

Dolphin fin pose correction using ICP in application to photo-identification

Andrew Gilman, Ting Dong, Krista Hupman, Karen Stockin and Matthew Pawley
 Institute of Natural and Mathematical Sciences
 Massey University
 Albany Campus, Auckland, New Zealand
 Email: a.gilman@massey.ac.nz

Abstract—Photo-identification of dolphin individuals is a commonly used technique in ecological sciences to monitor state and health of individuals, as well as to study the social structure and distribution of a population. Uniqueness of some forms of damage sustained by the trailing edge of the dorsal fin throughout an individual's life allows it to be used to identify the individual. Traditional photo-identification involves a laborious manual process of matching each dolphin fin photograph captured in the field to a catalogue of known individuals. A number of efforts have been made in the past to automate this process through the use of digital image processing; however, none of them correct for the different fin orientations (3D pose) prior to performing matching in a robust way. This paper provides an overview of these existing methods and offers a novel approach of utilising the iterative closest point (ICP) algorithm for performing the projective registration of two fin shapes and shows that this proposed method is robust to segmentation noise, partial occlusions and new nicks and notches appearing since last encounter.

I. INTRODUCTION

Individual identification within a population is often necessary when studying the ecology and behavior of a species. The size, social structure and distribution of a population are often estimated by tagging (or marking) a sample of the population which may be recaptured at a later date. In the case of cetaceans (whales, dolphins and porpoises), the marking and recapturing processes are often replaced by using photographs of individuals to create a catalogue. This allows researchers to monitor the status and health of individuals without the use of an invasive method (such as physical tagging) [1]. However, the technique requires individuals to be uniquely identifiable within the photograph. Dolphins are often identified using unique markings found on the dorsal fins [2]. They are highly active, social and sometimes aggressive marine mammals, and their dorsal fins are very likely to acquire permanent damage, such as nicks, tears and notches. The over-all shape of the dorsal fins can also differ between individuals [3], [4].

However, current manual identification of dolphin fin photos can be both time consuming and inefficient – each new photograph must be compared against a catalogue of photographs that grows over time. A computational method to automate the identification (and classification) of individuals relieves cetacean researchers of a considerable burden.

We are currently involved in ongoing research into the development of a semi-automatic computer assisted photo-identification system (consisting of segmentation, registration and matching stages) to classify photographs of New Zealand common dolphins (*Delphinus sp.*). The field images of dolphin fins are captured from a research vessel using a digital SLR camera with a telephoto lens. Identification is performed by comparing the unique markings on the dorsal fin against a catalogue of known individuals. If a match is found, the date and location records are updated. If no match is found, a new individual is added to the catalogue.

Researchers try to take photographs orthogonal to the dolphin dorsal fin orientation, but naturally this is not always possible. Figure 1 shows a common scenario, where two individuals (top and bottom rows) were captured with different relative orientations (left and right columns). Even with such distinctive markings, they would be hard to match automatically without first registering them to remove the perspective distortion. Because photographs are often captured under extreme lighting conditions and with partial occlusions (from the waves), performing traditional image registration on the fins is not straight forward and we opt for registering contours - the shape outlines (from segmented images).

The purpose of this paper is two-fold: (i) as a review of the existing efforts in automating photo-identification process and (ii) to describe a new pre-matching registration stage, based on Iterative Closest Point (ICP) algorithm.

II. RELATED WORK

Two previous efforts have been made to create a piece of software that would aid researchers to perform photo-identification and semi-automatic classification of dolphin dorsal fins by individual: FinScan (dating late 90s) [6] and DARWIN (under development since 1993) [7]. Both programs extract the contour outline of the fin shape and then either attempt to match the contours directly or to extract and match higher level features, as described below.

FinScan utilises three different matching techniques:

1) *Dorsal ratio*: This is a ratio of the distance between the two most prominent notches on the fin to the distance between the tip of the fin and the lower notch [8]. The notches are located by analysing a cubic spline-smoothed curvature function (as defined in [9]) of the contour, where the



Fig. 1. Typical dolphin fin images of two separate individuals. Photographs can be taken under varying lighting conditions and with various amount of perspective distortion due to fin 3D pose.

positions of the two most prominent notches correspond to the positions of the two strongest minima and the position of the tip corresponds to the strongest maximum [10]. The authors found the dorsal ratio to be useful in narrowing down the search, but alone it was not sufficient for exact identification.

