IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 7, 2020, accepted September 20, 2020, date of publication September 29, 2020,
date of current version October 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027708

A Hybrid Framework Based on Warped Hierarchical Tree for Pose Estimation of Texture-Less Objects

YONGQI GUO, JIANLIN WANG, XINJIE ZHOU, ZHENGUO TAN, AND KEPENG QIU
College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

Corresponding author: Jianlin Wang (wangjl@mail.buct.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2018YFC0808904.

ABSTRACT The pose of texture-less objects is very important for intelligent manufacturing and intelligent assembly. Existing methods cannot accurately estimate pose when partial features are missing or cluttered due to shading, reflection, and occlusion. We propose a hybrid framework based on the warped hierarchical tree for pose estimation which integrates template matching and sparse representation classification in this paper. Firstly, the template is formed by the prospectively cumulated orientation feature (PCOF), which is a probabilistic representation of orientations extracted from template images. And the warped hierarchical tree can be built offline according to the parameters for projecting the 3D object and the similarity between templates. Then the online searching can be repeated through the warped hierarchical tree until the template candidates have been found. Finally, the pose corresponding to the best-fitting template can be obtained by sparse representation classification based on the dictionaries consisted of the spreading orientations of template candidate images. The experiment results show the effectiveness of our method when partial features are missing or cluttered.

INDEX TERMS Pose estimation, template matching, sparse representation classification.

I. INTRODUCTION

Pose estimation is one of the hottest subjects in the field of computer vision, which is crucial for intelligent manufacturing and intelligent assembly [1]. The objects in intelligent manufacturing and intelligent assembly usually have shiny, highly reflective, and texture-less surfaces. The pose estimation of these texture-less objects has drawn attention due to their specificity [2], [3].

Recently, the RGB-D cameras have been used for pose estimation, but it is not suitable for texture-less objects. Because the shiny and highly reflective surfaces affect the measurement of depth [4]–[6]. Besides, the multocular cameras can be used for estimating the pose of texture objects [7], [8], but it is difficult to extract and match feature points accurately from the multiple images of texture-less objects [9]–[11].

The methods using monocular camera for pose estimation include CNN-based methods, geometry-based methods, and template-based methods. CNN-based methods use convolutional neural networks (CNN) to extract features and describe the texture-less objects by the features in different convolutional layers, but the network needs to be retrained using the new dataset when new objects are added [12]. Geometric features such as points, lines, and circles have excellent quality due to its invariance to scale, rotation, and illuminations. But the geometry-based methods rely on the quality of feature extraction and matching. Extracting and matching features is difficult because the features will change when the posture change [13]–[15]. Template-based methods are performed to obtain the pose corresponding to best-fitting template based on the similarity between the sub-block of test image and templates. The methods have high running speed and precision. Besides, learning new objects does not need to retrain, only requires extracting and storing features of the new objects, which is very convenient for practical applications. However, it cannot work well when partial features are missing or cluttered due to shading, reflection, and occlusion in local areas. Generally, the similarity between the sub-block of test image and templates will decrease with increasing the ratio of missing or cluttered features. Then the similarity

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenzhou Tang.

between different templates may not be significantly different. That will lead to wrong matching [16]–[18].

So we propose a hybrid framework which integrates template matching and sparse representation classification based on the warped hierarchical tree for pose estimation of texture-less objects in this paper. The main contributions are as follows: (1) We build the warped hierarchical tree based on the parameters for projecting the 3D object and similarity between templates to accelerate the search online. The warped hierarchical tree compared with the hierarchical pose tree [31] built just based on similarity is more reasonable and not easily trapped in a local optimum. Besides, the technique of building warped hierarchical tree with parameter constraint can greatly reduce training time. (2) A new framework integrating template matching and sparse representation classification based on the warped hierarchical tree for pose estimation of texture-less objects is proposed. The framework can work well when partial features are missing or cluttered due to sparse representation classification.

The paper is organized as follows. A total review of related work on pose estimation using monocular camera is given in section II. The warped hierarchical tree and the hybrid framework for pose estimation algorithm are introduced in section III. The proposed method is evaluated in section IV. The conclusion of the paper is given in section V.

II. RELATED WORK

The methods using monocular camera for pose estimation can be summarized in three classes.

A. CNN-BASED METHODS

The features extracted by CNN can comprehensively describe the objects compared to manual features. There are emerging algorithms for object detection such as Faster-RCNN, YOLO, SDD. They have achieved huge success in various fields. Some methods using CNN have been presented for 6D pose estimation. SSD-6D [19] was trained on synthetic data, which consists of an SSD network and autoencoder network. The method showed that the color information alone can already achieve a good detection rate. YOLO-6D [20] utilized a single-shot CNN architecture to predict the 3D-bounding box and the object's class without additional post-processing. Posenet [21] was a robust and fast CNN architecture by modifying GoogLeNet, which regresses the 6-DOF camera pose with no need of additional engineering or graph optimization. Crivellaro *et al.* [22] utilized CNN to detect parts and predicts the projections of the control points to estimate pose. These methods based on CNN are very fast on the graphics processing unit (GPU), but the weights of network describing the objects need to update when new objects are added.

B. GEOMETRY-BASED METHODS

Geometry features such as the corner points, lines, and curves on the contour are used to estimate the pose based on the 3D features and 2D features. Lepetit *et al.* [23] used four virtual control points to express the 3D points and estimated the coordinates of the virtual control points. Pribyl *et al.* [24] used the direct linear transformation to solve the Perspective-n-Line (PnL) problem. The redundant 3D points and 3D lines for expressing the 3D structure can reduce the minimum of required lines. For larger line sets, the method has high accuracy and small reprojection error. Meng *et al.* [25] utilized the perspective view of circle and line in the image to estimate the pose. The false pose was identified by re-projecting the random point on the line. However, these methods are fragile for cluttered background. The corner point, line, and curve features in background will interfere with feature matching. Then false correspondences will lead to wrong pose parameter [26], [27].

