Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Point matching based on affine invariant centroid tree for synthetic aperture radar image

Xingwei Yan
National University of Defense Technology https://orcid.org/0000-0002-6033-3638
Wei Wang (282067675@qq.com)
National University of Defense Technology https://orcid.org/0000-0002-9988-0532
Jianhua Shi
National University of Defense Technology
Jin Liu
National University of Defense Technology

Research

Keywords: synthetic aperture radar, point matching, centroid tree, affine invariant descriptor

Posted Date: April 26th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1561870/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Point matching based on affine invariant centroid tree for synthetic aperture radar image

Xingwei Yan[1], Wei Wang[2*], Jianhua Shi[2], Jin Liu[2]

1. College of Electronic Science and Technology, National University of Defense Technology, Changsha, China

2. College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha, China

Correspondence should be addressed to Wei Wang, wangwei8610@nudt.edu.cn

Abstract:

Point matching has attracted important attention in radar signal processing while the targets can be modeled by point sets to realize their recognition. Due to the different imaging parameters or viewpoints, the images captured by different synthetic aperture radar (SAR) sensors suffer from distortions. Since the distortions between images can be approximated by affine transformations, the key problem for point matching is to extract affine invariant descriptors. Moment, which has been widely used for point matching, limits to affine transformations as their support point set (SPS) can't keep invariant. To address this problem, AICT is proposed as a rigorous affine invariant SPS. It 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 which induce new partitioning. In addition, the centroids of the subsets are stored in order to form the AICT of the point. AICT, which represents the inherent structure of the point set, highly tolerant to noise

and outliers due to the global nature of our partitioning process. More importantly, it is affine invariant owing to the affine invariance of centroids and relative position. Therefore, we can get a new rigorous affine invariant descriptor while moments are computed based on the points in AICT. The experimental results on synthesized and real data validate verify that our proposed algorithm outperforms the state-of-the-art point matching methods including SC, ICP, and TPS-RPM.

Keywords: synthetic aperture radar, point matching, centroid tree, affine invariant descriptor

1 Introduction

Point matching, which can be used for target matching **Error! Reference source not found.**[2], image registration [3][4], autonomous navigation [5], and so on, plays an important role in radar application. The aim is to discover the correspondences between two point sets and/or recover the transformation that map one point set to the other. Broadly speaking, point matching can be formulated to be a chicken-and-egg problem. One variable can be determined while the other is known in advance. Several methods achieve point matching by alternatively discovering the correspondences and estimating the transformation. The iterative closet point (ICP) [6][7], one of the well-known heuristic approach, find the correspondences based on the nearest neighbor and converges to the nearest local minimum of mean square distance. Different to the ICP, where point matching probabilities jumps around in the binary space, TPS-RPM [8][8] updates the point matching probabilities gradually and continuously to improve the performance on point matching. The coherent point drift (CPD) [9] is another

method jointing an iterated estimation framework for point matching. In CPD, the alignment of two point set is formulated to be a probability density estimation problem as the model point are fitted to the target points by maximizing the likelihood. To overcome the complexity computation of iterated estimation algorithm, several point matching methods have been developed based on the invariant descriptors. Belongie et.al [10] proposed a descriptor named shape context (SC) to represent the coarse distribution of the rest points with respect to a given point. SC could be invariant to rotation while a relative frame is used. Restricted Spatial Order Constraints (RSOC) [11] is developed to generate an affine invariant descriptor based on the preserved adjacent spatial order. Similarly, RPM-LNS [12] is proposed assuming the neighbor is preserved. However, in practice, the neighbor may be quite different due to the transformation, noise and outliers. Besides, moments [13][14][15] also have been widely used for point matching under affine transformations. These approaches devote to discovering descriptors via using the algebraic methods on support point sets which are assumed to be affine invariant [16]. Unfortunately, the assumption is not always valid as the support point sets are generally composed of the neighboring points sampled by uniform spacing, arc length, affine length [17] and so on.

In this paper, we aim to develop an affine invariant support point set to enforce the invariance of the moments. For a given point p in the point set, the point set is first partitioned into three subsets (they are the negative subset, zero subset and positive subset) by the vector from the point p to the centroid of the point set, then the centroids of two subsets (i.e., the negative and positive subset) are extracted and

stored as the left and right son of p. Moreover, the vectors from the sons to their father can induce new affine invariant partitioning on the point set, and so on recursively. The centroids of each subsets obtained from each partitioning are stored in order to form a centroid tree. The centroid tree can be treated as a support point set, and a descriptor is obtained while the tree is coupled with moments. Consequently, the point matching can be achieved under affine transformations only by descriptor matching since the descriptor is affine invariant.