2) *Notch matching*: Notch matching involves finding all notches in the trailing edge, along with their position, width and depth and comparing these features to a database of stored values. The notches are located by computing the Euclidean distance between the detected contour and an approximation of the “undamaged” trailing edge [11]. The latter is constructed by utilising a deformable template (created by fitting a quartic polynomial to the trailing edge), similar to the ‘snakes’ – active contours algorithm [12].

The notches appear as local maxima in this distance curve and their position along the contour (relative to the tip), depth and width are approximated by the mean, height and standard deviation parameters of a Gaussian function fitted to each maxima. The parameters are normalised relative to the values of the largest notch to make them invariant to scale and image resolution. Hillman *et al.* found the number of detected notches and the pattern of their locations to be useful as a basis for identification; however, the depth and width of the peaks were found to be much less consistent [11].

3) *Curve matching*: This involves matching the entire distance curves, computed in much the same way as above, for each potential pair of fins. The distance curves are first scaled to account for possible image resolution and scale differences and then compared using the sum of square differences (SSD) [13]. But unlike the curvature function, the distance curves are not invariant to rotation.

Araabi *et al.* attempted to perform SSD matching of the curvature functions, but found it performed poorly [14]. The SSD

measure diminished the contribution of significant curvature structures, corresponding to prominent notches, and enhanced the contribution of insignificant ones – present because of the curvature’s sensitivity to small fluctuations in the contour due to discrete representation and segmentation noise. Instead, the authors opted for a string matching approach [14], utilising a syntactic/semantic representation of the curvature function. However, this resulted in invariance only up to similarity transformation. In an attempt to obtain a higher level of invariance, the authors then evaluated three families of invariant models (differential, algebraic and semi-differential invariants) [4]. However, they found the curvature based matching method outperformed all of these under realistic conditions.

The makers of DARWIN (Stewman *et al.*) explicitly accounted for the possibility of different dolphin body orientations (3D pose) by registering any unknown contour to each known one, prior to performing the matching [15]. They used a 2D shear transformation (normalisation was used to correct for scale separately), computed from correspondences of three pairs of ‘control points’. Dolphin fin contours contain few reliably identifiable salient features that can be used to compute correspondences for this transformation. Stewman *et al.* chose to use the fin tip, the beginning of the leading edge and the most prominent notch on the trailing edge as the control points for their registration. The locations of these points are located by analysing what the authors call an ‘absolute chain code’, but what is effectively a tangent angle function [16] of the contour. The fin tip location on the contour corresponds to the position of a large change in the tangent function, appearing as a large step-edge, whereas the location of the most prominent notch corresponds to the largest local minimum. The authors utilized a quadratic spline wavelet [17] (which produces large coefficients for step-edge type features) to pin point these locations. The beginning of the leading edge was located by adapting Otsu’s threshold selection method [18]. This method finds the tangent angle that discriminates between the tangent angles described by the leading edge of the fin and the angles associated with the dolphin body.

Stewman *et al.* found these feature detection mechanisms lacked robustness and had to employ additional optimisation of the registration parameters [15]. Unfortunately, because they used a matching mechanism based on SSD, even a small amount of misregistration may lead to a missed match of two contours that actually belong to the same dolphin fin.

Another problem with their approach is that unless the two contours that are being registered belong to the same fin, it is unlikely that using above point correspondences will result in correct registration, making any further metric computation meaningless. The implication of this is two-fold: possible matches cannot be ranked in the order of likeness, and, if a fin has sustained additional damage that affects detected feature points’ locations since last being photographed, it is unlikely to be chosen as a possible match.

Given fin shapes have very few features, feature based matching methods may not have enough information for

correct registration and one-dimension descriptor approaches which reduce the level of detail may also be not applicable. We propose the use of iterative closest point (ICP) technique [19] that solves the registration problem without the need to find salient features and their correspondences.

III. CONTOUR REGISTRATION

The purpose of the registration stage is not to recover the true 3D pose of the fin, but to align two fins that possibly had different 3D poses relative to the camera. Once the fins are aligned, spatial locations of the features of interest (nicks and notches) can be compared. The coordinate systems of two such fins are related to each other by planar homography; thus, the problem is to find the homography matrix H that transforms one perspective of a fin image into another. Because of quality issues associated with typical images captured in the field (extreme lighting conditions, severe blurring, reflections, etc), registration of fin images themselves does not provide a robust solution at a reasonable computational cost (catalogue matches need to be performed in a timely manner). The shape contours, however, stay relatively invariant to these problems and are also much faster to perform the registration on.

