

RESEARCH

Open Access



# Point matching based on affine invariant centroid trees

Wei Wang<sup>1\*</sup> , Xingwei Yan<sup>2</sup>, Ge Zhao<sup>3</sup>, Jianhua Shi<sup>1</sup> and Jin Liu<sup>1</sup>

\*Correspondence:  
wangwei8610@nudt.edu.cn

<sup>1</sup> College of Advanced  
Interdisciplinary Studies,  
National University of Defense  
Technology, Changsha, China

<sup>2</sup> College of Electronic  
Science and Technology,  
National University of Defense  
Technology, Changsha, China

<sup>3</sup> PLA Strategic Support Force,  
Beijing, China

## Abstract

Object detection can be formulated as a point matching problem when objects are modeled by point sets. Moments, which have been widely used for point matching, are limited to affine transformations as their support point sets cannot keep invariant. To address this problem, we developed an affine invariant centroid tree (AICT) to obtain a rigorous affine invariant support point set (SPS). The algorithm is constructed by a recursive process: the point set is first divided by the vector from the certain point to the centroid of the point set, and the centroids of subsets are used to generate vectors for renewed partitions. In addition, the centroids of the subsets are stored to form an AICT. The AICT represents the inherent structure of the point set. It is highly tolerant to noise and outliers due to the partitions on the whole point set. More importantly, it is affine invariant owing to the affine invariance of partition. Therefore, we can get rigorous affine invariant descriptors while moments are combined with AICT. The experimental results on synthesized and real data verify that our proposed algorithm outperforms the state-of-the-art point matching methods including shape context, iterative closet point, and the method adopting thin plate spline for rigid robust point matching (TPS-RPM).

**Keywords:** Point matching, Centroid tree, Support point set, Affine invariant descriptor

## 1 Introduction

Point matching plays an important role in computer vision, such as object matching [1, 2], image registration [3, 4], autonomous navigation [5], 3D modeling [6] and so on. It aims to find the correspondences between two point sets and/or recover the transformation that map one point set to the other. Several methods achieve point matching by alternatively finding the correspondences and estimating the transformation. The iterative closet point (ICP) [7, 8], one of the well-known heuristic approach, finds the correspondences based on the nearest neighbor and converges to the nearest local minimum of mean square distance. Different to ICP, in which point matching probabilities jump around in the binary space, TPS-RPM [9] updates the point matching probabilities gradually and continuously to improve the performance on point matching. Myronenko et al. proposed a coherent point drift (CPD) [10] while an iterated estimation framework is jointed. In CPD, the alignment of two point sets is formulated to be a probability density

estimation problem as the model points are fitted to the target points by maximizing the likelihood. Different to the above iterated estimation algorithms which are time consuming, several methods explore the correspondences based on invariant descriptors. Belongie et al. [11] proposed a descriptor named shape context (SC) to represent the coarse distribution of the rest points with respect to a given point. SC is invariant to rotation while a relative frame is used. Restricted spatial order constraints (RSOC) [12] are developed to generate an affine invariant descriptor based on the preserved adjacent spatial order. Similarly, assuming the neighbors are preserved, Zheng et al. [13] proposed a robust point matching algorithm for nonrigid shapes. However, in practice, the neighbors may be quite different due to the transformations, noise and outliers. Moments [14–16], which also have been widely used for point matching under affine transformations, share the similar limitation. These approaches devote to finding descriptors via using the algebraic methods on SPSs. The SPSs are assumed to be affine invariant [17]. Unfortunately, the assumption is not always valid as SPSs are generally composed of the neighbor points sampled by uniform spacing, arc length, affine length [18] and so on.

In this paper, to enforce the invariance of the moments, we developed an AICT. The algorithm is constructed by a recursive process: the point set is first divided into two subsets by the vector from the certain point to the centroid of the point set, and the centroids of subsets which are rigorous affine invariant are used to generate vectors for renewed partitions. In addition, the centroids are stored to form an AICT. Hence, a SPS which is constructed by the points in the AICT is affine invariant. A descriptor can be extracted from the SPS while a moment is adopted. The affine invariance of the SPS and the moment guarantees the invariance of the descriptor, and the issue of point matching under affine transformations can be formulated to descriptor matching.

The reminder of this paper is organized as follows: Sect. 2 introduces the AICT and its application in point matching. Section 3 compares the performance of our algorithm with three state-of-the-art algorithms, and followed by a conclusion in Sect. 4.

## 2 Method

### 2.1 Affine transformation and its properties

Before embarking on introducing the AICT, we briefly introduce the affine transformation, as well as its important properties based on which the AICT is proposed. A general 2D affine transformation  $T = \{A, b\}$  transforms the point  $p$  in the model point set into its corresponding point  $q$  in the target point set by  $q = Ap + b$ .  $b_{2 \times 1}$  is the translation vector and  $A_{2 \times 2}$  is the affine transformation matrix.  $A_{2 \times 2}$ , which includes rotation, scaling, and shearing transformations, can be represented as the following matrices separately:

$$A_{T \text{ scaling}} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}, \quad A_{T \text{ rotation}} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \quad A_{T \text{ shearing}} = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix} \quad (1)$$

Affine transformation has many properties, and two remarkable of them are introduced as follows.