C. TEMPLATE-BASED METHODS

The template images captured from different viewpoints can represent the 3D structure of the object. The pose corresponding to the best-fitting template can be obtained by searching in the template set. Note that the accuracy of these methods heavily depends on the number of templates. Tombari [18] proposed the object recognition approach using the sum of the normalized dot products to find the best-fitting template, which has the maxima of the similarity in the transformation space. Hinterstoisser *et al.* [28], [29] proposed the LINE-2D and LINE-MOD. LINE-MOD used complementary depth information compared with LINE-2D. They were based on the spread orientations which are robust to the background clutter and illumination change than gradient directions. Besides, the methods were accelerated by precomputing response maps and linearizing the memory. However, the spread orientation cannot deal with the feature changes caused by posture changes. Ulrich *et al.* [30] used the sum of the normalized dot products as similarity and built the hierarchical model by merging the views with high similarity to accelerate the exhaustive search. But the hierarchical model is easily trapped in a local optimum. Because they only consider the similarity between templates. Ren *et al.* [31] used structural symmetry and context constraint as prior-knowledge to estimate pose. The object was represented as combination of sub-objects. Then the pose estimation was implemented by the fitting algorithm. Zhang *et al.* [32] utlized the similarity of the input image and template images to find the most similar subset. Then they established 2D-3D correspondences by feature matching and performed pose estimation. Konishi *et al.* [33] proposed the perspectively cumulated orientation feature (PCOF) and hierarchical pose tree for pose estimation. PCOF is a probabilistic representation of orientations extracted from template images, which is not sensitive to the feature change caused by posture change. And the hierarchical pose tree built by clustering and downsampling can accelerate the exhaustive search. However, it cannot work well when partial features are missing or cluttered due to shading, reflection, and occlusion. Besides, the clustering approach is easily trapped in a local optimum, because they does not utilize the prior information of the template. And

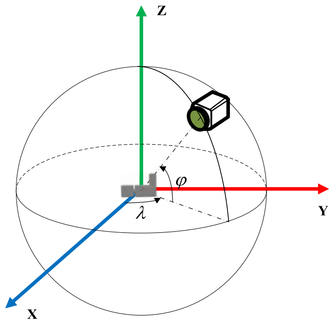


FIGURE 1. Spherical coordinate system fro projecting the CAD model. The viewpoint can be determined by the longitude, the latitude, the distance, and the rotation angle around the optical axis.

the training of hierarchical pose trees based on clustering and downsampling needs much computation and time.

III. PROPOSED METHOD

A. SIMILARITY MEASURE

The similarity is used for evaluating the fitting degree of the sub-block of test image and the templates in the template set. The template images can be generated by projecting the CAD model into the image plane. The spherical coordinate system for projecting the 3D object is shown as Figure 1. The viewpoint for projecting the CAD model is determined by four parameters those are the longitude (λ), the latitude (φ), the rotation angle (γ) around the optical axis, and the distance (d). To make the discrete template robust to feature changes caused by posture changes, the PCOF [33] which is a probabilistic representation of quantized gradient directions of template images is presented.

Assuming the parameters of the central viewpoint for projecting the CAD model to produce the template T are $p = \{\lambda, \varphi, \gamma, d\}$ and parameters of sub-viewpoints for computing the template T are $p+\Delta$. Konishi *et al.* [33] used the randomized sub-viewpoints within the defined range to generate template images. But the randomized sub-viewpoints are redundant, and it will lead to a lot of calculations. So we choose the sub-viewpoints whose parameters are at fixed intervals. In our research, the range of Δ were ±12 degrees for the longitude and latitude, ±10 degrees for the rotation angle and ±40 mm for distance. The parameters of sub-viewpoints are with longitude step of 2°, latitude step of 2°, rotation step of 2°, and distance step of 10mm. We can generate 720 template images for the template T.

The RGB channel values of each face of CAD model are set to its normal vector. The gradients are computed using CANNY operators on three channels separately. The maximum gradient which exceeds a given threshold in RGB channel is used.

$$C(\mathrm{x}) = \underset{C\in\{R,G,B\}}{\arg\max} \left\| \frac{\partial C}{\partial \mathrm{x}} \right\| > t \tag{1}$$

where R, G, and B are the RGB channels, $\mathrm{x} = [x, y]$ is the coordinate in image coordinate system.

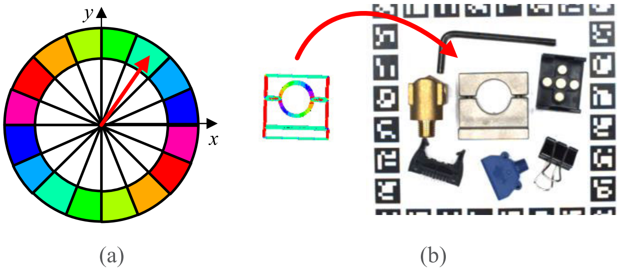


FIGURE 2. (a) Colored quantized gradient direction. (b) The similarity in between the sub-block of the test image and the template.

The gradient amplitude is determined by the angle between the normal vectors of neighboring faces. So the threshold t can be computed by the minimum face angle, and the quantized gradient directions are used as features discarding gradient magnitudes. The quantized gradient direction named as orientation is disregarded its polarities as Figure 2(a). The orientation histogram in each pixel is built by voting from the orientation of all the template images. Then the dominant orientations extracted from the histograms are represented by 8-bit binary number. In our researcher, the threshold for extracting domain orientations was 90. We use the maximum frequency of the orientation histograms in each pixel as weights to calculate the similarity.