Although, there is some research on estimating the homography that relates two planar contours [20], it is not directly applicable in this case as the dorsal fin contours are open shapes and there is no way of dealing with possible truncations at the contour's ends.

We propose to use an alternative approach – the iterative closest point (ICP) algorithm [19], which is ubiquitous in registration of 3D point clouds, where partial overlap of the data sets is quite common. Rather than treating contours as curves (as is in [20]), ICP treats them as point sets. It uses iterative optimisation to compute transformation parameters from point correspondences based on the nearest neighbour criterion, without the need to locate and match salient features. Algorithmically, ICP is extremely simple and consists of only two steps that are iterated until some stopping criteria (e.g. a maximum number of iterations or root-mean-square (RMS) error) is met (see algorithm 1).

Data: 2 point sets (fixed and moving) and an initial transformation (homography matrix)
Result: transformation that aligns the point sets
 Initialization;
 Transform the moving set using the initial homography matrix;
while *stopping criteria not met* **do**
 Associate each point in moving point set with the nearest neighbour in the fixed point set and compute the homography matrix H using these correspondences;
 Transform the moving point set using H ;
end

Algorithm 1: Iterative closest point (ICP) algorithm

A. Implementation details

1) *Contour smoothing and resampling:* The contours are extracted from segmented images and are represented as sets of (x, y) coordinates of various lengths - depending on the

resolution of the image and the extent to which it is occupied by the fin. Contours are first smoothed using a low-pass filter with a normalised cut-off frequency of 0.1. This removes any spurious high-frequency signals due to the segmentation procedure, but does not have much of a smoothing effect on nicks and notches. If the two contours being registered contain a different number of points, the longer one is linearly interpolated and resampled to the number of points in the shorter one.

2) *Initial transformation estimate:* To speed up algorithm convergence, the contours are pre-registered using the Procrustes algorithm. This is a set of three transformations used to minimise the distance between sets of points. Initially, it translates the moving contour such that its centroid matches that of the fixed contour; then, then it normalises the variation in the moving contour point set to match the fixed contour data. Finally, it orthogonally rotates the moving contour around the common centroid to minimise the RMS difference between the contours.

3) *Matching:* The points in the moving contour are matched with points in the fixed contour based on the nearest neighbour criterion. A kD tree [21] is utilised to speed up the nearest neighbour search.

4) *Rejection and weighting:* After matching, each point correspondence is given an equal weight; however, some correspondences are rejected in order to enhance the alignment of partially-truncated contours. Any points on the moving contour that have been matched with the first or last points of the fixed contour are rejected. This is a common practice in ICP-based registration to cope with partially-overlapping point sets [22].

5) *Computing the projective transform from point correspondences:* The direct linear transformation (DLT) method [23] is used to solve the planar homography from point correspondences. This solves an over-determined homogeneous system of linear equations $A\mathbf{h} = 0$, where \mathbf{h} is the homography matrix H rewritten in vector form: $\mathbf{h} = [h_{11}, h_{12}, h_{13}, \dots, h_{33}]^T$ and A is a $2n \times 9$ matrix constructed from the coordinates of n point pairs (see Hartley and Zisserman for more details). The solution (that minimises the algebraic residuals $|A\mathbf{h}|$ with respect to a constraint $|\mathbf{h}| = 1$, implemented to prevent a solution of all zeros) is obtained by computing the singular-value decomposition (SVD) of $A^T A$ and taking the eigenvector of least eigenvalue. To ensure numerical stability of the problem, the matrix $A^T A$ is pre-conditioned by translating both point sets to their respective centroids and then scaling them so that the average distance from the origin is equal to $\sqrt{2}$, as advocated by Hartley [24].

Originally, we attempted to utilise MLESAC method proposed by Torr and Zisserman [25], assuming that outlier rejection would result in faster convergence and hence faster registration. However, this did not prove to be the case. The convergence rates did not see much improvement, and the overall registration time increased dramatically, due to a higher computational cost of MLESAC, compared to DLT.

IV. EXPERIMENTAL METHOD

As discussed earlier, the chosen registration method must be robust to different levels of cropping of the fin (at the bottom), various segmentation errors, as well as possible new nicks and notches added over time. The proposed method was first tested using simulated data, with projective transformation, cropping and segmentation error applied artificially and then tested on some real fin images acquired in the field.

