



Pattern matching for industrial object recognition using geometry-based vector mapping descriptors

Oung Tak You¹ · Dong Sung Pae¹ · Sung Hee Kim¹ · Kyeong Eun Kim¹ · Myo Taeg Lim¹  · Tae Koo Kang²

Received: 18 July 2017 / Accepted: 19 July 2018 / Published online: 2 August 2018
© Springer-Verlag London Ltd., part of Springer Nature 2018

Abstract

Object recognition has always been a troublesome issue for computer vision. Despite continuous researches, it still remains a challenge to define features, match the corresponding features, and develop accuracy and precision concurrently while considering computational speed and robustness at the same time. In this paper, we propose a novel feature matching method called the vector mapping descriptor (VMD) to overcome existing issues. We implement sub-pixel units for edge detection to improve the accuracy of invariant features, after which sub-pixel unit edges are enhanced by least squares error estimation, and more accurate geometric features are extracted from the enhanced sub-pixel unit edges of an object's geometric shape. We defined two geometric features, namely a circle center and a line intersection, used to construct the VMD, which represents the correlation of features consisting of the Euclidean distance and angle. The geometry-based VMD for pattern matching is proposed to match salient feature points between different images effectively under geometric transformation irrespective of missing or additional feature points. The VMD enabled one-to-one feature matching of corresponding grouped feature points from different images resulting in complete object matching. The proposed matching algorithm was invariant to geometric transformation such as translation, rotation, and scale differences and was also able cope with partial distortion or occlusion. Experiments were conducted with an industrial camera to show that our system can be executed in real time.

Keywords Geometric features · Vector mapping descriptors · Matching · Geometric transformation · Partial distortion or occlusion

1 Introduction

The ability of human vision is excellent and accurate in object recognition, but certain limitations exist since a human being may become exhausted and unable to work as fast and efficiently as a computer [1]. Due to these limitations, computer vision is required to assist in image recognition. Object matching refers to detecting and recognizing the pose (position, rotational angle, and scale difference) of the target object in digital image form. The problem of object registration information is that, respectively, comparing the model image to the target image when they are not identical. Matching objects between two images with different appearances due to situations such as varying intensity of light, scale change, rotation of the object, and partial distortion or occlusion of the object is a challenging task.

There are many researches on object retrieval approaches and methods. Some researches have focused on complex image scenes like crowded places and many buildings in a city using algorithms based on obtaining certain points

✉ Myo Taeg Lim
mlim@korea.ac.kr

✉ Tae Koo Kang
tkkang@smu.ac.kr

Oung Tak You
ykystw1525@korea.ac.kr

Dong Sung Pae
paeds915@korea.ac.kr

Sung Hee Kim
haha217@korea.ac.kr

Kyeong Eun Kim
noni@korea.ac.kr

¹ Department of Electrical Engineering, Korea University,
1 Anam-dong, Sungbug-Gu, Seoul, South Korea

² School of Human Intelligent Robot Engineering, Sangmyung
University, 31 Sangmyungdae-Gil, Dongnam-Gu,
Cheonan-Si, Chungnam, South Korea

(features) from an image pyramid, in which the features are robust. Lowe [2] suggested scale-invariant feature transform (SIFT) algorithms, while Ker et al. [3] improved the SIFT algorithm by implementing principal components analysis to the normalized gradient patch, and Mikolajczyk et al. [4] proposed an extension of the SIFT descriptor named the gradient location and orientation histogram. Herbert Bay [5] proposed speeded up robust features (SURF) which is inspired from SIFT algorithm. Other studies have been performed on object recognition through the shape-based retrieval approach [6, 7]. Ling et al. [8] suggested the inner distance shape context which is an extension of shape contexts by comparing their similarity for corresponding points [9], and the inner distance is defined as the shortest path along the edge of an object shape and results in invariant shape articulation. Furthermore, Yang et al. [10] computed affine geometric invariants using convex hulls generated by an object to find the correspondences between the convex hull vertices, while Wang et al. [11] suggested that the nodes and edges of an object can be used to form a histogram descriptor for matching, and Caetano et al. [12] proposed graphical matching that works in Euclidean space to solve weighted graph matching problems.

Moreover, a method called the Fourier shape descriptor that improved the retrieval accuracy by a Fourier transform of object information has been suggested [13–15], and Foroosh et al. [16] proposed an extension of this method of phase correlation to sub-pixel registration by taking the means of the phase correlation on down-sampled images. Besides, Guest et al. [17] introduced a method called correspondence by sensitivity to movement which works by selecting features according to their reliability of possible matches in two and three dimensions in medical and biological applications, while Montesinos et al. [18] proposed using a first-order differential descriptor of the image function in the neighborhood of the detected control points. A

comparison of the basic mutual information registration with multi-resolution contests from coarse-to-fine speed up was introduced by Pluim et al. [19], and Suk et al. [20, 21] used the invariant shape descriptor to represent regions and further developed this method to incorporate point-based invariants. An approach using 3D depth sensor was also proposed [22–24].

Matching can be performed by an image segmentation method. To this end, Zhang and Ji [25] proposed image segmentation using a Bayesian network for object detection, and Ferrari et al. [26] suggested contour segments for object detection. Furthermore, dynamic programming was used for distorted and occluded object retrieval by Petrakis [27] in which the task of transform model estimation was solved in the geometric deformation of the target image. A method of image acquisition to the required accuracy of registration for rigid-body point-based registration was introduced by Fitzpatrick et al. [28], while Bentoutou et al. [29] used mutually shifted and blurred digital subtraction angiograph images for registration.

However, these previous methods can be at a disadvantage for simple images like industrial component recognition where the background is mono-color and the object is simple as executing them is too computationally slow for industrial situations where speed is one of the most important criteria, and previous works have not satisfied this requirement. A mono-color background and simple object may indicate insufficient information that results in less salient feature points, which causes problems when the objects are occluded since this relationship can be inadequate due to a lack of feature points, thus leading to object mismatching. The patents of Matrox [30] and Cognex [31] are verified tools in simple object recognition. However these tools have accuracy and precision problems specifically in solving occlusion and scale changed images.

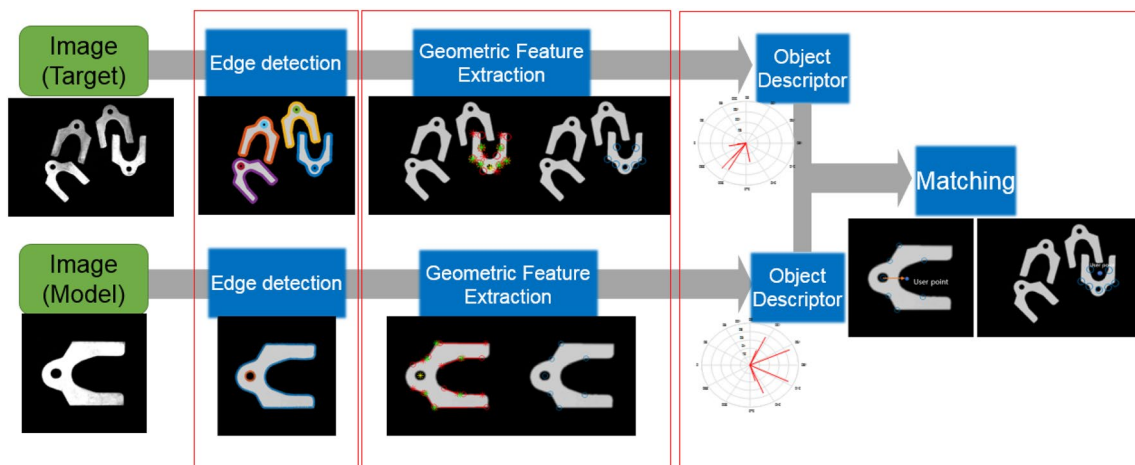


Fig. 1 Overview of entire system for geometric matching process

In this paper, we address the challenge of improving the efficiency and reliability of object matching in image processing. The process is divided into three sections, namely edge detection, feature extraction, and object matching, as shown in Fig. 1. Edge detection of objects is applied to input images at a low level of abstraction with its purpose being to reduce undesired outliers and enhance the useful image data important for further processing. The feature extraction process is an important procedure in the object matching system, in which the features contain unique, relevant information relating to the model and target objects. The geometry-based vector mapping descriptor (VMD) for pattern matching is proposed for object matching, and one for each feature is developed in order to obtain corresponding feature points between the model and target images. Object matching is performed by the constructed descriptor by finding corresponding feature points that cope with both distortion and occlusion, and the method is invariant to geometric transformation.

The rest of this paper is organized as follows. In Sect. 2, we introduce edge detection from edge enhancement using image sub-pixeling. Section 3 contains information on what kinds of geometric features can be detected, and in Sect. 4, we present the VMD algorithm to test the match between model and target images. In Sect. 5, we report on experiments conducted with three different cases of image datasets and a real-time camera, and Sect. 6 contains a discussion on the results of these, our conclusions on the proposed algorithm, and future work on what is left to do and on how to improve our system.

2 Edge enhancement using image sub-pixeling

Geometric features are obtained from the boundaries of an object; thus, we need edge enhancement using sub-pixeling to obtain accurate edge information. Edge detection is divided into two steps. First, image filtering is necessary and is applied to the raw image to reduce any undesired effects, and then, the edge where the geometric features are defined is extracted. In this step, sub-pixel units are applied for accurate feature information. The second step is edge linking and thinning whereby the pixel sequence of the edge and a single layer of edge pixels are required for further processing in edge enhancement as a smooth edge is required for geometric feature extraction performed at a later stage.