The reminder of this paper is organized as follows: Section 2 introduces the centroid tree and explains its affine invariant. Section 3 compares the performance of our algorithm with four state-of-the-art algorithms, and followed by a conclusion in Section 4.

2 Method

2.1 Affine transformation and its properties

Before embarking on introducing the affine invariant centroid tree (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 $q=Ap+b$. $b_{2\times1}$ is the translation vector and $A_{2\times2}$ is the affine transformation matrix including rotation, scaling, and shearing transformations which can be represented as the following matrices separately:

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

Affine mapping have many properties and two remarkable of them are introduced

as follows.

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

$$\left.\begin{aligned} c'_x &= \frac{\sum_{i=1}^{N} x'_i}{N} = \frac{\sum_{i=1}^{N} (Ax_i + b)}{N} = A\frac{\sum_{i=1}^{N} x_i}{N} + b = Ac_x + b \\ c'_y &= \frac{\sum_{i=1}^{N} y'_i}{N} = \frac{\sum_{i=1}^{N} (Ay_i + b)}{N} = A\frac{\sum_{i=1}^{N} y_i}{N} + b = Ac_y + b \end{aligned}\right\} \Rightarrow (c'_x, c'_y) \leftrightarrow (c_x, c_y). \quad (2)$$

'$\leftrightarrow$' denotes the corresponding relationship.

2) **Relative position invariance**: points keep their relative position with respect to the vector $\mathbf{v}$ under affine transformations. Note that, The property is premised on the affine invariance of the vector $\mathbf{v}$, which has affine invariant start point and end point.

The relative position $r_{\mathbf{v}}^{p}$ of the point p with respect to the vector $\mathbf{v}$ is

$$r_{\mathbf{v}}^{p} = \begin{cases} 1 & y_{\mathbf{v}}^{p} > 0 \\ 0 & y_{\mathbf{v}}^{p} = 0 \\ -1 & y_{\mathbf{v}}^{p} < 0 \end{cases} \quad (3)$$

In Eq. (3), $y_{\mathbf{v}}^{p}$ is the $y-$axis coordinate of the point p in the positively oriented orthogonal frame $(O_{\mathbf{v}}, \mathbf{x}_{\mathbf{v}}, \mathbf{y}_{\mathbf{v}})$, in which the vector $\mathbf{v}$ is taken as the $x-$axis $\mathbf{x}_{\mathbf{v}}$.

Proof:

As shown in Fig. 1, to describe the relative position of p_k with respect to the vector from p_i to p_j (i.e., $\mathbf{v}_{\mathbf{ij}}$) clearly, we first establish a positively oriented orthonormal frame $(O_{\mathbf{ij}}, \mathbf{x}_{\mathbf{ij}}, \mathbf{y}_{\mathbf{ij}})$, in which $\mathbf{v}_{\mathbf{ij}}$ is taken as $\mathbf{x}_{\mathbf{ij}}$. Then, the $y-$axis

coordinate of p_k in $(O_{\mathbf{ij}}, \mathbf{x}_{\mathbf{ij}}, \mathbf{y}_{\mathbf{ij}})$ can be computed by

$$y_{ij}^{k} = (\hat{\mathbf{v}}_{\mathbf{ij}} \times \mathbf{v}_{\mathbf{ik}} \times \hat{\mathbf{v}}_{\mathbf{ij}}) \cdot \hat{\mathbf{v}}_{\mathbf{ik}} = \frac{(\mathbf{v}_{\mathbf{ij}} \times \mathbf{v}_{\mathbf{ik}} \times \hat{\mathbf{v}}_{\mathbf{ij}}) \cdot \hat{\mathbf{v}}_{\mathbf{ik}}}{\left|\mathbf{v}_{\mathbf{ij}}\right|} \propto (\mathbf{v}_{\mathbf{ij}} \times \mathbf{v}_{\mathbf{ik}} \times \hat{\mathbf{v}}_{\mathbf{ij}}) \cdot \hat{\mathbf{v}}_{\mathbf{ik}} = \det\left(\begin{bmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{bmatrix}\right) \tag{4}$$

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 ξ, and $|\xi|$ is the magnitude of the vector ξ. (x_i, y_i) is the coordinates of p_i in $(O_{\mathbf{ij}}, \mathbf{x}_{\mathbf{ij}}, \mathbf{y}_{\mathbf{ij}})$.

