Aligning Concave and Convex Shapes

Silke Jänichen and Petra Perner

Institute of Computer Vision and applied Computer Sciences, IBaI,
Körnerstr. 10, 04107 Leipzig
pperner@ibai-institut.de, www.ibai-institut.de

Abstract. There are plenty of different algorithms for aligning pairs of 2D-shapes and point-sets. They mainly concern the establishment of correspon-dences and the detection of outliers. All of them assume that the aligned shapes are quite similar and belonging to the same class of shapes. But special problems arise if we have to align shapes that are very different, for example aligning concave shapes to convex ones. In such cases it is indispensable to take into account the order of the point-sets and to enforce legal sets of correspondences; otherwise the calculated distances are incorrect. We present our novel shape alignment algorithm which can handle such cases also. The algorithm establishes legal one-to-one point correspondences between arbitrary shapes, represented as ordered sets of 2D-points and returns a distance measure which runs between *0* and *1*.

Keywords: Shape Alignment, Correspondence Problem, Aligning Convex to Concave Shapes and vise-versa.

1 Introduction

The analysis of shapes and shape variation is of great importance in a wide variety of disciplines. It is especially interesting for biologists, since shape is one of the most concise features of an object class and may change over time due to growth or evolution. The problems of shape spaces and distances have been intensively studied by Kendall [1] and Bookstein [2] in a statistical theory of shape. In digital image processing the statistical analysis of shape is a fundamental task in object-recognition and classification [3] .

In all these applications, shapes of the same class are aligned and compared. The mapping of convex to concave pieces of the shapes rather indicates that wrong correspondences between elements have been established or that there are outliers [4]. However, there is a number of applications where we have to study the similarity between shapes of different classes. In that case we are faced with the problem to determine the similarity between convex and concave shapes.

We are describing our work on aligning arbitrary shape to each other and determining the similarity between them. It can happen that we have to compare convex to concave shapes. The natural shapes are acquired manually from real images [5]. The object shapes can appear with varying orientation, position, and scale in the

D.-Y. Yeung et al. (Eds.): SSPR & SPR 2006, LNCS 4109, pp. 243 – 251, 2006.

image. The shapes are arbitrary and there is nothing special about them. Our algorithm establishes symmetric and legal one-to-one point correspondences between arbitrary shapes, represented as ordered sets of 2D-points and returns a similarity value.

The paper is organized as follows. We describe the problem of shape alignment in Sect. 2. The algorithm for pair-wise alignment of the shapes and calculation of distances is proposed in Sect. 3 and evaluated in Sect. 4. Finally we give conclusions in Sect. 5.

2 The Problem of Alignment of 2-D Shapes

Consider two shape instances P and O defined by the point-sets $p_i \in R^2$, $i = 1, 2, \ldots, N_P$ and $o_k \in R^2$, $k = 1, 2, \ldots, N_O$ respectively. The basic task of aligning two shapes consists of transforming one of them (say P) so that it fits in some optimal way the other one (say O) (see Fig 1 left). Generally the shape instance $P = \{ p_i \}$ is said to be aligned to the shape instance $O = \{ o_k \}$ if a distance $d(P,O)$ between the two shapes cannot be decreased by applying a transformation ψ to P.

The problems of shape spaces and distances have been intensively studied [1], [2] in a statistical theory of shape. The well-known Procrustes distance [6], [7] between two point-sets P and O is defined as the sum of squared distances between corresponding points:

$$d(P,O) = \sum_{i=1}^{N_{PO}} \left\| \frac{(p_i - \mu_P)}{\sigma_P} - R(\theta) \frac{(o_i - \mu_O)}{\sigma_O} \right\|^2 , \quad (1)$$

where $R(\theta)$ is the rotation matrix, μ_P and μ_O are the centroids of the object P and O respectively, σ_P and σ_O are the standard deviations of the distance of a point to the centroid of the shapes and N_{PO} is the number of point correspondences between the point-sets P and O. This example shows that the knowledge of correspondences is an important prerequisite for calculation of shape distances.

