Cognitive Computation (2020) 12:834–843
https://doi.org/10.1007/s12559-020-09717-5

Cognitive Template-Clustering Improved LineMod for Efficient Multi-object Pose Estimation

Tielin Zhang[1] · Yang Yang[5] · Yi Zeng[1,2,3,4] · Yuxuan Zhao[1]

Received: 16 November 2019 / Accepted: 5 February 2020 / Published online: 17 March 2020

Abstract

Various types of theoretical algorithms have been proposed for 6D pose estimation, e.g., the point pair method, template matching method, Hough forest method, and deep learning method. However, they are still far from the performance of our natural biological systems, which can undertake 6D pose estimation of multi-objects efficiently, especially with severe occlusion. With the inspiration of the Müller-Lyer illusion in the biological visual system, in this paper, we propose a cognitive template-clustering improved LineMod (CT-LineMod) model. The model uses a 7D cognitive feature vector to replace standard 3D spatial points in the clustering procedure of Patch-LineMod, in which the cognitive distance of different 3D spatial points will be further influenced by the additional 4D information related with direction and magnitude of features in the Müller-Lyer illusion. The 7D vector will be dimensionally reduced into the 3D vector by the gradient-descent method, and then further clustered by K-means to aggregately match templates and automatically eliminate superfluous clusters, which makes the template matching possible on both holistic and part-based scales. The model has been verified on the standard Doumanoglou dataset and demonstrates a state-of-the-art performance, which shows the accuracy and efficiency of the proposed model on cognitive feature distance measurement and template selection on multiple pose estimation under severe occlusion. The powerful feature representation in the biological visual system also includes characteristics of the Müller-Lyer illusion, which, to some extent, will provide guidance towards a biologically plausible algorithm for efficient 6D pose estimation under severe occlusion.

Keywords Müller-Lyer illusion · Cognitive template-clustering · Brain-inspired computation · LineMod · 6D pose estimation

Introduction

The evolutionary procedure of the mammalian brain has resolved the problem of 6D pose estimation by integrating different related brain regions, hundreds of specifically designed neuron types, and functional microcircuits. However, a challenge remains in discovering the mysterious black box of the brain and designing the most efficient biologically plausible model for 6D pose estimation.

With significant development of computer science and cognitive robot theory, various types of theoretical algorithms have been proposed [1] in the research area of 6D pose estimation, e.g., the point pair method, template matching method, Hough forest method, and deep learning method. However, these efforts in machine learning and robotics are still a considerable distance from the performance of the natural biological system. They still face fundamental problems such as sensitivity to illumination changes, noise, blur, and occlusion.

The Müller-Lyer illusion is a special kind of functional phenomenon in the procedure of visual information processing of the brain [2], in which the cognitive distance of different 3D spatial points will be further affected by additional features related to direction and magnitude. Figure 1 shows the three basic types of Müller-Lyer illusion, e.g., the distance illusion in Fig. 1a, direction illusion in Fig. 1b, and angle illusion in Fig. 1c, respectively. These illusions in the visual system contribute to feature detection and object identification during 6D pose estimation, especially in cases of severe occlusion.

Tielin Zhang and Yang Yang contributed equally to this article and should be considered as co-first authors.

✉ Tielin Zhang
tielin.zhang@ia.ac.cn

Extended author information available on the last page of the article.

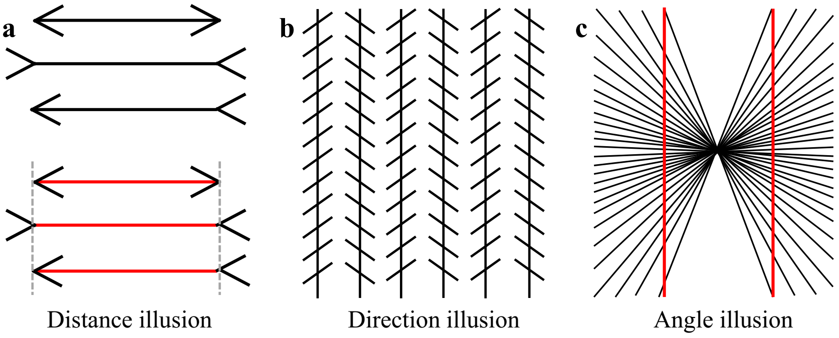

Fig. 1 **a**–**c** The Müller-Lyer illusion in the biovision system

In order to estimate 6D pose in the case of multiple objects with occlusion and motion blur, as well as decrease the computation cost as much as possible, we firstly focused on the standard Patch-LineMod method for its accuracy and efficiency. Patch-LineMod is an improved version of the LineMod method, which separates points into different patches by the K-means method and makes possible multi-object 6D pose estimation.

Inspired by the Müller-Lyer illusion, a cognitive template-clustering improved LineMod (CT-LineMod) model was proposed. The model uses a 7D cognitive feature vector to replace standard 3D spatial points in the clustering procedure of Patch-LineMod, in which the cognitive distance of different 3D spatial points is further affected by additional 4D features related to direction and magnitude in the Müller-Lyer illusion. The 7D vector is dimensionally reduced into the 3D vector by gradient-descent method, and then further clustered by K-means method to aggregately match templates and automatically eliminate superfluous clusters, which makes template matching possible on both holistic and part-based scales.

The CT-LineMod model can be considered as an integration of powerful 7D feature representation from Patch-LineMod and efficient cognitive template-clustering based on the Müller-Lyer illusion. The model was verified on the standard Doumanoglou dataset and performed perfectly, demonstrating the accuracy and efficiency of the proposed model on cognitive feature distance measurement and template selection on multiple pose estimation with severe occlusion.