2.1 Edge extracting by sub-pixeling

The raw data from an input image are obtained from the camera and other datasets and usually contain noise that interferes with edge detection. The noise in an image can cause strong intensity differences considered as edges, and so a Gaussian filter is applied to eliminate it [32]. Light

reflection can also cause deformation of an image that interferes with edge detection; thus, an intensity histogram is created over the whole image to investigate the light reflection effect since there will be a greater difference in the intensity histogram if part of an object is reflected by light [33]. The image becomes more suitable for extracting the edge once the noise and light reflection effects have been reduced.

In image processing, edges contain significant information for object detection which allows one to distinguish between an object and the background or between objects. Edges can be detected when a change in light intensity occurs in an image, which results in a steeper gradient. There are edge detection algorithms such as the Sobel operator, Robert's cross operator, Prewitt's operator, the Laplacian of Gaussian, and the Canny edge detector [34], the latter currently being the most accurate edge detection algorithm [35]. However, these algorithms calculate an edge's position information as pixel units that have limitations in accurately detecting features such as circles and lines. A sequence of pixels tends to be a straight line instead of a smooth curve, which makes detecting the information on a feature's position less accurate.

Application of sub-pixel [36] units in image processing algorithms has been suggested to improve the accuracy of position information. In order to increase the accuracy of the edge position using a sub-pixel edge detection algorithm, the partial area effect is used to update the pixel units with the sub-pixel units. First, we obtain the edge using the Canny edge detector from the filtered raw input image, and then, the edge position information obtained and its neighbor information with the gradient are used to update the sub-pixel units.

2.2 Edge enhancement by least square error estimation

After the edges have been detected, each edge needs to be identified as a separate object, although if one edge is contained in another edge, they are considered as the same object. A group of edges is labeled as an object, and each object can contain more than one models in occlusion problem. For further processing, the edges need to be thin single lines so that the geometric feature extraction step can proceed with more accuracy [37].

Although the detected sub-pixel unit edge is more suitable than a pixel unit edge, it can be processed further to achieve a smoother edge, which improves geometric feature extraction. Figure 2 shows the result of edge enhancement, which clearly shows that the blue line is much smoother than the red line. The feature edge indicated by the red line is made up of many short lines instead of one long line since the red line zigzags, causing lots of line intersections that need to be dealt with in the geometric feature extraction section.

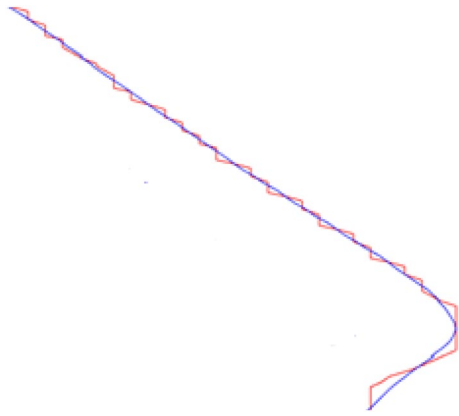


Fig. 2 Edge enhancement result. The red line is before and the blue line is after enhancement

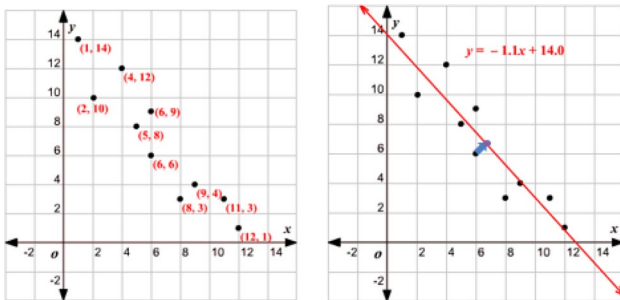
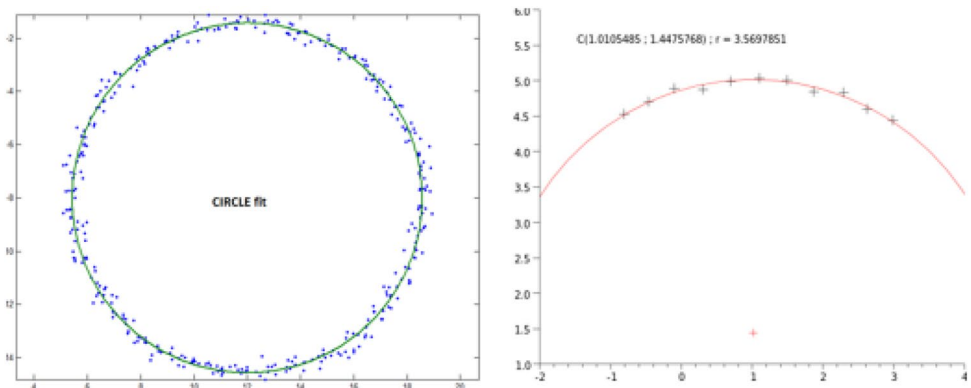


Fig. 3 An example of line fitting using the least squares method

Many features cause time-consuming computation, which we need to avoid. Another issue is that zigzagging causes randomness that makes it difficult to create certain rules, like descriptors. Therefore, the enhancement procedure needs to progress significantly if we want stable features.

To achieve this, the K th edge ($K: 1, 2 \dots N$) of the j th point is E_{kj} and $E_{kj} = (x_{kj}, y_{kj})$ where x_{kj} and y_{kj} are the respective x and y coordinates of the point. If the K th edge has N points, line fitting and circle fitting are carried out for

Fig. 4 An example of circle fitting



the $(2n + 1)$ edge points, $[E_{k(j-n)}, E_{k(j-n+1)}, \dots, E_{k(j+n)}]$, in the interval between $j=(n, N-n)$.

Figure 3 shows the results of the line fitting. As can be seen, there are dots scattered along the red line. Initially, we only have the scattered dots and we have to find the line of best fit from them whereby each dot is at a minimum distance from the red line, which means they have the least error.

The least squares line-fitting method is used to represent line $1 = ax + by$. According to the following equation, a and b values can be approximated as

$$\begin{bmatrix} x_{k(j-n)} & y_{k(j-n)} \\ \vdots & \vdots \\ x_{k(j+n)} & y_{k(j+n)} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{1}$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \text{pinv} \left(\begin{bmatrix} x_{k(j-n)} & y_{k(j-n)} \\ \vdots & \vdots \\ x_{k(j+n)} & y_{k(j+n)} \end{bmatrix} \right) \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{2}$$

The line fitting error E_l is defined as

$$E_l = \frac{\sum_{i=j-n}^{j+n} \frac{|1-by_{ki}-ax_{ki}|}{\sqrt{a^2+b^2}}}{2n + 1}. \tag{3}$$

Figure 4 shows the results of circle fitting in which the blue dots are scattered around the blue line. Initially, we only have scattered dots and we have to find the circle of best fit from them whereby each dot is at a minimum distance from the blue circle, which means they have the least error.

The least squares circle fitting method is used to represent circle $(x - x_c)^2 + (y - y_c)^2 = r^2$. According to the following equation, a , b , and r values are approximated as

$$\begin{bmatrix} -2x_{k(j-n)} & -2y_{k(j-n)} & 1 \\ \vdots & \vdots & \vdots \\ -2x_{k(j+n)} & -2y_{k(j+n)} & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ c \end{bmatrix} = \begin{bmatrix} -x_{k(j-n)}^2 & -y_{k(j-n)}^2 \\ \vdots & \vdots \\ -x_{k(j+n)}^2 & -y_{k(j+n)}^2 \end{bmatrix}, \tag{4}$$

$$\begin{bmatrix} x_c \\ y_c \\ c \end{bmatrix} = \text{pinv} \left(\begin{bmatrix} -2x_{k(j-n)} & -2y_{k(j-n)} & 1 \\ \vdots & \vdots & \vdots \\ -2x_{k(j+n)} & -2y_{k(j+n)} & 1 \end{bmatrix} \right) \begin{bmatrix} -x_{k(j-n)}^2 & -y_{k(j-n)}^2 \\ \vdots & \vdots \\ -x_{k(j+n)}^2 & -y_{k(j+n)}^2 \end{bmatrix} \quad (5)$$

$$c = x_c^2 + y_c^2 - r^2, \quad (6)$$

$$r = \sqrt{x_c^2 + y_c^2 - c}. \quad (7)$$

The circle fitting error E_c is defined as

$$E_c = \frac{\sum_{i=j-n}^{j+n} \left| \sqrt{(x_{ki} - x_c)^2 + (y_{ki} - y_c)^2} - r \right|}{2n + 1}. \quad (8)$$

If the circle fitting error is larger than the line fitting error $E_l < E_c$, E_{kj} belongs to the line; then, the point position is corrected as follows:

$$x'_{kj} = \frac{y_{kj} + x_{kj} - y_c}{2x_c}, \quad y'_{kj} = \frac{y_{kj} + x_{kj} + y_c}{2x_c}. \quad (9)$$

Otherwise, if $E_l > E_c$, E_{kj} belongs to the circle, and then, the point position is corrected as follows:

$$\begin{aligned} x'_{kj} &= x_c + \frac{(x_{kj} - x_c)r}{\sqrt{(x_{kj} - x_c)^2 + (y_{kj} - y_c)^2}}, \\ y'_{kj} &= y_c + \frac{(y_{kj} - y_c)r}{\sqrt{(x_{kj} - x_c)^2 + (y_{kj} - y_c)^2}}. \end{aligned} \quad (10)$$

All the edges are tested with the enhancement process; then, point positions are corrected close to either the line or circle. This corrected edge is more suitable for the next step: geometric feature extraction.