The template T is represented as follows.

$$T : \{x_j, y_j, ori_j, w_j | j = 1, \cdots, m\} \tag{2}$$

where m is the number of PCOF in the template T. x_j, y_j, ori_j, and w_j are x-coordinate, y-coordinate, quantized gradient direction and weight of the jth PCOF in the template T.

The similarity score between the subblock of test image I and the template T is given by the following equation.

$$S = \frac{\sum_{j=1}^{m} \delta(ori^{I}_{(x+x_j, y+y_j)} \in ori^{T}_{j})}{\sum_{j=1}^{m} w_j} \tag{3}$$

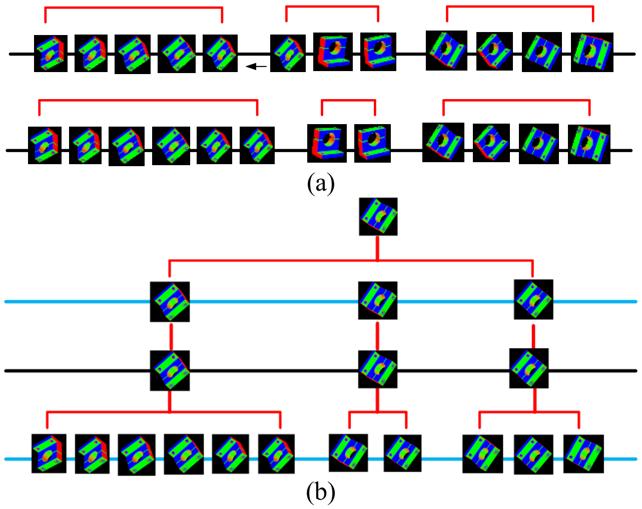
where m is the number of PCOF. x and y are the coordinates of the top left corner of the sub-block in the test image. x_i and y_i are the coordinates of jth PCOF in template. The delta function is defined as equation (4). If the orientation in the template T includes the orientation in the test image I as Figure 2(b), the weights which are the maximum frequencies of the histograms are added to the score. The similarity score is the ratio of weights of the matching features to total weights.

$$\delta(ori^{I} \in ori^{T}_{j}) = \begin{cases} w_j & \text{if } ori^{I} \wedge ori^{T} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\wedge$ represents the bitwise AND operation.

B. BWARPED HIERARCHICAL TREES

The runtime complexity of the exhaustive search is linearly dependent on the number of templates. The hierarchical tree is the key to accelerating exhaustive search.


(a)

(b)

FIGURE 3. (a) Updating to the neighboring cluster. (b) Part of warped hierarchical tree.

Algorithm 1 Building Warped Hierarchical Tree

```
Input: templates set $X_0$, the parameters of central viewpoint for each template
Output: warped hierarchical tree
  Initialize the hierarchical level $k \leftarrow 1$
  loop
    Initialize the cluster center $X_k$ with parameter constraint
    for each template do
      Computer the similarity with the cluster center
      Update the cluster with parameter constraint
    end for
    for each cluster center do
      $X'_k \leftarrow$ downsampling the cluster center $X_k$
    end for
      $N'_k \leftarrow$ the minimum number of feature point
    if $N'_k > N_{\min}$ then
      $X_{k++} \leftarrow X'_k$
    else
      break
    end if
  end loop
```

Warped K-means [34] is a clustering procedure with sequence constraints. It is applied to the clustering of sequential data generated from motion sensors, eye trackers, and e-pens. The sequential information reflects the relationship between data. Clustering based on sequence information can quickly converge to a good local minimum due to checking neighboring clusters in each step. In our application, the parameters of the central viewpoint for each template also reflect the relationship between templates. Therefore, we use this idea to build the warped hierarchical tree with parameter constraint.

Assuming the template set can be represented as $X_0 = \{T_1, \cdots, T_n\}$ and the parameters corresponding to template T_i is $p_i = \{\lambda_i, \varphi_i, \gamma_i, d_i\}$. The four parameters can be regarded as the four dimensions in the 4D space. Each template is a single point in the 4D space. Neighboring points have similar parameters. And the neighboring templates have high similarity. So we choose the central point as an initial cluster center, and the neighboring points are classified into the cluster. Note that 180° and $-180°$ are the same points. The initial cluster determined by the parameter constraint is conducive to convergence. And the template is only allowed to move to neighboring clusters as Figure 3(a). And the cluster center is the template with the smallest maximum similarity to the other templates in each cluster.

Except for clustering with parameter constraint, the scale-space effects should also be taken into account. The image pyramid building by downsampling can result in higher robustness and speed. Therefore, our warped hierarchical pose can be built by clustering and downsampling. The process of building the warped hierarchical tree is shown in algorithm 1. Firstly, we initialize the cluster center with parameter constraints in 4D space. Secondly, the clusters and centers are updated based on the similarity. Then the cluster centers are downsampled to get new templates in different resolutions. We repeat the steps until the minimum number of feature points is less than the predefined threshold. Part of the warped hierarchical tree is shown in Figure 3(b). The blue horizontal line represents the level obtained by downsampling. The dark horizontal line represents the level obtained by clustering.

C. SPARSE REPRESENTATION CLASSIFICATION

The best-fitting template can be found by scanning the warped hierarchical tree when features are complete. However, the similarity between the sub-block of test image and templates will decrease with increasing of the ratio of feature loss and clutter, which will lead to wrong matching. The sparse representation which assumes the test sample can be linearly represented by the dictionary is robust to noise, clutter, and occlusion. It has been widely used in face recognition and object tracking because it is robust to feature loss and clutter [35], [36]. The orientation can reflect the relationships between neighboring pixels points and the underlying inherent structure of images. So we use sparse representation classification based on orientation to find the best-fitting template.