Related Works

Four main types of models have been proposed in the research area of 6D pose estimation, including the point pair method, template matching method, Hough forest method, and deep learning method.

For point pair–based algorithms, Drost et al. proposed a point pair feature algorithm, which successfully integrates global description, local point pair features, and local-global matching conversion for better recognition performance in the case of noise, clutter, and partial occlusions [3]. However, it performs poorly in detecting objects with similar background clutter and ignores valuable edge information of objects. Hinterstoisser proposed a new sampling and voting point pair scheme [4] to reduce the harmful effects of clutter and sensor noise. However, this method is extremely sensitive to occlusion and cannot recover the complete object location information. A cognitively inspired 6D motion estimation method is proposed based on solving the Perspective-n-Point problem and the Kalman filter [5].

For template matching–based algorithms, Hinterstoisser et al. proposed the LineMod algorithm, which utilizes the gradient information and normal features of the surface of an object for template matching [6, 7]. However, LineMod shows poor performances on real-time template matching. One possible reason for this is it only focuses on the strong edges during feature extraction. Hodan et al. proposed BOP [8] as a novel and standard benchmark to use with current datasets. However, it focuses more on different single objects rather than multiple overlapping objects in a single scene.

For Hough forest–based algorithms, Gall et al. proposed a target detection algorithm [9], which constructs a random forest to extract image blocks, and then makes a template judgment within each decision tree, and votes in the Hough space. Tejani et al. proposed the latent-class Hough forest model, which integrates a new template-based segmentation function into the regression forest [10]. However, this is limited by the manually designed features for different objects in different overlapped scenes.

For deep neural network (DNN)–based algorithms, Kehl et al. proposed a single-shot multi-box detection algorithm, which uses the DNN-based model for depth learning of 2D images and then uses the projection properties to analyze the inferred points and in-plane rotation scores [11]. A similar convolutional neural network (CNN) based on category detectors has also been used [12]. A robust 3D object detection and pose estimation pipeline based on RGB-D images has been constructed, which can detect multiple objects simultaneously while reducing false positives [13]. For heavy clutter scenes and occlusion problems, Bonde et al. proposed a highly robust real-time object recognition framework, which uses an iterative training scheme to classify the position and posture of 3D objects [14]. Xiang et al. proposed a new pose-CNN for 6D pose estimation by introducing a new loss function named ShapeMatch-Loss [15]. Feifei et al. proposed heterogeneous DenseFusion architecture based on PoseCNN, which uses an end-to-end iterative gesture fine-tuning program and performs excellently on both YCB-Video and LineMod datasets [16]. Euclidean distance, scalable nearest neighbor search method, and CNN are integrated as an efficient model to capture both the object identity and 3D pose [17, 18]. Park et al. proposed a novel architecture Pix2Pose based on CNN [19], which predicted the coordinates of each pixel after feature extraction, and then calculated the position and orientation by voting. Although this effort largely improved the robustness of pose estimation, especially under heavy occlusion, the computation cost was relatively expensive considering a comparable accuracy can be achieved with other methods.

Besides 6D pose estimation, many clustering methods have also been proposed for intelligent patch identification, especially in cases involving severe occlusion. Nazari et al. proposed a clustering ensemble measurement framework, which is based on the cluster-level weighting, can assign weights into each cluster with different reliability, and has high robustness, clustering quality, and time complexity [20]. Rashidi et al. proposed a clustering ensemble framework, which is based on the integration of undependability concepts and cluster weighting, and also uses hierarchical agglomerative clustering and bi-partite graph formulation to estimate the cluster dependability and certainty [21]. Some semi-supervised clustering methods have also been proposed for better information representation [22].

Unfortunately most of the models mentioned above are not as efficient and robust as the human brain. Further inspiration from the biological system is necessary for human-comparable algorithm on efficient 6D pose estimation.

Standard LineMod and Patch-LineMod

LineMod is a template-based approach for 6D pose estimation [6]. It can handle untextured objects under massive clutter by taking a short training time. However, this template-based effort will inevitably fail on the identification of "multiple" objects in complex scenes. The number of features in this method cannot be greater than a predesigned parameter, for example, 64, which dramatically limits the higher performance of the algorithm. Besides, the recognition rate of the LineMod algorithm will drop rapidly under occlusion. The training procedure of LineMod is shown in Fig. 2a.

Different with LineMod which makes the pose estimation mostly based on the similarity of target object and whole template, the Patch-LineMod [8] separates the whole template into different small templates (e.g., patch templates) by K-means clustering, and then makes the similarity between training patches and target patches for the identification of 6D pose with some parts of object under occlusion. The training procedure of the Patch-LineMod is shown in Fig. 2b. Both the LineMod and Patch-LineMod algorithms contain training and testing phases:

– The training phase is shown in Fig. 2a and b: LineMod and Patch-LineMod load the RGB-D images of a specific object from different training directions and preprocess it with Gaussian blur and Sobel operator. Then both of them calculate the gradient direction and magnitude above the predefined threshold as the 7D cognitive feature. Finally, the template matching is used for the learning procedure of pose identification.
– The test phase is shown in Fig. 2d: A predefined sliding window is used for the matching process, which is from both horizontal and vertical directions, respectively. The similarity of 7D cognitive features between trained objects (or patches with K-means in Patch-LineMod) and target objects is calculated, and then the trained object with the biggest similarity will be selected as the estimated pose.

Cognitive Template-Clustering Improved LineMod

Comparisons between the different processing steps of LineMod, Patch-LineMod, and CT-LineMod are shown in Fig. 2. All three methods contain training and test phases.