3 Geometric feature extraction with circle centers and line intersections

In this section, we define the geometric features that are used for creating vector mapping descriptors (VMDs) and object matching. We defined two types of geometric features: (a) a circle and its center and (b) line segments and their intersection points. This section is divided into two parts as two main geometric features are employed. We begin by detecting the circle information, and once this step has been completed, the remaining contours are examined to determine whether they consist of any lines. The defined line segments are extended imaginarily to produce the line intersections used for salient feature points. Figure 5 shows

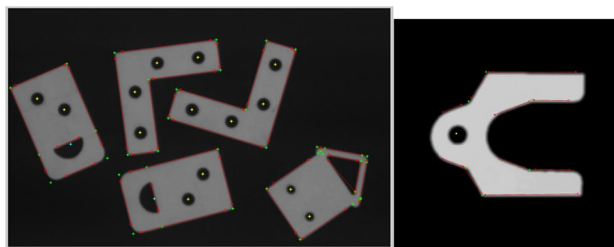


Fig. 5 Results of geometric feature extraction. The yellow dots represent the circle centers detected by the least squares circle fitting algorithm, the green dots represent line intersections, and the red line represents detected line segments from the least squares line-fitting algorithm

the results of various geometric features from geometric feature extraction.

3.1 Circle feature detection

The input contour contains information about the sequence of edges. One object has k contours $\{C_1, C_2, C_3, \dots, C_k\}$, and contour C_i has n edge points $\{C_i(x_1, y_1), C_i(x_2, y_2), C_i(x_3, y_3), \dots, C_i(x_n, y_n)\}$. The least squares circle fitting method (4) is used to represent circle $(x - x_c)^2 + (y - y_c)^2 = r^2$ along the contours [38]. According to the following equation, a , b , and c values are approximated where $c = r^2 - x_c^2 - y_c^2$. If a circle-like region is defined, it needs to be confirmed at a certain threshold with defined parameter

$$\begin{cases} \text{Circle,} & \sum_{j=1}^n r_j > \text{Circle_Error_Threshold} \\ \text{Not circle,} & \text{Otherwise} \end{cases}$$

Hence, there are two types of regions: circular and non-circular, and through this process, we obtain information on the circle radii and centers in the defined regions.

3.2 Line feature detection

3.2.1 Line-like region

Once identification of circles from the object has been completed, the remaining contour regions are indicated as line segments. Again, the object has k remaining contours $\{C_1, C_2, C_3, \dots, C_k\}$, and contour C_i has n edge points $\{C_i(x_1, y_1), C_i(x_2, y_2), C_i(x_3, y_3), \dots, C_i(x_n, y_n)\}$. The least squares line-fitting algorithm in Eq. (1) is used to detect line segments. First, one point $C_i(x_p, y_p)$ is selected from the remaining contours as the starting point for line segments if the point and its sequence of neighboring points

$\{C_i(x_{p+1}, y_{p+1}), C_i(x_{p+2}, y_{p+2}), \dots, C_i(x_{p+q}, y_{p+q})\}$ construct line L_i using Eq. (1). p represents an arbitrary point that is usually the first point of the contour, and q represents the initial number of neighboring points set by the user. The line segments need to be formed by satisfying Eq. (1) within the condition of an initial number of neighboring points q . Once the starting point $C_i(x_p, y_p)$ is established, the initial line L_i is formed and its seed starts to grow: $\{C_i(x_{p+1}, y_{p+1}), C_i(x_{p+2}, y_{p+2}), \dots, C_i(x_{p+(q+1)}, y_{p+(q+1)}), C_i(x_{p+(q+2)}, y_{p+(q+2)}), \dots, C_i(x_{p+(q+m)}, y_{p+(q+m)})\}$ until Eq. (1) is no longer satisfied. This procedure is repeated until all of the remaining contours have been inspected.

3.2.2 Line segments with least errors

Noise can cause the separation of one long line segment into several line segments, as shown in Fig. 6a. In order to detect a complete long line segment among the contours, the broken line segments need to be combined, as shown in Fig. 6b. A line segment is compared with neighboring line segments using Eq. (1) to determine whether it can be combined or left separate until detection of the line segment has completed. However, the line fitting still contains error which needs to be minimized by shortening the length of the line segments until they contain the least number of errors. The shortening process is carried out by cutting both sides of the edge pixel points $\{C_i(x_{p+1}, y_{p+1}), C_i(x_{p+2}, y_{p+2}), \dots, C_i(x_{p+(q+m-1)}, y_{p+(q+m-1)})\}$ until the error has been minimized. The value of the least squares error parameters can be determined to minimize the range of cutting of the edge pixel points [39].

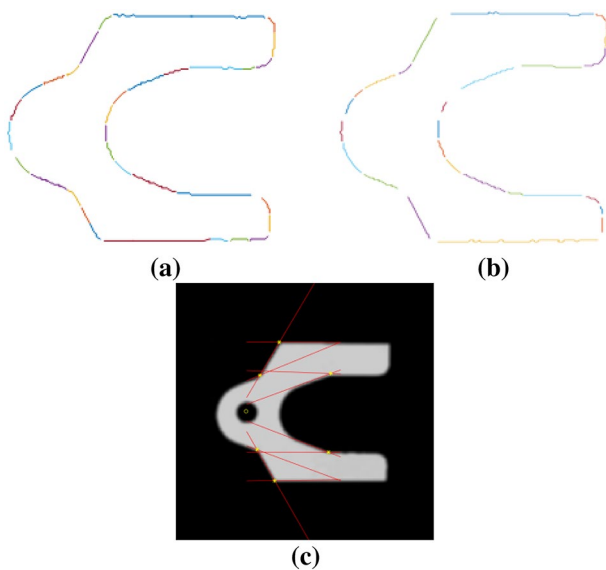


Fig. 6 **a** Line-like regions detected by the least squares line-fitting algorithm, **b** line segments containing the least errors, and **c** line intersections

3.2.3 Line Intersections

Although we now have lines with the least errors, the shortening process causes the line segments to have different lengths, which means they have different starting and ending points and results in line segments being unable to be used as salient features as well as the vertices being unstable. Hence, we apply line intersections by creating another concept of vertex. The defined extended line segments cause the intersections, and the intersection points are robust with regard to different line lengths and thus can be used as salient feature points.

4 Constructing VMDs and the matching algorithm

VMD matching is a sufficient algorithm for transformation invariance. Distance information is invariant to rotation and angle information is invariant to scale, and so both distance and angle are selected for constructing the descriptor. In this section, we describe the three steps of our proposed geometric matching method: constructing the descriptor, one-to-one feature matching, and finding the desired point with completed object matching. First, we need a descriptor that is unique for an object, then we compare each descriptor from one feature to the other features, and lastly, the position of the objects in the target are calculated according to the user setting of points for the object in the model.

4.1 Constructing VMDs

The VMD contains unique information about the object and the uniqueness of each VMD is used to recognize the objects in different scenes. In this case, the target objects could be single or multi-target or could be occluded. Our descriptors are created from geometrical features F defined at the previous section. The relationships between the defined features are represented as vectors which can be separated to form distance descriptors and angle descriptors. Figure 7 represents feature number 1 in relation to the other feature vectors that contain distance and angle. Likewise, the rest of the features from numbers 2 to 7 form the VMD. The vector of feature number 1 itself is 0, and so all the descriptors contain vector 0 for each feature.

A vector between two feature points can be represented as follows:

$$\vec{v}_i = F_i - F_j, \quad (11)$$

where $i \in (1, \dots, n)$ and $j \in (1, \dots, n)$, in which n represents the number of features.

