Robust Identification of Locally Planar Objects Represented by 2D Point Clouds under Affine Distortions

Dominic Mai, Thorsten Schmidt, and Hans Burkhardt

Univeristy of Freiburg, Computer Science Department
79110 Freiburg i. Br., Germany
{maid, tschmidt, burkhardt}@informatik.uni-freiburg.de

Abstract. The matching of point sets that are characterized only by their geometric configuration is a challenging problem. In this paper, we present a novel point registration algorithm for robustly identifying objects represented by two dimensional point clouds under affine distortions. We make no assumptions about the initial orientation of the point clouds and only incorporate the geometric configuration of the points to recover the affine transformation that aligns the parts that originate from the same locally planar surface of the three dimensional object. Our algorithm can deal well with noise and outliers and is inherently robust against partial occlusions. It is in essence a GOODSAC approach based on geometric hashing to guess a good initial affine transformation that is iteratively refined in order to retrieve a characteristic common point set with minimal squared error. We successfully apply it for the biometric identification of the bluespotted ribbontail ray *Taeniura lymma*.

1 Introduction

Euclidean motion of planar objects in 3D is equivalent to affine transformations in 2D if we assume parallel projection neglecting occlusion. It is often possible to robustly extract interest points from images that suffice to uniquely identify a class of objects or even individual entities [1–3]. The identity of two clouds under some transformation model can be established by a global invariant feature [2, 4] or by aligning the two clouds [5]. Global features for point clouds like shape contexts [6] or features derived by integrating a local feature over the whole structure [4] are fast to compare as we only need to compute distances in the feature space. Such features, however, cannot deal well with outliers as every point affects the value in the feature space. Point registration performs much better in the presence of outliers: once a valid transformation is found, a similarity measure based on point correspondences is not affected. Rigid motion of non-planar objects generally requires the construction of a 3D model to be able to model the transformation in two dimensions [2]. In this paper we exploit the fact that many objects possess partly planar surfaces and therefore can be partly modeled with an affine transformation[7]. Sample applications are depicted in Fig. 1.

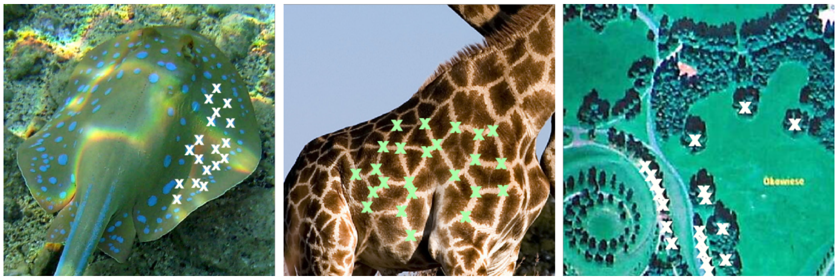


Fig. 1. Possible applications of the point registration algorithm include noninvasive wildlife monitoring (a,b) and place recognition in mobile robotics (c).

Formalization of the Problem Let $\mathcal{C}$ be a point cloud containing points $p_i = (x, y, 1)^{\mathrm{T}}$ in homogeneous coordinates. Let the common point set of two clouds $\mathcal{C}_i$ under the affine transformation $\boldsymbol{A}$ be:

$$\mathcal{X}_{\boldsymbol{A}}(\mathcal{C}_1, \mathcal{C}_2) = \{(p_1, p_2) \quad | \quad p_1 \in \mathcal{C}_1, p_2 \in \mathcal{C}_2 : \|\boldsymbol{A}p_1 - p_2\| < \delta_{corr}\} \quad . \tag{1}$$

We assume a correspondence (p_1, p_2) if the Euclidean distance of p_2 and the transformed point p_1 is smaller than a threshold δ_{corr}. We define the *characteristic common point set*

$$\hat{\mathcal{X}}_{\boldsymbol{A}}(\mathcal{C}_1, \mathcal{C}_2) \subseteq \mathcal{X}_{\boldsymbol{A}}(\mathcal{C}_1, \mathcal{C}_2), \quad |\hat{\mathcal{X}}_{\boldsymbol{A}}(\mathcal{C}_1, \mathcal{C}_2)| \geq \#_{min} \tag{2}$$

as a subset of $\mathcal{X}_{\boldsymbol{A}}$, holding enough correspondences to identify the object that the point clouds originate from.