- 1) **Centroid invariance:** centroid maps into centroid.
- 2) **Relative position invariance:** point keeps its relative position with respect to the vector  $\mathbf{v}$  under affine transformations.

As shown in Fig. 1, the relative position  $r_v^p$  of the point  $p$  with respect to the vector  $\mathbf{v}$  is

$$r_v^p = \begin{cases} 1 & y_v^p > 0 \\ 0 & y_v^p = 0 \\ -1 & y_v^p < 0 \end{cases}. \quad (2)$$

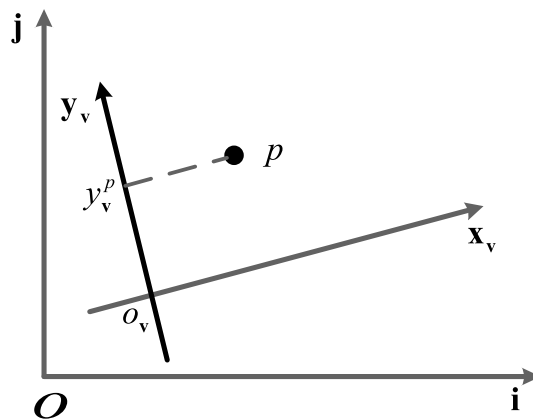
In Eq. (2),  $y_v^p$  is the  $y$ -axis coordinate of the point  $p$  in the positively oriented orthogonal frame  $(O_v, \mathbf{x}_v, \mathbf{y}_v)$ , in which the vector  $\mathbf{v}$  is taken as the  $x$ -axis  $\mathbf{x}_v$ .

Note that, this property is premised on the assumption that the unit vector of  $\mathbf{v}$  is affine invariant. For simplicity, in this paper, the assumption can be restricted to the affine invariance of the vector  $\mathbf{v}$ . In other words, the start point and end point of the vector  $\mathbf{v}$  keep invariant under affine transformations.

