huang Y and Guan X P 2017 Model-based active viewpoint transfer for purposive perception Proc. Conf. on Automation Science and Engineering (Xi’an: IEEE) pp 1085–89
[13] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition Computer Science http://arxiv.org/abs/1409.1556v6
[14] pytorch.org
[15] https://github.com/tensorflow/tensorboard/
[16] Morán A, Fuertes J J, Prada M A, Alonso S, Barrientos P and Diaz I 2012 Analysis of electricity consumption profiles by means of dimensionality reduction techniques Communications in Computer & Information Science 311(Suppl 1) pp 152–61