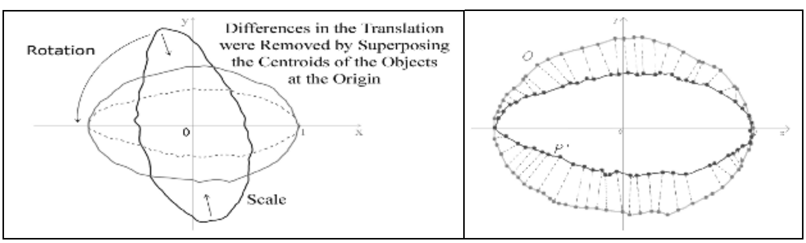


Fig. 1. Alignment of shape instances, superimposition, and calculation of correspondences

Various alignment approaches are known [8][9]. They mainly differ in the kind of mapping (i.e. similarity, rigid, affine) and the chosen distance measure. A survey of different distance measures used in the field of shape matching can be found in [10].

For calculating a distance between two shape instances the knowledge of corresponding points is required. If the shapes are defined by sets of landmarks [11], the knowledge of point correspondences is implicit. However, at the beginning of many applications this condition does not hold and often it is hard or even impossible to assign landmarks to the acquired shapes. Then it is necessary to automatically determine point correspondences between the points of two aligned shapes P and O, see (see Fig 1 right).

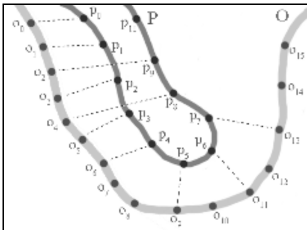
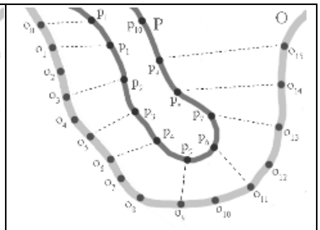
There has been done a lot of work concerning the problem of automatically finding point correspondences between two unknown shapes. An extension of the classical Procrustes alignment to point-sets of differing point counts is known as the *Softassign Procrustes Matching* algorithm [6]. It alternates between solutions for the correspondence, the spatial mapping, and the Procrustes rescaling.

Hill *et al.*[9] presented a greedy algorithm used as an iterative local optimization scheme to modify the correspondences, in order to minimize the distance between two polygon segments of shapes. Another popular approach to solving the correspondence problem is called *Iterative Closest Point* (*ICP*) developed by Besl and McKay [12]. In the original version of the *ICP* the complexity of finding for each point p_k in P the closest point in the point-set O is $O(N_P N_O)$ in the worst case. Marte *et al.*[13] improved this complexity by applying a spatial subdivision of the points in the set O. Fitzgibbon [14] replaced the closed-form inner loop of the *ICP* by the Levenberg-Marquardt algorithm, a non-linear optimization scheme. Another solution of the correspondence problem was presented by Belongie *et al.*[15]. He added to each point in the set a descriptor called shape context. In our work we solved the correspondence problem by a nearest-neigbor search algorithm [5].

One of the most essential demands on these approaches is symmetry. Symmetry means obtaining the same correspondences when mapping instance P to instance O and vise versa instance O to instance P. This requirement is often bound with the condition to establish one-to-one correspondences. This means a point o_k in shape instance O has exactly one corresponding point p_k in shape instance P. If we compare point sets with unequal point numbers under the condition of one-to-one mapping, it is clear that some points will not have a correspondence in the other point set. These points are called outliers.

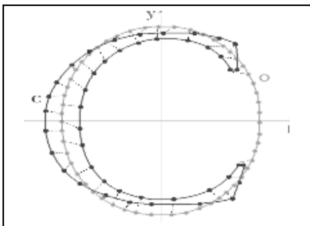
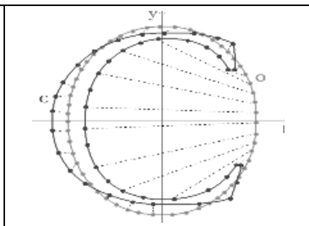
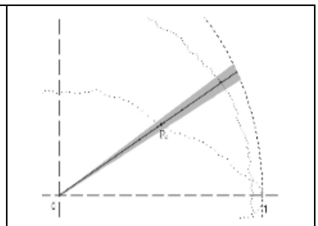
Special problems arise if we have to align shapes that are very different, for example aligning concave to convex shapes. In these cases it is indispensable to take into account the order of the point-sets and to enforce legal sets of correspondences by not allowing inverse mapping of the points. To demonstrate this, see points o_2 and o_4 in Table 2(a). Suppose that a concave shape representing the letter C is compared with the shape of the letter O (see Table 1). If the pair-wise correspondences were established between nearest neighbored points by one-to-one mapping and by allowing inverse mapping, the resulting distance between both shape instances will be very small (Table 1 a).