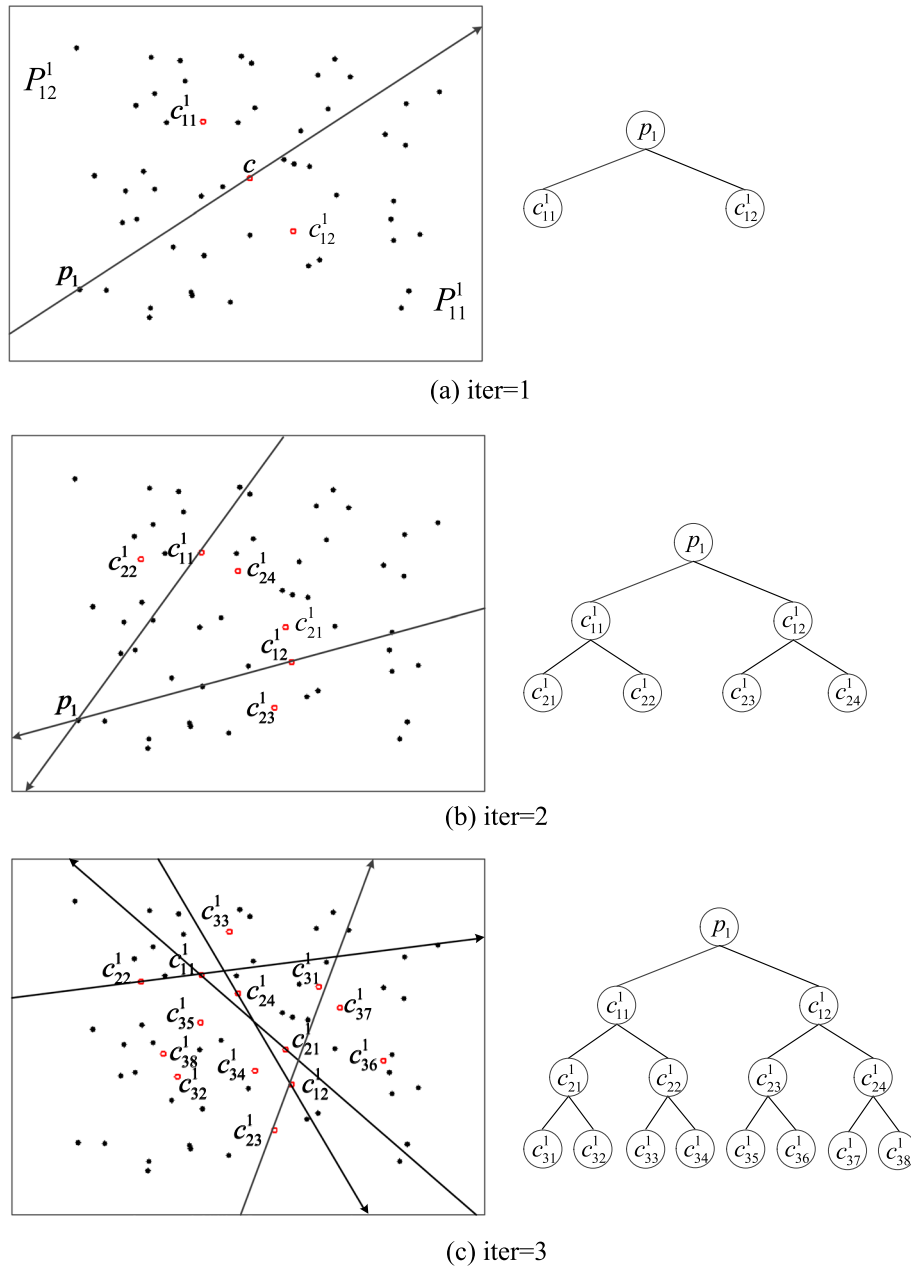
## 2.2 The construction of AICT

Relative position invariance implies that a point set  $P$  can be partitioned into three affine invariant subsets by an affine invariant vector. The three subsets are the positive subset  $P_v^+ = \{p \in P | r_v^p = 1\}$ , the negative subset  $P_v^- = \{p \in P | r_v^p = -1\}$  and the zero subset  $P_v^0 = \{p \in P | r_v^p = 0\}$ . The key problem of the partition is how to get affine invariant vectors. Fortunately, due to the centroid invariance of the affine transformation, the vector from the centroid of one subset to another fits the bill. Consequently, AICT is built via a recursive operation: the point set is first partitioned by an affine invariant vector, then the centroids of the positive subset  $P_v^+$  and negative subset  $P_v^-$  are used to generate other vectors to induce new partitions on the point set, and so on. The centroids of  $P_v^+$  and  $P_v^-$  obtained from each partition process are stored in order to construct the AICT.

Figure 2 illustrates the above operation on the point set  $P = \{p_1, p_2, \dots, p_N\}$ . To build the AICT for  $p_i$ , the key is how to find an affine invariant vector for the first iteration. Due to the affine invariance of the centroid, the centroid of  $P$  (i.e.,  $c$  in Fig. 2) is adopted and the vector from  $p_i$  to  $c$  is used to divide  $P$  into three subsets  $P_1^{i0}$ ,  $P_{11}^{i+}$  and  $P_{12}^{i-}$ . Apparently, the centroids  $c_{11}^i$  and  $c_{12}^i$  of  $P_{11}^{i+}$  and  $P_{12}^{i-}$  are both affine invariant. They are extracted and stored as the left and right son of  $p_i$ , respectively. In the AICT of  $p_i$ ,  $p_i$  is the root node.  $c_{11}^i$  and  $c_{12}^i$  are at the second level while the level of the root node is defined to be



**Fig. 1** The relative position of  $p$  with respect to the vector  $\mathbf{x}_v$



**Fig. 2** The illustration of the recursive point set partition

one. The recursive partition process can go on while the vector from the node  $c_{ij}^i$  to its father  $c_{(t-1)\text{ceil}(j/2)}^i$  is adopted to induce renewed partition, and the centroids  $c_{(t+1)(2j-1)}^i$  and  $c_{(t+1)(2j)}^i$  of the positive subset  $P_{(t+1)(2j-1)}^{i+}$  and negative subset  $P_{(t+1)(2j)}^{i-}$  are stored as the left and right son of  $c_{ij}^i$ , respectively.  $\text{ceil}(\xi)$  converts  $\xi$  to the nearest integers greater than it. The partition process needs to continue until the AICT achieves a given depth. Here, the depth is defined as the number of levels included by the AICT. For example, the depth of the AICT in Fig. 2c is 4.

### 2.3 AICT for point matching

Our point matching algorithm can be divided into two stages: descriptor calculation and matching.

The first stage devotes to getting a descriptor for each point. Since the points in the AICT are affine invariant, we can get an affine invariant SPS when the points in AICT are arranged in order. Then, an affine invariant descriptor is gotten while a moment is combined with the SPS. For example, for the point  $p_i$ , if the points in its AICT are arranged from top to down and left to right, we can get a SPS  $\{p_i, c_{11}^i, c_{12}^i, c_{21}^i, c_{22}^i, \dots, c_{n(2n)}^i\}$ . Once the SPS is obtained, the moment such as cross weighted (CW), affine invariant Fourier moment (AIFM), or diagonals of orthogonal projection matrices (DOPM) could be acquired from the SPS as the descriptor of  $p_i$ .

In the matching stage, the correspondences between points are established by descriptor matching, and target recognition is achieved based on point-to-point matching. Considering two points  $p_i$  and  $q_j$  from the model and target point set, respectively, the cost of matching two points, represented as  $c(p_i, q_j)$ , is measured via the  $\chi^2$  test statistic between the descriptors, that is

$$c(p_i, q_j) = \frac{1}{2} \times \sum_{k=1}^N \frac{|dp_i(k) - dq_j(k)|^2}{dp_i(k) + dq_j(k)}. \quad (3)$$

where  $dp_i$  and  $dq_j$  denote the descriptors of  $p_i$  and  $q_j$ , respectively. Given the set of costs between all point pairs, we measure the cost of object matching by

$$C = \min_{\rho} \sum_i c(p_i, q_{\rho(i)}). \quad (4)$$

Equation (4) is a weighted bipartite matching problem. It can be solved by the Hungarian method subjecting to the constraint that the matching is one-to-one, i.e.,  $\rho$  is a permutation. In order to have robust handling of outliers, we add “dummy” points to each point set with a constant matching cost of  $\varepsilon_d$ . In the meantime,  $C$  is treated as the matching cost between two objects, and the smaller the  $C$ , the better the localization of the corresponding object pairs is.

