https://doi.org/10.20965/jaciii.2021.p0204

Paper:

An Improved Algorithm for Detection and Pose Estimation of Texture-Less Objects

Jian Peng[*,**] **and Ya Su**[*,**]

[*]School of Automation, China University of Geosciences
388 Lumo Road, Hongshan, Wuhan, Hubei 430074, China
[**]Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems
388 Lumo Road, Hongshan, Wuhan, Hubei 430074, China
E-mail: {pengjian, suya}@cug.edu.cn
[Received September 29, 2020; accepted December 22, 2020]

This paper introduces an improved algorithm for texture-less object detection and pose estimation in industrial scenes. In the template training stage, a multi-scale template training method is proposed to improve the sensitivity of LineMOD to template depth. When this method performs template matching, the test image is first divided into several regions, and then training templates with similar depth are selected according to the depth of each test image region. In this way, without traversing all the templates, the depth of the template used by the algorithm during template matching is kept close to the depth of the target object, which improves the speed of the algorithm while ensuring that the accuracy of recognition will not decrease. In addition, this paper also proposes a method called coarse positioning of objects. The method avoids a lot of useless matching operations, and further improves the speed of the algorithm. The experimental results show that the improved LineMOD algorithm in this paper can effectively solve the algorithm's template depth sensitivity problem.

Keywords: computer vision, object detection and pose estimation, LineMOD algorithm

1. Introduction

Object detection and pose estimation are important components in computer vision systems and research problems in the field of computer vision. Research on object detection and pose estimation algorithms is also crucial to the field of robotics and augmented reality. In order to improve the degree of automation and production efficiency, many factories use a large number of robots instead of manual workers [1]. For example, in the mutual fields of augmented reality and object sorting, robots are already the most important part. Augmented reality is a hot research field in recent years. Its purpose is to fuse and interact the virtual world with the real world. The realization of augmented reality technology is based on object detection and pose estimation, and on this basis, a virtual world is established and enabled to interact with the real world. Although robots have high demands in the industry, there are still some technical problems in the robot system that need to be effectively solved, such as the robot grabbing scattered and disordered objects. A good vision system is the prerequisite for the robot to complete any operation, and the robot's vision system mainly includes the recognition and pose estimation technology of the target object.

Due to the wide application of object detection and pose estimation, in recent years, many scholars have conducted in-depth research on it and proposed many excellent algorithms. For example, Guo et al. [2] proposed the method of local features, Rusu et al. [3] proposed the VFH feature descriptor method, Brachmann et al. [4] proposed a method based on dense feature learning. However, most of these algorithms are for the detection and pose estimation of textured objects. There are still many problems to be solved for the detection and pose estimation of texture-less objects. Because of the rich pattern information on the surface of textured objects, effective features can be extracted from these rich patterns, so the detection and pose estimation of textured objects is relatively simple. Texture-less objects cannot extract effective features from their surfaces because their surfaces are smooth and non-textured. Therefore, this paper mainly studies the detection and pose estimation of texture-less objects. LineMOD method was proposed by Hinterstoisser et al. [5] in 2011. It mainly solves the problem of real-time detection and positioning of 3D objects in complex background. It uses the information of RGB-D, and can deal with the situation without texture, and does not need lengthy training time. Based on the LineMOD algorithm, an improved strategy is proposed for the depth sensitivity of its template, which enables it to perform object detection and pose estimation quickly and accurately in scenes with a wide range of object depth variation.

This research has two main contributions. First, the template invocation method of LineMOD algorithm was improved to solve the template depth sensitivity problem, and the template invocation strategy based on the Scene-Patch (Scene area) was proposed. The strategy of tem-

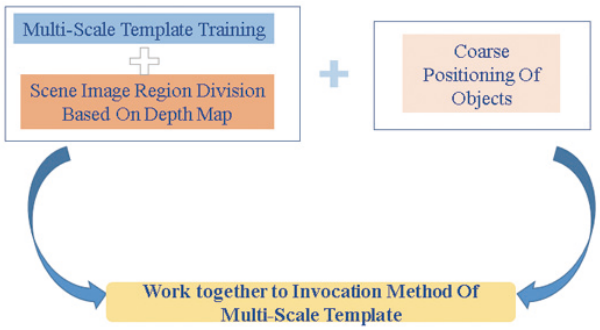

Fig. 1. Overview of the proposed recognition algorithm.

plate invocation based on Scene-Patch is firstly to train the multi-scale template of the object. Then, according to the depth value of the scene image, the scene image is divided into several areas. The depth of images in these areas is kept within a certain range, and the multi-scale template is flexibly called in each scene image area when template matching is carried out. In the case that the depth of the target object in the scene can always be similar to the depth of the template, the target object can still be quickly and accurately identified.

And then in order to reduce the number of template matching to improve the speed of the algorithm, we proposed a method called coarse positioning of objects. It can filter out the scene image area which obviously does not contain the target object, and can locate the possible location of the target object in the scene image, subsequent template matching operations will only be performed at these locations to achieve accurate identification of the target object. Experimental results show that the improved strategy in this paper enables the LineMOD algorithm to perform object detection and pose estimation quickly and accurately in scenes with a wide range of object depth variation. The overview of the proposed algorithm is shown in **Fig. 1**.