Table 1. Illegal and legal sets of correspondences

	
(a) Illegal correspondences inversions at o_2 and o_4	**(b)** Legal correspondences without any inversions

But intuitively we would say that these shapes are not very similar. Particularly in such cases it is necessary to regard the order of point correspondences and to remove correspondences if they produce inversions (see Table 1 b). Ultimately it can be seen that big distances are arising between corresponding points which leads to an increased distance measure.

Table 2. Establishing correspondences while mapping a concave and convex shape

		
(a) Illegal correspondences with inversions	**(b)** Enforced legal correspondences	**(c)** Range for finding potential correspondents for p_k

3 Our Algorithm

<u>The input</u> into our algorithm (see table 3) is the rescaled shape P and O translated into its origin. This normalization ensures that the centroids are identical and that our similarity measure is running between *0* and *1*. The Euclidean distance between the two shapes P and O is calculated. We are also calculating the maximum distance and a score based on the sum between the maximum distances and the mean distance.

The algorithm is comprised of three main steps: (A) rotate shape, (B) calculate point correspondences, and (3) calculate the similarity score. The differences in rotation will be removed during our iterative alignment algorithm. In each iteration of this algorithm, the first shape is rotated stepwise by an angle $\nabla\psi$, while the second

Table 3. Outline of our shape alignment algorithm

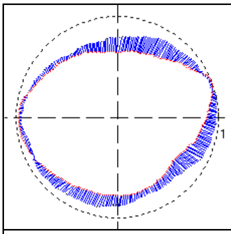
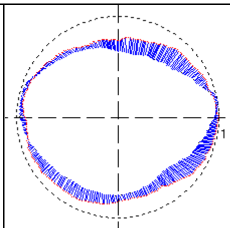
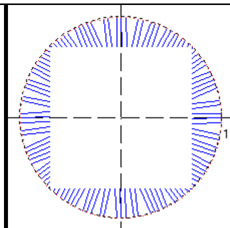
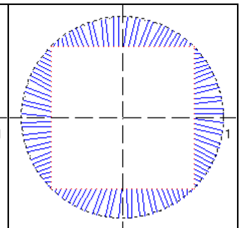
```
Initialize ψ          /* stepwise rotation angle */
SET ψ_i = 0           /* actual rotation angle */

Input: Normalized Shape O and Shape P
Output: min{SCORE(P,O_i)}
REPEAT UNTIL ψ_i ≥ 2π  or  SCORE(P,O_i) = 0
   (A) Rotate O with ψ_i = ψ_{i-1} + ψ
   (B) CalcCorrespondences( P , O_i )
   (C) CalcScore SCORE(P,O_i)
RETURN SCORE(P,O) = min{SCORE(P,O_i)}


SUB (B) CalcCorrespondences( P , O ) BEGIN
   Calculate γ_dev
   FOR EACH point p in P DO
       -Calculate orientation angle γ_p of p
       -Put into {CorrList(p)} all points o with angle γ_o
        where (γ_p − γ_dev) ≤ γ_o ≤ (γ_p + γ_dev)
       -IF {CorrList(p)} = EMPTY THEN
           Mark p as Outlier
       -ELSE
          -QuickSort{CorrList(p)} with ascending
           distances in relation to p
          -FOR EACH item k in CorrList(p)
             -IF k has no Correspondence on P THEN
                  SET Correspondence between k and p
   If t + x < i < t THEN Remove p_i
END
```

shape is kept fixed. For every transformed point in the first shape we try to find a corresponding point on the second shape. For the establishment of point correspondences we demand the following facts: a. produce one-to-one point correspondences, remove illegal point-correspondences from the list of one-to-one point correspondences, c. determine points without a correspondence as outlier, and d. produce symmetric results, which is obtaining the same results when aligning instance P to instance O as when aligning instance O to P.