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., $y_{i'j'}^{k'}$) of p_k' in the positively oriented orthonormal frame in which $\mathbf{v}_{\mathbf{i'j'}}$ is taken as the $x-$axis is

$$\begin{aligned} y_{i'j'}^{k'} &= (\hat{\mathbf{v}}_{\mathbf{i'j'}} \times \mathbf{v}_{\mathbf{i'k'}} \times \hat{\mathbf{v}}_{\mathbf{i'j'}}) \cdot \hat{\mathbf{v}}_{\mathbf{i'k'}} \propto (\mathbf{v}_{\mathbf{i'j'}} \times \mathbf{v}_{\mathbf{i'k'}} \times \hat{\mathbf{v}}_{\mathbf{i'j'}}) \cdot \hat{\mathbf{v}}_{\mathbf{i'k'}} \\ &= \det\left(\begin{bmatrix} x_i' & y_i' & 1 \\ x_j' & y_j' & 1 \\ x_k' & y_k' & 1 \end{bmatrix}\right) = \det(A_T) \det\left(\begin{bmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{bmatrix}\right) \end{aligned} \tag{5}$$

Correspondingly, the mathematical relationship between y_{ij}^{k} and $y_{i'j'}^{k'}$ can be deduced as

$$y_{i'j'}^{k'} \propto \det(A_T) \cdot y_{ij}^{k} \tag{6}$$

According to the fact $\det(A_T) > 0$, Eq. (6) implies that the relative position of p_k' with respect to $\mathbf{v}_{\mathbf{i'j'}}$ is directly proportional to the one of p_k with respect to $\mathbf{v}_{\mathbf{ij}}$. Substituting Eq. (6) into Eq. (3) yields

$$r_{ij}^{k} = r_{i'j'}^{k'} \tag{7}$$

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

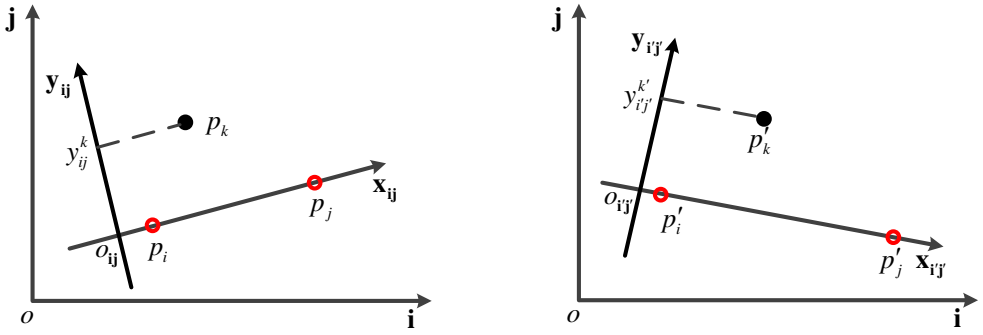

Fig.1 The illustration of relative position . a) The relative position of p_k with respect to the vector from p_i to p_j and b) The relative position of p'_k with respect to the vector from p'_i to p'_j.

2.2 The construction of AICT

Relative position invariance implies that a point set P will be partitioned into three affine invariant subsets by the affine invariant vector $\mathbf{v}$, i.e., $P = P_{\mathbf{v}}^{+} \cup P_{\mathbf{v}}^{-} \cup P_{\mathbf{v}}^{0}$, in which, the positive subset $P_{\mathbf{v}}^{+} = \{ p \in P \mid r_{\mathbf{v}}^{p} = 1 \}$, the negative subset $P_{\mathbf{v}}^{-} = \{ p \in P \mid r_{\mathbf{v}}^{p} = -1 \}$ and zero subset $P_{\mathbf{v}}^{0} = \{ p \in P \mid r_{\mathbf{v}}^{p} = 0 \}$. The key problem of the affine invariant partitioning is how to get the affine invariant vector $\mathbf{v}$. Fortunately, due to the centroid affine invariance property, the vector from one centroid of subset to another fit 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 $P_{\mathbf{v}}^{+}$ and negative subset $P_{\mathbf{v}}^{-}$ are used to generate new vectors to induce partitioning on the point set, and so on. The centroids of $P_{\mathbf{v}}^{+}$ and $P_{\mathbf{v}}^{-}$ obtained from each partitioning process are stored in order to construct the AICT.