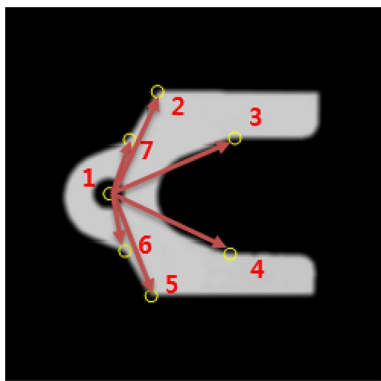


Fig. 7 The vector mapping descriptor of feature number 1

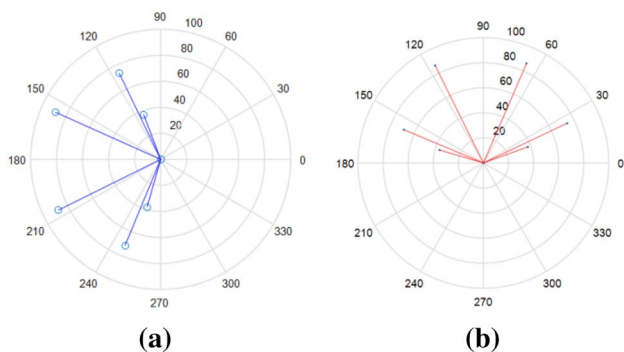


Fig. 8 Representation at one feature point for a model VMD (a) and a target VMD (b)

The Euclidean distance between corresponding feature points F_i and F_j is denoted by

$$d_{ij} = \sqrt{(F_{xi} - F_{xj})^2 + (F_{yi} - F_{yj})^2}, \tag{12}$$

and its distance descriptor is defined as

$$D = [d_1, d_2, \dots, d_n], \text{ where } d_1 = [d_{11}, d_{12}, \dots, d_{1n}]. \tag{13}$$

The angle between corresponding feature points F_i and F_j can be calculated as

$$\theta_{ij} = \tan^{-1} \frac{F_{yi} - F_{yj}}{F_{xi} - F_{xj}}. \tag{14}$$

and its angle descriptor is defined as

$$\Theta = [\theta_1, \theta_2, \dots, \theta_n], \text{ where } \theta_1 = [\theta_{11}, \theta_{12}, \dots, \theta_{1n}]. \tag{15}$$

4.2 One-to-one Feature Matching using VMD

Object matching indicates that one feature in an object can only have one corresponding feature in another object. Thus,

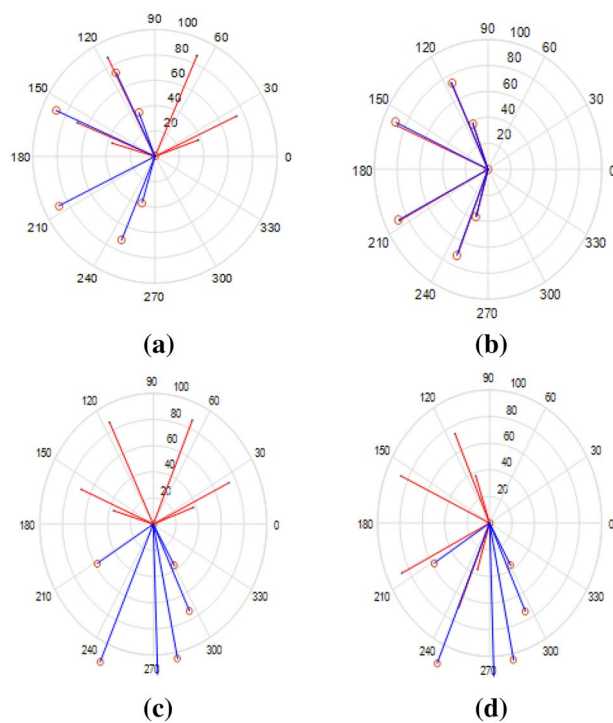


Fig. 9 Arbitrary one-to-one feature matching: a the true corresponding features, b correct matching by scale factor and rotational angle, c false corresponding features, and d incorrect matching by scale factor and rotational angle

the corresponding feature set should only contain unique conformity between features. Figure 8 shows the VMD of one feature point in which the circle segments represent the equality of distance and the bearing represents the angle.

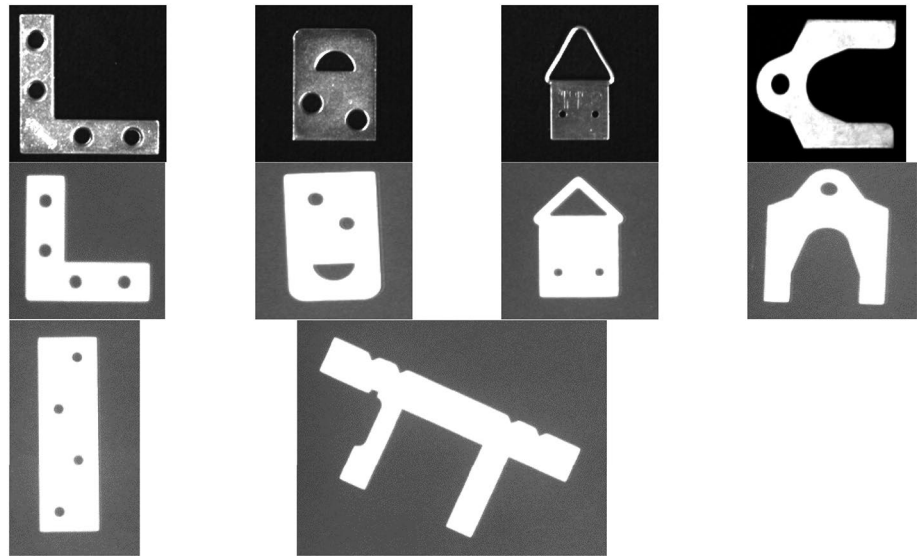
To achieve one-to-one feature matching, the object descriptor in Eqs. (13) and (15) is divided into feature descriptor distance D_i and feature descriptor angle Θ_i and thus

$$D_i = [d_i, \dots, d_n], \quad \Theta_i = [\theta_i, \dots, \theta_n]. \tag{16}$$

1. Input (Model & Target) Geometric Features
2. Calculate Model VMD, (D_M, Θ_M)
3. Calculate Target VMD, (D_T, Θ_T)
4. Calculate (σ_i, θ_i)
5. Checking scale factor and rotational angle
 - 5.1 if $(d_{Mi1} = \sigma_i d_{Ti1}) \ \&\& \ (\theta_{Mi1} = \theta_{Ti1} + \Delta\theta_i)$
 - 5.2 increase matching accumulator, $A(x,y)$
6. Check accumulator condition
 - 6.1 if $(A(x,y) > TH)$, where $TH < 0.5 * A(x,y)$
 - 6.1.1 Calculate average (σ_i, θ_i)
 - 6.1.2 Repeat from step 2 with another model descriptor
 - 6.2 else repeat from step 2 with another target descriptor
7. Collect average registration result with minimum matching error

Fig. 10 The VMD algorithm

Fig. 11 The tested model images



The model image I_M has distance descriptor D_M and angle descriptor Θ_M , and similarly, the target image I_T also has distance descriptor D_T and angle descriptor Θ_T . Hence,

$$D_{M_i} = [d_{M_{i1}} \dots d_{M_{im}}], \quad \Theta_{M_i} = [\theta_{M_{i1}} \dots \theta_{M_{im}}], \quad (17)$$

$$D_{T_i} = [d_{T_{i1}} \dots d_{T_{im}}], \quad \Theta_{T_i} = [\theta_{T_{i1}} \dots \theta_{T_{im}}], \quad (18)$$

where $i \in (1, \dots, m)$ and $j \in (1, \dots, m)$, and n and m represent the number of features in the model and target object, respectively.

The rotation and scaling relationship for arbitrary feature F from image I to feature \tilde{F} in image \tilde{I} are, respectively, denoted by

$$\sigma_i = d_{M_{ia}} / d_{T_{ib}}, \quad (19)$$

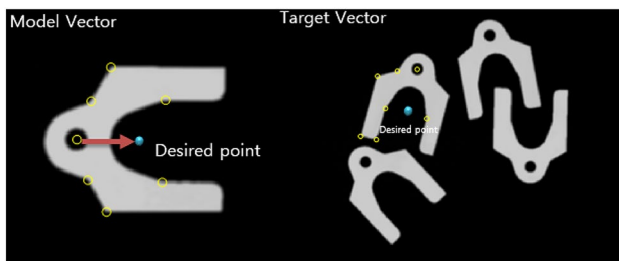


Fig. 12 Finding the desired point using the vector from each corresponding feature (the model and target have different feature points, target has two missing feature point on its right shoulder and two unwanted feature points on its left leg)

$$\theta_i = \theta_{M_{ia}} - \theta_{T_{ib}}. \quad (20)$$

Once the arbitrary scale factor and rotation angle are calculated, it is assumed that the transformed image \tilde{I} is enlarged by σ_i and rotated by θ_i , and then

$$D_{M_i} = \sigma_i [d_{T_{i1}} \dots d_{T_{im}}], \quad \Theta_{M_i} = \theta_i + [\theta_{T_{i1}} \dots \theta_{T_{im}}]. \quad (21)$$

$$E = \sqrt{(D_M - \sigma_i D_T)^2 + (\Theta_M - (\theta_i + \Theta_T))^2}. \quad (22)$$

The steps in the VMD algorithm are shown in Fig. 9. If arbitrary features correspond to each other, the feature descriptor distance and angle are similar to the enlarged and rotated feature descriptor distance and angle. Therefore, we calculate the least error E for pairwise corresponding feature points after which the counter in the accumulator increases and the process continues for the remaining elements in the feature descriptor. Otherwise, we calculate the scale factor and rotation angle for the remaining elements in the feature descriptor. The process repeats until all the elements in one feature descriptor have participated in the calculation, and then until all of the feature descriptors have been compared. Every iteration leads to one-to-one feature point matching. In addition, we define the type of geometric features as detailed in Sect. 3, and the matching calculation is only performed with the same type of geometric feature; thus, we can reduce unnecessary iterations, which reduces computational time. Figure 9b shows the true matches for four features relations out of six possible ones, and Fig. 9a, c, and d shows an example of false corresponding features where none of them match.

