

A Pseudo-Metric for Weighted Point Sets

Panos Giannopoulos and Remco C. Veltkamp

Department of Computer Science, Utrecht University
Padualaan 14, 3584 CH Utrecht, The Netherlands
{panos,Remco.Veltkamp}@cs.uu.nl

Abstract. We derive a pseudo-metric for weighted point sets. There are numerous situations, for example in the shape description domain, where the individual points in a feature point set have an associated attribute, a weight. A distance function that incorporates this extra information apart from the points' position can be very useful for matching and retrieval purposes. There are two main approaches to do this. One approach is to interpret the point sets as fuzzy sets. However, a distance measure for fuzzy sets that is a metric, invariant under rigid motion and respects scaling of the underlying ground distance, does not exist. In addition, a Hausdorff-like pseudo-metric fails to differentiate between fuzzy sets with arbitrarily different maximum membership values. The other approach is the Earth Mover's Distance. However, for sets of unequal total weights, it gives zero distance for arbitrarily different sets, and does not obey the triangle inequality. In this paper we derive a distance measure, based on weight transportation, that is invariant under rigid motion, respects scaling, and obeys the triangle inequality, so that it can be used in efficient database searching. Moreover, our pseudo-metric identifies only weight-scaled versions of the same set. We demonstrate its potential use by testing it on two different collections, one of company logos and another one of fish contours.

Keywords: pseudo-metric, weighted point set, shape recognition, indexing, triangle inequality

1 Introduction

In this paper we derive a distance measure for weighted point sets that is invariant under rigid motion, respects scaling, and obeys the triangle inequality, so that it can be used in efficient database searching.

There are numerous situations where the individual points in a point set have an associated value, a weight. It would be very interesting to try to define distance functions that, apart from the position of the points, incorporate this extra information as well.

The meaning of a point weight and the way it affects matching depends on the particular application. For example, the weight could be a certainty measure of the correctness of the points' coordinates. The interpretation for matching could be that points with low weight values match easier than those with high

weight values, since their *position* is uncertain, and they have more freedom to move around. Alternatively, a small weight could mean that the *existence* of the point is uncertain; the smaller its weight the less its influence in the matching.

Another example of the meaning of a weight is the amount of some property such as the curvature of points along a contour. Matching weighted point sets is then a matter of matching one amount of curvature spread over a number of locations, with another one.

If the weights are in $[0, 1]$ they could be interpreted as “membership values” used in fuzzy sets. For the support set S , a fuzzy set U is a function $U : S \rightarrow [0, 1]$. The value $U(x)$ is thought of as the membership value of x in S . In other words, the points have weights from $[0, 1]$, where weight 0 means the point is not contained in the set and weight 1 means the point is completely contained in the set. An ordinary or “crisp” set is a fuzzy set whose membership values are either 0 or 1.

A similarity measure is a function defined on pairs of patterns indicating the degree of their resemblance. Formally speaking, a similarity measure d on a set S is a nonnegative valued function $d : S \times S \rightarrow \mathbb{R}^+ \cup \{0\}$. For many pattern matching applications, it is desirable that d has some of the following properties:

- i. *Self-identity*: For all $x \in S$, $d(x, x) = 0$.
- ii. *Positivity*: For all $x \neq y$ in S , $d(x, y) > 0$.
- iii. *Symmetry*: For all $x, y \in S$, $d(x, y) = d(y, x)$.
- iv. *Triangle inequality*: For all $x, y, z \in S$, $d(x, z) \leq d(x, y) + d(y, z)$.
- v. *Transformation invariant*: For a chosen transformation group G , for all $x, y \in S$, $g \in G$,
 $d(g(x), g(y)) = d(x, y)$. This also implies that $d(g(A), B) = d(A, g^{-1}(B))$.

A function d having properties (i)–(iv) is called a metric. Other combinations are possible: a pseudo-metric is a function that has properties (i), (iii) and (iv), while a semi-metric is a function that obeys only (i), (ii) and (iii). In addition, an invariant metric (pseudo-metric or semi-metric) for a group G can be used to define a new metric (pseudo-metric or semi-metric) on shapes using minimization under all transformations in G .