A. Simulated projective transformation

The algorithm's accuracy and convergence rate were tested on simulated contours with approximately the same level of projective transformation extent that researchers find in the field. Four separate experiments with sequentially increasing degrees of complexity and potential error were performed:

- 1) A single contour (shown in figure 3) was transformed 400 times using a random projective transformation and registered with the original. The transformation was defined by moving the four corners of the original image towards the centre of the image by a random offset of between 0 and 20 % of the image dimensions.
- 2) Experiment 1 was repeated with the addition of segmentation error (noise), which was simulated by adding low-pass filtered (normalised cut-off frequency = 0.05) Gaussian noise ($\mu = 0$, $\sigma = 3$) to contour coordinates.
- 3) Experiment 2 was repeated with an addition of a random truncation of between 0 and 10 % of the contour length at each end.
- 4) In addition to 3, a new notch was added at a random location on the contour. The Gaussian function with a random height (between 10 and 20) and width (σ between 3 and 8) was used to approximate the shape of a notch (see figure 3 for example).

B. Real contours

To test how the algorithm would cope with contours extracted from real images, it was used to register contours extracted from 76 photographs of 36 individuals (32 individuals with 2 sightings and 4 individuals with 3 sightings). For each individual, all pair-wise permutations of sightings were registered. The photographs were a typical sample of what is obtained in the field, capturing individuals from different angles and under different lighting and environmental conditions.

C. Error calculation

Because of the nature of the projective transformation, it is difficult to make a meaningful comparison of estimated parameters to the ground-truth parameters. In the case of the experiment with real contours, the ground truth is also not available. Instead, the registration error is assessed using the same metric as that employed by ICP - the RMS distance between each point on the moving contour and the closest point on the fixed contour.

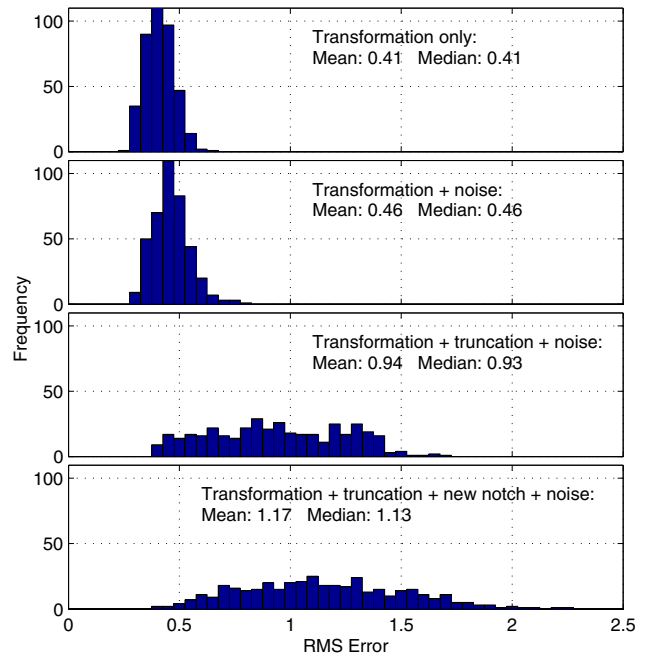


Fig. 2. RMS error distribution (RMS distance in pixels between two contour point sets after registration) from 400 simulated registrations. RMS error shown was computed on noise-free contours (that were transformed using the estimated registration parameters) to remove the effect of noise on RMS error directly (i.e. not through mis-registration).

V. RESULT AND DISCUSSION

A. Simulated projective transformations

The distribution of RMS error from 400 simulated contour registrations is shown in figure 2 in a form of a histogram. The top histogram shows the results of experiment 1. The algorithm resulted in almost perfect registration in the absence of noise or truncation and converged fairly quickly - under 30 iterations in most cases. Most RMS errors were under 0.5, with distribution mean of 0.41. The RMS error here is expressed in units of pixels (spacing) and with error this small, the registered contours would perfectly overlay each other when plotted.

The second histogram shows the results of experiment 2 - adding simulated segmentation noise. As adding noise to the contours effectively increases RMS error (i.e. even perfectly registered contours result in non-zero error) it is difficult to compare error distributions between experiments 1 and 2 directly. To make this comparison more sensible, the presented RMS error was calculated using noise-free contours that were transformed using the registration parameters estimated from noise-contaminated contours. Adding noise did not have any significant effect on either the RMS error or the convergence rate. Registered contours (noise-free versions) still perfectly overlay each other when plotted.