The minimum cardinality $\#_{min}$ of $\hat{\mathcal{X}}_{\boldsymbol{A}}(\mathcal{C}_1, \mathcal{C}_2)$ depends on the application. For biometric identification, $\#_{min}$ usually is a small value: The Battley System reports two fingerprints identical when seven correspondences between minutiae have been found[8], the seven most significant *Eigenface* coefficients suffice to describe a face[9].

A fast and well understood method for aligning point clouds is the *Iterative closest Point* algorithm[10] (ICP). The ICP needs a good initial alignment of the point clouds and is hence not suitable for our problem by itself, but it is useful to refine a transformation once a coarse initial guess has been made.

For recovering an affine transformation we need to find at least three corresponding points. Local descriptors like SIFT[1], Spin Images[7] or orientation of surface normals[11] are popular features to solve this task. In absence of such descriptors, one can use invariant features based on the geometric configuration of the points. [3, 12, 13] encode the points of a cloud relative to all possible 3 point bases and use a generalized hough voting to find a corresponding basis. Aiger et al.[14] use a feature based on area ratios to find a corresponding basis to a fixed basis of 4 points. A popular philosophy to speed up the search for correspondences is is geometric hashing [3, 11, 12]: Points or groups of points are indexed with with a feature that remains invariant under the assumed transformation. When using groups of points two additional problems arise: *1.* One has to make sure that the sampling strategy remains unaffected by the allowed transformations of the cloud. Sampling all $\binom{n}{3}$ combinations [3, 12] is only practicable for small clouds. *2.* The feature for indexing also has to be invariant against permutations of the input – to the best of our knowledge none of the existing affine invariant features fulfills this property.

Contribution and Overview In this paper we present a novel algorithm to find a characteristic common point set of two point clouds under the assumption of an affine transformation. Our contributions are:

1. a family of affine invariant descriptors $T_{\boldsymbol{\rho}}$ for sets of four points in arbitrary order based on area ratios,
2. a novel strategy to partition the point cloud into a linear number of those local four-point-neighborhoods based on a Voronoi decomposition, and
3. a point registration algorithm based on geometric hashing to identify a characteristic common point set.

The remainder of the paper is organized as follows: In Section 2 we will introduce the invariant mapping $T_{\boldsymbol{\rho}}$, in Section 3 we present our cloud partition algorithm, and in section 4 we introduce the point registration based on the former two sections. In section 5 we show the applicability of our algorithm for the biometric identification of blue spotted ribbontail rays and conclude in Section 6.

2 Affine Invariants for 4 Points

An invariant mapping T of a pattern x is a function that maps all members of an equivalence class ϵ_G under a transformation group G into one point of the feature space:

$$x_i \overset{G}{\sim} x_j \Rightarrow T(x_i) = T(x_j) \quad . \tag{3}$$

As this necessary condition for invariance can already be achieved by a simple constant function that maps the same value to every input, we also require

$$T(x_1) = T(x_2) \Rightarrow x_1 \overset{G}{\sim} x_2 \quad , \tag{4}$$

which would assure completeness [15]. This is difficult to achieve and might also be difficult to prove in a domain with possibly infinite different patterns. We therefore aim to construct invariants with a high degree of *separability*, which is completeness on a subset of relevant patterns. Furthermore, we demand *continuity* of the invariant mapping T to be able to deal with noisy data, i.e. small changes in the pattern result in small changes in the feature space.

Four not all collinear points p_i[1]

$$\boldsymbol{P} = \{p_1, p_2, p_3, p_4\} \tag{5}$$

[1] in homogeneous coordinates like in Eqn. 1

define four triangles $\triangle_i$ with at most one degenerate triangle having an area of zero. We will call such a point set $\boldsymbol{P}$ a *4 point affine set* (4PAS). Without loss of generality we order the triangles ascending based on the area:

$$\text{Area}(\triangle_i) \leq \text{Area}(\triangle_j) \Leftrightarrow i < j. \tag{6}$$

The sequence of triangles with equal areas is not important. We compute six area ratios

$$R_{ij} = \frac{\text{Area}(\triangle_i)}{\text{Area}(\triangle_j)} \quad \forall i < j. \tag{7}$$