The triangle inequality is very useful for making searching more efficient [1, 25]. This is based on the following observation. Consider a shape or point set A_1 that closely matches a query A_q : $d(A_1, A_q)$ is small. Let A_r be some reference shape. If the triangle inequality holds, $d(A_r, A_q) \leq d(A_r, A_1) + d(A_1, A_q)$, then we know that $d(A_r, A_q) - d(A_r, A_1)$ is small as well. We can measure the distance between a database shape A_1 and a query A_q by comparing their distances from a reference shape A_r . The computation of the latter distances can be done off-line for all A_i . So at run-time the computation of a single $d(A_r, A_q)$ yields all A_i that resemble A_r as much as A_q , including those that resemble A_q .

It is tempting to demand of any distance defined for fuzzy sets to be an invariant under rigid motion metric and respect scaling of the underlying distance. In [3] it is proved that there is no metric that satisfies all these properties. So it makes sense to take a close look at distance properties, and choose those that are relevant for the application in mind. In this paper we are interested

in a distance measure for weighted point sets that is invariant under rigid motion, respects scaling and obeys the triangle inequality. These properties makes it suitable for shape based image database retrieval: transformation invariance is then used to recognize shape (represented as weighted point sets), and the triangle inequality is used to efficiently search the database. Depending also on the application, the property of *positivity* can be dropped as long as this does not result in counter-intuitive identifications of arbitrarily different sets.

In the next two sections we survey related work on devising distance measures for weighted point sets. We expose advantages and disadvantages of proposed distances in the literature and show that either the properties of interest don't hold, or when they do, they can produce extremely counter-intuitive results. Then in section 4 we introduce a pseudo-metric, the proportional transportation distance (PTD), which has the desired properties and identifies only weight-scaled versions of the same set. Finally we exhibit its potential use by testing it on a company logo collection and a fish contour database.

2 Fuzzy Sets and Hausdorff Distance

An often used metric for general point sets is the Hausdorff distance. The directed Hausdorff distance between two general sets A, B is given by $h^p(A, B) = \sup_{a \in A} \inf_{b \in B} d(a, b)$ where d is one of the L_p metrics. Then the Hausdorff metric for A, B is defined by $h(A, B) = \max\{h^p(A, B), h^p(B, A)\}$. The rest of the section gives an overview of some extensions of the Hausdorff metric to weighted point sets that have been proposed. All of them use the notion of level sets. Let S be the support set and a non-empty compact metric space. For a weighted point set w , the set $\sigma_t(w) = \{x \in S | w(x) \geq t\}$ is the “internal” t -level set of w ; the set $\xi_t(w) = \{x \in S | w(x) \leq t\}$ is called the “external” t -level set of w .

An extension of the Hausdorff distance that is a metric, respects scaling of the sets and is invariant under isomorphisms is defined in [4]. However the serious drawback of this definition is that it is not always valid, since not all the sets have the same maximum membership values. As a consequence the level sets are empty for some sets and membership values, and the Hausdorff distance between the empty and some other set is undefined. This problem can be solved by modifying the sets such that they have the same maximum membership value. This has the undesirable effects that the nature of the sets change, and that sets that initially differ only in their weights cannot be distinguished anymore, e.g. two sets that are equal at all support points except their maxima. In other words, *uniqueness* is not satisfied any more and the distance fails to differentiate between fuzzy sets with arbitrarily different maximum membership values. This problem can be tackled by adding a term that is a metric itself, denoting dissimilarity rather than geometrical distance based on weights of the original sets only. On the other hand, under this correction, the metric will no more respect scaling of the underlying distance. Both definitions coincide with the ordinary Hausdorff distance for crisp sets.

Another metric was defined in [2]. Here the problem of emerging null level-sets is solved by introducing an auxiliary compact set S' where $S \cap S' = \emptyset$, and extending every fuzzy set w defined on S to the fuzzy set w' defined on $S \cup S'$ such that $w'(x) = 0$ for all $x \in S'$. Under this definition the resulting distance is a metric, respects scaling but is not motion invariant. Another disadvantage of this metric is that it does not coincide with the ordinary Hausdorff distance in the case of ordinary sets. For a general compact set S , the selection of S' might be difficult and different selections have different effect on the metric H . The arbitrary choice of S' makes this metric less appealing than it seems, though in the simple case of S being the set of pixels of an image, S' can naturally be the border of the image.