## 3 Results and discussion

In this section, AICT is coupled with DOPM to get a descriptor, which is named AICT-DOPM. We perform experiments both on synthetic data and real data to compare the performance of AICT-DOPM with state-of-the-art algorithms including SC, ICP, and TPS-PRM.

### 3.1 Fundamental experiments on synthetic data

In synthetic data experiments, we generate a point set, in which 100 points uniformly distribute in 2D space with mean distance between neighbor points is normalized to 1. The point set is treated as the model, and target point sets are obtained under different levels of affine transformations, noise and outliers, respectively. To get noisy target point sets, the model points are firstly transformed by a random affine transformation, and

then, the coordinates of deformed points are shifted in the range of  $[-e, e]$ . The level of noise is defined as

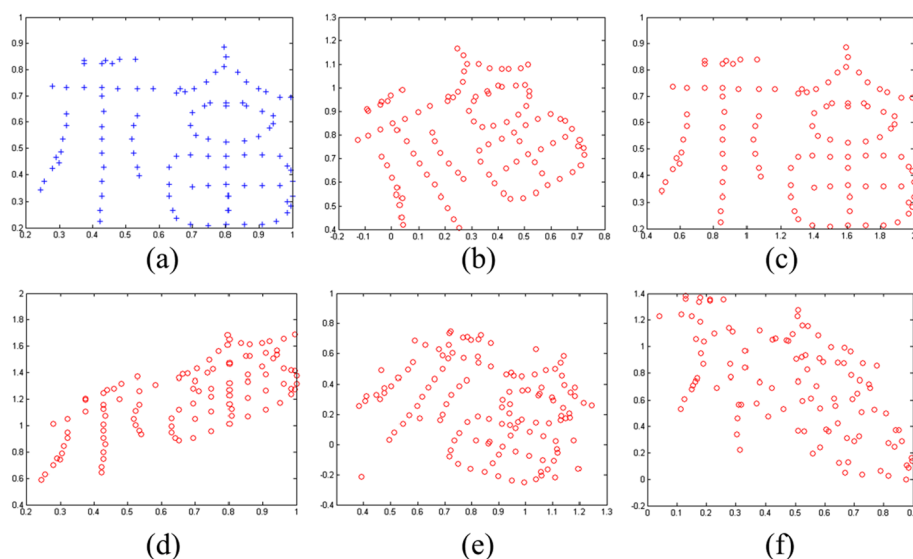
$$NSR = e/2d, \quad (5)$$

where  $d$  is the minimum distance between the point to the others in the point set. The outlier measurement, denoted by ODR, is defined as the ratio of the number of outlier to the number of original points. Figure 3 shows the distortions between the model and target point sets. The model is from Chui database.

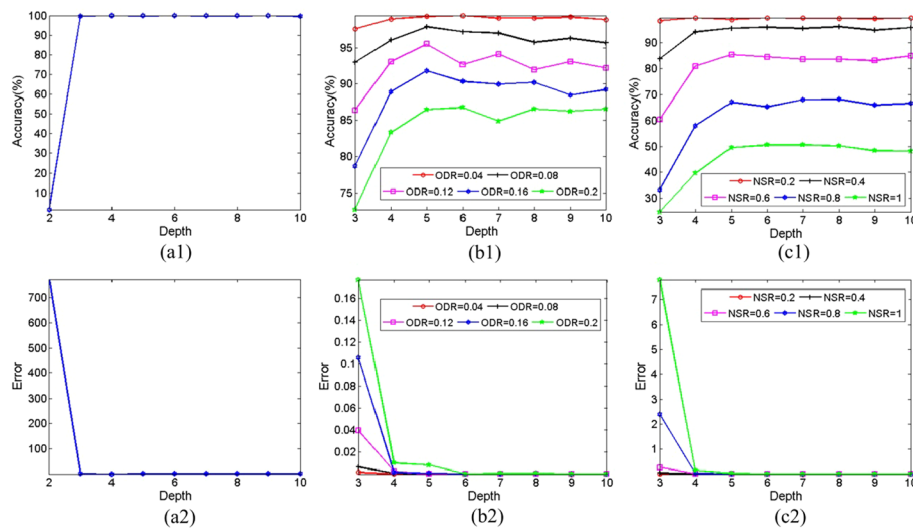
The matching accuracy of descriptors is evaluated by the number of correct matches with respect to the number of currently existing matches. In addition, the correspondences between two point sets are used to estimate the affine transformation  $T'$ . The matching error is quantified as the average Euclidean distance between the points in the transformed model point set under  $T'$  and  $T$ . All results given in this subsection are the average results based on 100 independent trials.

