An Integrated Target Acquisition Approach and GUI Tool for Parallel Manipulator Assembly

Article *in* Journal of Computing and Information Science in Engineering · November 2019

DOI: 10.1115/1.4045411

4 authors:

Haodong Chen
University of Maryland, College Park

Zhiqiang Teng
Heifei University of Technology

Zheng Guo
Shanghai Jiao Tong University

Ping Zhao
Hefei University of Technology

Haodong Chen
Department of Mechanical Design Engineering,
School of Mechanical Engineering,
Hefei University of Technology,
Hefei 230009, China;
Mechanical Engineering Department,
Missouri University of Science and Technology,
Rolla, MO 65401
e-mail: haodong-chen@mail.hfut.edu.cn

Zhiqiang Teng
Department of Mechanical Design Engineering,
School of Mechanical Engineering,
Hefei University of Technology,
Hefei 230009, China
e-mail: tzq199611@163.com

Zheng Guo
The Institute of Robotics,
School of Mechanical Engineering,
Shanghai Jiao Tong University,
Shanghai 200240, China
e-mail: guo_zheng@sjtu.edu.cn

Ping Zhao[1]
Department of Mechanical Design Engineering,
School of Mechanical Engineering,
Hefei University of Technology,
Hefei 230009, China
e-mail: ping.zhao@hfut.edu.cn

An Integrated Target Acquisition Approach and Graphical User Interface Tool for Parallel Manipulator Assembly

In this paper, two integrated target identification and acquisition algorithms and a graphical user interface (GUI) simulation tool for automated assembly of parallel manipulators are proposed. They seek to identify the target machine part from the workspace, obtain its location and pose parameters, and accomplish its assembling task while avoiding the collision with other items (obstacles). Fourier descriptors (FDs) and support vector machine (SVM) are adopted in this approach. The image of task area of workspace is obtained through machine vision, and the target assembling parts are identified. To acquire the location and pose information of the target, a modulus-shift matching (MSM) algorithm is proposed and integrated into the FD and SVM approaches, which could efficiently obtain the pose parameters while eliminating the effect of choice of starting point. The simulation results of two integrated algorithms, FD-MSM and SVM-MSM, are then compared and analyzed. In addition, a GUI is designed to visualize and assist the assembly process. An application on delta parallel robot with an extra rotational degree of freedom (DOF) is presented. [DOI: 10.1115/1.4045411]

Keywords: motion planning, parallel robot, machine vision, FD, SVM, image matching, computer aided manufacturing, machine learning for engineering applications, manufacturing automation, manufacturing planning

1 Introduction

Due to many mechanical advantages such as high stability, high accuracy, and compact structure, parallel manipulators are widely used in modern manufacturing industries, e.g., Steward Platform, DELTA robot (as shown in Fig. 1), etc. [1–3].

Automated assembly by parallel robots is one of the most important applications in intelligent manufacturing. Most of these assembly strategies are established on a scenario that all the components are placed at a fixed location [5]. However, if the coordinates of these locations are not accurate, or if the assembly scenario is arbitrarily set, then the robot needs to first acquire the location and pose of the assembly parts. In that case, machine vision needs to be adopted to identify the target machine parts of the current assembly stage from the workspace and reconstruct the environment [6].

Target recognition is one of the key issues in the development of machine vision [7]. Many approaches solve the recognition issue based on the feature of contours or other shape parameters of the targets. In 2012, a contour-based spatial model was proposed to detect geospatial targets accurately in high-resolution remote sensing images [8]. In 2017, an effective computer vision-based automatic detection and state recognition method for disconnecting switches is proposed by Chen et al. [9]. In recent years, machine learning is applied in the target recognition which not only detects the target based on the shapes but also the surface textures, etc. Typical machine learning algorithms that deal with such issues include the artificial neural network-based method [10] and hierarchical model and X model [11,12]. In this paper, two different approaches based on Fourier descriptors (FDs) and support vector machine (SVM) are applied in visual detection, respectively. These two algorithms have been used in the detection of targets, such as geometric shapes, human faces, digital numbers, etc. [13,14].

After identification of the correct target, its geometric parameters such as location (translational displacement) and pose (rotational displacement) of the target parts need to be obtained. Generally, this task is achieved by image matching between the target and the reference. These kinds of methods can be categorized into feature-based matching and gray-based matching [15]. In 2002, Chen et al. proposed a method to match two images of different shape, location, and pose on the basis of the Fourier–Mellin transform method [14]. Yet in the Fourier-based method, the choice of starting point on the contour shares the same effect (the phase part for each complex harmonic) with rotational angle, thus could not be decoupled. In 2009, a method for extracting the distinctive feature of rotation invariance from images was proposed by Zhang et al. [16]. In 2013, an adaptive thresholding approach was

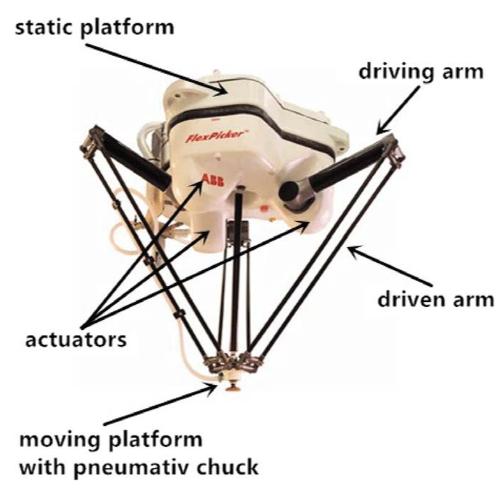

Fig. 1 An four-DOF delta parallel robot [4]

[1]Corresponding author.

Manuscript received July 8, 2019; final manuscript received October 24, 2019; published online November 6, 2019. Assoc. Editor: Ying Liu.

presented [17]. In 2015, a shape feature learning scheme is proposed by Xie et al. which could extract features and realize 3D shape matching [18]. In this paper, a modulus-shift matching (MSM) algorithm is proposed to accomplish image matching, so as to efficiently obtain the rotational angle and the location of the target parts, which eliminates the effect of starting point in vision detection. This algorithm is then integrated into FDs and SVM for a fast acquisition of geometric parameters of the target machine parts in the automatic assembly.