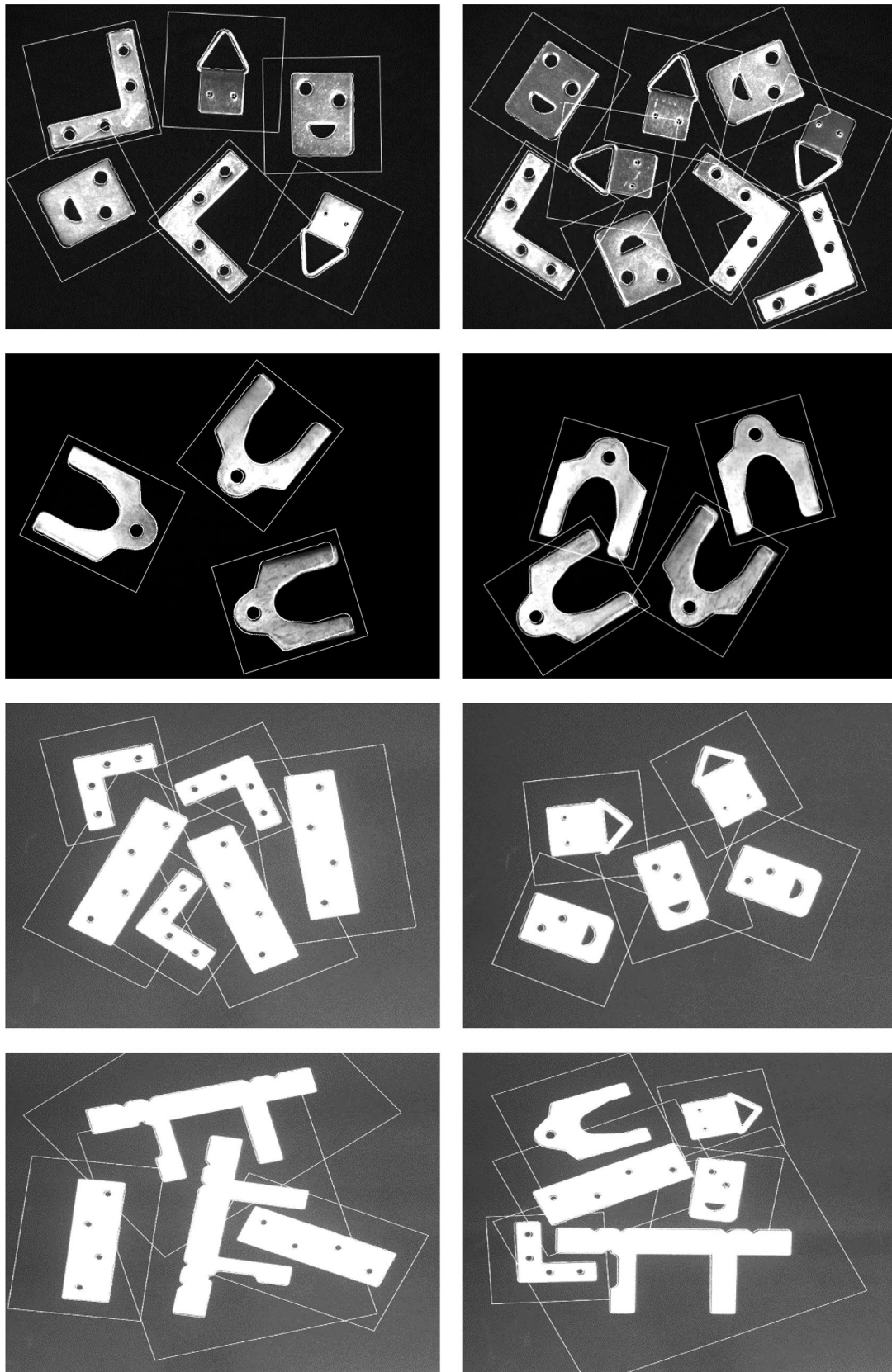


Fig. 13 Non-occluded object results

4.3 Completed object matching and desired detection point

The geometric matching between the set of corresponding feature points in two different images is derived from the rotational angle and distance (Fig. 10). Once all sets of corresponding feature points have been obtained, a comparison between the number of feature points in the model image and the target image is made, for which the features in the model image become the standard. The desired point \tilde{x}_i, \tilde{y}_i in the target image can be calculated from each corresponding feature point's characteristics: the x and y direction vectors \vec{v}_{xi} and \vec{v}_{yi} from the model, rotation angle $\Delta\theta_i$, and feature points F_{xi} and F_{yi} from the target. Hence, the desired point from the original image can be detected in transformed image by

$$\begin{aligned}\tilde{x}_i &= \cos(\theta_i) * \vec{v}_{xi} - \sin(\theta_i) * \vec{v}_{yi} + F_{xi}, \\ \tilde{y}_i &= \sin(\theta_i) * \vec{v}_{xi} + \cos(\theta_i) * \vec{v}_{yi} + F_{yi}.\end{aligned}\quad (23)$$

5 Experiments

The experiments were carried out in the following situations: clear objects on the image, deformed object features on the image, and occluded objects on the image. These three situations were conducted on both image datasets and a real-time camera.

5.1 Experimental environment

The experimental environment was set up as an industrial inspection system in which a mono-color background and simple objects for recognition were used. The mono-color background allowed for clear separation between the background and the foreground objects since clear separation helped the edge detection method described in Sect. 2. Clear edges have a close relationship with the feature accuracy described in Sect. 3, and this resulted in the desired feature detection (circle center and line intersection). In order to keep computational time at a minimum, we limited the line intersection feature to be performed on an edge and its neighbor only. Finally, the VMD algorithm was performed to identify the relationships between geometric features.

Table 1 Comparison of the three methods for a non-occluded object

	Proposed	MIL ^a	SURF ^b
Intersection over union	0.9171	0.8999	0.6149
Standard deviation	0.04214	0.06652	0.2003

^aMatrox Imaging Library

^bSpeeded up robust features

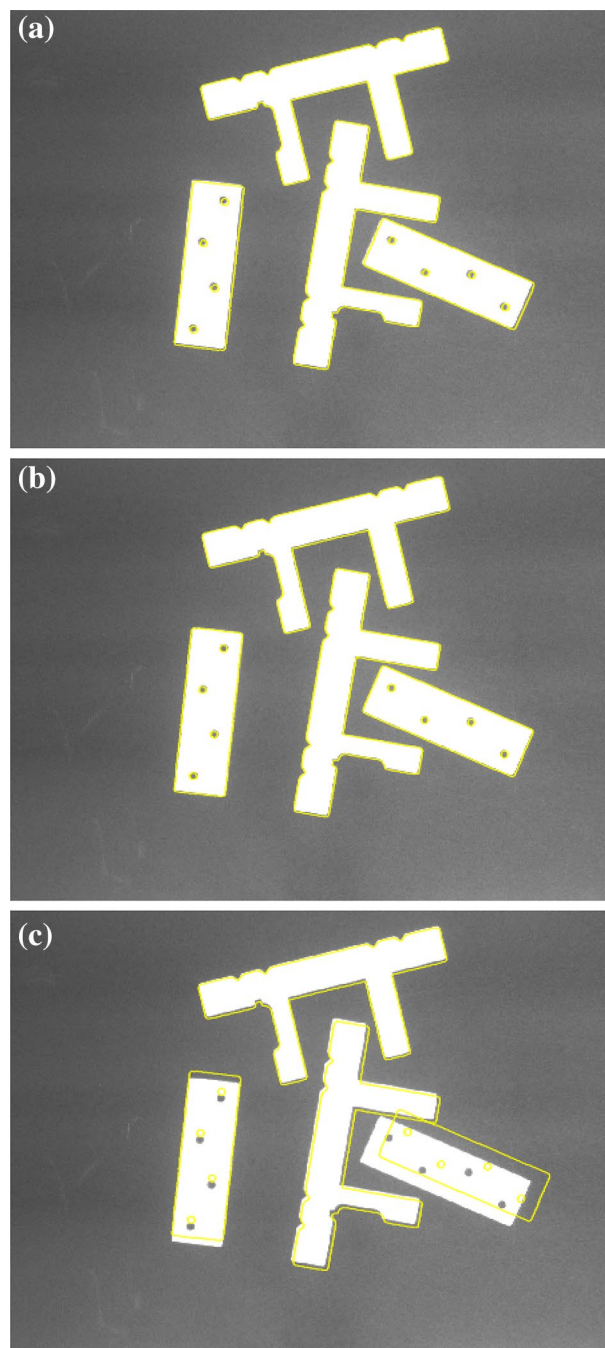


Fig. 14 An example of registration results for a non-occluded object: **a** proposed, **b** MIL, and **c** SURF

Table 2 An example of registration results of the three methods for a non-occluded object

Registration result (proposed)					Registration result (MIL)					Registration result (SURF)				
x	y	θ	σ	IoU	x	y	θ	σ	IoU	x	y	θ	σ	IoU
730.07	457.31	245.93	1	0.9673	747.76	434.55	65.93	1.001	0.9139	778.59	423.31	- 63.62	1.2066	0.4847
324.65	426.17	172.72	1	0.9505	350.58	436.64	353.11	1.002	0.9405	352.45	424.54	173.00	0.9995	0.5325
578.04	391.97	104.58	1	0.9735	578.71	391.17	104.64	1	0.9738	581.35	395.04	104.30	1.025	0.7618
551.22	139.77	38.24	1	0.9578	552.45	140.42	38	1.002	0.9518	551.77	142.65	37.38	0.9965	0.6354