### 3.1.1 Effect of depth on the performance

As described in Sect. 2, an AICT with depth  $n$  contains  $2^n - 1$  affine invariant points. Apparently, the deeper the AICT is, the stronger ability it has to capture the inherent structure of the point set. However, whether we can get better performance while the depth increases, and which depth is the best choice while the performance and computational complexity are both considered. To answer these questions, we test the effect of depth on the performance of AICT-DOPM when the points in AICT are all used to construct the SPS. The performance of AICT with various depth under affine transformations, outliers and noise are given in Fig. 4. Figure 4a1, a2, the matching accuracy and matching error of AICT-DOPM, denotes that the descriptor has excellent performance when the depth of the AICT is larger than 3. In these circumstances, the



**Fig. 3** Illustration of the effect of distortions, outliers and noise on a point set. **a** An original point set, and it is transformed by **b** rotation ( $\theta = 30^\circ$ ), **c** non-uniform scaling ( $s_x/s_y = 2$ ), **d** shearing ( $k = 1$ ) or contaminated by **e** outliers ( $ODR = 0.2$ ) and **f** noise ( $NSR = 0.5$ )



**Fig. 4** Performance of AICT-DOPM with various depth. Point matching accuracy and matching error with respect to affine transformations, outliers and noise are shown in the first column (**a1** and **a2**), second column (**b1** and **b2**), and the last column (**c1** and **c2**)

matching accuracy nearly all reach 100% while the matching error drop to 0. AICT with depth 3 has poor performance if the target point set is polluted by outliers (Fig. 4b1, b2). Though the performance is becoming bad as ODR increases, it is highly improved when the depth of the AICT is larger than 5. In addition, the performance just only fluctuates slightly if the depth continues to increase. The reason is that the AICT with depth 5 can represent the global structure of the point set well. Therefore, we prefer 5 to other larger value to be the depth of the AICT for point matching under outliers. Similarly, for point matching under noise (Fig. 4c1, c2), 5 is also an available option for the depth of the AICT.

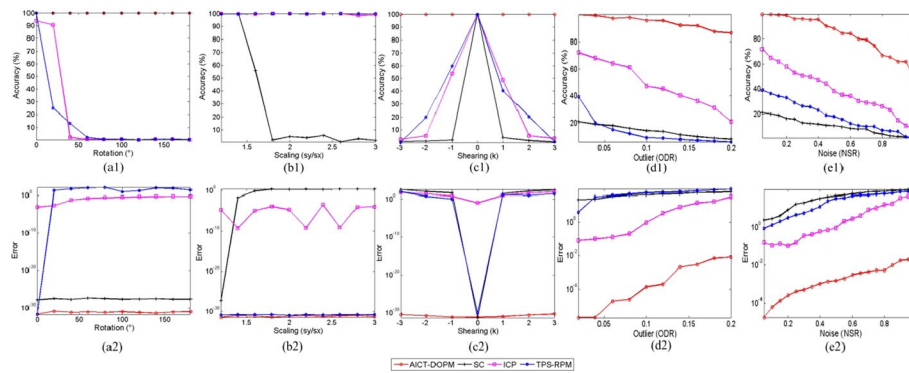
### 3.1.2 Performance to affine transformations, outliers and noise

In this subsection, the performance of AICT with depth 5 is compared with SC, ICP and TPS-RPM.

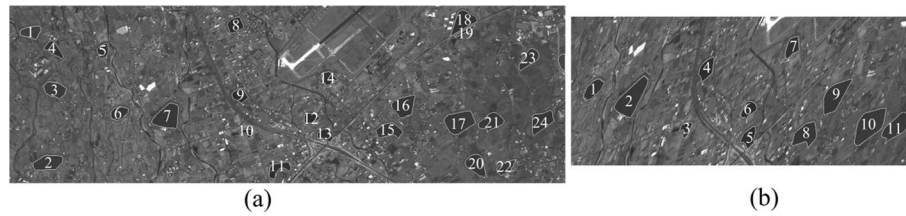
**3.1.2.1 Performance to affine transformations** An affine transformation includes rotation, scaling, and shearing transformations. The behavior of algorithms with respect to rotation is first tested. The target point sets are generated while the model point sets are rotated from  $0^\circ$  to  $180^\circ$  with  $20^\circ$  intervals. Experimental results in Fig. 5a1, a2 show that AICT-DOPM can nearly find all correspondences, whereas the matching accuracy and error of other algorithms fluctuate when the rotation angle changes. Especially for ICP and TPS-RPM, which highly depend on the initial correspondence, their performance get worse when the rotation angle is larger.

Then, the sensitivity of the descriptors with respect to scaling is evaluated. To obtain the target point sets, the model point sets are transformed while different non-uniform scaling values (i.e.,  $s_x/s_y$ ) change from 1.2 to 3 in step of 0.2. The performance of algorithms on point matching is compared in Fig. 5b1, b2, and they denote that AICT-DOPM is more robust to non-uniform scaling.