For the standard LineMod method, it will determine the similarity measurement of these features directly with the target object by template matching. However, this matching cannot detect an object when it is affected by an occlusion. For example, by using LineMod method in Fig. 2a, only one object (and also one pose) is identified in the test phase in Fig. 2d.

For Patch-LineMod method in Fig. 2b, the simple K-means clustering method is used based on purely spatial locations (e.g., the X, Y, and Z positions of points), which will generate different patches for different modalities, hence will detect more objects in the test phase, for example, two objects are detected in Fig. 2d.

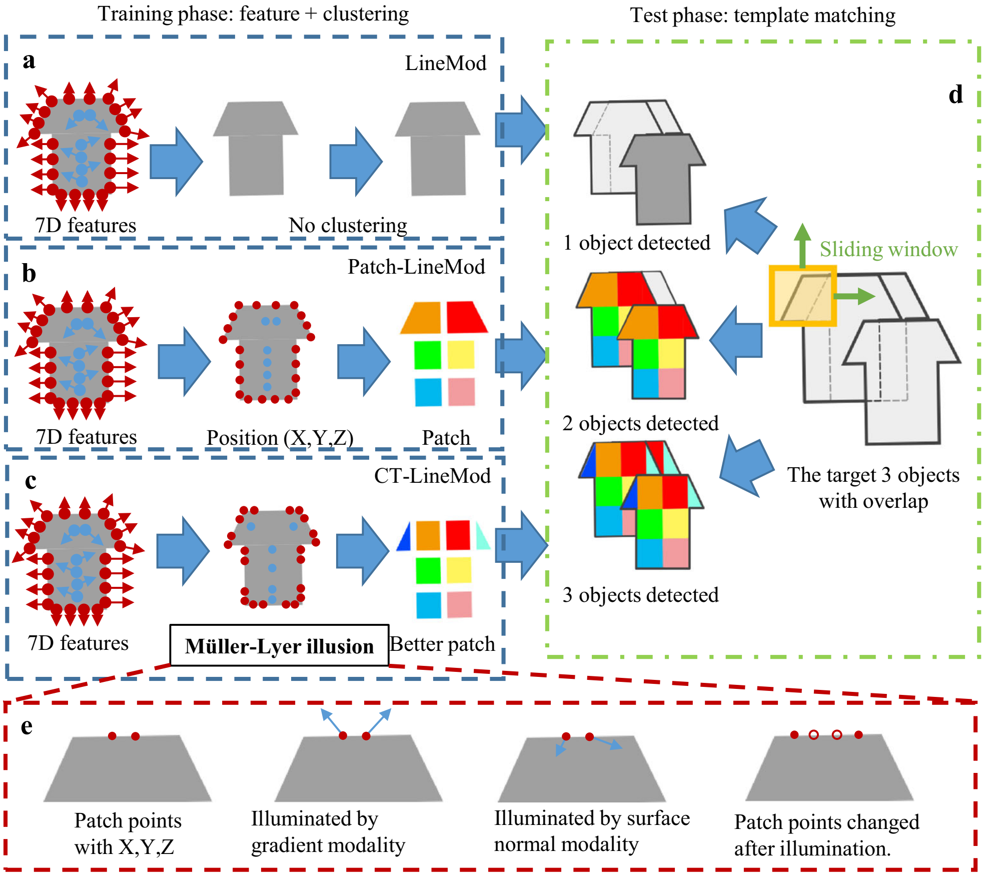

Fig. 2 **a–e** The comparisons between LineMod, Patch-LineMod, and CT-LineMod methods

The Müller-Lyer illusion can provide a new 3D cognitive feature vector to replace the original 3D spatial location. The cognitive feature vector is the dimensional reduction from the 7D cognitive feature vector. The distance between two points (e.g., the 3D spatial information) will be affected by their neighborhood features (e.g., the direction and magnitude of the gradient feature in another 4D vector), as shown in Fig. 2e. This illusion will change the feature characteristics of objects; for example, the features near the edges or borders will contribute more to the feature identification. Hence, after the cognitive template-clustering in Fig. 2c, the CT-LineMod will successfully detect all of the three objects in Fig. 2d.

The 7D Cognitive Feature Vector

The 7D cognitive feature vector contains the 3D spatial feature vector (with the X, Y, and Z spatial locations), and also another 4D vector for gradient direction, gradient magnitude, surface-normal direction, and surface-normal magnitude, as shown in Fig. 3.

The LineMod-type methods contain both gradient modalities and surface normal modalities. For good template matching algorithms, the templates after selection should be robust to scale change, color variance, and severe occlusion. The 7D cognitive feature vector has the potential of integrating different types of cognitive features. For example, the distances between different features will all contribute to the patch allocation. The intrinsic links between these feature points will further help the next-step multiple 6D pose estimation.

Cognitive Template-Clustering

The cognitive template-clustering is the improvement of patch clustering in Patch-LineMod with Müller-Lyer illusion, as shown in Fig. 2e. The templates are aggregated based on the similarity matching of all 7D cognitive feature vectors, e.g., locations (with three dimensions), magnitudes (with two dimensions), and directions (with two dimensions).