The next histogram shows the results of experiment 3: adding end-truncations. The RMS error distribution became more variable, but was still relatively low. The convergence rate also got slower, requiring around 60-80 iterations.

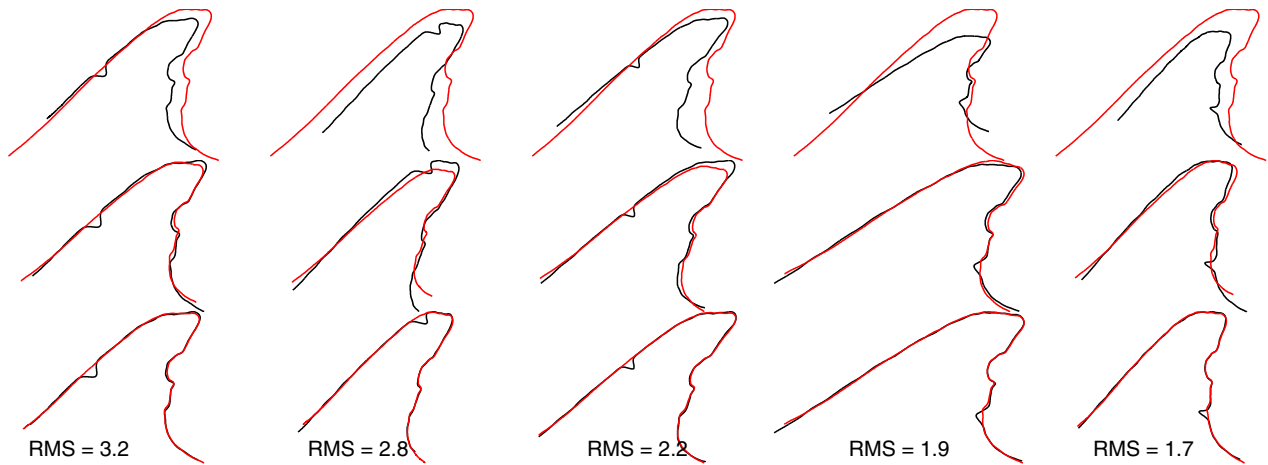


Fig. 3. A selection of contours used in the experiment with simulated contours. Top row shows the initial contour pairs to be registered. Middle row shows the contours after pre-registration step (Procrustes algorithm). Bottom row shows the contours after registration.

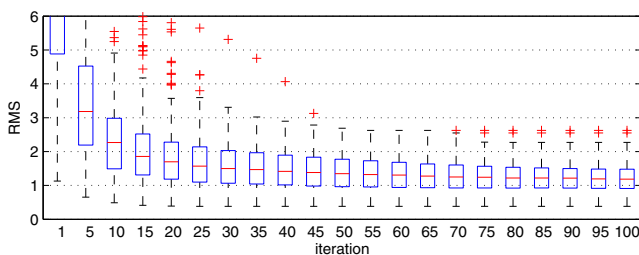


Fig. 4. Convergence of RMS error during registration of 400 simulated transformations.

The bottom histogram shows the results of experiment 4: effect of adding a new notch to the fixed contour. Again, the RMS error reported here was computed on noise-free contours without the extra notch to allow for a sensible comparison of the results to previous experiments. Addition of a notch slightly increased the registration error, but it was still relatively small (mean of 1.17 and most errors under 2 pixels). Visual inspection of registered contours showed good registration results in all inspected cases. Even contours with higher registration error (between 1.5 and 2) still almost perfectly overlay each other (when plotted without the noise and added notches). Figure 3 shows five randomly selected examples of pre- and post-registration contours. Figure 4 demonstrates the convergence rate from experiment 4.