This paper is organized as follows. In Sec. 2, the image of task area of workspace is obtained through machine vision, and target identification with FD and SVM is described. Acquisition of the target parameters, including location and pose, is accomplished with MSM algorithm in Sec. 3, which is then integrated into FD and SVM. The simulation of these two integrated algorithms is conducted in Sec. 4, where the results are also compared and analyzed. Section 5 briefly introduces the motion planning and obstacle-avoidance strategy in real applications, and a graphical user interface (GUI) software is designed to visualize and assist the process. In the end, simulations in the GUI are conducted on a delta parallel robot with four degrees of freedom (4-DOFs).

2 Target Identification Based on Machine Vision

Generally, to identify the correct target machine parts from an arbitrarily set workspace, the visual information need to be obtained and analyzed. In this section, the workspace image is obtained and preprocessed, followed by two recognition methods, i.e., Fourier descriptors and support vector machine, which are adopted to identify the target from the given workspace.

2.1 Image Preprocessing. After obtaining the image of workspace through cameras, the preprocessing is first implemented. The gray value of each pixel is set to the median value of the gray value of all pixels in a neighborhood of the point, and the gray scale images are obtained. And then, binary images are extracted after binarization processing. The opening–closing operation, a fundamental operation in mathematical morphology, is applied to binary images to realize noise removal. Opening operation removes small objects from the foreground of an image, placing them in the background, while closing operation removes small holes in the foreground, changing small islands of background into foreground [19]. Then, the holes of parts are eliminated. Each part includes one connected domain, respectively. After that, edges of images are extracted by Canny operator [20,21], and this process consists of four steps:

(1) Gaussian convolutions are adopted to smooth the image and remove noise. The generating equation of Gaussian filter kernel with the size of $(2k+1)\times(2k+1)$ is shown as follows:

$$H_{ij} = \frac{1}{2\pi\sigma^2}\exp\left\{-\frac{[i-(k+1)]^2+[j-(k+1)]^2}{2\sigma^2}\right\}; \quad 1 \le i, j \le (2k+1) \tag{1}$$

where σ is the variance, and $2k+1$ is the dimension of the kernel matrix.

(2) The horizontal, vertical, and diagonal edges in the image are detected, and the first derivative values in the horizontal component Gx and vertical component Gy directions are returned, from which the gradient and the angle of the pixel can be determined as follows:

$$G = \sqrt{G_x^2 + G_y^2} \tag{2}$$

$$\theta = \arctan(G_y/G_x) \tag{3}$$

(3) Based on non-maximum suppression, all gradient values that equals to zero other than the local maximum are suppressed. The maximums are determined as accurate edges, and the remained pixel is represented by G_p.

(4) Edges obtained above are connected based on the double threshold algorithm. High threshold and low threshold should be considered in this step. The decision strategy of pixels is shown in Table 1.

The strong edge above will be regarded as real edge. Yet, the weak edge will also be regarded as real edge only if this pixel is connected to at least one strong edge in eight neighborhood directions. Generally, the high threshold and low threshold of the canny edge detection is set based on the gray-scale image. With adoption of Otsu's method, this threshold is determined by minimizing intra-class intensity variance or, equivalently, by maximizing inter-class variance. This ensures the universality and feasibility of Canny algorithm [20,21].

Since each part includes one connected domain, in this way, only one edge exists for the coming target recognition.

2.2 Target Recognition Based on the Fourier Descriptors Method. Figure 2 shows a general overview of the FDs method. After the image acquisition, a basic image process of the target workspace is carried out. Then, the contours of the existing items are obtained through an edge detection algorithm (e.g., canny operator). The shape contour coordinates $(x(n), y(n))$, $n = 0, 1, \ldots N-1$ are extracted and represented in complex form. The points are sorted based on the phase angle of its complex representation [22,23].

Through a Fourier transformation, the FDs of $Z(n)$ are defined as follows:

$$F(k) = \sum_{n=0}^{N-1} Z(n)e^{\left(\frac{-j2\pi kn}{N}\right)}, \quad 0 \le k \le N-1 \tag{4}$$

where k denotes the index of FDs.

All the geometric information (translation, rotation, scale, and choice of starting point P_0) are combined in the above expression. Suppose that shapes are prescribed for the machine parts to be assembled, then what we need is to identify the actual parts in the workspace that have the same shape as the prescribed ones. Based on Eq. (4) and Table 2, the FDs of the workspace contours are normalized by eliminating the effect of translation, scale, rotation, and choice of starting point. Since the translation information is only contained in the zero order of FDs, just set it as zero. Then, the scale factor exists in all magnitudes of FDs; it could be eliminated after dividing by the first order. Also, the rotation and starting

Table 1 Decision strategy of edge detection

Condition	Decision
$G_p \ge$ high threshold	Strong edge
$G_p <$ low threshold	Suppression
High threshold $> G_p \ge$ low threshold	Weak edge

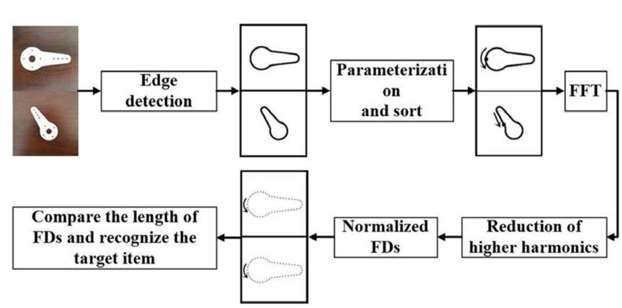

Fig. 2 FDs method: basic steps

Table 2 Normalized FDs

Geometric invariance	Operation in normalized FDs
Translation	$F(0) = 0$
Scale	$F(k) = F(k)/\|F(1)\|, \quad k = 0, 1, \ldots, N-1$
Rotation and starting point	$F(k) = \|F(k)\|, \quad k = 0, 1, \ldots, N-1$

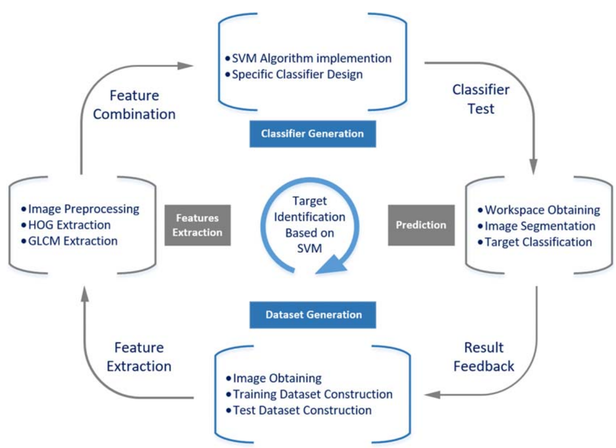

Fig. 3 HOG/GLCM-based SVM method adopted for target recognition of machine parts

point can be reflected by the phase angle of each harmonic component that can be eliminated by using only the magnitude of $F(k)$. Table 2 shows the operation in normalized FDs [24].