Last, another proposal comes from [8]. If S is a compact metric space then it is bounded in the Euclidean space. We arbitrarily choose the diameter d of S ; then $d = \sup h(U, V)$ exists for all non-empty compact subsets $U, V \subset S$. Then the Hausdorff metric can be extended such that $h(\emptyset, \emptyset) = 0$ and $h(U, \emptyset) = h(\emptyset, U) = d$ for all non-empty compact subsets U of S . Under this convention the distance can be defined as in [4]. This metric is motion invariant, coincides with the Hausdorff distance for crisp sets but does not respect scaling since d is fixed. An intuitive drawback is that the value of the metric can be heavily biased by the diameter d .

3 The Earth Movers Distance

3.1 Definition

Let $A = \{a_1, a_2, \dots, a_m\}$ be a weighted point set such that $a_i = \{(x_i, w_i)\}$, $i = 1, \dots, m$, where $x_i \in \mathbb{R}^k$ with $w_i \in \mathbb{R}^+ \cup \{0\}$ being its corresponding weight. Let also $W = \sum_{j=1}^n w_i$ be the total weight of set A .

The Earth Movers Distance between two weighted point sets measures the minimum amount of work needed to transform one to the other by moving weight under certain conditions which are discussed later on. Intuitively speaking, a weighted point a_i can be seen as an amount of earth or mass, equal to w_i units, situated at x_i ; alternatively it can be taken as an empty hole of w_i units of mass capacity. We can arbitrarily assign the role of the supplier to one set and that of the receiver/demander to the other one, setting, in that way, the direction of weight movement. The EMD then, measures the minimum amount of work needed to fill the holes with earth.

Formally, the EMD can be expressed as a linear programming problem. Given two weighted point sets A, B and a ground distance d , we denote as f_{ij} the elementary flow of weight from x_i to y_j , over the elementary distance d_{ij} . If W, U are the total weights of A, B respectively, the set of all feasible flows $\mathcal{F} = [f_{ij}]$ is defined by the following constraints: (i) $f_{ij} \geq 0$, $i = 1, \dots, m$, $j = 1, \dots, n$, (ii) $\sum_{j=1}^n f_{ij} \leq w_i$, $i = 1, \dots, m$, (iii) $\sum_{i=1}^m f_{ij} \leq u_j$, $j = 1, \dots, n$, (iv) $\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min(W, U)$. These constraints simply say that each particular flow is non-negative, a point from the “supplier” set does not give away more weight than it has and a point from the “receiver” set does not receive

more weight than it needs. Finally the total weight moved is the minimum of the total weights of the two sets.

The flow of weight f_{ij} over a distance d_{ij} , is penalized by its product with this distance. The sum of all these individual products-costs is the total cost paid in order to transform A to B . The $\text{EMD}(A, B)$ is defined as the minimum total cost over $\mathcal{F}$, normalized by the weight of the lighter set; a unit of cost or work corresponds to transporting a unit of weight by a unit of ground distance. That is: $\text{EMD}(A, B) = \frac{\min_{F \in \mathcal{F}} \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}}{\min(W, U)}$. Stated in a different way, the EMD is the average ground distance that weights travels during an optimal flow [6].

An extensive work on the history, theory, modifications and applications of general Mass Transportation Problems can be found in [17] and [18]. The earth movers distance has been used for colour-based image retrieval by Cohen [5] and Rubner [21] as well as shape-based image retrieval [5] and [10].

3.2 Properties and Computation

The most important properties of the EMD can be summarized below:

1. It is a metric if the ground distance is a metric and is applied on the space of equal total weight sets. A nice proof of this can be found in [21].
2. Its transformation invariance group coincides with that of the ground distance (assuming transformations that do not modify the weight of the points).
3. It respects scaling of the underlying ground distance.
4. It is invariant under uniform weight scaling.
5. It allows for partial matching by definition, thus showing some tolerance for occlusion, since the total weights need not be the same. In the case of two sets with unequal total weights, some of the weight of the heavier distribution remains unused or unmatched depending on the direction of the weight flow.
6. It is continuous, in other words, infinitesimal small changes in position and/or weight of existing points cause only infinitesimal change in its value. Moreover addition of a point, i.e noise (which can be seen as increasing its weight from zero to a positive value) can lead to an arbitrary small change in the EMD's value by making the point's weight arbitrarily small. These two facts simply imply that the EMD is deformation and noise robust according to the corresponding robustness axioms presented in [11].