The Area ratio fulfills the following properties:

1. Due to the sorting, we have $R_{ij} \in [0, 1]$.
2. The area can be computed using the determinant which is a continuous function in the point coordinates. Hence the area ratio is also continuous. It can be shown that this also holds in the case of a change in the ordering of triangles caused by coordinate noise.
3. The area ratio $\frac{\text{Area}(\triangle_1)}{\text{Area}(\triangle_2)} = \frac{\text{Area}(\blacktriangle_1)}{\text{Area}(\blacktriangle_2)}$, with $\blacktriangle_i$ being an affine transformed version of $\triangle_i$, is invariant under affine transformations[16].

In the following, we will index the six ratios R_{ij} with a single index for ease of notation as the order is not important. We can now introduce the invariant mapping function $T_{\boldsymbol{\rho}} : \mathbb{R}^{3\times 4} \to [0, 1]$

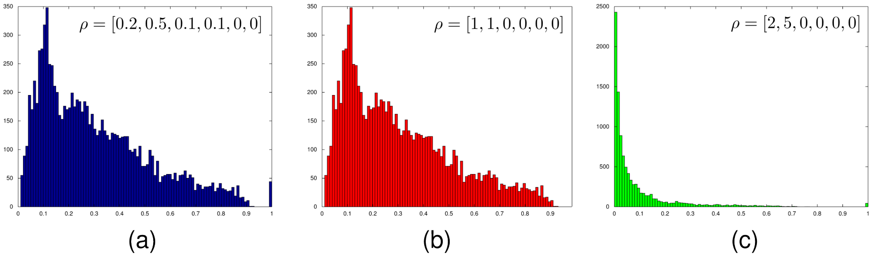
$$T_{\boldsymbol{\rho}}(\boldsymbol{P}) = T_{\boldsymbol{\rho}}(R_1, \ldots, R_6) = \frac{1}{6!} \sum_{\pi \in S_6} R_{\pi(1)}^{\rho_1} \cdot \ldots \cdot R_{\pi(6)}^{\rho_6} \quad . \tag{8}$$

The mapping $T_{\boldsymbol{\rho}}$ is a symmetric polynomial parameterized with a set of exponents $\boldsymbol{\rho} \in [0, \infty)^6$, i.e. the positive part of the $\mathbb{R}^6$. We need this restriction to assure that $R_{ij} \in [0, 1]$. In order to be invariant to a permutation of the input we integrate over the symmetric group S_6.

We want to use $T_{\boldsymbol{\rho}}$ as an indexing function for a hash table. Therefore we aim to find a parameterization $\boldsymbol{\rho}$ that yields an invariant distribution as uniform as possible, as this is optimal for hashing[13]. The shape of the distribution depends on the underlying population of 4PAS and the parameterization ρ. Fig. 2 illustrates the impact of the parameterization ρ. Note that the hashing will work with any kind of distribution, the degree of uniformity only has an impact on the number of candidates within a tolerance level. We did not elaborately analyze this part of the problem and choose $\rho = [1, 1, 0, 0, 0, 0]$ as parameterization for the further experiments, as it yields the most uniform distribution among the parameterizations tested.

3 Voronoi Decomposition

With geometric hashing, we want to establish an initial pairing between two 4PAS $\boldsymbol{P}$ that lie on a characteristic common point set (Eqn. 2) of two objects.



(a) (b) (c)

Fig. 2. The histograms show invariant distributions for 10.000 4PAS (see Sec. 3), sampled from a typical population of *Taeniura lymma* (see Sec. 5) under different parameterizations ρ. The distribution (c) is not suitable for hashing.

As we do not make any assumptions for the initial configurations of the point clouds, we need to partition the cloud into local neighborhoods[2] of four points independent of the cloud's orientation.