By comparing the least-square error of the nonzero parameters in FDs between the prescribed shapes and actual work space image contour, the actual machine parts to be assembled can be recognized.

2.3 Target Recognition Based on the Support Vector Machine Method. In this section, SVM method is discussed and implemented in target detection. As a supervised machine learning method, it mainly learns the features of samples and obtains the maximum margin hyper plane as its decision boundary. In this paper, two features adopted from image samples are histograms of oriented gradients (HOGs) descriptor and gray level concurrence matrix (GLCM). The HOG descriptor counts occurrences of gradient orientation in localized portions of an image, and local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The GLCM describes the joint distribution of the pixel gray levels with spatial position relationship [25–27].

To accomplish the target recognition of machine parts from the workspace, the HOG/GLCM-based SVM algorithm we are adopting in this paper is realized as in Fig. 3 and is described as follows: (1) Obtain images of different machine parts and construct the training dataset and the test dataset; the proportion of which is 9:1. (2) Carry out image preprocessing, extract the HOG and GLCM features, construct the image feature matrices and labels; the labels are input vectors of training dataset and the corresponding output vectors of feature matrices. (3) Implement SVM algorithm and generate the classification model fitting on training dataset. The test dataset will provide an unbiased evaluation of the classification model. (4) Obtain workspaces constructed with different parts and segment them to separate set of parts. Predict the class of these parts based on the classifier obtained in the last step.

3 Acquisitions of Target Parameters

After the recognition of the prescribed target machine parts with either FDs or SVM, the geometric parameters (location, pose, etc.) also need to be determined for the assembly process. Thus, in this section, the acquisition of translational and rotational parameters of the target parts is indicated.

3.1 Translational Parameters. To accomplish an assembly procedure, the manipulator needs to move toward the target machine parts and conduct a pick-and-place motion. The location (translational parameters) of the target hereby needs to be obtained first, which could be simply represented as the coordinates of the centroid. Note that the edge of the target shape is already obtained in Sec. 2.1, thus with the coordinates of each pixel on the target shape, we could easily determine the centroid:

$$C_x = \text{sum}(x)./\text{length}(x); \quad C_y = \text{sum}(y)./\text{length}(y) \tag{5}$$

where x and y are the coordinates of each pixel (point) of the target shape, and Cx and Cy are the coordinates of the centroid of the associated part. This provides a location for the manipulator to reach during the assembly process.

3.2 Rotational (Pose) Parameters. While obtaining the location parameters of the target machine parts is as simple as Eq. (5), the acquisition of pose parameters, though, is not as straightforward. For SVM, the classification results in the target recognition generally do not reflect the geometric parameters. On the other hand, as in Eq. (4), the phase angle of FDs shows some information of pose, but it could not separate the effect of rotation angle and choice of starting point of the given shape. To solve this issue and decouple the effects of rotation and starting point, here in this paper, a MSM algorithm is proposed to calculate the pose information after target detection. It is established by taking both the modulus and the phase of contour points into consideration. The detailed algorithm is described as follows.

Given the contour shape of a target part in workspace and the reference prescribed shape, we first convert their coordinates from Cartesian space to polar space and obtain the phase angles of all polar points. The origin of the polar space is set to be the centroid of the contour. Then, the points are sorted by the phase angle, and the amount of points is adjusted to be the same as prescribed shape. The resizing of the number of points is carried out by uniformly truncating. The points dropped are evenly distributed along the shape contour. Now, we have two series of complex numbers, denoting the prescribed shape and the identified machine part. The scale of two edges is obtained as the modules sum ratio of reference and target edge points.

To determine the pose (angle of rotation) between these two similar shapes, the most straightforward method is to take an arbitrary rotational transformation on one shape and compute the least-square error between two series of complex numbers, and then find the optimal rotational transformation that could minimize this error. However, due to the different choice of starting point on the contour, another optimization process also need to be conducted to find the closest starting point. To reduce the time complexity, in our MSM algorithm, instead of taking a rotational transformation on the whole series of complex numbers that represent the actual machine part contour, only the modulus part of the complex series is shifted by one point in each step so as to simultaneously realize the rotational transformation and the shift of starting point. Note that only the modulus part of each complex number is shifted, but the phase part (i.e., the angle) remains, which combines into a brand new series of complex numbers as shown in Fig. 4. Figure 4(*d*) shows the detailed process of adopting MSM algorithm in this application. After representing contours (a) and (b) in polar coordinates, we obtain the modulus of the contour points and plot them in Fig. 4(*d*). The shifting and matching process is then conducted, and the angles of rotation that minimize the errors between (a) and (b) are found. This new series of complex numbers is then compared with the prescribed shape. This step repeats until a minimal comparing error is found, and the angle of

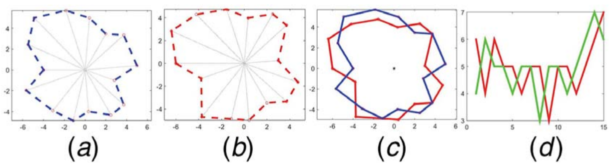

Fig. 4 An illustration of MSM algorithm

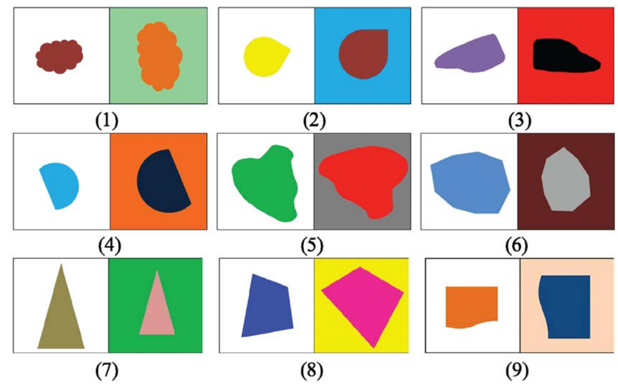

Fig. 5 A group of MSM algorithm tests with various given shapes

rotation between the target machine part contour and the prescribed shape contour can then be obtained while eliminating the effect of the starting point choice.