Assuming the atom of the dictionary can be represented as $a_{i,k} \in R^m$ and the test sample can be represented as $u \in R^m$. The dictionary can be represented as

$$A = [A_1, \cdots, A_h] = [a_{1,1}, a_{1,2}, \cdots, a_{h,n}] \in R^{m \times n} \quad (5)$$

where h is the number of classes of template candidate images. n is the number of template candidate images. m is the dimension of the atom.

The dictionary and the test sample are both transformed from two-dimensional matrices of the orientation of the template candidate images and sub-block of the test image.

We generate 16 atoms by spreading the orientation to neighboring ±2 and ±4 pixels in the X direction and Y direction for each class of template candidate image to build dictionary. To increase the robustness, we build an extended dictionary $B \in R^{m \times m}$, which is an m-dimensional identity matrix. Then the test sample can be linearly represented with A and B.

$$u = [A, B] \begin{bmatrix} \beta \\ c \end{bmatrix} + z \tag{6}$$

where $\beta = [\beta_1, \cdots, \beta_n]$ and $c = [c_1, \cdots, c_m]$ are the coefficients. The coefficients β_i reflect the degree of fitting with the ith template candidate image. The coefficients c reflect the degree of feature loss and clutter. z is the noise term.

To make the equation (6) undetermined, we utilize random projection [37] for dimension reduction. The elements $p_{i,j}$ of projection matrices P are defined as

$$p_{i,j} = \sqrt{s} \cdot \begin{cases} +1 & \text{with probability } 1/(2s) \\ 0 & \text{with probability } 1 - 1/s \\ -1 & \text{with probability } 1/(2s) \end{cases} \tag{7}$$

where s is the random value between 2 and 4.

The equation (6) can be expressed as

$$Pu = P[A_i, B] \begin{bmatrix} \beta_i \\ c \end{bmatrix} + z \tag{8}$$

The equation (8) can be solved by l^1 minimization.

$$\begin{bmatrix} \hat{\beta} \\ \hat{c} \end{bmatrix} = \arg \min_{\beta} \left\| \begin{bmatrix} \beta \\ c \end{bmatrix} \right\|_1 \quad \text{s.t.} \left\| Pu - P[A, B] \begin{bmatrix} \beta \\ c \end{bmatrix} \right\| \leq \varepsilon \tag{9}$$

where ε is the optimal error tolerance.

Then the best-fitting template can be directly determined by the sparse coefficients as follows.

$$class = \arg \min_{i=1,\cdots,h} r_i(u) = \arg \min_{i=1,\cdots,h} \left\| Pu - P[A, B] \begin{bmatrix} \hat{\beta}_i \\ \hat{c} \end{bmatrix} \right\|_2 \tag{10}$$

where $\hat{\beta}_i$ is the vector by setting the coefficients as zero expect for ith class. $r_i(u)$ is the difference between the original test sample and the reconstructed test sample.

D. HYBRID FRAMEWORK FOR POSE ESTIMATION

To improve the accuracy of pose estimation when partial features are missing or cluttered due to shading, we adopt a hybrid framework which integrates template matching and sparse representation classification for pose estimation. The region of object in image and the template candidate images can be got by template matching. Then sparse representation classification is used to find the best-fitting template based on the dictionary consisted of the spread orientation of template candidate images.

The difference between the templates is big in the high levels of warped hierarchical tree, and the difference between the templates is small in the low levels of warped hierarchical tree. when partial features are missing or cluttered due to shading, the wrong matching will occur in the low levels. So we use sparse representation classification to find the best-fitting template in the lowest level. Firstly, we extract the orientation features of the test image and build the pyramid. Secondly, template candidates can be got by scanning the warped hierarchical tree to the bottom. Then we can build the dictionary by spreading the orientation of template candidates in the X direction and Y direction. Finally, the best-fitting template can be directly determined by the sparse coefficients. The algorithm flow chart of our method is shown in Figure 4. The 6D object pose can be calculated by minimizing the distance between the edge points in the test image and edge points in the template image.

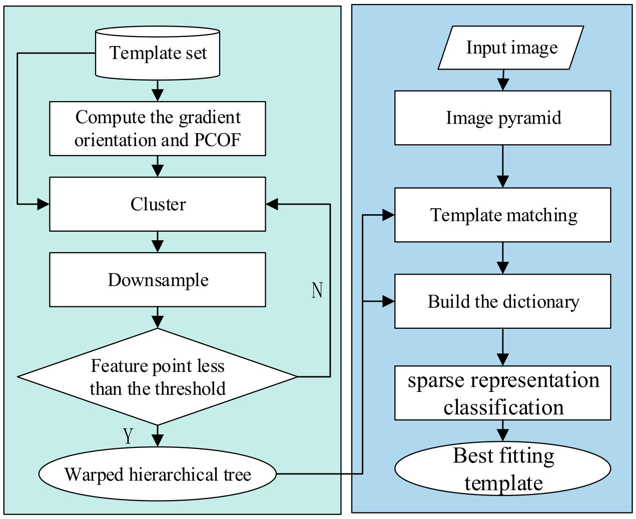


FIGURE 4. The flow chart of the hybrid framework.

IV. EXPERIMENTS

A. EXPERIMENTAL SETTING

We used the dataset made by Konishi and the dataset made by Hinterstoisser to evaluate our method.