The overall procedure of CT-LineMod is shown in Fig. 4. After loading of the raw images in Fig. 4a, e.g., the 3D cloud points, the four additional features are calculated for each 3D points in Fig. 4b. Features in the neighborhood area with size $s \times s$ are integrated as the templates in Fig. 4c. Then the patches of templates are allocated based on the K-means clustering method with updated spatial 3D vectors

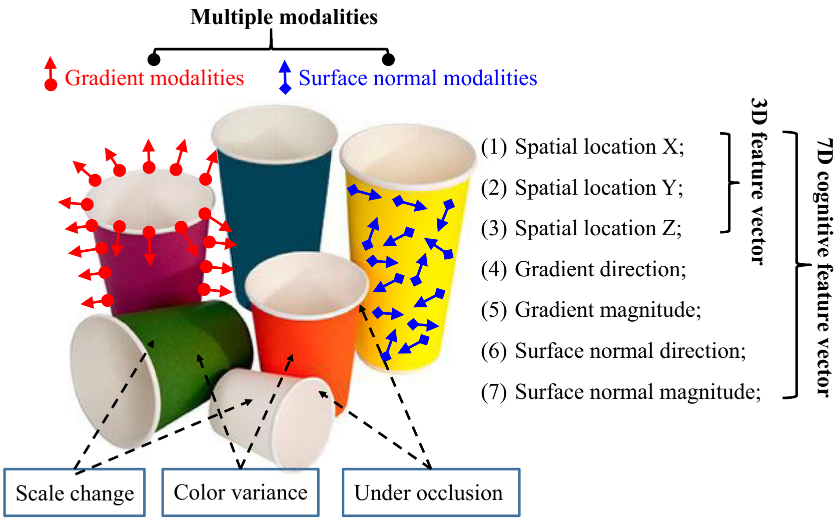

Fig. 3 The 7D cognitive feature vector

after Müller-Lyer illusion, as shown in Fig. 4d. Finally, the templates from input images will be matched with templates from training images in Fig. 4e, and the matched templates will contain the estimated 6D pose.

Information in Feature Level

The information in feature level is shown in Fig. 4b and Eq. (1). F_p^I and F_p^M are the features calculated from input image I and trained image M respectively in the position of point p.

$$\begin{cases} F_p^I = \left\{p_X^I, p_Y^I, p_Z^I, p_{\mathrm{gd}}^I, p_{\mathrm{gm}}^I, p_{\mathrm{sd}}^I, p_{\mathrm{sm}}^I\right\} \\ F_p^M = \left\{p_X^M, p_Y^M, p_Z^M, p_{\mathrm{gd}}^M, p_{\mathrm{gm}}^M, p_{\mathrm{sd}}^M, p_{\mathrm{sm}}^M\right\} \end{cases} \quad (1)$$

p^I is the raw pixel in the input 3D RGB-D image I, p^M is the raw pixel in the trained image M. p_X^I, p_Y^I, and p_Z^I are the spatial locations of point p in X, Y, and Z axes in image I. p_{gd}^I, p_{gm}^I, p_{sd}^I, and p_{sm}^I represent features from gradient direction, gradient magnitude, surface-normal direction, and surface-normal magnitude respectively.

Information in Template Level

The template-level information is shown in Fig. 4c and Eq. (2), in which s is the calculation step in template T_p, and p is the center point template T with an area of $s \times s$.

$$\begin{cases} T_p^I = \frac{1}{s^2} \sum_{p \in P}^{s \times s} F_p^I(\mathrm{gd}, \mathrm{gm}, \mathrm{sd}, \mathrm{sm}) \\ T_p^M = \frac{1}{s^2} \sum_{p \in P}^{s \times s} F_p^M(\mathrm{gd}, \mathrm{gm}, \mathrm{sd}, \mathrm{sm}) \end{cases} \quad (2)$$

Müller-Lyer Illusion from 7D to 3D Feature Space

The Müller-Lyer illusion can be considered as the dimension reduction of the information in feature level from the

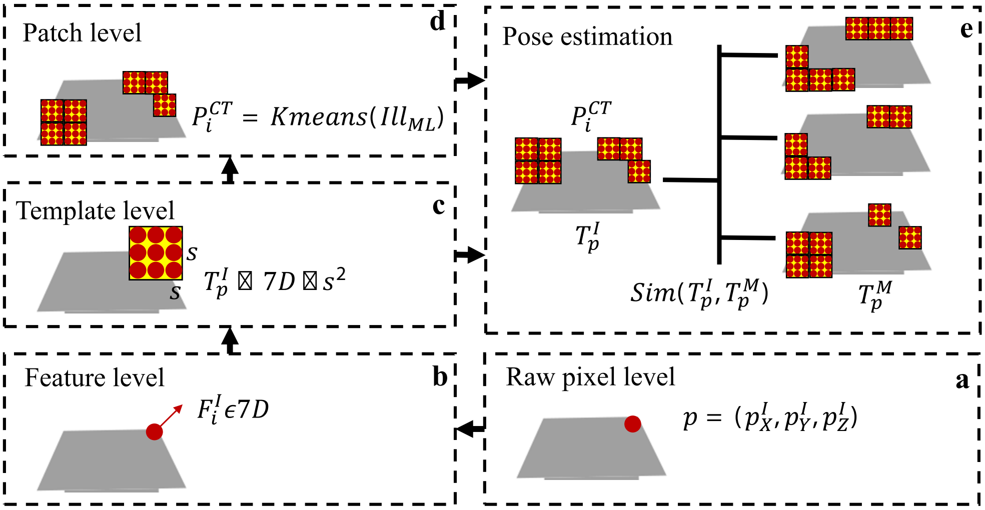

Fig. 4 **a–e** The architecture of hierarchical information processing in CT-LineMod

7D information to the 3D information. Equation (3) shows the Müller-Lyer illusion function Ill_{ML}, in which the input of the function is the 7D feature in Eq. (1) and the output will be the updated 3D pixel-level information from p^I to $\hat{p}^I$.