It is admitted that only shifting the modulus and combining it with the original phase could cause a distortion in the shape, but in industrial application, generally the amount of points on the contours is quite large. Thus, with such a large density of contour points, the phase angle could be viewed as uniformly distributed, and the distortion of shape is small enough to be neglected for this algorithm.

The pseudo code of this algorithm is as follows:

Algorithm Modulus-shift matching

```
Input: images Re (reference, i.e., the prescribed shape), Ta (target, i.e., the actual image of the parts in workspace);
Output: rotational angle and scale of Ta with respect to Re;
Begin:
1. Obtain the contour-point image of Re and Ta, set them as ReL and TaL;
2. Obtain nonzero points and centroids of ReL ((fry, frx), (Crx,Cry)) and TaL ((fty, ftx), (Ctx,Cty));
3. Convert the Cartesian coordinates of ReL and TaL to polar coordinates (Ur, Ut);
4. Sort Ur and Ut according to the phase angle;
5. Obtain the modulus and phase angles matrixes of all points of Ur(Mr, Ar) and Ut(Mt, At);
6. Resize the amount of points of Ur and Ut to be equal;
7. Shift the modulus series Mt by one point a time, obtain Mts, and combine Mts with At as Uts.
8. Compare the least-square error of the modulus of Uts with respect to Ur:
Num ← the number of points of Ut;
For i from 1 to Num
{
    Mt ← [Mt(2:Num), Mt(1)]; /*Move the first element of MT to the last to realize rotate contour points*/
    diff(i) ← sum of absolute values of (Mr-Mt);
    If diff(i) < threshold; /*the threshold is related to the size of the image*/
    mindiff ← i; /*record the position of the minimum difference*/
    threshold ← diff(i); /*update the threshold*/
    End
}
End
9. Rotational angle← (At(mindiff) – At(1))/pi*180;
    Scale← sum(Mr)/Sum(Mt)
```

To show the feasibility of MSM algorithm, we take various images of different shapes and conduct a group of tests (Fig. 5). And their MSM pose detection results are presented in Table 3. It could be observed that the error in the angle of rotation is less than 2 deg. Such a small error could be compensated in the assembly process with structural design such as a guiding slot. Thus, adopting MSM algorithm could successfully determine the pose of the target machine parts despite of the starting point choice, and its accuracy is acceptable.

3.3 Description of the Integrated Target Identification Algorithms. Now, combining with the acquisition method of target parameters proposed above, the two target identification methods (FD and SVM) in Sec. 2 could hereby be improved. Aiming at the acquisition of the location and pose parameters of the target assembly parts, two integrated algorithms, FD-MSM and SVM-MSM, are presented in this section.

Step-by-step procedure for FD-MSM algorithm

(1) Input the workspace image (gray scale), perform binarization and open–close operation, then extract the edge of part shapes through Canny detector as in Sec. 2.1.
(2) Represent the contour points in complex form $Z(n)=x(n)+jy(n)$ and sort the points based on the phase angle of its complex representation.
(3) Obtain the FDs of the contour points with Eq. (1) and normalize the FDs as in Table 1.
(4) Compare the normalized FDs of the workspace contours with those of the standard target shape and identify the target parts from the workspace image.
(5) Obtain the location parameters of the target parts by computing the centroid of its contour points as in Sec. 3.1.
(6) Shift and match the modulus to obtain the pose parameters of the target shape while eliminating the starting point effect as in Sec. 3.2.

Step-by-step procedure for SVM-MSM algorithm

(1) Input the workspace image and obtain the edge of each part shape as in Sec. 2.1.
(2) Obtain the centroid (*location parameters*) and the maximum radius of each shape and construct a square to extract each part separately from the original gray-scale image.
(3) Extract the HOG and GLCM features of each square image as Sec. 2.3 and Fig. 3 and construct the feature matrices for them.
(4) Conduct the classification of the square images of workspace parts using SVM and identify the target parts from the workspace.

Table 3 The prescribed rotational angles in Fig. 5 and the resulting angles from our MSM algorithm

Order	Prescribe (deg)	Results (deg)	Error (deg)
1	90.00	88.30	1.70
2	30.00	31.44	1.44
3	155.00	155.49	0.49
4	180.00	180.52	0.52
5	215.00	213.27	1.73
6	295.00	293.97	1.03
7	0.00	359.00	1.00
8	101.00	99.84	1.16
9	270	268.73	1.27

(5) Extract the contour of the target parts and conduct the modulus-shift matching process to obtain the *pose parameters* of the target shape.

With these two integrated algorithms, we could identify the target parts (shapes) from the workspace image and also obtain their locations and poses for the coming automated assembly process.

4 Simulation Results

In this section, to show the performance of the two integrated algorithms, 65 experiments of target acquisition from simulated workspace are conducted and the results are compared. First, 14 different types of machine parts as in Fig. 6 are placed randomly on the work space, and 65 different work space images are then generated. For the sake of best illustration of the performances of the two algorithms, the collection of reference machine parts covers a variety of different shapes, materials, and features while a number of them share similar properties to test the adaptation of the algorithms. For example, Nos. 6, 8, and 10 share the same contour shape, and Nos. 7 and 11 share the same material and texture, etc.

The preprocessed images results of reference machine parts are shown in Fig. 7. It needs to be noted that the holes are eliminated, so there is no multiple edge target.

After the image preprocessing, the edges of the items in the workspace are identified, and they could be extracted individually. Using FD-MSM, we could obtain the Fourier descriptors of each shape and compare it with the reference machine parts. After 910 (65×14) rounds of target recognition experiments, the results of successful recognition of each type of machine parts are shown in Fig. 8. The Y-axis denotes the number of successful recognition. The average time for the recognition of each target part is 35.1 ms in MATLAB.

It could be seen that while most of the parts have relatively high recognition accuracy, Nos. 6, 8, 10 as well as Nos. 7 and 13 have a much worse performance due to their similarity in the contour shapes.