2. Related Work

At present, object detection and pose estimation techniques are mainly divided into four categories: methods based on local features, methods based on deep learning, methods based on point pair and methods based on template matching. This section will summarize and explain the advantages and disadvantages of these four categories of methods.

2.1. Methods Based on Local Features

Local features have been widely applied in object detection. SIFT [5], SURF [6], and ORB [7, 8] are the common local features. These features are carefully designed, and have some unique advantages, such as rotation, lighting and scale invariance, so these features have good robustness when dealing with scenes with occlusion and lighting changes. However, these features are only suitable for the recognition of textured objects, and the recognition effect for texture-less objects is not good. In many fields, such as industrial robot operation and augmented reality technology,operation objects are often texture-less artifacts, so these methods based on local features are not applicable to actual industrial scenes.

2.2. Methods Based on Deep Learning

Since deep learning technology can break through the limitations of some traditional methods, it has been widely applied in the field of object detection and recognition. In 2D object detection and recognition, more representative methods include CNN [9], RCNN, faster-RCNN [10], YOLO [11, 12], SSD [13]. On the basis of these 2D object detection and recognition methods, object detection and 6D pose estimation techniques have been gradually developed, which are mainly divided into two categories: SSD-6D and YOLO-6D. SSD-6D [14] is composed of two basic network structures, namely SSD network and automatic encoding network. Among them, SSD network is mainly responsible for object recognition and pose estimation with a single RGB image as input. The automatic coding network takes the CAD model of the object as the input to extract the high dimensional features of the object. The YOLO-6D [15] network takes RGB images as input and outputs the 3D bounding box and classification prediction of the target object in the test image. After obtaining the 3D bounding box of the object, the posture of the target object can be calculated by the PNP algorithm [16]. The advantage of the method based on deep learning is that the speed of object detection and pose estimation is fast and the accuracy is high. The disadvantage is higher requirements for hardware equipment.

2.3. Methods Based on Point Pair

Drost et al. [17] proposed a method based on matching directional point pairs between the point cloud of the test scene and the object model, which has become one of the classical 6D pose estimation methods. Prior to this, the accuracy and speed of the global-based method did not meet the satisfactory requirements, and was mainly limited to the classification and recognition of certain specific objects; in contrast, the local matching method based on local invariant features was proved very effective, but the production of local invariant features depends largely on the quality of the acquired data and model data. Compared with these methods, the method [17] adopts the idea of global modeling and local matching. During training, the points of the model are sampled, and similar features are grouped together and stored in a hash table; during the test, random reference points are found in the scene, similar model point pairs are searched in the hash table, and a fast voting method similar to Hough transform is used to vote for each matching point pair, the peak value in the accumulator is extracted as the candidate of pose, and the best pose is finally selected through ICP optimization.

Table 1. Pose estimation accuracy of LineMOD algorithm in different situations.

The accuracy of pose estimation (%)		The depth of template (cm)	
		70	90
The depth range of target object [cm]	68–72	98.2	63.2
	88–92	66.5	98.9

2.4. Method Based on Template Matching

LineMOD algorithm [18] is the most representative template matching algorithm and the algorithm with the best effect. Different from traditional template matching methods, in the template trained by the LineMOD algorithm, its features are discretized, so not all feature points participate in the template matching operation, which will greatly reduce the computational complexity of the algorithm. In the detection stage, the template and the test image are tested for similarity using a sliding window. If the similarity is higher than the set threshold, it indicates that there is a target object in the test image, and the pose of the corresponding template can be regarded as the initial pose of the target object. After the initial posture of the target object is obtained, the precise posture of the target object can be further solved by the ICP algorithm [19]. Fanelli et al. [20, 21] proposed the method of random forests. This method improves the matching efficiency by dividing the template and training the random forest. However, the size and number of template image blocks of this method are difficult to grasp, and its implementation is more complex. Zhang et al. [22] proposed a cascading template matching method, and in order to solve the problem of sensitivity to template scale, scale-independent technology was proposed. However, there is a zooming operation in this method. If the zooming is serious, the image information may be lost seriously.

In summary, considering the requirements of industrial reliability, real-time performance, and rapid training of newly added objects, the LineMOD algorithm is still the most suitable method among these methods, but the algorithm is sensitive to the depth of the template. As can be seen from **Table 1**, when the depth of the template is similar to the depth of the target object in the test image, the LineMOD algorithm has a high pose estimation accuracy rate; otherwise, the pose estimation accuracy of this algorithm decreases seriously. This paper proposes an improved solution to the depth-sensitive problem of the template of LineMOD algorithm.