The *Voronoi decomposition* partitions the $\mathbb{R}^2$ into disjunct cells based on a set of centroids $\mathbf{c}_i$. For each cell holds that every point of the cell is closer to its centroid $\mathbf{c}_i \in \mathcal{C}$ than to every other centroid $\mathbf{c}_j \in \mathcal{C}$:

$$\mathcal{V}_{cell}(\mathbf{c}_i) = \{p \quad | \quad \forall j \neq i \quad \|\mathbf{c}_i - p\| < \|\mathbf{p}_j - p\|\} \quad . \tag{9}$$

Note that all the $\mathcal{V}_{cell}(\mathbf{c}_i)$ are disjoint, but their union does not equal the $\mathbb{R}^2$ as the borders of the cells are not part of the cell. These borders are referred to as *Voronoi segments* and can be defined in the following way:

$$\mathcal{V}_{seg}(\mathcal{C}) = \mathbb{R}^2 \setminus \bigcup_{\mathbf{c}_i \in \mathcal{C}} \mathcal{V}_{cell}(\mathbf{c}_i) \tag{10}$$

These segments represent the borders of the distinct Voronoi cells. All points on a segment are equidistant to at least two Voronoi sites. The intersection points of the borders define a *Voronoi node*:

$$\mathcal{V}_{nodes}(\mathcal{C}) = \{\mathbf{v} \quad | \quad \exists \mathbf{c}_i, \mathbf{c}_j, \mathbf{c}_k \in \mathcal{C} : \|\mathbf{c}_i - \mathbf{v}\| = \|\mathbf{c}_j - \mathbf{v}\| = \|\mathbf{c}_k - \mathbf{v}\|\} \, . \tag{11}$$

We will construct the set $\mathcal{P}$ containing the 4PAS $\boldsymbol{P}$ by extracting the four nearest neighbors for every Voronoi node $\boldsymbol{v} \in \mathcal{V}_{nodes}(\mathcal{C})$ (See Fig. 3 for an illustration). The construction of a Voronoi decomposition takes $O(n \log(n))$ time and contains $O(n)$ Voronoi nodes. The distance queries are issued on a *kd-tree* that also takes $O(n \log(n))$ for construction and $O(log(n))$ for a nearest neighbor query. The overall preprocessing time for a point cloud hence is in $O(n \log(n))$.

This partitioning strategy is canonical for similarity transformations as relative point distances do not change. It therefore is also valid for affine distortions that do not contain strong skews or strong anisotropic scaling. The number of 4PAS $\in \mathcal{P}$ is linear in the number of points in the point cloud.

[2] The points of the characteristic common point set originate from a locally planar part of the object.

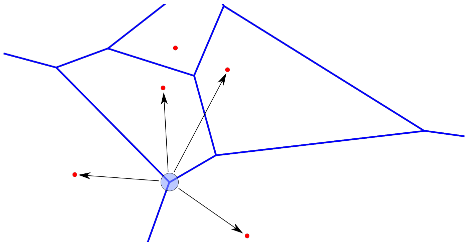


Fig. 3. A Voronoi node with its four nearest neighbors.

4 Point Registration

The goal of the point registration is to find the affine transformation $\boldsymbol{A}$ that defines a characteristic common point set (Eqn.2) of two point clouds $\mathcal{C}_i$. We use *geometric hashing* to establish an initial correspondence between two 4PAS that are indexed by the invariant T_ρ (Eqn. 8). This seed is used to compute an initial transformation which is iteratively refined to obtain the transformation that minimizes the quadratic error between the assumed point correspondences. This can be seen as a GOODSAC [17] approach as we make an informed guess (through hashing) for a good initial alignment. To achieve robustness against bin-jumping, we use a *kd-tree* for indexing the set of 4PAS instead of fixed bin sizes. For a 4PAS $\boldsymbol{P}_1$ we consider every $\boldsymbol{P}_2$ that lies within a certain tolerance δ as a candidate correspondence: $|T_{\boldsymbol{\rho}}(\boldsymbol{P}_1) - T_{\boldsymbol{\rho}}(\boldsymbol{P}_2)| \leq \delta$.