Fig. 2 illustrates the above operation using the point p_i from the point set $P = \{p_1, p_2, ..., p_N\}$. To build the AICT for p_i, the key is how to find an affine invariant vector for the first partition. Due to the affine invariance of centroid, the

centroid of P (i.e. c in Fig. 2) is adopt and the vector from c to p_i is used to divide P into three subsets P_1^{i0}, P_{11}^{i+} and P_{12}^{i-}. Then, the centroids c_{11}^i and c_{12}^i of P_{11}^{i+} and P_{12}^{i-}, which are both affine invariant, are extracted and stored as the left and right son of the root node p_i in its AICT respectively. In the AICT of p_i, c_{11}^i and c_{12}^i are at the second level while the level of root node p_i is defined to be one. The recursive partitioning process can go on while the vector from the node c_{tj}^i to its father $c_{(t-1)ceil(j/2)}^i$ is adopt to induce partitioning and new centroids $c_{(t+1)(2j-1)}^i$ and $c_{(t+1)(2j)}^i$ of the positive $P_{(t+1)(2j-1)}^{i+}$ and negative subsets $P_{(t+1)(2j)}^{i-}$ are stored as the left and right son of c_{tj}^i. $ceil(\xi)$ converts ξ to the nearest integers greater than it. The partitioning stop if a specified depth of the AICT is achieved. Here, the depth is defined as the number of levels included by AICT. For example, the depth of AICT in fig. 3(c) is 4.

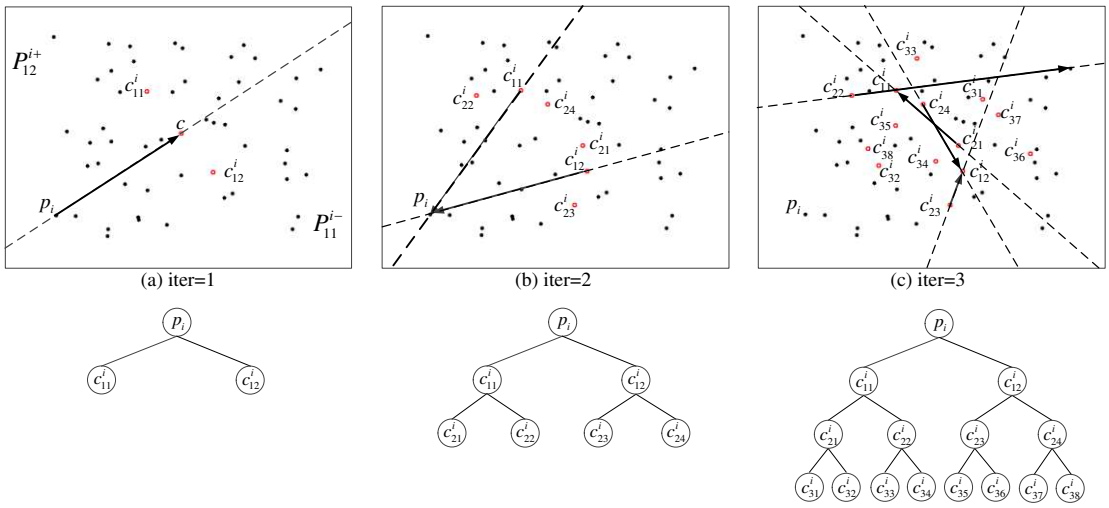

Fig.2 The illustration of the recursive point set partitioning

2.3 AICT for point matching

The process for point matching can be divided into two stages: descriptor calculation and matching.

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

In the matching stage, the correspondences between points are established by descriptor matching, and object recognition is achieved based on point-to-point matching. Consider 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 χ^2 test statistic between the descriptors, that is

$$c(p_i, q_j) = \frac{1}{2} \times \sum_{k=1}^{N} \frac{\left| dp_i(k) - dq_i(k) \right|^2}{dp_i(k) + dq_i(k)}, \tag{8}$$

where, dp_i and dq_i denote the descriptors of the p_i and q_j, respectively. Given the set of costs $c(p_i, q_j)$ between all point pairs (p_i, q_j) of two point sets, we measure the cost of object matching by

$$C = \min_{\rho} \sum_{i} c(p_i, q_{\rho(i)}). \tag{9}$$

Eq. (9) is a weighted bipartite matching problem, and can be solved by Hungarian method, subject to the constraint that the matching is one-to-one, i.e., ρ 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 ε_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 moment DOPM to get the new descriptor denoted 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 such as SC, ICP, and TPS-PRM.