3. Proposed Method

In order to improve the recognition accuracy, the LineMOD algorithm needs to train the template at various depths of the target object. Since the LineMOD algorithm needs to traverse all the templates of the target object during template matching, the speed of the algorithm will decrease as the depth of the object template increases. If the depth of the target object in the scene varies in a large range, the number of templates will be greatly increased by training the templates at various depths of the target object, and the speed of the LineMOD algorithm will inevitably decrease. Aiming at this problem of LineMOD algorithm, this paper improves the template invocation mode of this algorithm when template matching, and proposes a template invocation strategy based on Scene-Patch. The strategy mainly includes three key technologies: multi-scale template training method, scene image region division based on depth map, and coarse positioning of objects, which enables the algorithm to select the template specifically when the template depth type increases. Therefore, the algorithm can still quickly identify the target object in the scene when the depth of the target object changes in a wide range.

3.1. Multi-Scale Template Training

The templates trained at multiple depths are called multi-scale templates. Since the template depth is a discrete quantity, it is impossible to train the templates at all depths when training multi-scale templates. Therefore, it is necessary to determine the step size of the depth change and the depth range to be covered when training the template. The step size of the depth change can be set according to the size of the target object. In practical application, the distance between the target object and the camera is a variable within a certain range. In the actual scene, the maximum and minimum depth of the target object in the scene image can be determined, thus the depth range that the template needs to cover can be determined. The training method of the multi-scale template is as follows.

Determine the maximum and minimum depth that the target object can reach in the actual scene, denoted as D_{max} and D_{min}, then the depth range covered by the template is $[D_{min}, D_{max}]$. The radius of the circumscribing sphere of the target object is denoted as r, and several depth layers are set in steps size of r, and the depth of each depth layer is denoted as D_i, where $i \in (1, 2, \ldots, m)$, $D_{min} \leq D_i \leq D_{max}$, $m = (D_{max} - D_{min})/r$. Train the template sequences at the corresponding depths of all the depth layers, then a total of m template sequences can be obtained. The depth of the template in each template sequence is the same, and the depth is equal to the depth D_i of the corresponding depth layer. The process of multi-scale template training is shown in **Fig. 2**.

3.2. Scene Image Region Division Based on Depth Map

The purpose of the scene image area division based on the depth map is to divide the scene image into several areas according to the depth value of each pixel of the scene image, and the depth value in each area is within an approximate range. In the subsequent template matching operation in each scene image area, the template trained at this depth can be targeted according to the depth of the corresponding depth layer of each scene image area, and only the template trained at this depth can be matched.

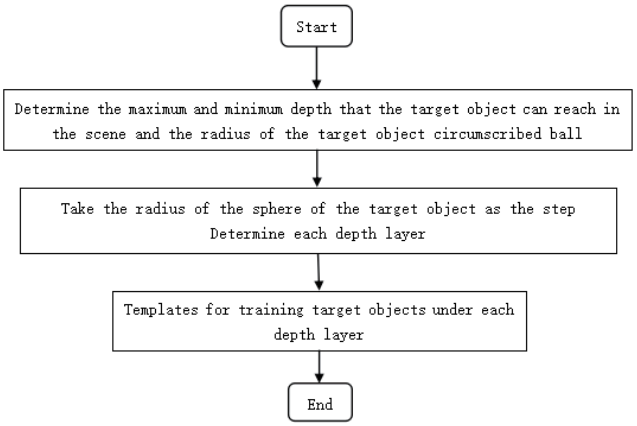

Fig. 2. Multi-scale template training flowchart.

In this way, it does not need to traverse all templates of the target object, but also ensures the normal use of linear storage acceleration technology. The strategy of scene image region division based on depth map is described as follows.

Set the initial points at the same interval on the scene depth map, denoted as c_i, $i \in (1,2,\ldots,n)$, the size of n depends on the size of the scene image, and the radius of the circumscribing sphere of the target object is denoted as r. Taking each initial point as the center, spread the range of the surrounding area in a breadth-first search strategy. When the maximum depth value β_{max} and the minimum depth value β_{min} in the area satisfy Eq. (1), the diffusion is stopped. At this time, the area to which each initial point diffuses during the diffusion process is a divided scene image area, denoted as C_i, $i \in (1,2,\ldots,n)$.

$$\beta_{max} - \beta_{min} \geq 2r. \quad \ldots \quad (1)$$

In this way, the scene image is divided into several regions with approximately equal depths. The color image is divided in the same way according to the division result of the depth image. After the scene image is divided into several areas C_i with substantially the same depth, there is a lot of overlap between the image areas. If template matching is performed directly on these image areas, many repeated matching operations will be generated, which greatly reduces the speed of the algorithm. Therefore, in order to integrate each region block of the scene image, it is also necessary to merge the region blocks of the scene image. When merging scene image area blocks, first calculate the average depth of each scene area block as d^i_{avg}, attach each scene image area block to a depth layer that minimizes the difference between d^i_{avg} and D_i. Then the scene image blocks attached to the same depth layer are merged to form a new scene image block. In each new scene image area block, there may be multiple small area blocks that are not connected to each other. If the scene images are not connected, subsequent template matching operations cannot be performed. Therefore, it is necessary to separate the unconnected area blocks in each new scene image area. After separation, the final scene image area blocks are recorded as W_i, $i \in (1,2,\ldots,m)$. Each scene image area block corresponds to the depth layer to which it is attached, and its depth is represented by the depth of the depth layer. When template matching is carried out on each scene image area block later, the template under this depth can be directly called. The scene RGB image is divided in the same way according to the division result of the depth image.