The dataset made by Konishi consisted of nine texture-less metal objects. The images in the dataset were captured using monocular camera from viewpoints within the range of −60° ∼60° longitude and latitude, −180° ∼180° rotation and 660mm∼800mm distance. The resolution of these images is 640 × 480. Each object which is in cluttered backgrounds has approximately 500 images. Besides, The pose information of the object estimated based on the AR markers was given in the dataset [38].

The dataset made by Hinterstoisser consisted of fifteen texture-less objects. The images in the dataset were captured using RGB-D camera from viewpoints within the range of 0° ∼360° longitude, 0° ∼90° latitude, −45° ∼45° rotation and 650mm∼1150mm distance. We only used the color images whose resolution is 640 × 480. The ground truth pose estimated by the markers in the image was given in the dataset.

The existing methods proposed by Hinterstoisser *et al.* [29], Ulrich *et al.* [30], and Konishi *et al.* [33] were compared with our method on the dataset. All the programs were run on a PC with 64G RAM and Xeon E5-2630 2.60GHz CPU and developed using C++ language on the Visual Studio 2017 platform.

The range of the parameter corresponding to templates was same as the range of objects in the dataset. For objects from dataset made by Konishi, we generated 38025 templates from various viewpoints with longitude step of 10°, latitude step of 10°, rotation step of 12°, and distance step of 30mm. There were 3380 templates after first clustering, 383 templates after second clustering, and 23 templates after third clustering. For objects from dataset made by Hinterstoisser, we generated 38880 templates from various viewpoints with longitude step of 10°, latitude step of 10°, rotation step of 12°, and distance step of 30mm. There were 3000 templates after first clustering, 288 templates after second clustering, and 32 templates after third clustering.

B. SPEED

The runtime of train and search are independent of the structure of the object. They are related to the number of templates, the size of templates and the type of features. They are both critical for practical applications.

Although the hierarchical tree is constructed offline, the long training time is still a problem that cannot be ignored. The complexity of clustering or merging determines the speed of building the hierarchical tree. Assuming that the number of templates is N and the number of clusters is k. For merging [30], each template needs to compute similarity with the other template, so the computational complexity of merging is $O(N(N-1))$. For X-means [33], the computational complexity is $O(N \log k)$. For our clustering algorithm, the template only needs to compute similarity with the neighboring clusters in 4D space. So the computational complexity of our clustering algorithm is $O(16N)$, and it is much less than merging and X-means.

For the runtime of the search, we randomly selected 10 images from the datatset made by Konishi and calculated their average processing time by different methods. The average processing times (ms) were shown in Table 1.

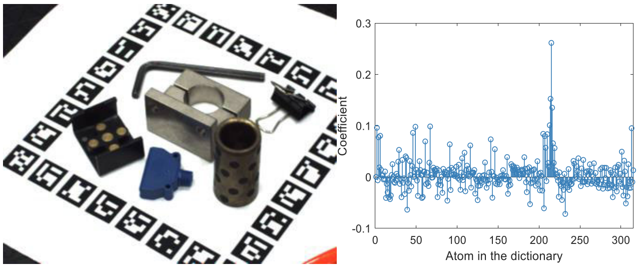
The search strategy of Hinterstoisser's method [29] is not effective for a large number of templates. But the similarity measure is optimized by precomputed responce maps, which is faster than the bitwise operation. Ulrich's method [30] used the floating-point arithmetic to measure similarity, so it is slower than other methods using bitwise operation or precomputed responce maps. Konishi's method [33] used the hierarchical pose tree to accelerate the search, so its speed is faster than Hinterstoisser's method. Our method which integrates template matching and sparse representation classification achieves a little bit slower than the Konishi's method [33].

TABLE 1. **The processing time.**

	Hinterstoisser[29]	Ulrich[30]	Konishi[33]	Ours
Bracket	416.3	838.4	**221.1**	273.2
Connector	412.3	1042.3	**198.1**	241.5
Flange	401.2	1024.7	**175.2**	252.4
HingeBase	432.5	1203.5	**190.1**	275.5
L-Holder	414.3	986.7	**169.4**	224.8
PoleClamp	408.3	1467.5	**153.4**	201.2
Sideclamp	402.5	2968.3	**225.8**	272.1
Stopper	456.4	2792.3	**152.5**	195.6
T-Holder	434.8	957.8	**156.7**	204.7

TABLE 2. **The success rate.**

	Hinterstoisser[29]	Ulrich[30]	PCOF[33]	Ours
Bracket	55.3%	57.7%	69.7%	**73.1%**
Connector	58.5%	60.5%	72.3%	**78.6%**
Flange	57.5%	63.4%	75.8%	**77.4%**
HingeBase	58.6%	60.1%	65.1%	**67.6%**
L-Holder	55.3%	62.1%	74.6%	**80.2%**
PoleClamp	55.7%	60.1%	77.8%	**79.8%**
SideClamp	44.8%	59.0%	74.3%	**75.7%**
Stopper	48.8%	58.6%	77.6%	**79.6%**
T-Holder	57.8%	57.4%	77.4%	**81.1%**



(a) (b)

FIGURE 5. **(a) The test images. (b) The coefficient of linearly represented by dictionary.**

C. ROBUSTNESS

For similarity measure, Hinterstoisser's method [29] used the summation of cosine based on spread orientation. Ulrich's method [30] used dot product based on gradient direction. But they will produce wrong matching in cluttered background and are not robust to the feature change caused by posture change. Just as the discussions mentioned in [33], PCOF compared the other similarity measure is robust to posture changes and cluttered background.