4.1 Pseudoinverse Matrix

We define the affine transformation $\boldsymbol{A}$ and two 4PAS:

$$\boldsymbol{A} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{P}_1 = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_4\} \subset \mathcal{C}_1, \quad \boldsymbol{P}_2 = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_4\} \subset \mathcal{C}_2. \tag{12}$$

Without loss of generality we align $\mathcal{C}_1 \to \mathcal{C}_2$, i.e. $\boldsymbol{A}(\boldsymbol{P}_1) = \boldsymbol{P}_2$, with point correspondences $\boldsymbol{p}_1^i \sim \boldsymbol{p}_2^i$. Therefore, we have to solve:

$$\underbrace{\begin{pmatrix} \boldsymbol{p}_{1x} & \boldsymbol{p}_{1y} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{p}_{1x} & \boldsymbol{p}_{1y} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{p}_{nx} & \boldsymbol{p}_{ny} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{p}_{nx} & \boldsymbol{p}_{ny} & 1 \end{pmatrix}}_{\boldsymbol{M}} \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix}}_{\boldsymbol{a}} = \underbrace{\begin{pmatrix} \tilde{\boldsymbol{p}}_{1x} \\ \tilde{\boldsymbol{p}}_{1y} \\ \vdots \\ \tilde{\boldsymbol{p}}_{nx} \\ \tilde{\boldsymbol{p}}_{ny} \end{pmatrix}}_{\boldsymbol{b}} . \tag{13}$$

This overdetermined linear system of equations (Eqn. 13) encodes a point correspondence in two lines, as the x– and y–coordinates of a point each impose a condition. For the initial pairing, the number of correspondences is $n = 4$. It can be solved using the *Moore-Penrose Pseudoinverse* $\boldsymbol{M}^{+}$, minimizing the quadratic Euclidean error between the corresponding points:

$$\boldsymbol{a} = \boldsymbol{M}^{+}\boldsymbol{b} \quad . \tag{14}$$

In general we do not know the corresponding points initially. Therefore we have to check all 4! permutations of $\boldsymbol{P}_1$ against $\boldsymbol{P}_2$. We return the candidate transformation with minimal error (Eqn. 15).

4.2 Iterative Refinement

Once we have an initial guess for the affine transformation $\boldsymbol{A}$, we iteratively add more correspondences to the Eqn. 13. We continue adding correspondences as long as a *maximum likelihood* classification of the error of $\boldsymbol{A}$ indicates its validity. We find the next correspondence by checking a small number k of candidates in the neighborhood of the already found correspondences and take the one with the smallest error (Alg. 1, lines 3 and 5). With this strategy, we find the next most likely corresponding points without explicitly specifying a threshold δ_{corr} (Eqn. 1). The goal of the refinement is to estimate the size of the common point set $\mathcal{X}_{\boldsymbol{A}}(\mathcal{C}_1, \mathcal{C}_2)$ that is implied by $\boldsymbol{A}$. We accept two objects as identical once its size reaches the application specific threshold $\#_{min}$. Naturally, the more common points we find, the more confident we can be about the identity of the objects.

We define the error of a mapping of two sets of n correspondences $\boldsymbol{K}_1 \to \boldsymbol{K}_2$ with respect to $\boldsymbol{A}$ as

$$e_{\boldsymbol{A}}(\boldsymbol{K}_1, \boldsymbol{K}_2) = \frac{1}{\eta^2} \sum_{i=1}^{n} \|\boldsymbol{A}(\boldsymbol{K}_1^i) - \boldsymbol{K}_2^i\|^2 \quad , \tag{15}$$

with normalizer η being the average distance of neighboring points in $\mathcal{C}_2$. We need to normalize the error to account for different scalings in the data. The error rises if the assumed correspondences can not be modeled well by an affine transformation which usually is the case if they do not originate from a common point set. This error measure enables us to to deal with noisy data well: By learning the distributions of the expected error of $\boldsymbol{A}$ for a given number of correspondences n, the registration algorithm adapts optimally to the noise-level present in the application. With *noise* we refer to errors in the point coordinates due to detector inaccuracies and not perfectly planar surfaces.

We have two classes: ω_{pos} for correspondences that originate from a common point set and ω_{neg} for assumed correspondences that do not originate from a common point set. We continue adding correspondences as long as maximum likelihood classification of the error $e_{\boldsymbol{A}}$ of the transformation

$$p(e_{\boldsymbol{A}}| \quad \omega_{pos}, n) < p(e_{\boldsymbol{A}}| \quad \omega_{neg}, n) \tag{16}$$

indicates its validity. The distributions $p(e_{\boldsymbol{A}}|\omega_{\{pos,neg\}}, n)$ for the positive and negative cases with n points have to be learned on a set of labeled correspondences originating from the population that we wish to work on.