B. Real images

The RMS error of real contour registration is shown in figure 5. On average, the RMS error is only slightly higher than that of the simulated data. There are some cases of higher error; though, these are primarily due to segmentation error, rather than registration error (segmentation error cannot be excluded from the RMS in this case as it was for the simulated data). Visual inspection of results showed that all of them registered successfully. Figure 7 shows a sample of five contour pairs before and after registration. It can be clearly observed that notches on the contours that do not match after pre-registration using Procrustes algorithm are positioned

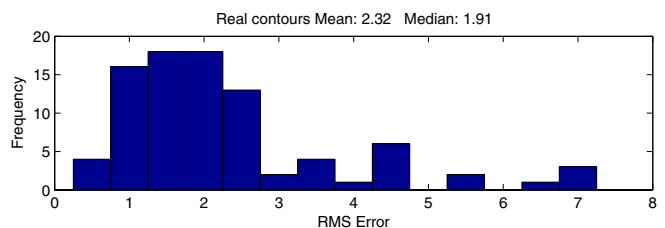


Fig. 5. Final RMS error of registration of 36 real contours.

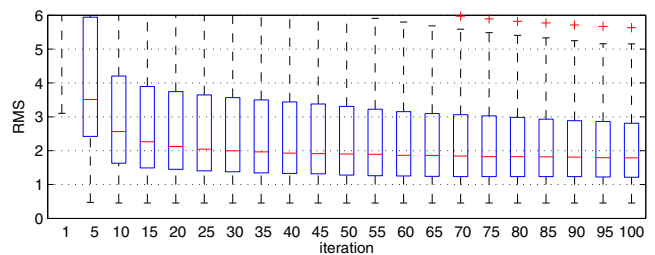


Fig. 6. Convergence of RMS error during registration of contours extracted from 76 field photographs.

correctly after ICP registration. The convergence rate (shown in figure 6) is, on average, similar to the simulated data experiments, apart from few extreme cases requiring 100-110 iterations.

VI. CONCLUSION

The registration of fin contours using ICP algorithm was shown to work successfully on simulated data in the presence of fairly large projective transformations, simulated segmentation noise, end-truncations and new notches. The proposed method was also shown to work well on real data acquired from field photographs.

It was clear from the experiments that segmentation errors/noise over the length of the contours can quickly add up to a large contribution to overall RMS error; thus, using RMS distance between the contours is not a good metric to use for matching in the next stage of the system. A matching method based on size/location of the notches may be more appropriate.

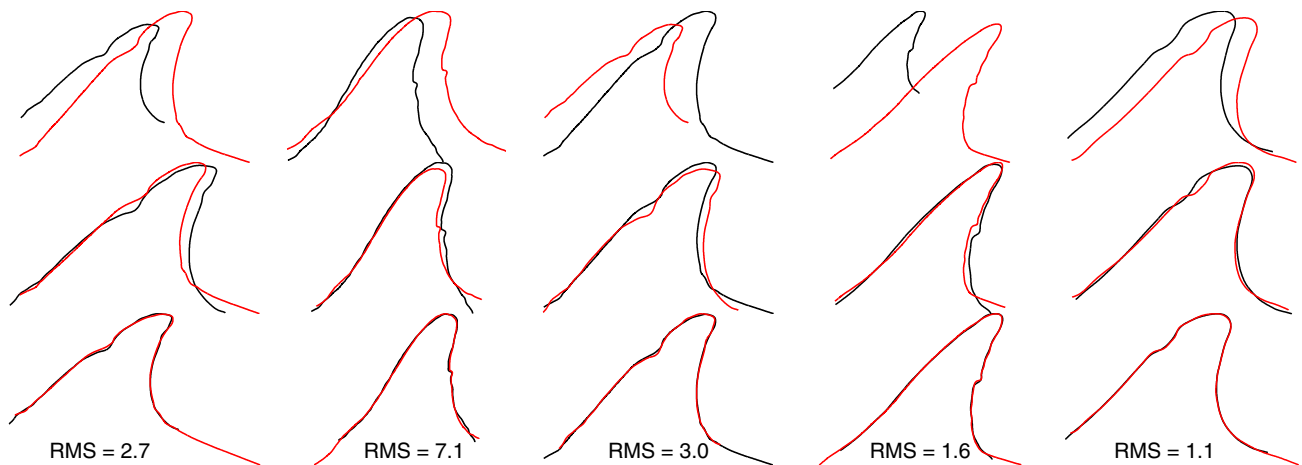


Fig. 7. A selection of contours used in the experiment with real images. Top row shows the initial contour pairs to be registered. Middle row shows the contours after pre-registration step (Procrustes algorithm). Bottom row shows the contours after registration.

Removing the effect of projective transformation through registration should make such a method more successful.