$$\hat{p}_X^I, \hat{p}_Y^I, \hat{p}_Z^I = Ill_{\mathrm{ML}}(p_X^I, p_Y^I, p_Z^I, p_{\mathrm{gd}}^I, p_{\mathrm{gm}}^I, p_{\mathrm{sd}}^I, p_{\mathrm{sm}}^I) \tag{3}$$

For the easier understanding, here we use x_i^{7d} to represent the F_p^I for the calculation of Ill_{ML}. As shown in Eq. (4), p_{ij} shows the conditional probability of two features between feature i and feature j in original 7D space. The feature vectors of x_i^{7d} and x_j^{7d} are with seven dimensions, and the Gaussian distribution centers on x_i^{7d} and x_j^{7d} are calculated. The σ_i is the variance, and n is the number of candidate points.

$$\begin{cases} p_{j/i} = \frac{\exp(-||x_i^{7d} - x_j^{7d}||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i^{7d} - x_k^{7d}||^2/2\sigma_i^2)} \\ p_{ij} = \frac{p_{j/i} + p_{i/j}}{2n} \end{cases} \tag{4}$$

Here we map the vector from 7D to 3D to represent the Müller-Lyer illusion, in which the original spatial 3D with p_X^I, p_Y^I, and p_Z^I, will be affected by p_{gd}^I, p_{gm}^I, p_{sd}^I, and p_{sm}^I, and during the mapping, we also need to reflect the similarity between high-dimensional 7D and low-dimensional 3D data points in the form of conditional probability q_{ij} in 3D space.

$$q_{ij} = \frac{\left(1 + ||y_i^{3d} - y_j^{3d}||^2\right)^{-1}}{\sum_{k \neq l} \left(1 + ||y_k^{3d} - y_l^{3d}||^2\right)^{-1}} \tag{5}$$

We then calculate all of the conditional probability p_{ij} in 7D space and q_{ij} in 3D space. Every two pairs are calculated to measure the minimal Kullback-Leibler divergence.

$$C = \mathrm{KL}(p_{ij}||q_{ij}) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{6}$$

Then, the stochastic gradient descent method is used for the information mapping from 7D to 3D space.

$$\frac{\partial C}{\partial y_i^{3d}} = 4 \sum_j \frac{(p_{ij} - q_{ij})\left(y_i^{3d} - y_j^{3d}\right)}{1 + \left\| y_i^{3d} - y_j^{3d} \right\|^2} \tag{7}$$

After iteratively learning of p_{ij} and q_{ij}, finally, we will get $\hat{p}_X^I$, $\hat{p}_Y^I$, and $\hat{p}_Z^I$ from the mapping of p_X^I, p_Y^I, and p_Z^I with Müller-Lyer illusion.

The function of $Ill_{\mathrm{ML}}()$ is similar with the traditional linear dimension-reduction algorithm PCA (the abbreviation of "principal component analysis") or t-SNE (the abbreviation of "t-distributed stochastic neighbor embedding") method [23], which is a non-linear dimension-reduction algorithm for mining high-dimensional data space into a lower-dimensional data space. The $Ill_{\mathrm{ML}}()$ can explain complex polynomial relations between features and perform well when they focus on dissimilar data points in lower-dimensional regions.

Information in Patch Level

The patch-level information is represented as P_i. For the standard Patch-LineMode method, the patch clustering will be based purely on the spatial information p_X^I, p_Y^I, and p_Z^I, as shown in Eq. (8).

$$\{\, P_i = K\mathrm{means}(p_X^I, p_Y^I, p_Z^I),\ p \in I, M \tag{8}$$

With the help of Müller-Lyer illusion, the new cognitive feature points $\hat{p}_X^I$, $\hat{p}_Y^I$, and $\hat{p}_Z^I$ will replace p_X^I, p_Y^I, and p_Z^I, as shown in Eq. (9).

$$\begin{cases} P_i^{\mathrm{CT}} = K\mathrm{means}(\hat{p}_X^I, \hat{p}_Y^I, \hat{p}_Z^I) \ \mathrm{if}(\hat{p} \in I, M) \\ P_i^{\mathrm{CT}} = K\mathrm{means}(p_X^I, p_Y^I, p_Z^I) \ \mathrm{if}(\hat{p} \notin I, M) \end{cases} \tag{9}$$

The P_i^{CT} represents the proposed cognitive template-clustering in the procedure of patch generation with K-means methods.

Similarity Measurement for Pose Estimation

As shown in Eq. (10), the pose estimation is looking for the max similarity template features from image I, which are most closely with the template features in dataset image M, with the condition of the patch search area identified by the K-means method. The function Sim() is the cosine similarity, which measures the angle of the two input vectors.

$$\mathrm{pose} = \arg \max_{P_i^{\mathrm{CT}}, i \in I, M} \sum_{p}^{p \in P_i^{\mathrm{CT},I}, P_i^{\mathrm{CT},M}} \mathrm{Sim}(T_p^I, T_p^M) \tag{10}$$

ICP Post Processing

We use the iterative closest point (ICP) algorithm to remove the duplicate poses based on the evaluation scores, which is the non-maximum suppression algorithm for pose calculation, pose correction, and pose verification [24].

The Training and Test Procedure of CT-LineMod

The two phases of training and test procedures for CT-LineMod-based 6D pose estimation are shown in Algorithm 1.