Now, we adopt SVM-MSM to conduct the target recognition experiments. After obtaining the centroid and the maximum radius, each part is extracted separately from the workspace image as in Fig. 9. After the extraction of HOG and GLCM features, the feature matrix of one sample image is $1260 * 3248$. The parameters adopted in the HOG algorithm is cellsize, which is set as $256 * 256$. For GLCM, the parameter offset is set to describe four angles, [0,1], [−1,1], [−1,0], and [−1,−1], which means 0 deg, 45 deg, 90 deg, and 135 deg, respectively. The hyper-parameters being used in SVM algorithm are C and gamma, which are automatically

Fig. 6 Fourteen types of reference machine parts to be identified from the simulated workspace

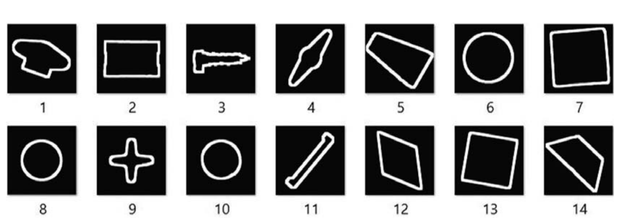

Fig. 7 Fourteen types of reference machine part contours obtained after image preprocessing

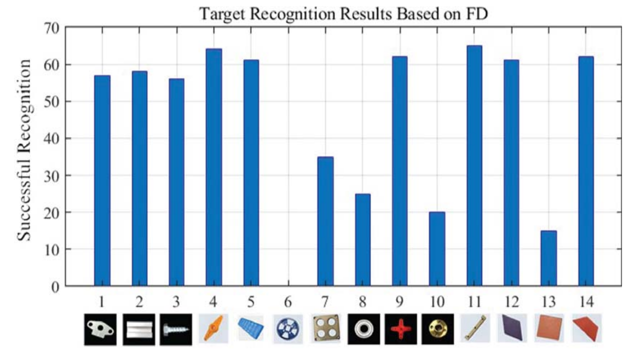

Fig. 8 Statistics of the number of successful recognition (*y*-axis) for the 14 machine parts in 65 simulated workspaces with FD-MSM

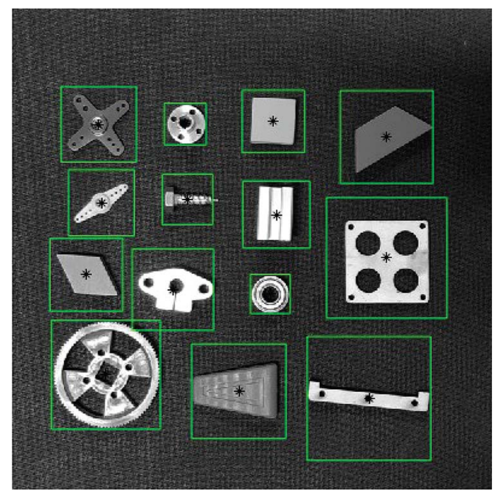

Fig. 9 Extraction of the image of individual target parts for SVM-MSM algorithm

set by MATLAB. To identify the corresponding reference machine parts, these images are classified and the statistics of successful recognition for each reference part is shown in Fig. 10. The average time for each part is 45.2 ms in MATLAB.

From the results, we could find that the performance of SVM-MSM is more stable for different types of machine parts. Yet, several parts still have a relatively low accuracy, such as Nos. 3, 7, and 13. After a detailed investigation of the classification results, we have found that a large number of part 3 are recognized as part 5 due to the unique stripe feature on both of them. Also, part 7 and part 13 are often recognized as 11 and 12/14, respectively, due to their similarity in materials and textures.

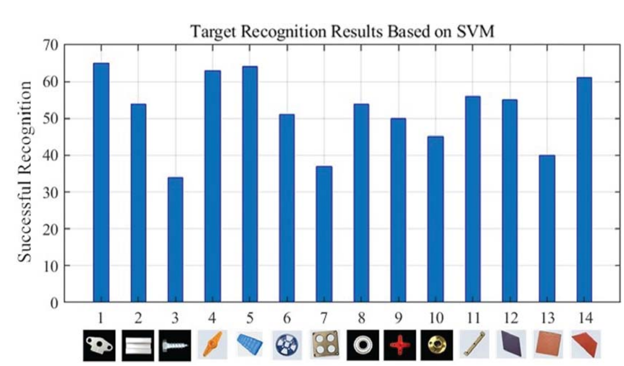

Fig. 10 Statistics of the number of successful recognition (*y*-axis) for the 14 machine parts in 65 simulated workspaces with SVM-MSM

Therefore, the simulation results of the two algorithms have shown that, for a regular target, they could both realize a good performance of recognition in real-time. However, if the target machine parts have a similar contour shape with other items on the workspace, SVM-MSM algorithm would be a preferable choice. On the other hand, FD-MSM yields better results when handling those with similar textures and materials.

5 Graphical User Interface Design for Automatic Assembly and Application

In this section, we seek to apply the proposed integrated algorithms on automatic assembly task in the manufacturing process. Generally, in an assembly task, there are target machine parts that need to be identified from the workspace, and their location/pose information need to be acquired. The assembly motion of the manipulator also needs to be planned such that collision with other items (usually marked as obstacles) could be avoided. Thus, we first proposed a feasible motion planning method for the parallel manipulators, and a GUI tool is then presented to visualize and assist the assembly process.

5.1 Modeling of Obstacles and Collision Detection. In the stage of assembling motion planning, there are obstacles that need to be avoided, and they are generally enveloped by cylinders for a simpler collision detection algorithm rather than the cubes and spheres [28]. For the parts that are placed on the platform, sphere envelope might cause redundancy envelope. For cylinder that has fixed radius in the horizontal direction, it is more efficient and convenient in horizontal distance detection than cube envelope.

Assume that the obstacles, parts, and end-effector (e.g., pneumatic chuck) are all enveloped as cylinders with radius being R_O, R_P, and R_E, respectively. The heights of them are also obtained according to the part benchmark. The detection of collision during the process of assembly can be converted into the intersection detection of cylinders. By calculating the shortest distance of the centroids and compare it with the radius of the cylinders, the collision status could be determined as in Table 4.

5.2 Obstacle-Avoidance Strategy and Graphical User Interface Design. In obstacle-avoidance path planning, many classic algorithms have been proposed, such as A star and RRT star. But for the high velocity of parallel robots, these algorithms are not very suitable [29]. In our application of parallel robot assembly, the length of the path is not as important as the execution speed. Since the obstacles are also machine parts from the benchmark, the height of them can be obtained easily. Therefore, instead of spending time to run a path optimization algorithm, here we adopt a simple strategy of lifting the end-effector of the parallel manipulator so as to get over the obstacles and guarantee a collision-free motion in the whole assembly process.