REFERENCES

- [1] B. Würsig and T. A. Jefferson, *Methods of photo-identification for small cetaceans*, 1990.
- [2] B. Würsig and M. Würsig, "The photographic determination of group size, composition, and stability of coastal porpoises (*tursiops truncatus*)," *Science*, vol. 198, no. 4318, pp. 755–756, 1977.
- [3] G. Hillman, N. Kehtarnavaz, B. Würsig, B. Araabi, G. Gailey, D. Weller, S. Mandava, and H. Tagare, "finscan", a computer system for photographic identification of marine animals," in *Engineering in Medicine and Biology, 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, vol. 2, 2002, pp. 1065–1066 vol.2.
- [4] B. Araabi, N. Kehtarnavaz, G. Hillman, and Würsig, "Evaluation of invariant models for dolphin photo-identification," in *14th IEEE Symposium on Computer-Based Medical Systems*, Bethesda, MD; USA, 2001.
- [5] P. V. A. Barros, N. T. M. Junior, J. M. M. Bisneto, B. J. T. Fernandes, B. L. D. Bezerra, and S. M. M. Fernandes, "Convexity local contour sequences for gesture recognition," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 34–39.
- [6] "Finscan: Computer-assisted photo-identification of bottlenose dolphins," <http://www.tamug.edu/mmrp/Software/Finscan/>, Jun. 2013.
- [7] "Darwin project," <http://darwin.eckerd.edu/>, Jun. 2013.
- [8] R. H. Defran, G. M. Shultz, and D. W. Weller, "A technique for the photographic identification and cataloging of dorsal fins of the bottlenose dolphin (*Tursiops truncatus*)," *Individual Recognition of Cetaceans: Use of Photo-Identification and Other Techniques to Estimate Population Parameters, Reports of the International Whaling Commission*, no. Special Issue 12, pp. 53–55, 1990.
- [9] F. MOKHTARIAN and A. MACKWORTH, "Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 34–43, Jan. 1986.
- [10] A. Kreho, N. Kehtarnavaz, B. Araabi, G. Hillman, B. Würsig, and D. Weller, "Assisting manual dolphin identification by computer extraction of dorsal ratio," *Annals of Biomedical Engineering*, vol. 27, no. 6, pp. 830–838, 1999.
- [11] G. Hillman, H. Tagare, K. Elder, A. Drobyshevski, D. Weller, and B. Würsig, "Shape descriptors computed from photographs of dolphin dorsal fins for use as database indices," in *Engineering in Medicine and Biology Society, Proceedings of the 20th Annual International Conference of the IEEE*, vol. 2, 1998, pp. 970–973 vol.2.
- [12] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [13] G. R. Hillman, B. Würsig, N. Kehtarnavaz, D. W. Weller, A. Kreho, A. Drobyshevski, K. Elder, B. N. Araabi, T. McKinney, G. Gailey, and H. Tagare, "Individual identification of dolphins in field photographs: a computer-based system," in *Proceedings of the 13th Biennial Conference on the Biology of Marine Mammals*. Wailea, Maui, Hawaii: Society for Marine Biology, 1999, p. 81.
- [14] B. Araabi, N. Kehtarnavaz, T. McKinney, G. Hillman, and B. Würsig, "A string matching computer-assisted system for dolphin photoidentification," *Annals of Biomedical Engineering*, vol. 28, no. 10, pp. 1269–1279, 2000.
- [15] J. Stewman, K. Debure, S. Hale, and A. Russell, "Iterative 3-d pose correction and content-based image retrieval for dorsal fin recognition," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science, A. Campilho and M. Kamel, Eds. Springer Berlin Heidelberg, 2006, vol. 4141, pp. 648–660.
- [16] D. Zhang and G. Lu, "A comparative study of fourier descriptors for shape representation and retrieval," in *Proc. of 5th Asian Conference on Computer Vision (ACCV)*. Springer, 2002, pp. 646–651.
- [17] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 7, pp. 710–732, 1992.
- [18] N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 1, pp. 62–66, 1979.
- [19] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, 1992.
- [20] P. Jain and C. Jawahar, "Homography estimation from planar contours," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, 2006, pp. 877–884.
- [21] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [22] R. Larkins, M. Cree, and A. Dorrington, "Analysis of icp variants for the registration of partially overlapping time-of-flight range images," in *Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of*, 2010, pp. 1–8.
- [23] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.
- [24] R. Hartley, "In defense of the eight-point algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 6, pp. 580–593, 1997.
- [25] P. H. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.