Based on the distance between these corresponding points the alignment score is calculated for this specific iteration step. When the first shape is rotated once around its centroid, finally that rotation is selected and applied where the minimum alignment score is calculated.

In this respect the algorithm is similar to our nearest neighbor-search algorithm proposed in [5]. The main difference is the way we calculate point correspondences. It was shown in Sect. 2 that the establishment of legal sets of correspondences is an

Table 4. Evaluation of symmetric property

			
(a) *shape_12* (340 points) align to *shape_13* (340 points)	**(b)** *shape_13* (340 points) align to *shape_12* (340 points)	**(c)** *rect_mid* (116 points) *is* aligned to *circle* (144 points)	**(d)** *circle* (144 points) aligned to *rect_mid* (116 points)
$\psi = 0.2094$	$\psi = -0.2094$		
$\bar{\varepsilon} = 0.0842$;	$\bar{\varepsilon} = 0.0842$;	$\bar{\varepsilon} = 0.1848$;	$\bar{\varepsilon} = 0.1887$;
$\varepsilon_{max} = 0.1835$	$\varepsilon_{max} = 0.1835$	$\varepsilon_{max} = 0.2921$	$\varepsilon_{max} = 0.2904$
$\underline{\underline{Score = 0.1339}}$	$\underline{\underline{Score = 0.1339}}$	$\underline{\underline{Score = 0.2384}}$	$\underline{\underline{Score = 0.2395}}$

important fact to distinguish between concave and convex shapes. The drawback of this requirement is that the set P of contour points p_i of the acquired shapes have to be an ordered set $(P, \leq)$.

Before the iterative algorithm starts we define a range where to search for potential correspondences. This range is defined by a maximum deviation of the orientation according to the centroid (see Table 2 c). This restriction will help us to produce legal sets of correspondences. The maximum permissible deviation of orientation γ_{dev} will be calculated in dependence of the amount of contour points n_O of the shape O, which is the instance that has more points than the other one. Our investigations showed that the following formula leads to a well-sized range

$$\gamma_{dev} = \pm \frac{4\pi}{n_O} \; . \tag{2}$$

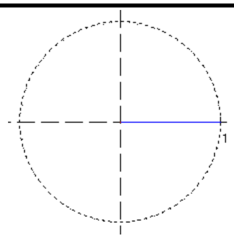
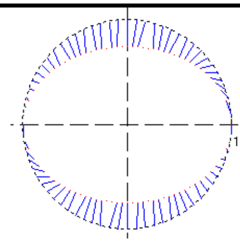
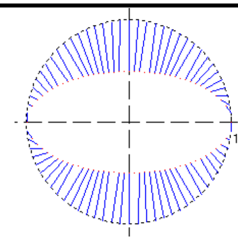
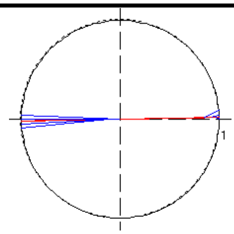
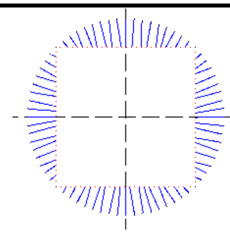
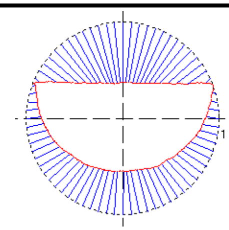
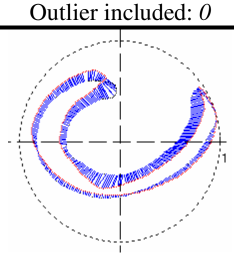
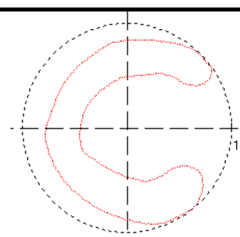
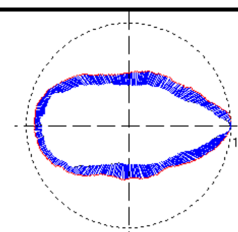
Let $t + x$ be an upper bound in the search area for a subset I of P if for every $i \in I$, we have $i \leq t + x$ and similarly, a lower bound in the search area for a subset I is an element t such that for every $i \in I, t \leq i$. Now, if we find more the one mapping between the point o and the points p_i within the search area, we remove the points $\{o, p_i\}$ having an ordering number i larger than the considered interval $\{t, t + x\}$ with $x = \frac{\gamma_{dev} \cdot n_o}{4\pi}$.