The obstacle-avoidance motion in a work cycle of the end-effector is shown as $P_0P_1P_2P_3$ in Fig. 11.

Overall, the procedure of machine vision-based motion planning algorithm for parallel manipulators is illustrated in Fig. 12.

To visualize and assist the assembly process, a GUI software is designed as in Fig. 13 [30,31]. Users could choose the target parts to be assembled from the benchmark, and based on the characteristics of the target parts, FD-MSM or SVM-MSM algorithm is selected to conduct the assembly task. After obtaining the workspace image through machine vision, the software will provide a collision-free assembly motion for the manipulators to execute. The interface also provides an interpolated simulation of the resulting assembly motion.

Table 4 Conditions of collision detection

Collision type	Collision condition
Obstacle part in transit	$D_{\min}(O, P) \leq R_O + R_P$
Obstacle end-effector	$D_{\min}(O, E) \leq R_O + R_E$

Note: O, obstacle; P, part E, end-effector; $D_{\min}$, shortest distance; R, radius.

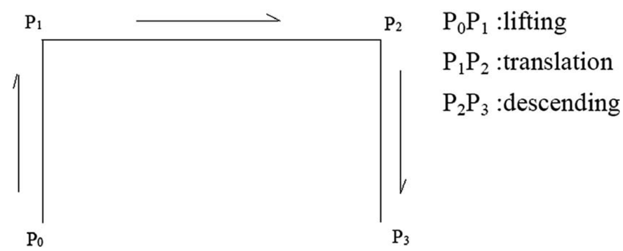

Fig. 11 Pick-and-place motion cycle

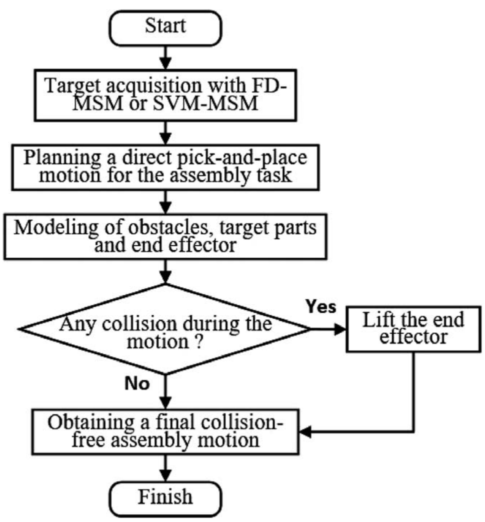

Fig. 12 A general illustration of the proposed collision-free motion planning based on motion vision

5.3 Application. In this section, based on a four-DOF delta parallel robot (as shown in Fig. 14), we have conducted the application of our proposed target acquisition and motion planning

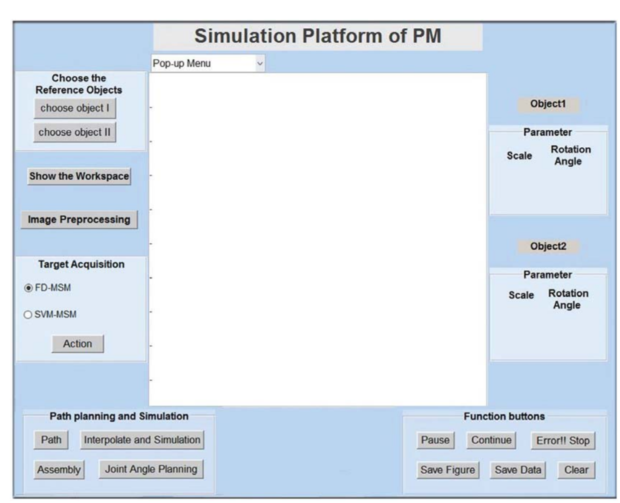

Fig. 13 The GUI software that visualizes the collision-free assembly process

Fig. 14 The four DOFs delta parallel robot

algorithm in the GUI. The parameters of four DOFs parallel robot are shown in Table 5. For a better legibility, only the image-display window of the GUI is shown in this section.

Suppose the machine parts to be assembled in the current task are those in Figs. 15(*a*) and 15(*b*). Figure 15(*c*) is the image (500×500 pixels) of the workspace, which is consisted by target parts as in (*a*) and (*b*), as well as two other machine parts for the upcoming assembly task. The other two parts are viewed as obstacles in the current assembly task, thus after target recognition, the workspace is plotted as in Fig. 16, where the target parts A and B are recognized respectively, while the two obstacles are enveloped by cylinders.

Meanwhile, the location and pose parameters of the two target machine parts are also achieved as in the right part of Fig. 16. The MATLAB computational time for this example is less than 40 ms.

Suppose that the assembly task is to grasp machine part A (Fig. 15(*a*)) and move it to part B (Fig. 15(*b*)), we can divide the whole assembly task into three steps. Step I is moving the end-effector from initial location to part A and grasping it, then Step II is to deliver part A to the location of part B and accomplish the assembly action. In the end, Step III takes the end-effector back to the initial location. Figure 17(*a*) shows the original motion that does not take obstacle-avoidance into consideration. The trajectory

Table 5 Parameters of the delta robot

Symbol	Numerical value (mm)
D	300
d	78
L_a	800
l	2000

Note: D, scale of static platform; d, scale of dynamic platform and C_i; L_a, length of driving arm; l, length of driven arm.