Algorithm 1 $[\boldsymbol{K}_1, \boldsymbol{K}_2, \boldsymbol{A}] = \text{refineTransformation}(\boldsymbol{A}, \mathcal{C}_1, \mathcal{C}_2, \boldsymbol{K}_1, \boldsymbol{K}_2)$

```
for $n \in 4, \ldots, \#_{min}$ do
  $\mu = \text{mean}(\boldsymbol{K}_2)$ {center of gravity}
  $B \leftarrow \text{KNN}(\mathcal{C}_2, \boldsymbol{K}_2, k)$ {k nearest neighbors to $\boldsymbol{K}_2$ on $\mathcal{C}_2$}
  for $c_2 \in B$ do
    $c_1 = \text{KNN}(\boldsymbol{A} \circ \mathcal{C}_1, c_2, 1)$
    $\boldsymbol{K}_1 \leftarrow c_1$,$\boldsymbol{K}_2 \leftarrow c_2$ {add to correspondences}
    $\boldsymbol{A} = \text{computeAFF}(\boldsymbol{K}_1, \boldsymbol{K}_2)$ {using Eqn. 14}
    if $p(e_{\boldsymbol{A}}| \quad \omega_{pos}, n) < p(e_{\boldsymbol{a}}| \quad \omega_{neg}, n)$ then {ML classification of quality}
      remove $c_i$ from $\boldsymbol{K}_i$
    else
      break
    end if
  end for
  if no new correspondence found then
    return $\boldsymbol{K}_1, \boldsymbol{K}_2, \boldsymbol{A}$
  end if
end for
```

4.3 Complexity

The complexity of the registration algorithm is directly proportional to the size of the queue holding the candidate pairings. For each pairing we have to run the iterative refinement (Alg. 1) that runs at most $\#_{min}$ iterations. We can assume that this is a rather small value in real world applications (See Sect. 5). Therefore we estimate the cost of one refinement with $O(\#_{min})$. The size of the queue is dependent on the discrimination power of $T_{\boldsymbol{\rho}}$ and the noise level of the data. The best case, i.e. a characteristic common point set exists, is achieved in constant time. The worst case (negative matching on a repetitive pattern) is quadratic in the number of points as we have to try all possible pairings. The average case for the negative matching is linear in the number of points, if the hashing yields few possible candidate pairings. This usually is the case for random patterns.

5 Experiments

We test the algorithm for the biometric identification of the bluspotted ribbontail ray, working with a total of 42 underwater pictures from 6 different individuals (7 pictures each). We extract the point pattern using a multi scale LoG detector. The extracted point data contains nearly no false positives on the surface of the ray, but may contain lots of *outliers*. The average point cloud contains 140 points and is partitioned into 180 4PAS. We learn the positive distribution $p(e_{\boldsymbol{A}}|\omega_{pos}, n)$ on one hand labeled set of 7 pictures for $n \in 4, \ldots, 21$, retrieving around 500 error measurements $e_{\boldsymbol{A}}$ for each n. The negative case ω_{neg} is learned on point clouds of two different individuals by sampling iteratively increasing neighborhoods of n points on the clouds.

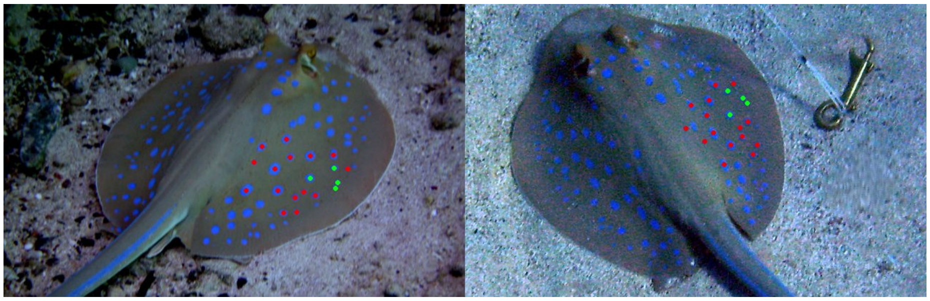


Fig. 4. A positively identified sting ray with its characteristic common point set (*red*) and the initial pairing of 4PAS (*green*).