The complexity of the algorithm is $n\, N_O\, O(k \log_2 k)$. By introducing Bucket Sort instead of QuickSort we can reduce the complexity to linear complexity ($n\, N_O\, O(k)$).

4 Results

Table 5 shows some results of the alignment process. A point aligned to a circle gives the expected maximum dissimilarity value of *one* (Table 5 a), since *zero* means identity. If we align an ellipse to the circle and let this ellipse converge to a line, we get an increasing dissimilarity value which reaches the value 0.5 in case of a line (see Table 5 b- e). It can be seen that the dissimilarity value between the line and the circle is not exactly 0.5 (see Table 5 e). This is a small approximation error of the algorithm

Table 5. Exemplary results of our alignment process

 **(a)** *point* aligned to *circle* $\bar{\varepsilon}=1$; $\varepsilon_{max}=1$ $\underline{\underline{Score=1}}$ Outlier included: *0*	 **(b)** *circle* aligned to *ellipse_1* $\bar{\varepsilon}=0.1643$; $\varepsilon_{max}=0.2532$ $\underline{\underline{Score=0.2088}}$ Outlier included: *0*	 **(c)** *circle* aligned to *ellipse_2* $\bar{\varepsilon}=0.3165$; $\varepsilon_{max}=0.5016$ $\underline{\underline{Score=0.4090}}$ Outlier included: 0
 **(e)** *circle* aligned to *diameter* $\bar{\varepsilon}=0.5112$; $\varepsilon_{max}=1$ $\underline{\underline{Score=0.7556}}$ Outlier included: *0*	 **(f)** *circle* aligned to *rect_mid* $\bar{\varepsilon}=0.1924$; $\varepsilon_{max}=0.2907$ $\underline{\underline{Score=0.2415}}$ Outlier included: *0*	 **(g)** *circle* al. to *semicircle* $\bar{\varepsilon}=0.3642$; $\varepsilon_{max}=0.6509$ $\underline{\underline{Score=0.5076}}$ Outlier included: 0
 **(i)** *concave3* al. to *concave6* $\bar{\varepsilon}=0.0777$; $\varepsilon_{max}=0.1617$ $\underline{\underline{Score=0.1197}}$ Outlier included: *0*	 **(j)** concave1 al. to concave1 $\bar{\varepsilon}=0$; $\varepsilon_{max}=0$ $\underline{\underline{Score=0}}$ Outlier included: *0*	 **(k)** *shape_2* al. to *shape_1* $\bar{\varepsilon}=0.1015$; $\varepsilon_{max}=0.1557$ $\underline{\underline{Score=0.1286}}$ Outlier included: *0*

caused by the allowed search area for the correspondences. The alignment of other arbitrary shapes is shown in Table 5 f-p. The alignment of a concave object to the convex shape of a circle is shown in Table 5 h. The established correspondences are legal and a set of outliers was detected. Finally, this results in a high dissimilarity value.

In case both shapes have the same number of points, the symmetry of the similarity is given (see Table 4 a and Table 4 b). But the symmetry property does not exactly hold if a shape that consists of m points is aligned to a shape that consists of n points where $m > n$ (see Table 4 c and Table 4 d). The similarity value has a small deviation. This is because there are multiple choices to establish correspondences among the larger number of points of shape P to the smaller number of points of shape O. If the shape with the larger number of points has to be aligned to the shape with a lower number of points so that the symmetry criterion holds, some constraints are necessary that will be developed during further work.