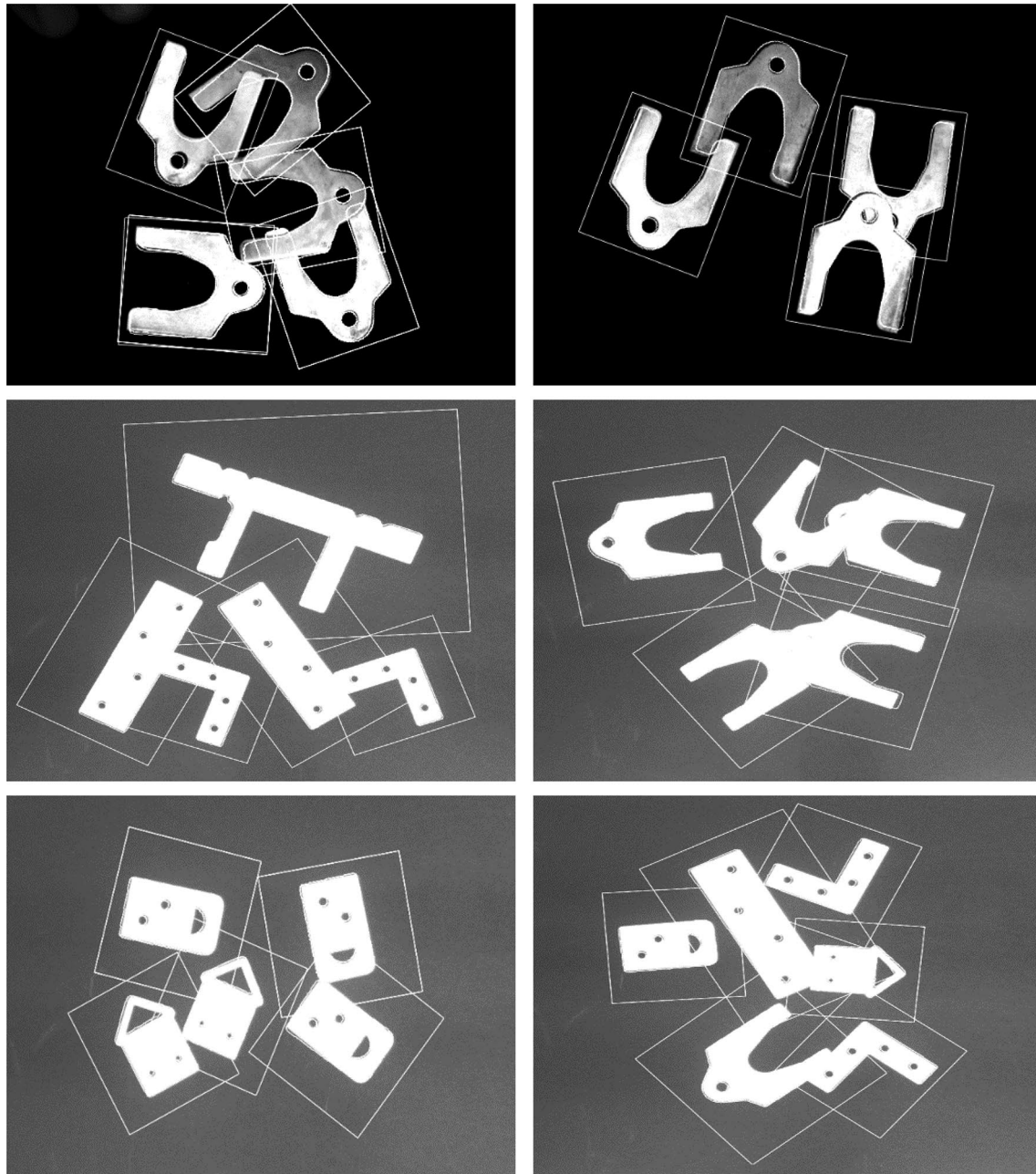


Fig. 15 Distinguishing multiple objects. The model definition is on the left and the occluded target on the right

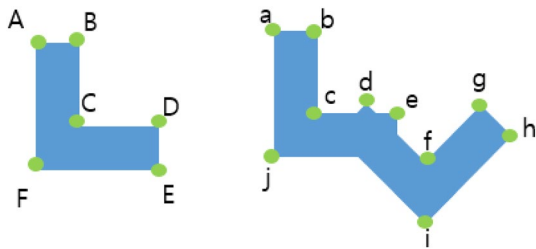


Fig. 16 Occluded object results

5.2 Matching accuracy experimental results

The matching experimental process began with non-occluded inputs separated into two types: full-featured objects and deformed objects with missing or additional features. Next, occluded input was created from the deformed objects, and various scale-different objects were tested as well. Figure 11 presents the tested models.

5.2.1 Matching Accuracy for Non-Occluded Input

The first experiment was based on a situation where the features on the model were similar to those on the target, which infers clear object recognition. The same number of features and the location of features on the objects were very close but not identical since the different images contained noise and uncertainty. There were allowable ranges of deformed features on the target matched with the model, and missing feature points resulting in less feature information caused fewer corresponding features in perhaps different locations. Figure 12 shows that an object in the model image has symmetric features located from the circle center where one of the objects in the target image has different feature locations (two features located on its left leg and missing two features at the top right). This deformation was due to the noise that each image contained and different uncertainty, which led to interference in the feature detection process. Figure 12 shows that the matching process still performed well regardless of the situation, and so we were able to discover the desired points. According to the matching process (Fig. 10), the model and target objects were defined as identical, and so the user-desired points were detected, as shown in Fig. 13 (Table 1).

The proposed algorithm had the highest intersection over union (IoU) rate with the lowest standard deviation (SD), as shown in Table 1 compared to the other methods. Even though the Matrox Imaging Library (MIL) [30] geometric model finder is a strong tool, the proposed algorithm showed

an improvement of 1.72% in IoU rate over it (Fig. 14). The speeded up robust features (SURF) algorithm had the lowest IoU rate, which is as expected from its limitations. The lower SD of the proposed method also indicates that it tended to be closer to the mean registration results (Table 2).

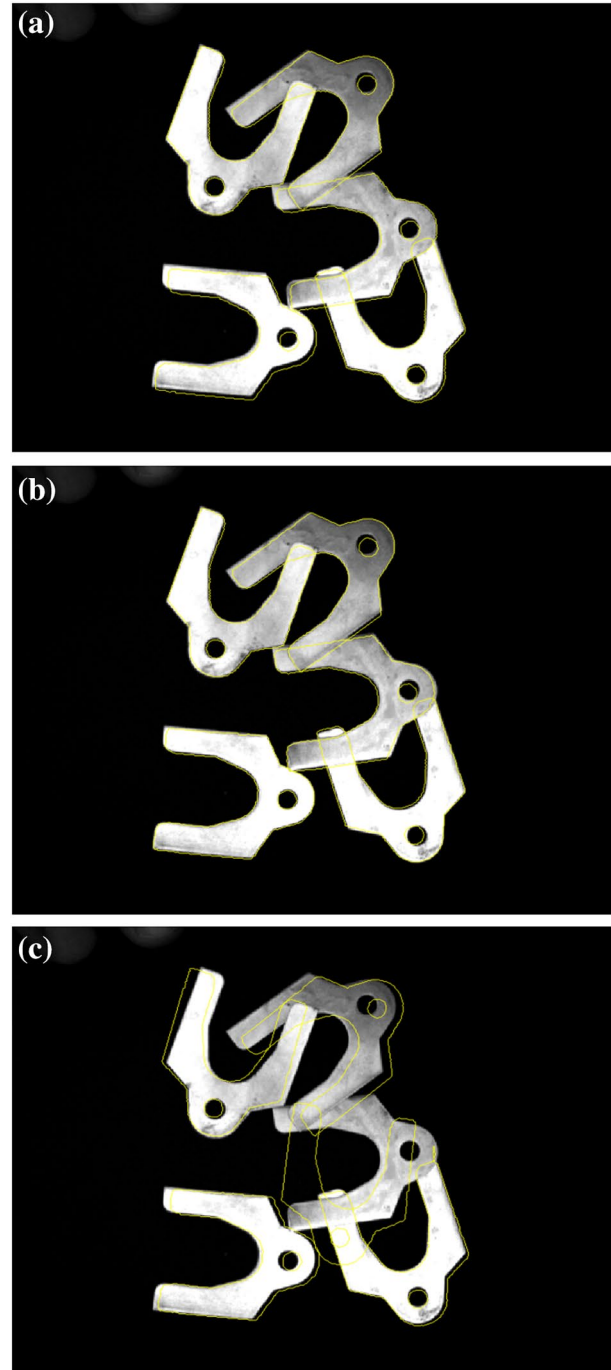


Fig. 17 An example of registration results for an occluded object: a proposed, b MIL, and c SURF

Table 3 Comparison of the three methods for an occluded object

	Proposed	MIL	SURF
IoU rate	0.8590	0.8355	0.6601
SD	0.05230	0.07122	0.18763

5.2.2 Matching accuracy for occluded input

The last experiment was conducted in a situation where the objects were overlapping and within the allowable range of feature detection, which is an extension of the second experiment. If the objects overlap, desired features might be missing and unwanted features might result from a high level of deformation. Another difference from the second experiment was that multi-object recognition was carried out instead of single-object recognition.

In this case, multiple detection is required in the region of interest. Initially, we needed to check whether the occlusion was true or not, and we used a parameter for the maximum and minimum scale factor because this needs to be reasonable in an industrial inspection system. Therefore, the maximum distance in the model by applying the parameter was compared to the distance in the target. This can be used to distinguish between multiple and single objects by whether the target distance is within the maximum distance of the model. As shown in Fig. 15, the maximum distance between model features was used as the standard for distinguishing between multiple objects. The group of features a, b, c, d, e, f, and j were candidates according to the maximum distance (A–E) in the model, while the group of features d, e, f, g, h, and i were other candidates. Thus, we were able to define that the target possibly contained two objects (Fig. 15).