**Fig. 5** Performance of algorithms for point matching with respect to rotation (first column), scaling (second column), shearing (third column), outliers (fourth column) and noise (last column). The point matching accuracy and error are shown in the first row (a1–e1) and the bottom row (a2–e2), respectively



**Fig. 6** Real data taken over areas of Taipei by different sensors

To evaluate the behavior of the algorithms in relation to shearing, the target point sets are obtained when the model point sets are transformed according to different shearing factor  $k$ , which are  $-3, -2, -1, 0, 1, 2, 3$ . The matching results summarized in Fig. 5c1, c2 verify the invariance of AICT-DOPM to shearing.

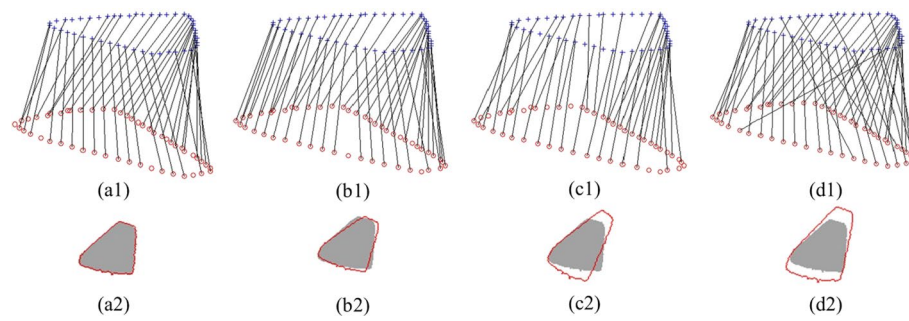
**3.1.2.2 Performance to outliers** The sensitivity of algorithms to outliers is tested when different numbers of outliers are added onto the random affine transformed model. Figure 5d1, d2 shows the matching accuracy and error against outliers, respectively. They depict that the accuracy of all algorithms decreases as the ODR increases, and AICT-DOPM has the best performance against outliers.

**3.1.2.3 Performance to noise** Finally, the effect of noise on algorithms is observed. Figure 5e1, e2, the matching accuracy and error, denotes that AICT-DOPM is most robust to noise.

### 3.2 Extended experiments on real data

The proposed algorithm can be used for object recognition once the objects are represented by point sets. In this subsection, the template image (Fig. 6a) and input image (Fig. 6b) are adopted to test the performance of our proposed algorithm on water region recognition. The two images, the real data taken over areas of Taiwan, were acquired by different sensors. The 11 water regions in the input image all have correspondences in the template image which has 24 regions. The closed water regions, which are extracted





**Fig. 7** The matching results between the No. 7 water region in the template image (blue plus sign) and the No. 2 water region in the input image (red cycle) using **a** AICT-DOPM, **b** SC, **c** ICP and **d** TPS-RPM

**Table 1** Comparisons of object recognition for real data

Input	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	Accuracy (%)
Matched	T6	T7	T10	T9	T13	T12	T14	T15	T16	T17	T21	
AICT-DOPM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100
SC	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	72.73
ICP	✓	✗	✓	✓	✓	✗	✓	✗	✗	✗	✓	54.55
TPS-RPM	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	45.45

automatically by a simply threshold segmentation, are numbered, and their contours are labeled by white color in Fig. 6. In the experiment, the water regions are treated as point sets while the contours are sampled with 100 points by uniform spacing. For each region pair, the correspondences between contour points are found by algorithms to estimate the transformations between the two images, and then, the input region is transformed to be close to the template region. Finally, the registration accuracy is measured via the ratio of the area of common domain between the template and transformed input regions to the area of the template region. The larger the registration accuracy, the similar the two regions are. Figure 7 shows the matching results between the No. 7 water region in the template image (blue plus sign) and the No. 2 water region in the input image (red cycle) using different algorithms. The point matching results are given in the top row, and in the bottom row, contours of transformed regions are plotted on the templates to show the performance of algorithms intuitively. Furthermore, the water region recognition results of algorithms are summarized in Table 1. It demonstrates that AICT-DOPM is much better at point-based object recognition than SC, ICP and TPS-RPM.

#### 4 Conclusion

In this paper, a novel AICT is proposed to generate an affine invariant SPS to refine the performance of moments. For a given point, the point set is first partitioned into two subsets by the affine invariant vector from the point to the centroid of the point set, the centroid of subsets is stored as the sons of the point. The sons and their father can form vectors to induce renewed partitions. The process will go on until the AICT achieves a given depth. The points in the AICT are arranged in order to construct a rigorous affine invariant SPS. Then, a descriptor of the point can be captured while a moment is combined. Finally, the similarity between two points is measured between the descriptors,

and the point matching-based object recognition can be achieved. The comparative analysis has been performed against three state-of-the-art algorithms including SC, ICP, and TPS-RPM on synthetic and real data, and the results denote that our proposed algorithm outperforms others in the presence of affine transformations, outliers and noise.

## Appendix

### 1. Centroid invariance: centroid maps into centroid.

**Proof** Let  $(c_x, c_y)$  and  $(c'_x, c'_y)$  are the coordinates of centroids of the model and target point sets under an affine transformation  $T = \{A, b\}$ , then we have.