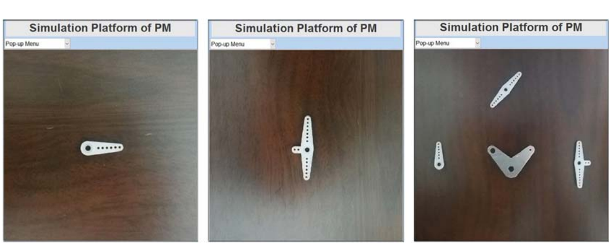

Fig. 15 The target machine parts to be assembled and the workspace

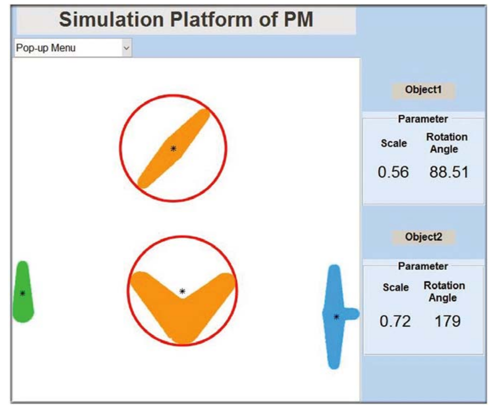

Fig. 16 Target acquisition results in GUI

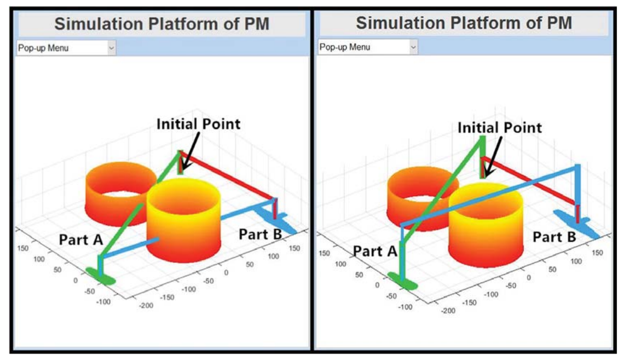

Fig. 17 Obstacle-avoidance strategy

for the end-effector of Steps I, II, and III are colored by green, blue, and red, respectively. Clearly, the original motion would collide with the two obstacles. Thus, by adopting our simple obstacle-avoidance strategy, we obtain Fig. 17(*b*), which illustrates that the collision-free path is obtained by lifting the end-effector in

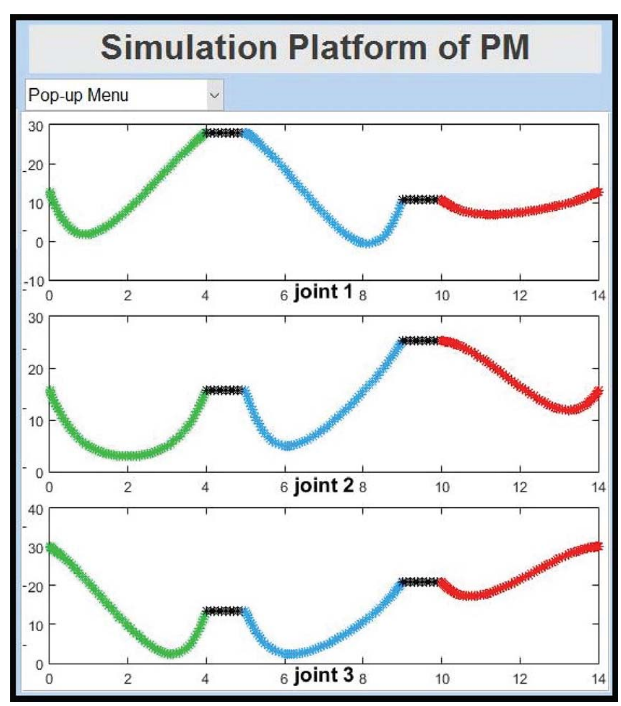

Fig. 18 Motion interpolation of the three driving joints based on B-spline

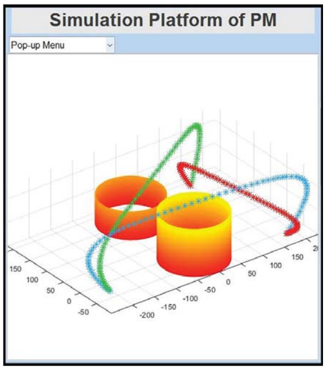

Fig. 19 A simulation of the actual performance of the resulting collision-free motion on our four-DOF delta platform

Steps I and II. It needs to be pointed out that the total time consumption from image input till the acquisition of a collision-free assembly motion in our MATLAB simulation is less than 0.2 s, thus our algorithm has a good potential to be applied in real-time assembly task.

In the end, to show a simulation of the actual performance that the resulting collision-free motion does on our four-DOF delta platform, we adopt the B-spline method to interpolate the resulting assembly motion [32]. The variation of three joint angles during the three steps is illustrated in Fig. 18. Each joint angle curve is separated by two intermissions of the grasping and releasing action. The simulated path of end-effector in the whole assembly task is illustrated in Fig. 19. The interpolated smooth B-spline path is above the original path. Therefore, if original path is obstacle-avoidance, the B-spline paths will guarantee to be collision-free. The results indicate that the assembly task is accomplished, and collision with any obstacles is successfully avoided.

6 Conclusion

This paper solves the problem of target acquisition and motion planning for the assembly task in manufacturing. Two integrated algorithms, FD-MSM and SVM-MSM, are proposed based on the machine vision technique. The prescribed assembling parts and obstacles could be recognized, and the geometric parameters could also be efficiently acquired. The simulation results of two integrated algorithms are compared and their advantages and shortages are analyzed. A simple collision-avoidance motion planning algorithm is proposed afterwards, and a GUI tool is designed to visualize and assist this process. In the end, experiments are conducted on a four-DOF delta parallel robot. A simulation of the assembly motion is shown in the GUI. The results show that the pick-and-place assembly motion could be established in almost real-time between the prescribed assembly parts, while the obstacles could be successfully detected and avoided.

Acknowledgment

The work has been financially supported by National Natural Science Foundation of China (NNSFC) (Ping Zhao, Project No. 51775155; Funder ID: 10.13039/501100001809) and Anhui Provincial Key R&D Project (Ping Zhao, No. 201904b11020035); all findings and results presented in this paper are those of the authors and do not represent those of the funding agencies.