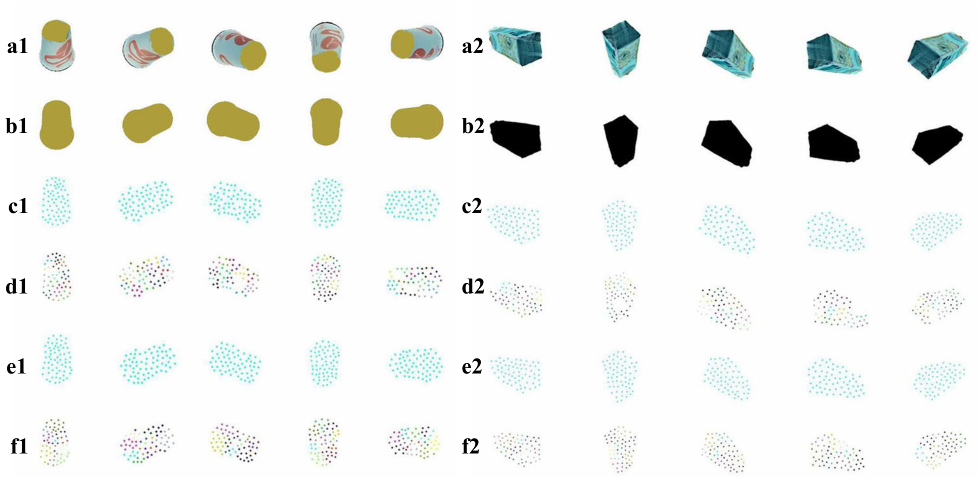

Fig. 5 The RGB-D images of cup and box in Doumanoglou dataset and the feature maps during training procedure (a1, a2). The RGB-D raw images for cup and box (b1, b2). The masks of objects (c1, c2). 3D spatial positions in Patch-LineMod (d1, d2). Clustering results in Patch-LineMod (e1, e2). 3D spatial positions in CT-LineMod (f1, f2). Clustering results in CT-LineMod

Algorithm 1 The algorithm for CT-LineMod model.

```
1. Raw samples generation from 3D cloud point; the 7D cognitive vector assignment; Template-Clustering variables configuration; the setting of feature size $s$.
2. Start training procedure:
2.1. Load training samples;
2.2. The 7D cognitive feature generation with Eq. (1);
2.3. The templates calculation with Eq. (2);
2.4. The Müller-Lyer illusion from 7D to 3D feature space with Equations from (3) to (7);
2.5. The patch-level information calculation with Eq. (9);
2.6. Pose estimation with Eq. (10);
2.7. The ICP post processing of estimated 6D poses.
3. Start test procedure:
3.1. 3D sliding window for target selection with feature size $s$.
3.2. Template matching for feature identifiers.
3.3. Multiple pose estimation.
4. End pose learning and estimation.
```

Experimental Results

RGB-D Doumanoglou Dataset

The 3D point cloud cup and box datasets are the main parts of the Doumanoglou dataset [25], as shown in Fig. 5. It contains the training set and test set, in which the training set contains 4740 rendered images, and the test set contains 177 images. The range of object distances is from 455 to 1076 mm, the azimuth range is from 0 to 360°, and the elevation range is from −58 to 88°.

Figure 5a1 shows the raw images of a mini coffee cup in 360° at different observation angles. The clustered feature maps during the training procedure include the RGB-D image, mask image in Fig. 5b1, and initial feature map in Fig. 5c1 and e1. Figure 5d1 shows the clustered feature map based on the Patch-LineMod method, in which the K-means method is used for the 3D spatial location clustering. Figure 5f1 shows the clustering of the Müller-Lyer illusion from 7D to 3D feature space. It is easy to find out that the cognitive template-clustering method will generate better feature point clustering, which could well separate the corners, edges, and plane of the cups. Additionally, it can generate more uniformly distributed feature points compared with the Patch-LineMod, which also contributes to the performance of 6D pose estimation under occlusion.

Similar with the coffee cup, the box dataset is also processed by both the Patch-LineMod and CT-LineMod from Fig. 5a2 to f2. The source code in this paper is forked from Patch-LineMod project[1], and then updated into the CT-LineMod[2].

6D Pose Estimation

6D pose estimation is the task of detecting 6D poses of cups and boxes, which contain the locations and orientations of different objects. The awareness of position and orientation of objects in a scene is sometimes referred to as six degrees of freedom pose.

After feature calculation and patch generation, we use the segmented template to perform the sliding window matching in the horizontal and vertical directions, respectively, and calculate the similarities in these windows. Figure 6 shows the improved CT-LineMod compared with