3.1 Experiments on synthetic data

In synthetic data experiments, we generate a cloud of 100 points, uniformly distributed in 2D space, with mean distance between neighboring points is normalized to 1. The point cloud is treated as the model, and target data is obtained under different levels of affine transformations, noise and outliers respectively. To get the noisy target point set, the model points are firstly transformed by a random affine transformation, then the coordinates of deformed points are shifted in the range of $[-e, e]$. The level of noise is defined as

$$NSR = e / 2d \,, \tag{10}$$

where d is the minimum distance between the point to the others of 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. Fig. 3 shows the distortions between the model and target using the data 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', then matching error is quantified as the average Euclidean distance between the points in the transformed model point set under T' and their correspondences in the transformed model point set under the real transformation T. All results given in this subsection are the average results based on 100 independent trials.

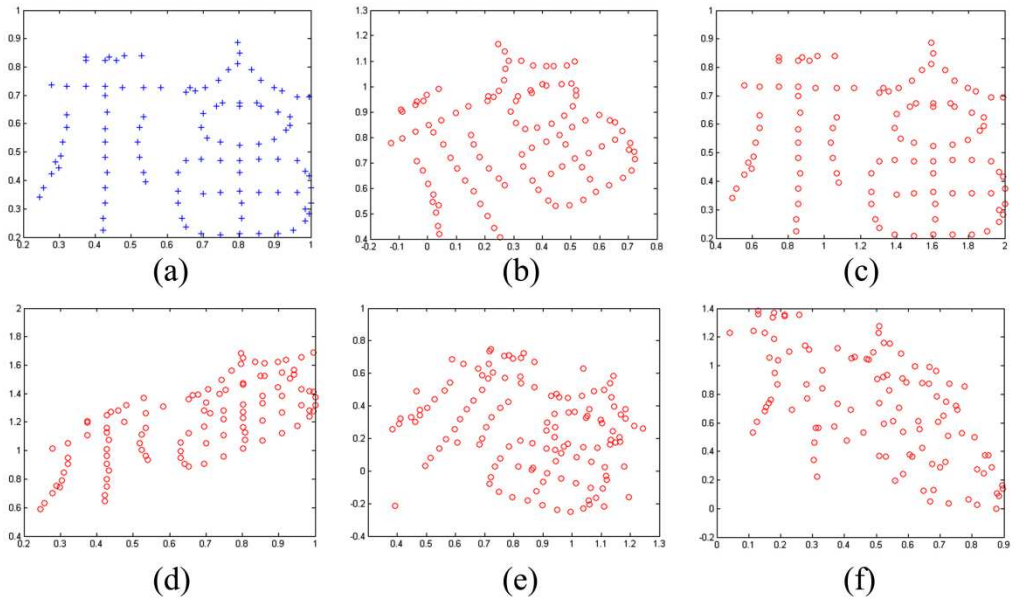

Fig. 3 a) The original point set is transformed by b) rotation ($\theta = 30^{o}$), c) non-uniform scaling ($sx / sy = 2$), d) shearing ($k = 1$) or contaminated by e) outliers ($ODR = 0.2$) and f) noise ($NSR = 0.5$).

3.1.1 The performance of AICT with various depth

As described in Section 2, a 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 the performance on point matching gets better with the increase of the depth, 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 while

all 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. Fig.4 (a1) and Fig. 4 (a2), the matching accuracy and error of AICT-DOPM under affine transformations, denote that, AICT has excellent performance when the depth is larger than 2. In these circumstances, the matching accuracy nearlly all reach 100% while the matching error drop to 0. However, AICT with depth 2 has poor performance if the target point set is polluted by outliers (Fig. 4(b1) and Fig. 4(b2)). Though the performance is becoming bad as *ODR* increases, under outliers with the same level, it is highly improved when the depth reaches 5, and it just only fluctuates slightly if the depth continue to increase. It denotes that, the AICT with depth 5 can represent the inner structure of point set well. As a result, we prefer 5 to other larger value to be the depth of AICT for the point matching under outliers. Similarly, for the point matching under noise (Fig. 4(c1) and Fig. 4 (c2)), 5 is an available option for the depth of AICT.