3.3. Coarse Positioning of Objects

After the scene image is divided into blocks whose depth difference is within a certain range according to its depth value, there are many area blocks in which the target object obviously does not exist. Inspired by [22], this section proposes a filtering method based on depth edge detection, which can not only filter out the scene image block which obviously does not contain the target object, but also locate the possible position of the target object in the reserved scene image block.

In a certain area block W of the scene image, the detection points are set with a fixed step, each detection point is denoted as w_i, $i \in (1,2,\ldots,n)$, where n represents the number of detection points, and its size is determined by the size of the area block and the sampling step of the detection point, the actual depth of each detection point is denoted as z_i, $i \in (1,2,\ldots,n)$. According to the depth D corresponding to the area block W, the template of the target object trained at this depth is retrieved. The size l_i of the target object imaged at the actual depth of the detection point w_i can be obtained from Eq. (2):

$$l_i = f \cdot \frac{R}{z_i}, \quad \ldots \quad (2)$$

where f is the focal length of the camera and R is the diameter of the circumscribing sphere of the target object. When the detection point w_i is on the target object in the scene, l_i is just the side length of the largest 2D bounding box of the target object. Similarly, the size L of the template image with the depth D can be obtained from Eq. (3):

$$Ł = f \cdot \frac{R}{D}. \quad \ldots \quad (3)$$

The test image block with the detection point w_i as the center and l_i as the side length is scaled to the same size as the template in the proportional relationship of Eq. (4). The main purpose of this operation is to scale the imaging of the target object at the depth of each inspection point in the scene image to the size imaged by the template at the depth D, in preparation for the coarse positioning of the object. Although there are image zoom operations here, the depth values contained in the scene image area blocks are within a certain range, and the corresponding depths are not much different from the depth D of the called template, even if there is an image zoom operation, the scale is not too large, and the impact of image scaling can be ignored.

$$\frac{l_i}{L} = \frac{D}{z_i}. \quad \ldots \quad (4)$$

Table 2. The pseudo code for coarse positioning of an object.

Object coarse positioning
Input: the scene image area block W and its corresponding depth D, template T with depth D
Output: there may be a set $P\{\}$ of target object positions in area W
Parameter: Sampling step size of detection point s, R, η
$(w_1,w_2,\ldots,w_n)\leftarrow$(W, s) // Determine the detection points in the area block W
$(N_{max}, N_{min})\leftarrow$(T, η) // Calculate the maximum and minimum number of deep edges in the template
for i $\leftarrow$ 1 to n
$l_i \leftarrow (w_i, \mathrm{R})$ // Calculate the imaging size of the target object at the detection point
$w_i \leftarrow (l_i, D, w_i)$ // Zoom the image block at the detection point
Calculate the number of depth edges num_i of the image block
if $80\% \cdot N_{min} \leq num_i \leq 150\% \cdot N_{max}$ then
$P\{\} = P\{\} + r_i$ // r_i is the position of the detection point w_i
end for

After scaling the test image block at the detection point w_i to the same size as the template image, in order to simplify the symbol marking, the test image block corresponding to each detection point is still marked with w_i, $i \in (1,2,\ldots,n)$. The coarse positioning technique in this section is based on depth edge detection. In fact, the depth edge of the image is calculated by the Soble operator. The depth edge appears at pixels where the calculation result of the Soble operator is greater than the set threshold. In this paper, the threshold is set to $30\%\eta$, where η is the side length of the maximum bounding box of the target object. The maximum and minimum values of the number of depth edges of the template at depth D are calculated as N_{max} and N_{min}, respectively. If the number of depth edges num_i in the image block w_i satisfies the relationship (5), it indicates that there may be a target object at the detection point w_i.

$$80\% \cdot N_{min} \leq num_i \leq 150\% \cdot N_{max}. \quad \ldots\ldots \quad (5)$$

The detection points that satisfy the relationship (5) are called target detection points. In this paper, only the area blocks of the scene image that contain the most target detection points are retained, and the other area blocks are directly discarded. Since the position of the target detection point in the scene image is known, the possible position of the target object in the test image is also determined. Subsequently, the template matching operation will only be performed at the target detection point in the reserved scene image area block. Avoid a lot of useless matching, thereby improving the speed of object recognition. Taking the scene image area block W as an example, the pseudo code for coarse positioning of an object is shown in **Table 2**.