In our study we are interested in determining the pair-wise similarity for clustering the set of acquired shapes into groups of similar shapes. The main goal is to learn for each of the established groups a generalized, representative shape. Finally, the set of generalized shapes is used for object recognition. From this point of view we do not need to enforce symmetric results ad hoc. The requirement was to result in a proper dissimilarity measure which holds under a wide variety of different shapes.

5 Conclusions

We have proposed a method for the acquisition of shape instances and our novel algorithm for aligning arbitrary 2D-shapes, represented by ordered point-sets of varying size. Our algorithm aligns two shapes under similarity transformation; differences in rotation, scale, and translation are removed. It establishes one-to-one correspondences between pairs of shapes and ensures that the found correspondences are symmetric and legal. The method detects outlier points and can handle a certain amount of noise. We have evaluated that the algorithm also works well if the aligned shapes are very different, like i.e. the alignment of concave and convex shapes. A distance measure which runs between *0* and *1* is returned as a result.

The methods are implemented in the program *CACM* (case acquisition and case mining)[16] *which runs on a* Windows *PC.*

References

1. D.G. Kendall, A Survey of the Statistical Theory of Shape, Statistical Science, Vol. 4, No. 2, pp. 87-120, 1989.
2. F.L. Bookstein, Size and Shape Spaces for Landmark Data in Two Dimensions, Statistical Science, Vol. 1, No. 2, pp. 181-242, 1986.
3. S. Belongie, J. Malik and J. Puzicha, Shape Matching and Object Recognition Using Shape Contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 24, pp. 509-522, 2002

4. N. Ueda and S. Suzuki, Learning Visual Models from Shape Contours Using Multiscale Convex/Concave Structure Matching, IEEE Trans. on Pattern Analysis and Machine Learning, vol. 15, No. 4, April 1993, p. 307-352.
5. P. Perner and S. Jänichen, Learning of Form Models from Exemplars, In: Ana Fred, Terry Caelli, Robert P. W. Duin, Aurelio Campilho, and Dick de Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Proceedings of the SSPR 2004, Lisbon/Portugal, Springer Verlag, lncs 3138, pp. 153-161, 2004.
6. A. Rangarajan, H. Chui and F.L. Bookstein, The Softassign Procrustes Matching Algorithm, Proc. Information Processing in Medical Imaging, pp. 29-42, 1997.
7. S. Sclaroff and A. Pentland, Modal Matching for Correspondence and Recognition, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 17, No. 6, pp. 545-561, 1995.
8. J. Feldmar and N. Ayache, Rigid, Affine and Locally Affine Registration of Free-Form Surfaces, The International Journal of Computer Vision, Vol. 18, No. 3, pp. 99-119, 1996.
9. A. Hill, C.J. Taylor and A.D. Brett, A Framework for Automatic Landmark Identification Using a New Method of Nonrigid Correspondence, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 3, pp. 241-251, 2000.
10. R.C. Veltkamp, Shape Matching: Similarity Measures and Algorithms, Shape Modelling International, pp. 188-197, 2001.
11. S.R. Lele and J.T. Richtsmeier, An Invariant Approach to Statistical Analysis of Shapes, Chapman & Hall / CRC 2001.
12. P Besl and N. McKay, A Method for Registration of 3-D Shapes, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239-256, 1992.
13. O.-C. Marte and P. Marais, Model-Based Segmentation of CT Images, In South African Computer Journal, Vol. 28, pp. 54-59, 2002.
14. A.W. Fitzgibbon, Robust Registration of 2D and 3D Point Sets, In Proc. British Machine Vision Conference, Vol. II, pp. 411-420, Manchester, UK, 2001
15. S. Belongie, J. Malik and J. Puzicha, Shape Matching and Object Recognition Using Shape Contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 24, pp. 509-522, 2002.
16. P. Perner, S. Jähnichen, Case Acquisition and Case Mining for Case-Based Object Recognition, , In: Peter Funk, Pedro A. González Calero (Eds.), Advances in Case-Based Reasoning, Springer Verlag 2004, lnai 3155, pp. 616-629