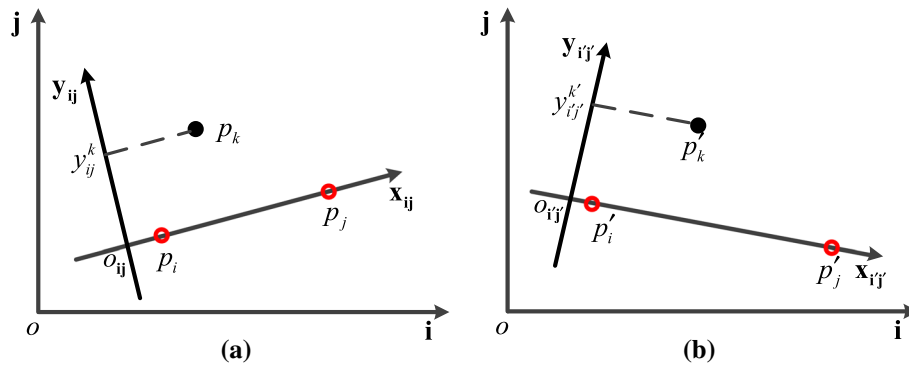
$$\left. \begin{aligned} c'_x &= \frac{\sum_{i=1}^N x'_i}{N} = \frac{\sum_{i=1}^N (Ax_i + b)}{N} = A \frac{\sum_{i=1}^N x_i}{N} + b = Ac_x + b \\ c'_y &= \frac{\sum_{i=1}^N y'_i}{N} = \frac{\sum_{i=1}^N (Ay_i + b)}{N} = A \frac{\sum_{i=1}^N y_i}{N} + b = Ac_y + b \end{aligned} \right\} \Rightarrow (c'_x, c'_y) \leftrightarrow (c_x, c_y). \quad (6)$$

' $\leftrightarrow$ ' denotes the corresponding relationship.

**2. Relative position invariance:** point keeps its relative position with respect to the vector  $\mathbf{v}$  under affine transformations.

**Proof** As shown in Fig. 8, to describe the relative position of  $p_k$  with respect to the vector from  $p_i$  to  $p_j$  (i.e.,  $\mathbf{v}_{ij}$ ) clearly, we first establish a positively oriented orthonormal frame  $(O_{ij}, \mathbf{x}_{ij}, \mathbf{y}_{ij})$ , in which  $\mathbf{v}_{ij}$  is taken as  $\mathbf{x}_{ij}$ . Then, the  $y$ -axis coordinate of  $p_k$  in  $(O_{ij}, \mathbf{x}_{ij}, \mathbf{y}_{ij})$  can be computed by

$$y_{ij}^k = (\hat{\mathbf{v}}_{ij} \times \mathbf{v}_{ik} \times \hat{\mathbf{v}}_{ij}) \cdot \hat{\mathbf{v}}_{ik} = \frac{(\mathbf{v}_{ij} \times \mathbf{v}_{ik} \times \hat{\mathbf{v}}_{ij}) \cdot \hat{\mathbf{v}}_{ik}}{|\mathbf{v}_{ij}|} \propto (\mathbf{v}_{ij} \times \mathbf{v}_{ik} \times \hat{\mathbf{v}}_{ij}) \cdot \hat{\mathbf{v}}_{ik} = \det \begin{pmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{pmatrix}. \quad (7)$$



**Fig. 8** The illustration of the relative position. **a** The relative position of  $p_k$  with respect to the vector from  $p_i$  to  $p_j$  and **b** The relative position of  $p_k'$  with respect to the vector from  $p_i'$  to  $p_j'$

where  $\hat{\mathbf{v}}$  denotes the unit vector of  $\mathbf{v}$ , ' $\times$ ' denotes the vector multiplication cross,  $\det(\xi)$  is the determinant of the matrix  $\xi$ , and  $|\xi|$  is the magnitude of the vector  $\xi$ .  $(x_i, y_i)$  is the coordinates of  $p_i$  in  $(O_{ij}, \mathbf{x}_{ij}, \mathbf{y}_{ij})$ .

Assuming the points  $p_i$ ,  $p_j$  and  $p_k$  are mapped into  $p'_i$ ,  $p'_j$  and  $p'_k$  under the affine transformation  $T = \{A, b\}$ , then the  $y$ -axis coordinate (i.e.,  $y'_{ij'}$ ) of  $p'_k$  in the positively oriented orthonormal frame in which  $\mathbf{v}_{i'j'}$  is taken as the  $x$ -axis is

$$\begin{aligned} y'_{ij'} &= (\hat{\mathbf{v}}_{i'j'} \times \mathbf{v}_{i'k'} \times \hat{\mathbf{v}}_{i'j'}) \cdot \hat{\mathbf{v}}_{i'k'} \propto (\mathbf{v}_{i'j'} \times \mathbf{v}_{i'k'} \times \hat{\mathbf{v}}_{i'j'}) \cdot \hat{\mathbf{v}}_{i'k'} \\ &= \det \left( \begin{bmatrix} x'_i & y'_i & 1 \\ x'_j & y'_j & 1 \\ x'_k & y'_k & 1 \end{bmatrix} \right) = \det(A_T) \det \left( \begin{bmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{bmatrix} \right). \end{aligned} \quad (8)$$