[1] https://github.com/meiqua/patch_linemod

[2] https://github.com/brain-cog/Brain_inspired_Batch

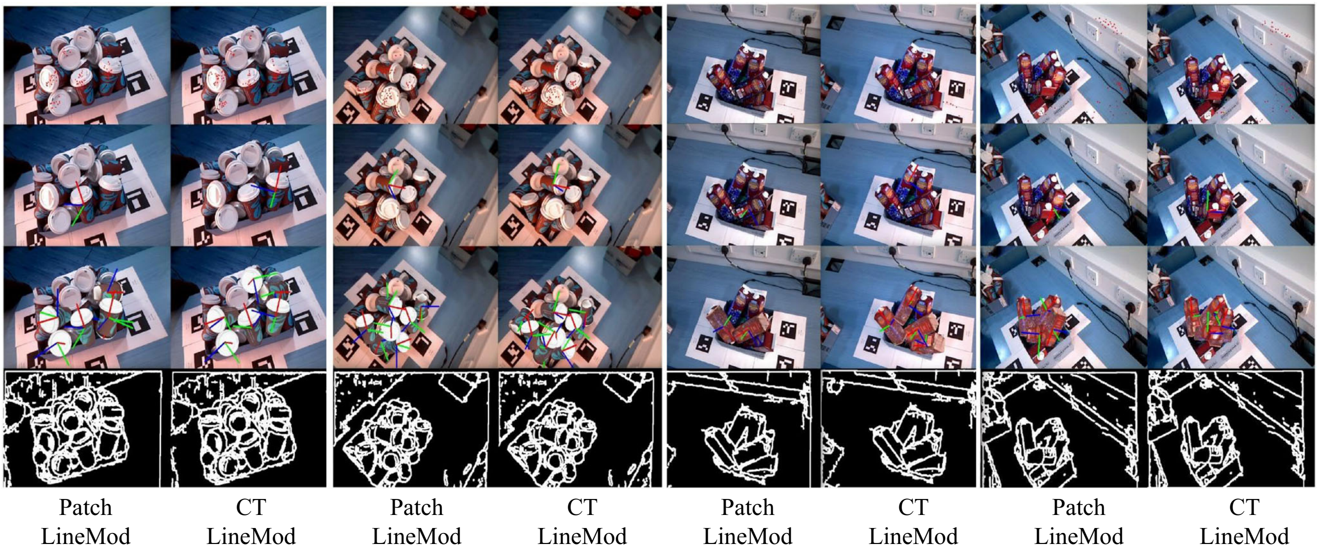

Fig. 6 6D pose estimation for cups and boxes in Doumanoglou dataset

standard Path-LineMod on identifying the number of targets and the accuracy of pose estimation of coffee cups and boxes.

- The first line of Fig. 6 shows the raw RGB-D images and detected feature points after the candidate-feature filtering. This image contains 12 cups, and the AprilTag is used for the spatial calibration [26].
- The second line of Fig. 6 shows the detected 6D pose with the top-1 target, which shows the one with the highest confidence in all of the candidate object poses. For a different view point, the detected best pose candidate object is different.
- The third line of Fig. 6 shows the multiple pose estimation of the images, in which more than one candidate poses are calculated by both the CT-LineMod and also the Patch-LineMod methods. Moreover, for most of the cases, the CT-LineMod will get more pose candidates. This result will also be verified in the compared results in Tables 1 and 2.
- The fourth line of Fig. 6 shows raw detected modalities, and the white color represents gradient modalities.

The Experimental Comparisons

We use the recall and F_1 values to show the performance of proposed CT-LindMod on the dataset with severe occlusion, which includes detected-correct, detected-incorrect, and undetected results of 6D pose estimation. N_{top} is the top N pose estimates with highest confidence, which is evaluated by two kinds of objects in each scene (i.e., 56 RGB-D pictures for coffee cup, 60 RGB-D pictures for juice box, and 61 RGB-D pictures for both of them, and the total is 177 test samples) in the Doumanoglou-dataset [25].

As shown in Table 1, we select three kinds of scenes, including the coffee cup scene, the juice box scene, and the mixed scene. Moreover, the three methods are compared with the configuration of N_{top} maximum, e.g., the standard LineMod, Patch-LineMod, and CT-LineMod methods. From the figure, the proposed CT-LineMod method will largely improve the recall performance compared with other methods. The CT-LineMod can identify most of the targets, except the ones hidden in the bottom or under severe occlusion.

Table 1 The comparison of recall values for three algorithms on three datasets

Scenes	LineMod	Patch-LineMod	CT-LineMod
Coffee cup	0.051	0.412	0.443
Juice box	0.253	0.439	0.459
Mixed scene	0.017	0.373	0.384
Mean values	0.107	0.408	0.422

Table 2 The comparison of F_1 scores between our method and other state-of-the-art methods

Methods	Coffee cup	Juice box	Mean
LineMod	0.819	0.494	0.656
PPF	0.867	0.604	0.735
Hough forest	0.877	0.870	0.873
Doumanoglou	0.932	0.819	0.875
Ours	0.947	0.913	0.93

Besides, we have tested our method on full Doumanoglou dataset (with heavy occlusion scenes) in the “Sixd Challenge”, and make the further F_1 score comparison with other states of the art methods with the configuration of $N_{\text{top}} = 1$. The LineMod algorithm [6], PPF (point pair Feature) algorithm [3], Hough forest algorithm [10], Doumanoglou algorithm [25], and our proposed model are verified. The mean is the average of the performance on two datasets. The F_1 scores of these methods are calculated, as shown in Table 2, which shows the power of the proposed CT-LineMod model compared with other state-of-the-art methods.

Conclusions

The LineMod method is a kind of template matching method which is more efficient for 6D pose estimation compared with other methods based on point pair, Hough forest methods, and DNNs. However, the standard LineMod method can only detect a single target (i.e., the one with the maximum confidence value) and cannot engate with multiple object occlusion and motion blur. One of the primary motivations of the paper is trying to find an efficient way to make possible multiple 6D pose estimation under occlusion.

Hence, inspired by the Müller-Lyer illusion, a cognitive template-clustering improved LineMod model, i.e., CT-LineMod model, is proposed. The model uses a 7D cognitive feature vector to replace standard 3D spatial points in the clustering procedure of Patch-LineMod, in which the cognitive distance of different 3D spatial points is further influenced by the additional 4D information related to direction and magnitude of features in Müller-Lyer illusion. The Müller-Lyer illusion is a kind of sensation illusion in the visual system, which contributes to a better feature generation, feature representation, and also 6D pose estimation under severe occlusion. Finally, the CT-LineMod method has been verified by its performance of 6D pose estimation on the Doumanoglou dataset, in which the model’s accuracy and efficiency were demonstrated.