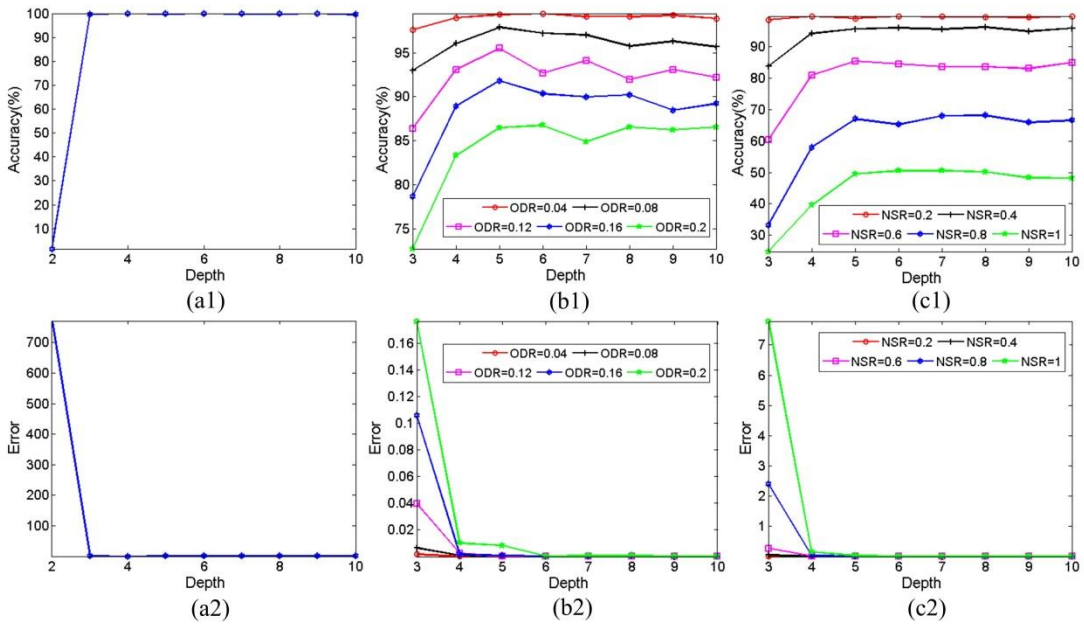

Fig. 4 Performance of AICT-DOPM with varying depth. Point matching accuracy and error with respect to affine transformations, outliers and noise are shown in the first column ((a1) and (a2)), second colunm ((b1) and (b2)), and the last column ((c1) and (c2)).

3.1.2 The 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.

The behavior of algorithms with respect to rotation are first tested. The target point sets are generated while the model point sets are rotated from 0^o to 180^o with 20^o intervals. Experimental results in Fig. 5(a1) and Fig. 5(a2) show that AICT-DOPM nearly find all correspondences, whereas the matching accuracy and error of other algorithms fluctuate when the rotation θ changes. Especially for ICP and TPS-RPM, which are highly depend on the initial correspondence, perform worse when the rotation 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 with different non-uniform scaling values (i.e., s_x / s_y) changing from 1.2 to 3 in step of 0.2. The performance of algorithms on point matching is compared in Fig. 5(b1) and Fig. 5(b2), and it denotes the robustness of AICT-DOPM against non-uniform scaling.

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. 5(c1) and Fig. 5(c2) verify the invariance of AICT-DOPM to shearing.

The sensitivity of algorithms to outlier is tested when different numbers of outliers are added onto the random affine transformed model. Fig. 5(d1) and Fig. 5(d2) show the matching accuracy and error against outlier with various levels respectively, depicting that the accuracy of all algorithms decreases as the *ODR* increases and AICT-DOPM has the best performance against outlier.

Finally, the effect of noise on algorithms is observed. Fig. 5(e1) and Fig. 5(e2), the matching accuracy and error, denote that AICT-DOPM is most robust to noise.