If occlusion exists, the VMD of the model object and objects on target objects are compared until nothing left to compare. If two objects overlap, then the first iteration of the comparing descriptor will ascertain the one object and the other object will be matched during the second iteration. If more than two objects are occluded,

the same procedure is executed until there is nothing left to compare. Therefore, our method can handle occlusion, as shown in Fig. 16 and 17.

Table 3 exhibits that the proposed algorithm had the highest IoU rate although the object was occluded. However, the occlusion caused a lower IoU rate than the non-occlusion results due to the loss in image data (Fig. 17), which led to a higher SD as well. The proposed algorithm still outperformed the others in both accuracy and precision (Table 4).

5.2.3 Matching accuracy at various scale levels

This experiment was conducted to determine the effect of varying scale on object recognition and registration. Changes of scale may result in different positions of features. However, their distance ratio is proportional to original model, and the advantage of using angles is that they are scale invariant. Therefore, we used angles to determine the correlation of features and distance to discover the scale factor (Figs. 18, 19). We tried both scaling up and down of the model images with the following different scale levels: ± 10 , ± 20 , and $\pm 30\%$. Recognition still succeeded at ± 10 and $\pm 20\%$ applied to the model image before recognition and registration, and the scale level at $\pm 20\%$ had a higher IoU rate than at $\pm 10\%$. Furthermore, our method was successful at $\pm 30\%$ scale level of the model image with a higher IoU rate than the other scale levels (Tables 5, 6, and 7).

The proposed algorithm had the highest IoU rate at 3.75% higher than MIL and 42.13% higher than SURF, and once again, showed the lowest SD among the three algorithms. The scale difference of $\pm 20\%$ incurred the highest IoU rate, indicating that image quality was more significant than ratio of scale. Therefore, the proposed algorithm outperformed the other methods in both accuracy and precision with images at various scale levels.

Table 4 An example of registration results of the three methods for an occluded object

Registration result (proposed)					Registration result (MIL)					Registration result (SURF)				
x	y	θ	σ	IoU	x	y	θ	σ	IoU	x	y	θ	σ	IoU
238.41	353.15	174.81	1.010	0.9124	238.13	352.5	174.95	1.013	0.8995	242.29	352.92	174.79	0.9950	0.3343
232.82	148.38	68.70	1.011	0.8813	234.94	144.037	70.29	1.017	0.8689	232.36	145.65	73.86	0.9925	0.9194
410.57	345.26	109.65	1.001	0.8670	411.51	344.74	108.01	1.002	0.8722	413.99	347.70	108.49	1.0234	0.5595
335.49	119.49	218.55	1.018	0.9200	334.03	121.07	216.21	1.021	0.9192	–	–	–	–	–
369.59	251.32	189.08	1.004	0.9385	367.78	251.19	188.24	1.016	0.9425	390.1196	267.76	82.963	0.878452	0.8760

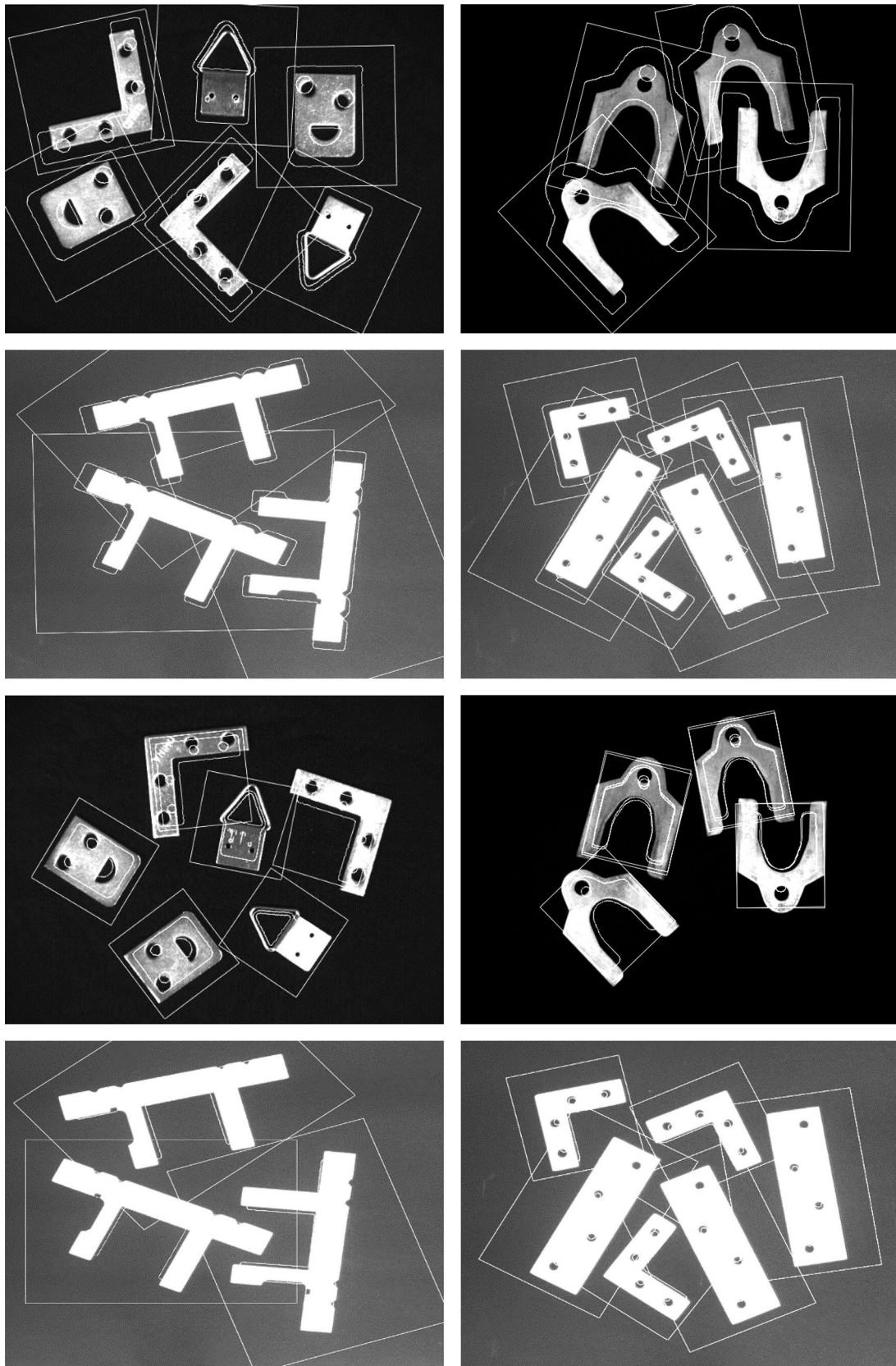


Fig. 18 Results of the objects at various scale levels

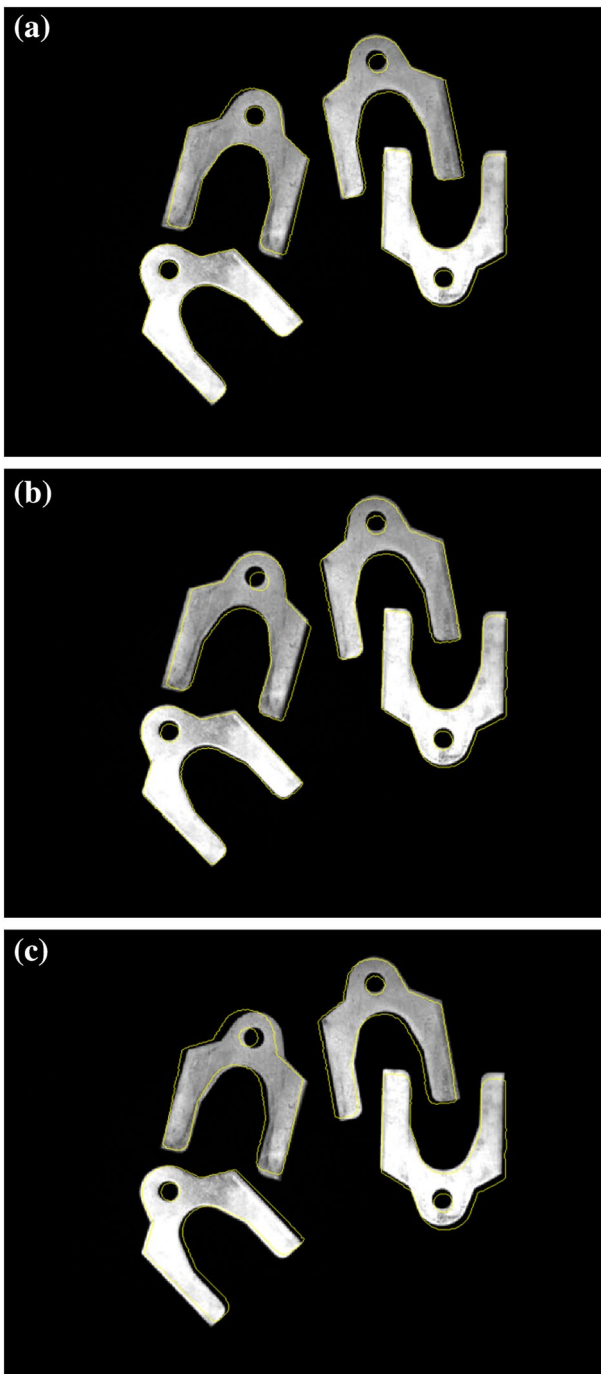


Fig. 19 Example of the registration results for a scale-altered object: **a** proposed, **b** MIL, and **c** SURF