Correspondingly, the mathematical relationship between  $y'_{ij'}$  and  $y'_{ij'}$  can be deduced as

$$y'_{ij'} \propto \det(A_T) \cdot y_{ij}. \quad (9)$$

According to  $\det(A_T) > 0$ , Eq. (9) implies that the relative position of  $p'_k$  with respect to  $\mathbf{v}_{i'j'}$  is proportional to the one of  $p_k$  with respect to  $\mathbf{v}_{ij}$ . Substituting Eq. (9) into Eq. (2) yields

$$r_{ij}^k = r_{i'j'}^{k'} \quad (10)$$

Hence, the relative position invariance of the affine transformation is proved.

#### Abbreviations

SPS	Support point set
AICT	Affine invariant centroid tree
ICP	Iterative closet point
TPS-RPM	Thin-plate spline for robust point matching
CPD	Coherent point drift
SC	Shape context
RSOC	Restricted spatial order constraints
CW	Cross weighted
AIFM	Affine invariant Fourier moment
DOPM	Diagonals of orthogonal projection matrices

#### Acknowledgements

The authors would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions for this paper.

#### Author contributions

WW proposed the algorithm, WW and XY wrote the majority of the manuscript, and GZ, JS, and JL revised the content. All authors read and approved the final manuscript.

#### Funding

This work was supported by the National Natural Science Foundation of China under Grant 61975235.

#### Availability of data and materials

The datasets used during the current study are available from the corresponding author on reasonable request.

#### Declarations

#### Ethics approval and consent to participate

This work does not involve human participants, human data or human tissue.

**Consent for publication**

This work does not contain any individual person's data in any form.

**Competing interests**

The authors declare that they have no competing interests.

Received: 15 April 2022 Accepted: 25 August 2022

Published online: 06 September 2022

**References**

1. J. Wang, B. Wu, Z. Wang, N. Yao, J. Wu, Y. Gao, A ship radar matching method based on target attributes and point pair topological characteristic. In *2021 International Conference on Computer, Remote Sensing and Aerospace (CRSA 2021)*, Tokyo, Japan, 23–25 July 2021
2. E.B. Baruch, Y. Keller, Joint detection and matching of feature points in multimodal images. *TPAMI* (2021). <https://doi.org/10.1109/TPAMI.2021.3092289>
3. Q. Shu, X. He, C. Wang, Y. Yang, Z. Cui, Fast point cloud registration in multidirectional affine transformation. *Optik* (2021). <https://doi.org/10.1016/j.jileo.2020.165884>
4. M. Zhao, G.X. Zhang, M. Ding, Heterogeneous self-supervised interest point matching for multi-modal remote sensing image registration. *Int. J. Remote Sens.* **43**(3), 915–931 (2022)
5. S. Shi, Z. You, K. Zhao, Z. Wang, C. Ouyang, Y. Cao, A 6-DOF navigation method based on iterative closest imaging point algorithm. *Sci. Rep.* **7**(1), 17414 (2017)
6. C.C. Lin, Y.C. Tai, J.J. Lee, Y.S. Chen, A novel point cloud registration using 2D image features. *EURASIP J. Adv. Signal Process.* **2017**(1), 1–11 (2017)
7. P.J. Besl, N.D. McKay, A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992)
8. F.A. Maken, F. Ramos, L. Ott, Stein ICP for uncertainty estimation in point cloud matching. *IEEE Robot. Autom. Lett.* **7**(2), 1063–1070 (2022)
9. H. Chui, A. Rangarajan, A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* **89**(2), 114–141 (2003)
10. A. Myronenko, X. Song, Point set registration: coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(12), 2262–2275 (2010)
11. S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(4), 509–522 (2002)
12. Z. Liu, J. An, Y. Jing, A simple and robust feature point matching algorithm based on restricted spatial order constraints for aerial image registration. *IEEE Trans. Geosci. Remote Sens.* **50**(2), 514–527 (2012)
13. Y. Zheng, D. Doermann, Robust point matching for nonrigid shapes by preserving local neighborhood structures. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(4), 643–649 (2006)
14. H. R. Su, S. H. Lai, Non-rigid registration of images with geometric and photometric deformation by using local affine fourier-moment matching. Paper presented at the *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7–12 June 2015
15. Z.W. Yang, F.S. Cohen, Cross-weighted moments and affine invariants for image registration and matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(8), 804–814 (1999)
16. Z.Z. Wang, M. Liang, Y.F. Li, Using diagonals of orthogonal projection matrices for affine invariant contour matching. *Image Vis. Comput.* **29**(10), 681–692 (2011)
17. W. Wang, Y.M. Jiang, B.L. Xiong, L.J. Zhao, G.Y. Kuang, Contour matching using the affine-invariant support point set. *IET Comput. Vis.* **8**(1), 35–44 (2014)
18. F. Mokhtarian, S. Abbasi, Shape similarity retrieval under affine transforms. *Pattern Recogn.* **35**(1), 31–41 (2002)

**Publisher's Note**

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