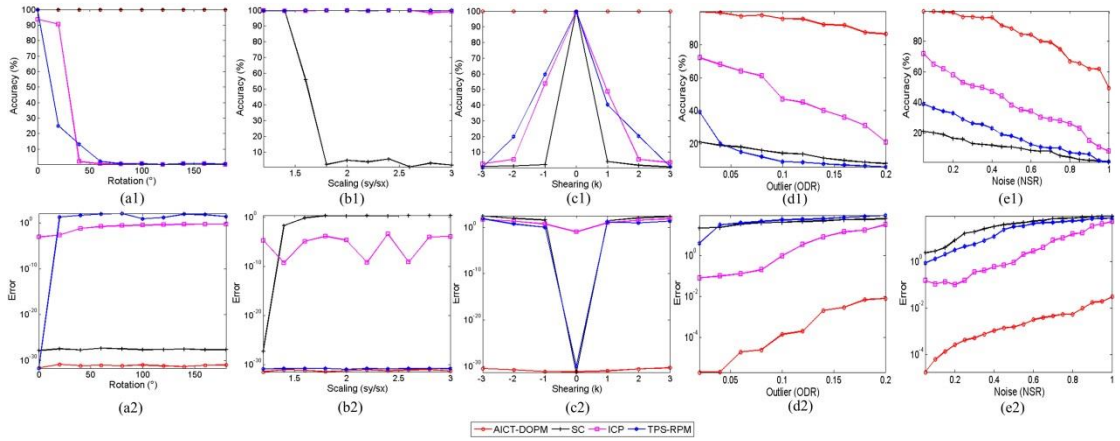

Fig. 5 Performance of algorithms for point matching with respect to rotation (first column), scaling (second column), shearing (third column), noise (fourth column) and outliers (last column). The point matching accuracy and error are shown in the first row ((a1-e1)) and the bottom row (a2-e2) respectively.

3.2 Experiments on real data

The proposed algorithm can be used for object recognition once the object can be represented by a point set. In this subsection, the template image (Fig. 6(a)) and input image (Fig. 6(b)) 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 SAR 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 automatically by a simply threshold segmentation, are numbered and their contours are labeled by white color in Fig. 6. In the experiments, the water regions are treated as point sets while the contours are sampled with 100 points by uniform spacing. For each pair of two regions from the template and input images, the corresponding contour points are discovered by algorithms to estimate the transformations between the two images, and then the input region is transformed to be close to the template region under the affine transformation. Finally, the registration accuracy is measured via the ratio of the area of common domain between the templates and transformed input regions to the area of the template region. The smaller the registration accuracy, the similar the two regions are. Fig. 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. Moreover, the registration accuracy in blacks indicates the best performance of AICT-DOPM on point matching. Furthermore, the water region recognition results of algorithms summarized in Table 1 demonstrate that, AICT-DOPM is much better at point based object recognition than SC, ICP and TPS-RPM.