We showed the robustness of sparse representation classification using the example as Figure 5(a), in which partial features are missing or cluttered. Firstly, we got the candidate templates and position of the object in the test image by template matching. Then the dictionary was built by the spread orientation of the template candidate images. The sub-block of the test image was linearly represented by the dictionary

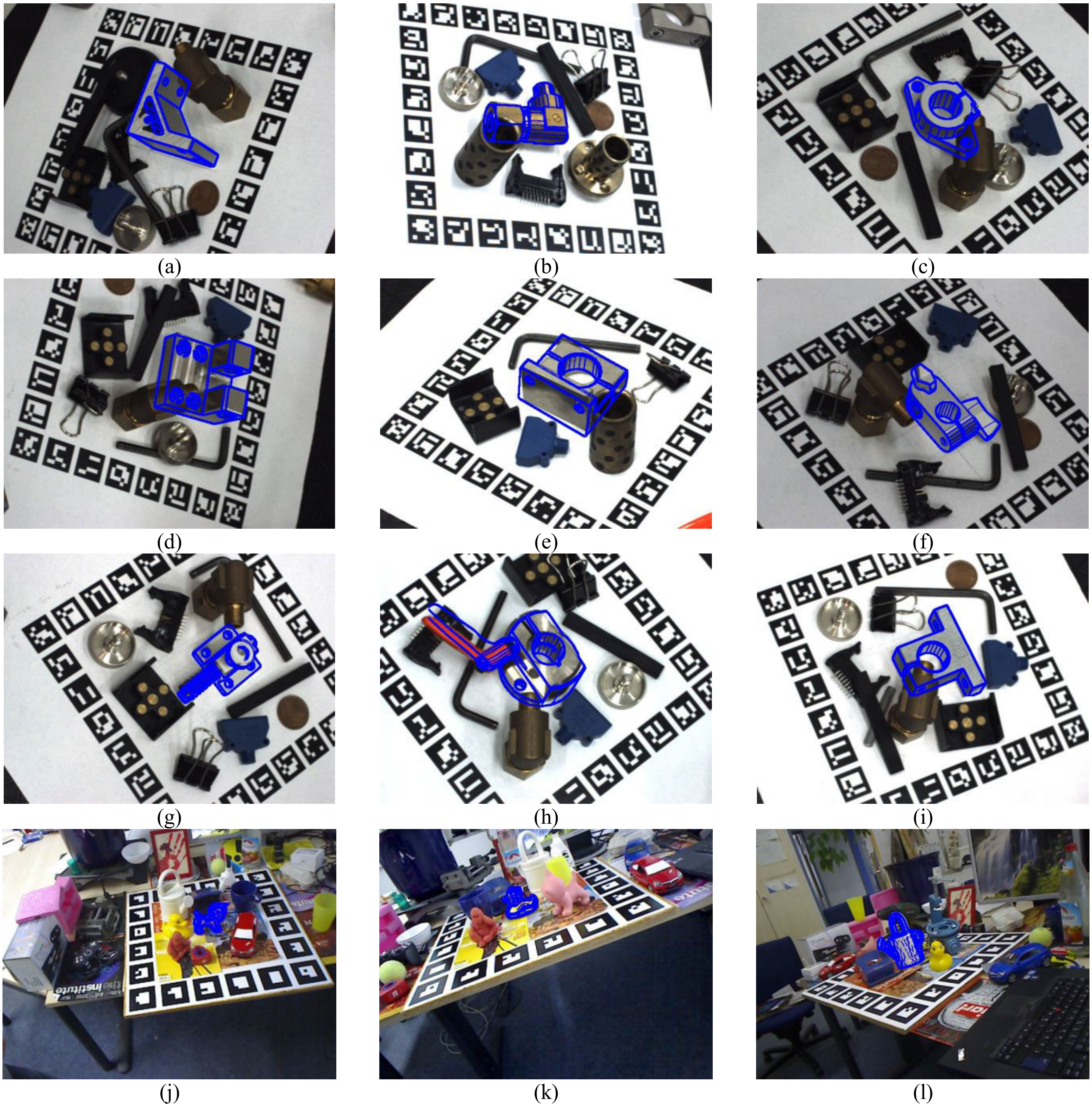


FIGURE 6. The example images of the dataset are presented. (a) Bracket. (b) Connector. (c) Flanger. (d) HingeBase. (e) L-Holder. (f) PoleClamp. (g) SideClamp. (h) Stopper. (i) T-Holder. (j) Cat. (k) Duck. (l) Can.

as Figure 5(b). There were 320 atoms in the dictionary and 20 classes of template. As we observe, the coefficient is sparse and corresponds to the correct class whose coefficient is dense. Besides, some example images with partial occlusion from the dataset made by Konishi and the dataset made by Hinterstoisser were shown in Figure 6. For texture-less objects with different colors, shapes, and backgrounds, our method can correctly find the best-fitting template using hybrid framework integrating template matching and sparse representation classification

D. ACCURACY

We evaluated the accuracy of our method compared with the other methods using the dataset made by Konishi.

We presented some images of Connector from the dataset made by Konishi with different partial occlusion dealt by differents methods in Figure 7. The first row were the results of Hinterstoisser's method. The second row were the results of Ulrich's method.The thrid row were the results of Konishi's method. The forth row were the results of our method. Our method got better results, which are best-fitting compared