3.4. Invocation Method of Multi-Scale Template

After the coarse positioning technology of the object, the target detection point in the scene image can be located, and only the test image area block containing the most target detection points is retained. However, the target object may exist at the detection point of the target, so it is necessary to accurately identify the object through template matching later. When accurate object recognition is performed, the template matching operation will be performed only at the target detection points of the remaining scene image area blocks to identify the target object. When performing template matching at the target detection point, the template trained at the depth can be retrieved according to the depth of the depth layer corresponding to the scene image area block, and template matching is only performed with the template trained at the depth, without traversal all templates of target objects, which will greatly improve the speed of the algorithm. Therefore, the template invocation strategy based on scene-patch enables the LineMOD algorithm to be used in scenes with a wide range of depth changes of target objects. In the case that the depth of the template is always similar to the depth of the target object in the scene, the target object in the scene can be quickly identified, so as to solve the problem that the LineMOD algorithm is sensitive to the depth of the template. The flow of the template invocation method based on Scene-Patch is shown in **Fig. 3**.

4. Experiments

4.1. Introduction to the Data Set

In this paper, we will test the improved strategy on the LineMOD dataset. This dataset was created by Hinterstoisser et al. and has become the most commonly used dataset for evaluating the performance of object pose estimation algorithms [23]. The LineMOD data set contains 15 kinds of 3D models of non-textured objects. During the experiment, the template images can be directly ren-

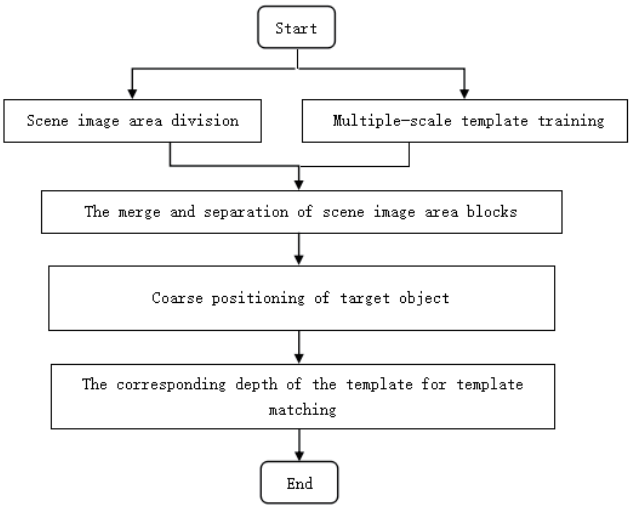

Fig. 3. Flowchart of template invocation method based on Scene-Patch.

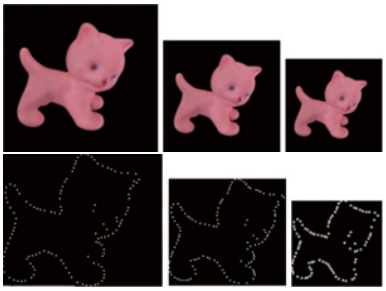

Fig. 4. Template images at different depths and their corresponding templates.

dered from the 3D models of these objects, and then the templates are trained from the template images. In addition, each object in the data set has more than a thousand test image sequences, and each test image carries information such as the pose of the target object in the test image, the distance between the target object and the camera, and camera parameters. The distance between the target object and the camera in the test image is called the depth of the target object in the test image, and the depth of the target object in the test image of this data set varies from 65–115 cm.

4.2. Multi-Scale Template Training

The biggest difference between this paper and the original LineMOD algorithm when training the template is the different step sizes of depth change. This paper uses the radius of the circumscribed spherel of the target object as the depth change step, and the original LineMOD algorithm takes a fixed value of 0.1 m as the depth change step. The template images and corresponding templates of toy cats in the LineMOD data set collected at different depths are shown in **Fig. 4**. The first three are template images collected at different depths, while the last three are templates trained by their own template images. The white dots represent feature points in the template. The other parameter settings when training the template are respectively:the azimuth step length when acquiring the template image is 15°, the pitch angle range is −45°–45°, the step length is 10°, and the rotation range in the plane is −45°–45°, the rotation angle step is 10°. Since the original LineMOD algorithm uses a fixed depth step when training templates, the number of templates trained for each object is 11,664. It can be seen from **Table 3** that the number of templates of most objects trained in this paper is greater than the number of templates trained by the original LineMOD algorithm. As can be seen from **Table 3**, the improved LineMOD algorithm can effectively solve the depth sensitivity problem, and the object recognition rate has a certain improvement compared with the original algorithm.

4.3. Test Results and Analysis

This paper first tests the accuracy of the improved LineMOD algorithm for object pose estimation. In the experiment of object pose estimation based on LineMOD algorithm, it is necessary to determine the method to evaluate whether the estimated pose is correct or not. This paper uses the evaluation method proposed in [18] to judge whether a pose estimation result is correct.

Suppose an object model M, which is composed of points n, denoted as $\{m_1,\ldots,m_n\}$. The object pose estimated by the algorithm is denoted as (R,t), and the real pose of the target object in the scene image is denoted as $(\bar{R},\bar{t})$, then the calculation formula of accuracy s of estimation pose is shown in Eq. (6).

$$s = avg\|(R\cdot m+t)-(\bar{R}\cdot m+\bar{t})\|. \quad (6)$$

As for the target object with symmetrical geometric structure or rotating structure, the same template image may be obtained from different perspectives, so the calculation method of accuracy s, of pose estimation needs to be modified, as shown in Eq. (7).

$$s = avg\min\|(R\cdot m_1+t)-(\bar{R}\cdot m_2+\bar{t})\|. \quad (7)$$

If the accuracy of pose estimation satisfies Eq. (8), it indicates that pose estimation is correct.

$$k_m\cdot d > s. \quad (8)$$

In Eq. (8), k_m is the control coefficient, which is set as 0.1 in this paper, and is the diameter of the target object's packet catching ball.