We perform a pairwise comparison on the remaining 5 sets conducting a total of 595 tests. Fig. 5 shows the precision–recall diagram of our classification results. We achieve a precision of 1 with a recall of 0.75 which validates our algorithm for the biometric identification. With $\#_{min} = 17$ we would achieve perfect confidence on this population. The points that lie outside of the characteristic common point set can be regarded as outliers, although they most likely represent a real blue spot on the ray's surface – but not from the same planar surface patch. To the best of our knowledge, no other point registration algorithm exists that can handle point clouds under affine distortions with $\sim 90\%$ outliers in arbitrary initial positions.

For the positive cases we find the solution after an average of 69 pairings. The algorithm was implemented using MATLAB R2009a on a Intel Core 2 Duo. Absolute timings for the positive matching are ~ 3 sec. and ~ 15 sec. for the negative case. The learning of $p(e_{\boldsymbol{A}}|\omega_{\{pos,neg\}}, n)$ takes less than one second for every n.

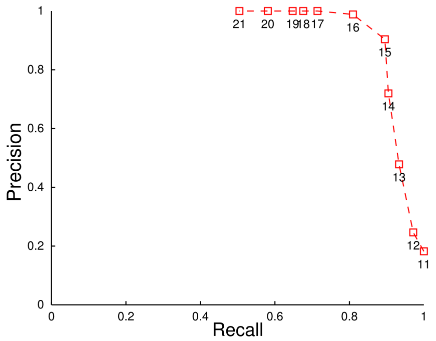


Fig. 5. A cardinality of $\#_{min} = 17$ for the characteristic common point set identifies an individual of *Taeniura lymma* unambiguously on our dataset.

6 Conclusion

We presented a novel point registration algorithm for matching two dimensional point clouds originating from partly planar surfaces under affine distortions. It can handle a huge amount of outliers and is able to deal with noisy data well. We successfully apply it for the biometric identification of *Taeniura lymma*.

7 Acknowledgements

We would like to thank Jessica Weidenbörner[3] for the underwater pictures of Taeniura lymma and the anonymous reviewers for their helpful comments.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
2. Burghardt, T., Campbell, N.: Individual animal identification using visual biometrics on deformable coat patterns. In: ICVS07, DOI:10.2390 (March 2007)
3. Costa, M., Haralick, R., Shapiro, L.: Object recognition using optimal affine-invariant matching (1990)
4. Burkhardt, H., Reisert, M., Li, H.: Invariants for discrete structures - an extension of haar integrals over trf. groups to dirac delta functions. In: DAGM. (2004)
5. Speed, C.W., Meekan, M.G., Bradshaw, C.J.a.: Spot the match - wildlife photo-identification using information theory. Frontiers in zoology **4** (2007) 2
6. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4) (2002) 509–522
7. Lazebnik, S., Schmid, C., Ponce, J.: Semi-local affine parts for object recognition. In: In BMVC. (2004) 959–968
8. Coetzee, L., Botha, E.C.: Fingerprint recognition in low quality images. Volume 26. (1993) 1441–1460
9. Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: CVPR '91., IEEE Computer Society Conference on. (1991) 586–591
10. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: International Conference on 3D Digital Imaging and Modeling. (2001)
11. Winkelbach, S.: Efficient methods for solving 3d-puzzle-problems. it - Information Technology **50**(3) (2008) 199–201
12. Lamdan, Y., Schwartz, J., Wolfson, H.: Object recognition by affine invariant matching. In: CVPR88. (1988) 335–344
13. Wolfson, H., Rigoutsos, I.: Geometric hashing: An overview. CalSE **4**(4) (October 1997) 10–21
14. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust surface registration. ACM Transactions on Graphics **27**(3) (2008) #85, 1–10
15. Burkhardt, H.: Transformationen zur lageinvarianten Merkmalgewinnung. Habilitation (1979) Fortschrittbericht(Reihe 10, Nr.7), VDI-Verlag.
16. Weiss, I.: Geometric invariants and object recognition. Int. J. Comput. Vision **10**(3) (1993) 207–231
17. Michaelsen, E., von Hansen, W., Kirchhof, M., Meidow, J., Stilla, U.: Estimating the essential matrix: Goodsac versus ransac. In: PCV06. (2006)

[3] jessica.weidenboerner@t-online.de