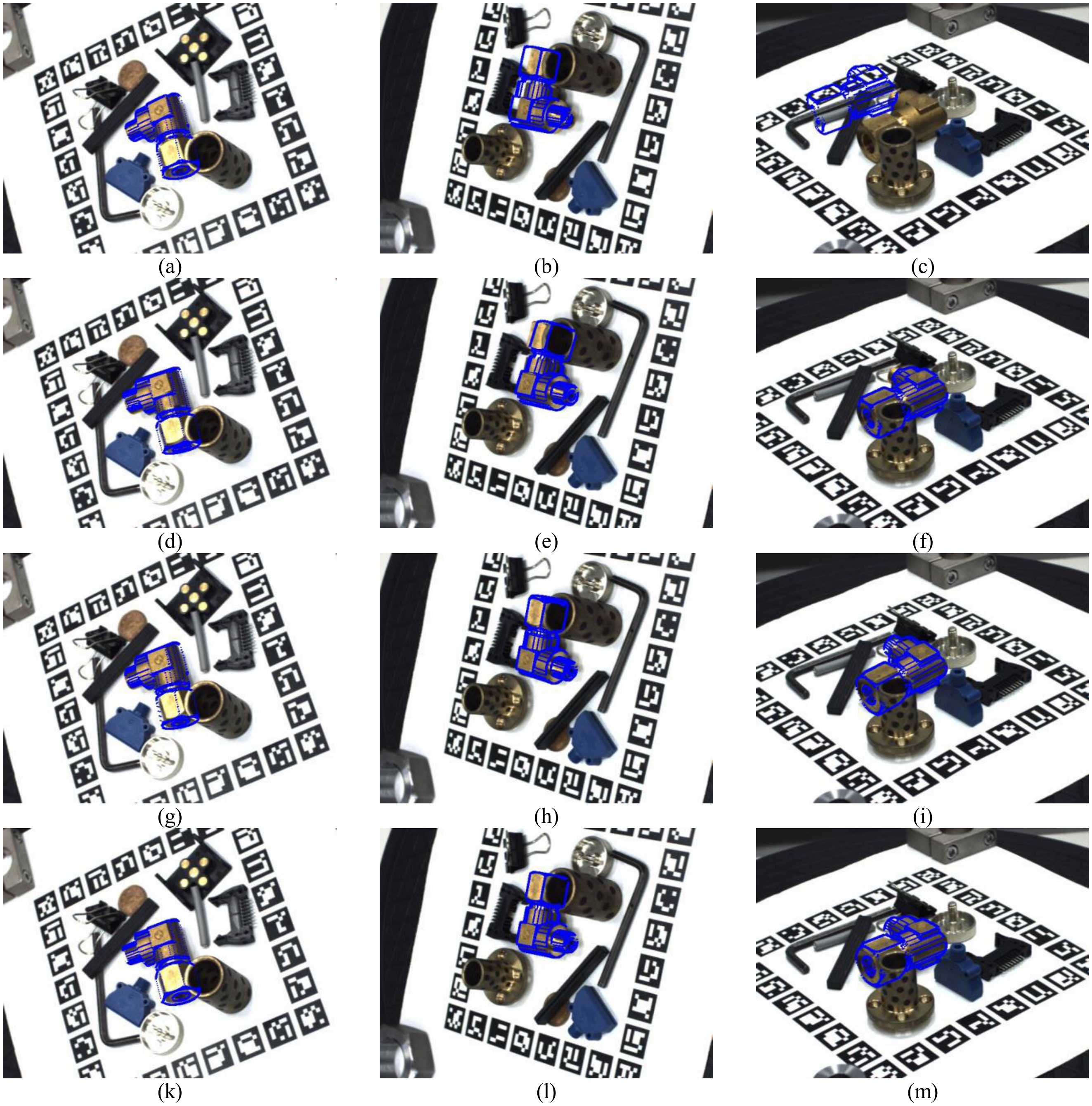


FIGURE 7. The first row are the results of Hinterstoisser's method. The second row are the results of Ulrich's method. The thrid row are the results of Konishi's method. The forth row is the results of our method.

with the other method's results. The region in 2D image and pose of object are interdependent. The correct positions in 2D image are beneficial to pose estimation. And the correct pose is also beneficial to determine the region in 2D images. The cluttered background often affects determining the region of object in 2D image. The false region often occurs when the feature of template and background is similar. Table 2 showed the rate of the correct pose estimation for the dataset made by Konishi. We regarded the template with the highest similarity as the best-fitting template and defined that the pose whose errors were within 12° for the longitude and latitude, 10° for the rotation angle, and 40mm for distance were correct. The results showed that our method has higher accuracy compared with the other methods. It is robust to the feature change caused by posture change, partial feature loss, and clutter.

V. CONCLUSION

A hybrid framework based on the warped hierarchical tree which integrates template matching and sparse representation classification was proposed for improving the accuracy of pose estimation in this paper. The warped hierarchical tree built with parameter constraint can accelerate the exhaustive search. The hybrid framework which integrates the template matching and sparse representation classification is robust to shading, reflection, and occlusion. The experiments on the texture-less datasets show that our method can achieve better results.

Large-scale variances still constrain the improvement of accuracy. In the future, the research for large-scale variances will be one of the important research contents.

REFERENCES

[1] Z. He, Z. Jiang, X. Zhao, S. Zhang, and C. Wu, "Sparse template-based 6-D pose estimation of metal parts using a monocular camera," *IEEE Trans. Ind. Electron.*, vol. 67, no. 1, pp. 390–401, Jan. 2020.

[2] E. Munoz, Y. Konishi, V. Murino, and A. D. Bue, "Fast 6D pose estimation for texture-less objects from a single RGB image," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 5623–5630.

[3] E. Munoz, Y. Konishi, C. Beltran, V. Murino, and A. D. Bue, "Fast 6D pose from a single RGB image using cascaded forests templates," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4062–4069.

[4] G. Du, P. Zhang, and D. Li, "Human–manipulator interface based on multisensory process via Kalman filters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5411–5418, Oct. 2014.

[5] H. Song, W. Choi, and H. Kim, "Robust vision-based relative-localization approach using an RGB-depth camera and LiDAR sensor fusion," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3725–3736, Jun. 2016.

[6] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis, "Single image 3D object detection and pose estimation for grasping," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2014, pp. 3936–3943.