Finally, the accuracy of pose estimation of the target object is taken as the evaluation standard. The accuracy of pose estimation is the ratio between the correct number of test images estimated by the target object and all the test images of the target object. The improved LineMOD algorithm visualizes the pose estimation of the target object in some test images in the data set as shown in **Fig. 5**, where the calculated pose of the target object in each picture is expressed in the form of a three-dimensional coordinate axis.

We compare our method to two state-of-the-art meth-

Table 3. The original LineMOD, Tejani et al. [24] and the improved LineMOD were used to recognize objects in the LineMOD data set. The recognition rate (%) is as follows.

Target object	The Number of depths			The total number of template			The recognition rate		
	Original LineMOD	Tejani et al. [24]	Improved LineMOD	Original LineMOD	Tejani et al. [24]	Improved LineMOD	Original LineMOD	Tejani et al. [24]	Improved LineMOD
Little monkey	6	5	11	11664	11664	21384	98.5	97.6	98.6
Vice	6	5	5	11664	11664	9720	99.1	99.1	99.3
Driller	6	5	5	11664	11664	9720	98.2	97.9	98.1
Camera	6	5	7	11664	11664	13608	99.3	99.0	99.5
Watering can	6	5	6	11664	11664	11664	98.7	99.0	98.9
Electric iron	6	5	5	11664	11664	9720	98.3	98.7	99.0
Table lamp	6	5	5	11664	11664	9720	99.0	98.9	98.6
Telephone	6	5	6	11664	11664	11664	97.2	98.2	98.4
Toy cat	6	5	7	11664	11664	13608	99.5	99.4	99.6
Hole puncher	6	5	7	11664	11664	13608	97.6	98.5	98.3
Toy duck	6	5	10	11664	11664	19440	98.1	98.1	98.0
Drinking glass	6	5	9	11664	11664	17496	98.6	97.9	97.5
Bowl	6	5	7	11664	11664	13608	99.8	98.9	98.6
Egg box	6	5	7	11664	11664	13608	99.6	98.8	99.8
Glue	6	5	7	11664	11664	13608	97.3	97.9	98.1

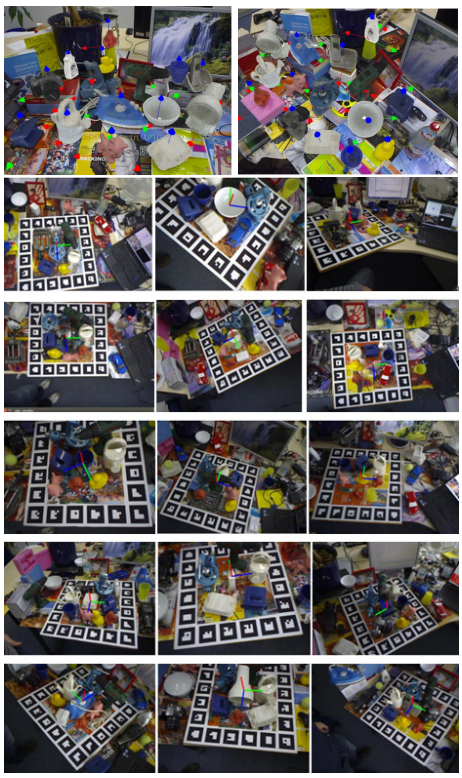

Fig. 5. Under the heavy clutter background of local occlusion, 15 textureless 3D objects with different attitudes were simultaneously detected by the LineMOD algorithm before and after the modification. Each detected object is augmented with its 3D model. We also showed the corresponding coordinate system.

ods, namely original LineMOD [18] and the method of Tejani et al. [24]. Tejani et al. [24] adapted the state-of-the-art template matching feature, LineMOD [18], into a scale-invariant patch descriptor and integrated it into aregression forest using a novel template-based split function. In training, rather than explicitly collecting representative negative samples, their method was trained on positive samples only and they treated the class distributions at the leaf nodes as latent variables. During the inference process they iteratively updated these distributions, providing accurate estimation of background clutter and foreground occlusions.

As shown in **Table 4**, the improved LineMOD algorithm has an average pose estimation accuracy rate of 95.8% on the LineMOD dataset, while the original LineMOD algorithm has an average pose estimation accuracy rate of 95.3% on the LineMOD dataset, the method of Tejani et al. [24] has an average pose estimation accuracy rate of 95.4% on the LineMOD dataset. The improved LineMOD algorithm improves the average pose estimation accuracy of the dataset by 0.5% compared with the original LineMOD algorithm and 0.4% compared with the method of Tejani et al. [24].