References

[1] Bai, P., Mei, J., Huang, T., and Chetwynd, D. G., 2016, "Kinematic Calibration of Delta Robot Using Distance Measurements," Proc. Inst. Mech. Eng. Part C, **230**(3), pp. 414–424.
[2] Zhang, N., and Shang, W., 2016, "Dynamic Trajectory Planning of a 3-DOF Under-Constrained Cable-Driven Parallel Robot," Mech. Mach. Theory, **98**, pp. 21–35.
[3] Pierrot, F., Reynaud, C., and Fournier, A., 1990, "DELTA: A Simple and Efficient Parallel Robot," Robotica, **8**(2), pp. 105–109.
[4] Vischer, P., and Clavel, R., 1998, "Kinematic Calibration of the Parallel Delta Robot," Robotica, **16**(2), pp. 207–218.
[5] Patel, Y. D., and George, P. M., 2012, "Parallel Manipulators Applications—A Survey," Mod. Mech. Eng., **2**(3), p. 57.
[6] Pérez, L., Rodríguez, Í., Rodríguez, N., Usamentiaga, R., and García, D. F., 2016, "Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review," Sensors, **16**(3), p. 335.
[7] Nash, S. R., Leddy, C. A., and Quevedo, H. A., 2019, "System for Real-Time Moving Target Detection Using Vision Based Image Segmentation," U.S. Patent Application 10/319,098, Raytheon Co.
[8] Fraser, S., Nikora, V., Williamson, B. J., and Scott, B. E., 2017, "Automatic Active Acoustic Target Detection in Turbulent Aquatic Environments," Limnol. Oceanogr.: Methods, **15**(2), pp. 184–199.
[9] Chen, H., Zhao, X., Tan, M., and Sun, S., 2017, "Computer Vision-Based Detection and State Recognition for Disconnecting Switch in Substation Automation," Int. J. Robot. Autom., **32**, p. 1.
[10] Van Gerven, M., and Bohte, S., 2017, "Artificial Neural Networks as Models of Neural Information Processing," Front. Comput. Neurosci., **11**, p. 114.
[11] Sufikarimi, H., and Mohammadi, K., 2017, "Speed Up Biological Inspired Object Recognition, HMAX," 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS), Shahrood, Iran, Dec. 20–21, IEEE, pp. 183–187.
[12] Zhang, H. Z., Lu, Y. F., Kang, T. K., and Lim, M. T., 2016, "B-HMAX: A Fast Binary Biologically Inspired Model for Object Recognition," Neurocomputing, **218**, pp. 242–250.
[13] Wang, C., Zhang, Y., Song, J., Liu, Q., and Dong, H., 2019, "A Novel Optimized SVM Algorithm Based on PSO With Saturation and Mixed Time-Delays for Classification of Oil Pipeline Leak Detection," J. Syst. Sci. Syst. Eng., **7**(1), pp. 75–88.
[14] Chen, Q. S., Defrise, M., and Deconinck, F., 1994, "Symmetric Phase-Only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition," IEEE Trans. Pattern Anal. Mach. Intell., **12**, pp. 1156–1168.
[15] Wang, L., Ye, X. F., Wang, G., and Wang, L., 2017, "A Fast Hierarchical MRF Sonar Image Segmentation Algorithm," Int. J. Robot. Autom., **32**(1), pp. 48–54.
[16] Zhang, X., He, G., and Yuan, J., 2009, "A Rotation Invariance Image Matching Method Based on Harris Corner Detection," 2nd International Congress on Image and Signal Processing, Tianjin, China, Oct. 17–19, pp. 1–5.
[17] Eid, A. H. M., and Cholewo, T. J., 2015, "Methods of Content-Based Image Identification," U.S. Patent 9,064,316, Lexmark International Inc.
[18] Xie, J., Fang, Y., Zhu, F., and Wong, E., 2015, "Deepshape: Deep Learned Shape Descriptor for 3d Shape Matching and Retrieval," 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, June 7–12, pp. 1275–1283.
[19] Vincent, L., 1994, *Shape in Picture*, O. Ying Lie, A. Toet, D. Foster, H. J. A. M. Heijmans, and P. Meer, eds., Springer, Berlin, pp. 197–208.
[20] Gong, S., Li, G., Zhang, Y., Li, C., and Yu, L., 2019, "Application of Static Gesture Segmentation Based on an Improved Canny Operator," J. Eng., **2019**(15), pp. 543–546.
[21] Thilagamani, S., and Shanthi, N., 2014, "Gaussian and Gabor Filter Approach for Object Segmentation," ASME J. Comput. Inf. Sci. Eng., **14**(2), p. 021006.
[22] Afzali, A., Mofrad, F. B., and Pouladian, M., 2018, "Inter-Patient Modelling of 2D Lung Variations From Chest X-Ray Imaging Via Fourier Descriptors," J. Med. Syst., **42**(11), p. 233.
[23] Demir, B., Sayinci, B., Çetin, N., Yaman, M., and Çömlek, R., 2019, "Shape Discrimination of Almond Cultivars by Elliptic Fourier Descriptors," Erwerbs-Obstbau, **61**(3), pp. 1–12.
[24] Zhang, D., and Lu, G., 2003, "A Comparative Study of Curvature Scale Space and Fourier Descriptors for Shape-Based Image Retrieval," J. Vis. Commun. Image Represent., **14**(1), pp. 39–57.
[25] Qi, S., Ma, J., Lin, J., Li, Y., and Tian, J., 2015, "Unsupervised Ship Detection Based on Saliency and S-HOG Descriptor From Optical Satellite Images," IEEE Geosci. Rem. Sens. Lett., **12**(7), pp. 1451–1455.
[26] Pare, S., Bhandari, A. K., Kumar, A., and Singh, G. K., 2017, "An Optimal Color Image Multilevel Thresholding Technique Using Grey-Level Co-Occurrence Matrix," Expert. Syst. Appl., **87**, pp. 335–362.
[27] Varish, N., and Pal, A. K., 2018, "A Novel Image Retrieval Scheme Using Gray Level Co-Occurrence Matrix Descriptors of Discrete Cosine Transform Based Residual Image," Appl. Intell., **48**(9), pp. 2930–2953.
[28] Shirai, K., Madokoro, H., Takahashi, S., and Sato, K., 2014, "Parallel Implementation of Saliency Maps for Real-Time Robot Vision," 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), KINTEX, Seoul, South Korea, Oct. 22–25, IEEE, pp. 1046–1051.
[29] Raja, P., and Pugazhenthi, S., 2012, "Optimal Path Planning of Mobile Robots: A Review," Int. J. Phys. Sci., **7**(9), pp. 1314–1320.
[30] Moran, K., Li, B., Bernal-Cárdenas, C., Jelf, D., and Poshyvanyk, D., 2018, "Automated Reporting of GUI Design Violations for Mobile Apps," The 40th International Conference on Software Engineering, Gothenburg, Sweden, May 27–June 3, ACM, pp. 165–175.
[31] Bayasgalan, M., 2018, "Graphical User Interface Design of FIR Filter," Open Access Library J., **5**(12), p. 1.
[32] Wang, D. G., Hu, Y. L., Sun, J. J., and Li, Y., 2016, "Estimation of the Water–Oil–Gas Relative Permeability Curve From Immiscible WAG Coreflood Experiments Using the Cubic B-Spline Model," Petrol. Sci., **13**(3), pp. 507–516.