[7] X. Qin, J. Shen, X. Mao, X. Li, and Y. Jia, "Structured-patch optimization for dense correspondence," *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 295–306, Mar. 2015.

[8] H. Zhao, H. Guo, X. Jin, J. Shen, X. Mao, and J. Liu, "Parallel and efficient approximate nearest patch matching for image editing applications," *Neurocomputing*, vol. 305, pp. 39–50, Aug. 2018.

[9] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, Nov. 1992.

[10] H. Naruse, A. Nobiki, T. Yabuta, and M. Tateda, "High-accuracy multi-viewpoint stereo measurement using the maximum-likelihood method," *IEEE Trans. Ind. Electron.*, vol. 44, no. 4, pp. 571–578, Aug. 1997.

[11] Z. Luo, K. Zhang, Z. Wang, J. Zheng, and Y. Chen, "3D pose estimation of large and complicated workpieces based on binocular stereo vision," *Appl. Opt.*, vol. 56, no. 24, pp. 6822–6836, 2017.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[13] F. Tombari, A. Franchi, and L. Di, "BOLD features to detect texture-less objects," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1265–1272.

[14] J. Chan, J. A. Lee, and Q. Kemao, "BORDER: An oriented rectangles approach to texture-less object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2855–2863.

[15] J. Chan, J. A. Lee, and Q. Kemao, "BIND: Binary integrated net descriptors for texture-less object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2068–2076.

[16] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2257–2264.

[17] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 858–865.

[18] C. Steger, "Occlusion, clutter, and illumination invariant object recognition," *Int'l Arch. Photogramm. Remote Sens.*, vol. 34, pp. 345–350, 2002.

[19] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1521–1529.

[20] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 292–301.

[21] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2938–2946.

[22] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "A novel representation of parts for accurate 3D object detection and tracking in monocular images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4391–4399.

[23] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EP*n*P: An accurate $O(n)$ solution to the P*n*P problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, p. 155, 2009.

[24] B. Přibyl, P. Zemčík, and M. Čadík, "Absolute pose estimation from line correspondences using direct linear transformation," *Comput. Vis. Image Understand.*, vol. 161, pp. 130–144, Aug. 2017.

[25] C. Meng, Z. Li, H. Sun, D. Yuan, X. Bai, and F. Zhou, "Satellite pose estimation via single perspective circle and line," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 6, pp. 3084–3095, Dec. 2018.

[26] C. Xu, L. Zhang, L. Cheng, and R. Koch, "Pose estimation from line correspondences: A complete analysis and a series of solutions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1209–1222, Jun. 2017.

[27] G. Wang, J. Wu, and Z. Ji, "Single view based pose estimation from circle or parallel lines," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 977–985, May 2008.

[28] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 548–562.

[29] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of texture-less objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 876–888, May 2012.

[30] M. Ulrich, C. Wiedemann, and C. Steger, "Combining scale-space and similarity-based aspect graphs for fast 3D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1902–1914, Oct. 2012.

[31] X. Ren, L. Jiang, X. Tang, and J. Zhang, "Single-image 3D pose estimation for texture-less object via symmetric prior," *IEICE Trans. Inf. Syst.*, vol. E101.D, no. 7, pp. 1972–1975, Jul. 2018.

[32] X. Zhang, Z. Jiang, H. Zhang, and Q. Wei, "Vision-based pose estimation for textureless space objects by contour points matching," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 5, pp. 2342–2355, Oct. 2018.

[33] Y. Konishi, Y. Hanzawa, M. Kawade, and M. Hashimoto, "Fast 6D pose estimation from a monocular image using hierarchical pose trees," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 398–413.

[34] L. A. Leiva and E. Vidal, "Warped K-means: An algorithm to cluster sequentially-distributed data," *Inf. Sci.*, vol. 237, pp. 196–210, Jul. 2013.

[35] E.-J. Cheng, K.-P. Chou, S. Rajora, B.-H. Jin, M. Tanveer, C.-T. Lin, K.-Y. Young, W.-C. Lin, and M. Prasad, "Deep sparse representation classifier for facial recognition and detection system," *Pattern Recognit. Lett.*, vol. 125, pp. 71–77, Jul. 2019.

[36] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Nonlocal patch tensor sparse representation for hyperspectral image super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 3034–3047, Jun. 2019.

[37] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2001, pp. 245–250.

[38] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014.

YONGQI GUO received the M.S. degree in control science and engineering from the Beijing University of Chemical Technology, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree with the College of Information Science and Technology.

His research interests include visual detection and image processing.

JIANLIN WANG received the M.S. degree in measurement technology and instrumentation and the Ph.D. degree in instrumentation science and technology from Tianjin University, Tianjin, China, in 1993 and 1997, respectively.

He is currently a Professor with the College of Information Science and Technology, Beijing University of Chemical Technology. His current research interests include image processing, visual detection, and complex industrial process intelligent detection technology. He is one of the Council Members of the China Instrument and Control Society.

XINJIE ZHOU received the B.E. degree in measurement and control technology and instruments from the Beijing University of Chemical Technology, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree with the College of Information Science and Technology.

His research interests include visual detection and image processing.

ZHENGUO TAN received the B.E. degree in measurement and control technology and instruments from the Beijing University of Chemical Technology, Beijing, China, in 2018, where he is currently pursuing the M.S. degree with the College of Information Science and Technology. His research interests include texture-less objects pose estimation and machine vision.

KEPENG QIU received the Ph.D. degree in control science and engineering from the Beijing University of Chemical Technology, Beijing, China, in 2020.

His current research interests include data-driven soft sensing, visual detection, and image processing.

• • •