In terms of speed, in the environment where the computer hardware is configured with an Intel core i7 quad-core processor and 8G of running memory, the average time required for the original LineMOD algorithm to complete the object pose estimation of a test image of the LineMOD dataset is 0.2 s, the average time required for the improved LineMOD algorithm to complete the object pose estimation for a pair of test images in the LineMOD data set is 0.15 s, the average time required for the method

Table 4. The original LineMOD, Tejani et al. [24] and the improved LineMOD were used to estimate the pose of objects in the LineMOD data set. The accuracy rate (%) and time of the estimation are as follows.

	The accuracy rate (%)			The time (s)		
Target object (number of test images)	Original LineMOD	Tejani et al. [24]	Improved LineMOD	Original LineMOD	Tejani et al. [24]	Improved LineMOD
Little monkey (1235)	94.6	95.5	95.3	0.199	0.183	0.149
Vice (1214)	97.3	97.7	98.1	0.194	0.195	0.146
Driller (1187)	91.5	92.1	92.6	0.191	0.194	0.143
Camera (1200)	96.1	96.3	98.5	0.192	0.187	0.149
Watering can (1195)	95.6	94.6	96.7	0.191	0.186	0.146
Electric iron (1151)	95.9	96.1	98.4	0.189	0.175	0.138
Table lamp (1226)	94.5	95.5	94.3	0.202	0.155	0.147
Telephone (1224)	93.1	91.8	95.1	0.205	0.176	0.148
Toy cat (1178)	98.6	96.6	97.9	0.192	0.166	0.141
Hole puncher (1236)	93.2	95.8	92.6	0.211	0.193	0.155
Toy duck (1253)	94.5	96.1	96.0	0.215	0.172	0.158
Drinking glass (1239)	96.8	95.5	95.8	0.198	0.187	0.157
Bowl (1232)	99.0	98.6	94.8	0.203	0.192	0.148
Egg box (1252)	99.2	96.4	98.2	0.219	0.201	0.169
Glue(1219)	90.6	92.7	93.0	0.199	0.188	0.156
The average (18241)	95.3	95.4	95.8	0.2	0.18	0.15

of Tejani et al. [24] is 0.18 s. As can be seen we outperform both state-of-the-arts in both datasets. The depth of the target object in the test image of the LineMOD dataset is not constant, and the variation range is 65–115 cm. Therefore, it is necessary to train templates at various depths to ensure the accuracy of the algorithm. It can be seen from Section 4.2 that the improved LineMOD algorithm has trained more types of templates for most objects in the LineMOD data set. The number of templates trained is greater than the number of templates trained by the original LineMOD algorithm and the method of Tejani et al. [24]. When the number of templates increases, the improved LineMOD algorithm's pose estimation speed is faster than the other two methods. This is because the other two methods need to traverse all templates of the target object when they perform pose estimation for the target object, and the improved LineMOD algorithm can specifically call the template at the corresponding depth for template matching, which greatly improves the speed of the algorithm, so the speed of the improved LineMOD algorithm will not be affected when the depth of the target object template increases. Furthermore, when the depth of the target object in the scene varies in a large range and more templates need to be trained, the improved LineMOD algorithm can still quickly estimate the pose of the target object, thus solving the problem that the other two methods are sensitive to the depth of the template.

5. Conclusions

The LineMOD algorithm can realize the recognition and pose estimation of texture-less objects in a messy background. It can adapt to the needs of different scenes by adding different templates, and has the advantages of fast speed and high precision. At present, it is still widely used in the field of pose estimation of texture-less objects in industry. However, this algorithm has the problem of depth sensitivity to the template, which makes it difficult to achieve satisfactory results in some special industrial scenarios. This paper proposes corresponding improvement schemes for the defects of the LineMOD algorithm, improves the template invocation method of the algorithm, and proposes a template invocation strategy based on Scene-Patch, which can make the LineMOD algorithm in scenes where the depth of the target object varies widely and the depth of the target object in the scene is always the same as the depth of the template, the target object can still be quickly and accurately identified. Experiments show that this strategy can solve the problem of LineMOD algorithm's sensitivity to the depth of the template.

References:

[1] T. P. Caudell and D. W. Mizell, "Augmented reality: An application of heads-up display technology to manual manufacturing processes," Proc. of the 25th Hawaii Int. Conf. on System Sciences, Vol.2, pp. 659-669, 1992.

[2] Y. Guo, M. Bennamoun, F. Sohel et al., "A Comprehensive Performance Evaluation of 3D Local Feature Descriptors," Int. J. of Computer Vision, Vol.116, No.1, pp. 66-89, 2016.