Table 5 IoU rates comparison of the three methods at various scale levels

	Proposed	MIL	SURF
Scale ($\pm 10\%$)	0.8796	0.8557	0.4922
Scale $\pm 20\%$	0.9187	0.8898	0.5394
Scale $\pm 30\%$	0.8988	0.8386	0.4010
Avg IoU rate	0.8988	0.8613	0.4775

Table 6 SD comparison of the three methods at various scale levels

	Proposed	MIL	SURF
Scale $\pm 10\%$	0.03295	0.11774	0.14309
Scale $\pm 20\%$	0.04860	0.10581	0.17346
Scale $\pm 30\%$	0.03743	0.03619	0.20521
Avg SD	0.04407	0.10002	0.18183

6 Conclusions

Image matching and registration is the foundation for many computer vision systems and their application to areas such as navigation and security surveillance by recognition of desired objects. In this study, we started from edge detection to demonstrate how to obtain fine images from coarse images, resulting in identifying robust geometric features for creating the VMD algorithm proposed to solve the object recognition problem for which the parameter for occlusion percentage can be altered. If this parameter is set to be high, the method can handle this situation, but the accuracy decreases, and vice versa. Based on the results, we showed that the proposed matching method was invariant to arbitrary geometric transformation: translation, rotation, scale change, extra or missing feature points, and occlusion. The computational speed of the matching algorithm is also fast enough for an industrial recognition

Table 7 Example of registration results for the three methods for a scale-altered object

Registration result (proposed)					Registration result (MIL)					Registration result (SURF)				
x	y	θ	σ	IoU	x	y	θ	σ	IoU	x	y	θ	σ	IoU
408.77	111.84	280.95	0.827	0.8721	409.00	112.79	280.35	0.839	0.8597	417.84	101.81	277.61	0.5583	0.2980
465.13	235.41	89.45	0.824	0.9237	465.38	233.56	88.26	0.835	0.9017	446.90	302.07	91.61	0.2692	0.3694
255.50	170.84	256.34	0.845	0.9126	256.33	174.19	254.75	0.870	0.9153	273.39	164.98	255.45	0.8672	0.4310
214.82	320.32	313.58	0.854	0.8931	215.43	320.83	313.44	0.852	0.8731	144.51	389.13	323.09	0.7987	0.4932

system. Moreover, the VMD algorithm itself can perform in any case where the features of an object in the model and target are similar. The experimental results show that the proposed algorithm outperformed two previously reported methods in both accuracy and precision.

In future work, we will improve on the performance of VMD algorithm by finding the optimal parameters for the recognition system. Our research will focus on 3D datasets with and without occlusion, in which the distance and angle descriptors extended to the x , y , and z axes can be part of the solution. Another area of interest is representing 3D images as 2D images by projection onto a plane.

Acknowledgments The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This paper is supported by basic science research program through the National Research Foundation of Korea funded by the Ministry of Education under Grant (NRF-2016R1D1A1B01016071) and also (NRF-2016R1D1A1B03936281).

References

- Buhmann JM, Malik J, Perona P (1999) Image recognition: visual grouping, recognition, and learning. *Proc Natl Acad Sci* 96(25):14203–14204
- Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vision* 60(2):91–110
- Ke Y, Sukthankar R (2004) PCA-SIFT: a more distinctive representation for local image descriptors. In: *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition*, 2004, vol 2. CVPR, pp 2
- Mikolajczyk K, Schmid C (2005) A performance evaluation of local descriptors. *IEEE Trans Pattern Anal Mach Intell* 27(10):1615–1630
- Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). *Comput Vis Image Underst* 110(3):346–359
- Horbert E, García GM, Frintrop S, Leibe B (2015) Sequence-level object candidates based on saliency for generic object recognition on mobile systems. In: *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 127–134
- Pezzementi Z, Hager GD (2017) Tactile object recognition and localization using spatially-varying appearance. In: Pezzementi Z, Hager GD (eds) *Robotics research*. Springer, Berlin, pp 201–217
- Ling H, Jacobs DW (2007) Shape classification using the inner-distance. *IEEE Trans Pattern Anal Mach Intell* 29(2):286–299
- Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. *IEEE Trans Pattern Anal Mach Intell* 24(4):509–522
- Yang Z, Cohen FS (1999) Image registration and object recognition using affine invariants and convex hulls. *IEEE Trans Image Process* 8(7):934–946
- Wang H, Xu D (2010) A novel geometry based matching method for images with multi objects. In: *2010 8th world congress on intelligent control and automation (WCICA)*, pp. 1337–1342
- Caetano TS, Caelli T, Schuurmans D, Barone DAC (2006) Graphical models and point pattern matching. *IEEE Trans Pattern Anal Mach Intell* 28(10):1646–1663
- Zhang D, Lu G (2002) Shape-based image retrieval using generic Fourier descriptor. *Sig Process Image Commun* 17(10):825–848
- Persoon E, Fu KS (1977) Shape discrimination using Fourier descriptors. *IEEE Trans Syst Man Cybern* 7(3):170–179
- Kunttu I, Lepistö L (2007) Shape-based retrieval of industrial surface defects using angular radius Fourier descriptor. *IET Image Proc* 1(2):231–236
- Foroosh H, Zerubia JB, Berthod M (2002) Extension of phase correlation to subpixel registration. *IEEE Trans Image Process* 11(3):188–200
- Guest E, Berry E, Baldock RA, Fidrich M, Smith MA (2001) Robust point correspondence applied to two-and three-dimensional image registration. *IEEE Trans Pattern Anal Mach Intell* 23(2):165–179
- Montesinos P, Gouet V, Deriche R, Pelé D (2000) Matching color uncalibrated images using differential invariants. *Image Vis Comput* 18(9):659–671
- Pluim JP, Maintz JA, Viergever MA (2001) Mutual information matching in multiresolution contexts. *Image Vis Comput* 19(1):45–52
- Suk T, Flusser J (1996) Vertex-based features for recognition of projectively deformed polygons. *Pattern Recogn* 29(3):361–367
- Suk T, Flusser J (2000) Point-based projective invariants. *Pattern Recogn* 33(2):251–261
- Gecer B, Azzopardi G, Petkov N (2017) Color-blob-based COSFIRE filters for object recognition. *Image Vis Comput* 57:165–174
- Lin CC, Tai YC, Lee JJ, Chen YS (2017) A novel point cloud registration using 2D image features. *EURASIP J Adv Signal Process* 2017(1):5
- Asif U, Bennamoun M, Sohel FA (2017) RGB-D object recognition and grasp detection using hierarchical cascaded forests. *IEEE Trans Rob* 33:547–564
- Zhang L, Ji Q (2008, June) Integration of multiple contextual information for image segmentation using a bayesian network. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on* (pp. 1–6)
- Ferrari V, Fevrier L, Jurie F, Schmid C (2008) Groups of adjacent contour segments for object detection. *IEEE Trans Pattern Anal Mach Intell* 30(1):36–51

27. Petrakis EGM, Diplaros A, Milios E (2002) Matching and retrieval of distorted and occluded shapes using dynamic programming. *IEEE Trans Pattern Anal Mach Intell* 24(11):1501–1516
28. Fitzpatrick JM, West JB (2001) The distribution of target registration error in rigid-body point-based registration. *IEEE Trans Med Imaging* 20(9):917–927
29. Bentoutou Y, Taleb N, El Mezouar MC, Taleb M, Jetto L (2002) An invariant approach for image registration in digital subtraction angiography. *Pattern Recogn* 35(12):2853–2865
30. Cristian S, Djamel M (2006) (Matrox Electronic System Ltd., USA) Geometric hashing method for model-based recognition of an object. U.S. Patent 7,027,65
31. Karen BS, Ivan AB, Cyril CM, Yian LC (2001) (Cognex Corporation USA), Object image search using sub-models. US Patent 6,324,299
32. Haddad RA, Akansu AN (1991) A class of fast Gaussian binomial filters for speech and image processing. *IEEE Trans Signal Process* 39(3):723–727
33. Hong L, Wan Y, Jain A (1998) Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Trans Pattern Anal Mach Intell* 20(8):777–789
34. Maini R, Aggarwal H (2009) Study and comparison of various image edge detection techniques. *Int J Image Process (IJIP)* 3(1):1–11
35. Canny J (1986) A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 6:679–698
36. Trujillo-Pino A, Krissian K, Alemán-Flores M, Santana-Cedrés D (2013) Accurate subpixel edge location based on partial area effect. *Image Vis Comput* 31(1):72–90
37. Zhang TY, Suen CY (1984) A fast parallel algorithm for thinning digital patterns. *Commun ACM* 27(3):236–239
38. Gander W, Golub GH, Strelbel R (1994) Least-squares fitting of circles and ellipses. *BIT Numer Math* 34(4):558–578
39. York D (1968) Least squares fitting of a straight line with correlated errors. *Earth Planet Sci Lett* 5:320–324