Author Contributions Tielin Zhang, Yang Yang, and Yi Zeng conceived the idea and formulated the problem. Tielin Zhang and Yang Yang performed the simulations and wrote the paper. Tielin Zhang and Yuxuan Zhao analyzed the results.

Funding Information This study is financially supported by the Beijing Natural Science Foundation (No. 4184103), the National Natural Science Foundation of China (No. 61806195), the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No.XDB32070100), the Beijing Municipality of Science and Technology (Grant No. Z181100001518006), the CETC Joint Fund (Grant No. 6141B08010103), and the Beijing Academy of Artificial Intelligence (BAAI).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Luo B, Hussain A, Mahmud M, Tang J. Advances in brain-inspired cognitive systems. Cogn Comput. 2016;8(5):795–796.
2. Seel NM, (ed). 2012. Müller-lyer illusion. Boston: Springer.
3. Drost B, Ulrich M, Navab N, Ilic S. Model globally match locally: Efficient and robust 3d object recognition. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE; 2010. p. 998–1005.
4. Hinterstoisser S, Lepetit V, Rajkumar N, Konolige K. Going further with point pair features. In: European Conference on Computer Vision. Springer; 2016. p. 834–848.
5. Chen J, Luo X, Liu H, Sun F. Cognitively inspired 6d motion estimation of a noncooperative target using monocular rgb-d images. Cogn Comput. 2016;8(1):105–113.
6. Hinterstoisser S, Cagniart C, Ilic S, Sturm P, Navab N, Fua P, Lepetit V. Gradient response maps for real-time detection of textureless objects. IEEE Trans Pattern Anal Mach Intell. 2011;34(5):876–888.
7. Hinterstoisser S, Lepetit V, Ilic S, Holzer S, Bradski G, Konolige K, Navab N. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Asian Conference on Computer Vision. Berlin: Springer; 2012. p. 548–562.
8. Hodan T, Michel F, Brachmann E, Kehl W, GlentBuch A, Kraft D, Drost B, Vidal J, Ihrke S, Zabulis X, et al. Bop: Benchmark for 6d object pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 19–34.
9. Gall J, Stoll C, De Aguiar E, Theobalt C, Rosenhahn B, Seidel H-P. Motion capture using joint skeleton tracking and surface estimation. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2009. p. 1746–1753.
10. Tejani A, Tang D, Kouskouridas R, Kim T-K. Latent-class hough forests for 3d object detection and pose estimation. In: European Conference on Computer Vision. Springer; 2014. p. 462–477.
11. Kehl W, Manhardt F, Tombari F, Ilic S, Navab N. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 1521–1529.

12. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 779–788.
13. Kehl W, Milletari F, Tombari F, Ilic S, Navab N. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In: European Conference on Computer Vision. Springer; 2016. p. 205–220.
14. Bonde U, Badrinarayanan V, Cipolla R. Robust instance recognition in presence of occlusion and clutter. In: European Conference on Computer Vision. Springer; 2014. p. 520–535.
15. Xiang Y, Schmidt T, Narayanan V, Fox D. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. Robotics: Science and Systems (RSS). 2018.
16. Wang C, Xu D, Zhu Yuke, Martín-martín R, Lu C, Fei-Fei L, Savarese S. Densefusion: 6d object pose estimation by iterative dense fusion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. p. 3343–3352.
17. Wohlhart P, Lepetit V. Learning descriptors for object recognition and 3d pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015. p. 3109–3118.
18. Tompson JJ, Jain A, LeCun Y, Bregler C. Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems; 2014. p. 1799–1807.
19. Park K, Patten T, Vincze M. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision; 2019. p. 7668–7677.
20. Nazari A, Dehghan A, Nejatian S, Rezaie V, Parvin H. A comprehensive study of clustering ensemble weighting based on cluster quality and diversity. Pattern Anal Applic. 2019;22(1):133–145.
21. Rashidi F, Nejatian S, Parvin H, Rezaie V. Diversity based cluster weighting in cluster ensemble: an information theory approach. Artif Intell Rev, pp 1–28. 2019.
22. Qin Y, Ding S, Wang L, Wang Y. Research progress on semi-supervised clustering. Cognitive Computation, pp 1–14. 2019.
23. van der Maaten L, Hinton G. Visualizing data using t-sne. J Mach Learn Res. 2008;9:2579–2605.
24. Besl PJ, McKay ND. Method for registration of 3-d shapes. In: Sensor fusion IV: Control Paradigms and Data Structures. International Society for Optics and Photonics; 1992. p. 586–606.
25. Doumanoglou A, Kouskouridas R, Malassiotis S, Kim T-K. Recovering 6d object pose and predicting next-best-view in the crowd. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 3583–3592.
26. Olson E. Apriltag: A robust and flexible visual fiducial system. In: 2011 IEEE International Conference on Robotics and Automation. IEEE; 2011. p. 3400–3407.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Tielin Zhang[1] · Yang Yang[5] · Yi Zeng[1,2,3,4] · Yuxuan Zhao[1]

✉ Yi Zeng
yi.zeng@ia.ac.cn

Yang Yang
1701210376@pku.edu.cn

[1] Research Center for Brain-inspired Intelligence,
Institute of Automation, Chinese Academy of Sciences,
Beijing, China

[2] Center for Excellence in Brain Science and Intelligence
Technology, Chinese Academy of Sciences, Shanghai, China

[3] National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
Beijing, China

[4] University of Chinese Academy of Sciences, Beijing, China

[5] School of Software and Microelectronics,
Peking University, Beijing, China