[3] R. B. Rusu, G. Bradski, R. Thibaux et al., "Fast 3D recognition and pose using the Viewpoint Feature Histogram," Int. Conf. on Intelligent Robots and Systems, pp. 2155-2162, 2010.
[4] E Brachmann, A Krull, F. Michel et al., "Learning 6D Object Pose Estimation Using 3D Object Coordinates," European Conf. on Computer Vision, pp. 536-551, 2014.
[5] S. Hinterstoisser, S. Holzer, C. Cagniart et al., "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," Int. Conf. on Computer Vision, pp. 858-865, 2011.
[6] D. G. Lowe, "Object recognition from local scale-invariant features," Proc. of Int. Conf. on Computer Vision, Vol.2, pp. 1150-1157, 1999.
[7] R. Sun, J. Qian, R. H. Jose et al., "A Flexible and Efficient Real-Time ORB-Based Full-HD Image Feature Extraction Accelerator," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, Vol.28, No.2, pp. 565-575, 2020.
[8] E. Rublee, V. Rabaud, K. Konolige et al., "ORB: An efficient alternative to SIFT or SURF," Int. Conf. on Computer Vision, pp. 2564-2571, 2011.
[9] Y. Ren, C. Zhu, and S. Xiao, "Object Detection Based on Fast/Faster RCNN Employing Fully Convolutional Architectures," Mathematical Problems in Engineering, Vol.2018, Article ID 3598316, 2018.
[10] S. Ren, K. He, R. Girshick et al., "Faster R-CNN: Towards real-time object detection with region proposal network," Advances in Neural Information Processing Systems, Vol.39, No.6, pp. 91-99, 2015.
[11] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 6517-6525, 2017.
[12] J. Redmon, S. Divvala, R. Girshick et al., "You only look once: Unified, real-time object detection," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 779-788, 2016.
[13] W. Liu, D. Anguelov, D. Erhan et al., "SSD: Single shot multibox detecton," European Conf. on Computer Vision, pp. 21-37, 2016.
[14] W. Kehl, F. Manhardt, F. Tombari et al., "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," Proc. of the IEEE Int. Conf. on Computer Vision, pp. 1530-1538, 2017.
[15] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition, pp. 292-301, 2018.
[16] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," Int. J. of Computer Vision, Vol.81, No.2, Article No.155, 2009.
[17] B. Drost, M. Ulrich, N. Navab et al., "Model globally, match locally: Efficient and robust 3D object recognition," IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, pp. 998-1005, 2010.
[18] S. Hinterstoisser, C. Cagniart, S. Ilic et al., "Gradient response maps for real-time detection of textureless objects," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.34, No.5, pp. 876-888, 2012.
[19] J. Lee, M. Lee, S. Kang et al., "Real-time 3D Pose Estimation of Small Ring-Shaped Bin-Picking Objects Using Deep Learning and ICP Algorithm," J. of Institute of Control Robotics and Systems, Vol.25, No.9, pp. 760-769, 2019.
[20] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," CVPR 2011, pp. 617-624, 2011.
[21] G. Fanelli, M. Dantone, J. Gall et al., "Random Forests for Real Time 3D Face Analysis," Int. J. Comput. Vis., Vol.101, No.3, pp. 437-458, 2013.
[22] H. Zhang and Q. Cao, "Texture-less object detection and 6D pose estimation in RGB-D images," Robotics and Autonomous Systems, Vol.95, pp. 64-79, 2017.
[23] A. Aldoma, F. Tombari, R. B. Rusu et al., "OUR-CVFH – Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation," Joint 34th DAGM and 36th OAGM Symp. Proc., pp. 113-122, 2012.
[24] A. Tejani, D. Tang, R Kouskouridas et al., "Latent-class hough forests for 3D object detection and pose estimation," European Conf. on Computer Vision, pp. 462-477, 2014.

Name:
Jian Peng

Affiliation:
School of Automation, China University of Geosciences

Address:
388 Lumo Road, Hongshan, Wuhan, Hubei 430074, China

Brief Biographical History:
1982-1986 Bachelor Student in Industrial Automation, Wuhan University of Technology
1988-1991 Postgraduate/Master Student in Industrial Automation, Huazhong University of Science and Technology
1986-1988 Automatic Control Technician, Wuhan Changjiang Cable Power Plant
1991-1993 Electrical and Electronic Engineer, China Electronics Import and Export Hubei Company
1993-1997 Lecturer, China University of Geosciences
1997-2002 IT R & D Project Manager, Huawei Technology Co., Ltd.
2002- Associate Professor, China University of Geosciences
2014- Visiting Professor, Huanggang Normal University

Main Works:
• "Modelling saliency attention to predict eye direction by topological structure and earth mover's distance," PLOS ONE, Vol.12, No.7, 2017.
• "Extraction of most important influencing factors of cement energy consumption," 5th Int. Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII 2017), 2017.
• "Class Socialn Leaming Particle Swarm Optimization Algorithm," Proc. of the 37th Chinese Control Conf. (CCC), 2018.

Membership in Academic Societies:
• Sensor Journal, Evaluation Expert

Name:
Ya Su

Affiliation:
School of Automation, China University of Geosciences

Address:
388 Lumo Road, Hongshan, Wuhan, Hubei 430074, China

Brief Biographical History:
2019 B.S. degree from Qufu Normal University
2019- Master degree Candidate, China University of Geosciences