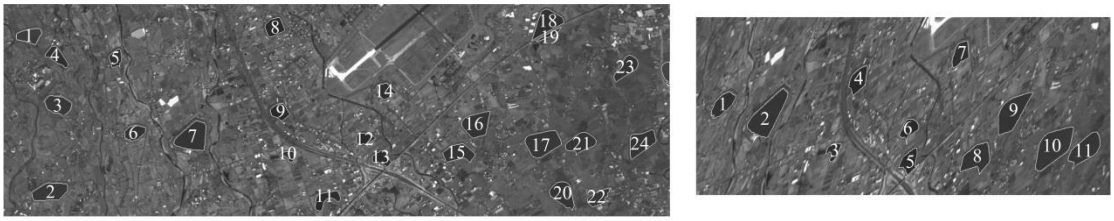

Fig.6 Real data taken over areas of Taipei by different sensors

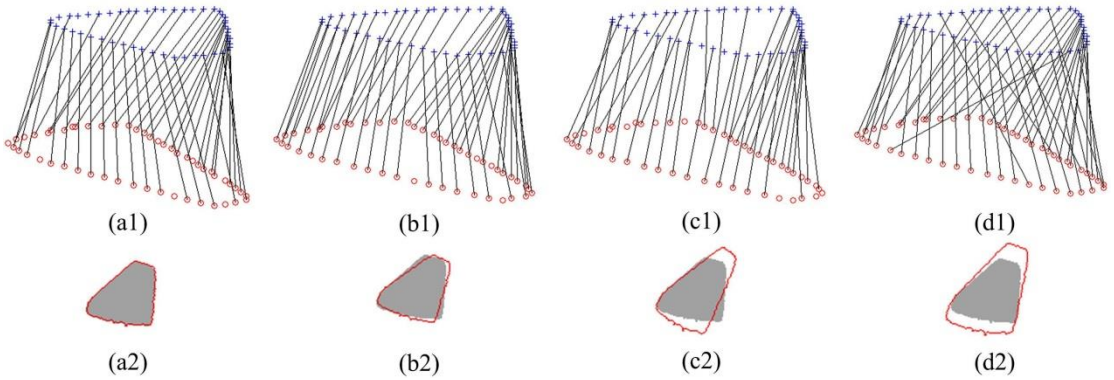

Fig. 7 The matching results between the No. 7 water region in the template image (blue plus sign) and the No. 2 water region in the input image (red cycle) using (a) AICT-DOPM, (b) SC, (c) ICP and (d) TPS-RPM.

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%

4 Conclusions

In this paper, a novel AICT, is proposed as a SPS to refine the performance of moments on point matching under affine invariant transformations. For an arbitrary

point, the point set is first partitioned into subsets by the affine invariant vector from the point to the centroid of the point set, the centroid of subsets are then used to generate new vectors which induce new partitioning on the point set, and also are ordered sequentially to form the AICT. Then the moment is generated from the AICT as the descriptor of the point. Finally, the similarity between two points is measured between the descriptors, and the point correspondence based object matching 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.

List of abbreviations

SAR: synthetic aperture radar; SPS: support point set; ICP: iterative closet point; SC: shape context; TPS-RPM: in-plate spline for robust point matching; CPD: coherent point drift; RSOC: Restricted Spatial Order Constraints; RPM-LNS: robust point matching by preserving local neighborhood structures; AICT: affine invariant centroid tree; CW: cross weighted; AIFM: affine invariant Fourier moment; DOPM: diagonals of orthogonal projection matrices.

Declarations

Availability of data and materials

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

Competing interests

The authors declare that they have no competing interests.

Funding

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

Authors' contributions

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

Acknowledgements

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

Ethics approval and consent to participate

This work doesn’t involve human participants, human data or human tissue.

Consent for publication

This work doesn’t contain any individual person’s data in any form.

Inference

[1] J. Wang, B. Wu, Z. Wang, N. Yao, J. Wu and 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, doi: 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 229, (2021). doi: 10.1016/j.ijleo.2020.165884.

[4] M. Zhao, G. X. Zhang, M. Ding, Heterogeneous self-supervised interest point matching for multi-modal remote sensing image registration, International Journal of Remote Sensing 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, Scientific Reports 7(1), 17414 (2017).

[6] P. J. Besl, N. D. McKay, A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239-256 (1992).

[7] F. A. Maken, F. Ramos, L. Ott, Stein ICP for uncertainty estimation in point cloud matching, IEEE Robotics and Automation Letters 7(2), 1063-1070 (2022).

[8] H. Chui, A. Rangarajan, A new point matching algorithm for non-rigid registration, Computer Vision and Image Understanding 89(2), 114-141 (2003).

[9] A. Myronenko, X. Song, Point set registration: coherent point drift, IEEE Transactions on Pattern Analysis and Machine Intelligence 32(12), 2262-2275 (2010).

[10] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence

24(4), 509-522 (2002).

[11] 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 Transactions on Geoscience and Remote Sensing 50(2), 514-527 (2012).

[12] Y. Zheng, D. Doermann, Robust point matching for nonrigid shapes by preserving local neighborhood structures, IEEE Transactions on Pattern Analysis and Machine Intelligence 28(4), 643-649 (2006).

[13] 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.

[14] Z. W. Yang, F. S. Cohen, Cross-weighted moments and affine invariants for image registration and matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 21(8), 804-814 (1999).

[15] Z. Z. Wang, M. Liang, Y. F. Li, Using diagonals of orthogonal projection matrices for affine invariant contour matching, Image and Vision Computing 29(10), 681–692 (2011).

[16] W. Wang, Y. M. Jiang, B. L. Xiong, L. J. Zhao, G. Y. Kuang, Contour matching using the affine-invariant support point set, IET Computer vision 8(1), 35-44 (2014).

[17] F. Mokhtarian, S. Abbasi, Shape similarity retrieval under affine transforms, Pattern Recognition. 35(1), 31–41 (2002).