The EMD can be computed efficiently by solving the corresponding linear programming problem, using for example a streamlined version of the simplex algorithm for the transportation problem [12]. In practice simplex performs well, but in theory it can perform an exponential number of steps before giving a solution. Theoretically better - polynomial time - algorithms for general linear programming, like an interior point algorithm could be used; however it is likely to perform better than the simplex method only for very large problem sizes. Since the transportation problem is a special case of the minimum cost flow problem in networks, a polynomial time algorithm that solves the latter can be used as well [19].

3.3 Drawbacks of the EMD

Let $\mathcal{N}$ denote the space of weighted point sets, in which any two sets can have unequal total weights. The EMD has the following drawbacks when applied on $\mathcal{N}$:

- 1. It does not obey the positivity property, see section 1. The EMD does not take into account the surplus of weight, if any, between two sets. As a result, there are cases where it does not distinguish between two non-identical sets. For example in figure 1, B' was constructed by adding just the point b'_i of arbitrary weight u to B . However $EMD(B, B')=0$. We can add new points or weight on existing points on the left of b'_2 thus making B' arbitrary different than B , but its distance to B will not increase.
- 2. It does not obey the triangle inequality. Figure 1 gives a simple counter-example. Moving point c_1 to the right, thus increasing ε , can invalidate the triangle inequality between sets A, B and C : The inequality holds for $\varepsilon = 1$ but not for $\varepsilon = 10$. As a result the EMD prevents the triangle inequality from being used in speeding up database retrieval.

Consequently the EMD on $\mathcal{N}$ is not a metric.

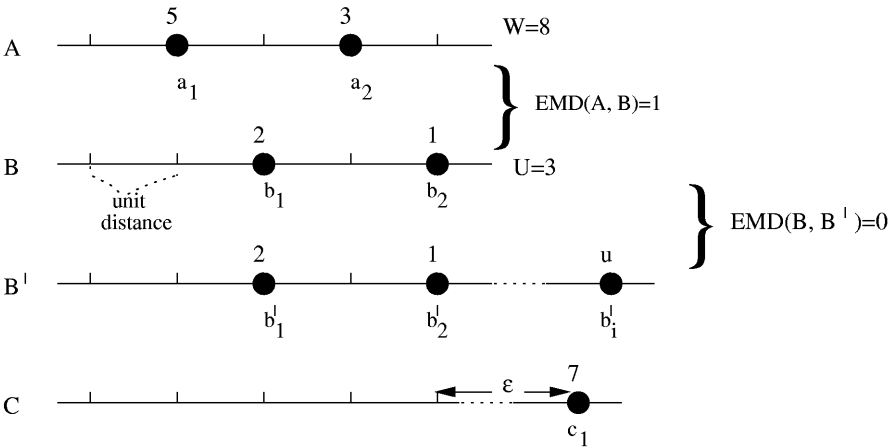


Fig. 1. The EMD does not distinguish between sets B and B' , and does not satisfy the triangle inequality for sets A, B and C .

4 The Proportional Transportation Distance

An interesting question, that naturally arises, is the following: is there a similarity measure based on weight transportation such that the surplus of weight

between two point sets is taken into account and the triangle inequality still holds?

In the sequel we present a new distance for weighted point sets in $\mathcal{N}$. Let $A, B \in \mathcal{N}$. When measuring the distance from A to B , rather than taking A as the supplier and B as the demander moving only as much weight as needed, trying to fill the ‘holes’ with ‘earth’, we move the total weight of A to the positions of the points in B . What we measure then, is the minimum amount of work needed to transform A to a new set A' that resembles B . In particular, we redistribute A ’s total weight from the position of its points, to the position of B ’s points leaving the old percentages of weights in B the same.

We call this distance the Proportional Transportation Distance (PTD); it is defined as follows. Let A, B be two weighted point sets and d a ground distance. The set of all feasible flows $\mathcal{F} = [f_{ij}]$ from A to B , is now defined by the following constraints: (i) $f_{ij} \geq 0, i = 1, \dots, m, j = 1, \dots, n$, (ii) $\sum_{j=1}^n f_{ij} = w_i, i = 1, \dots, m$, (iii) $\sum_{i=1}^m f_{ij} = \frac{u_j W}{U}, j = 1, \dots, n$, (iv) $\sum_{i=1}^m \sum_{j=1}^n f_{ij} = W$.

The $\text{PTD}(A, B)$ is given by: $\text{PTD}(A, B) = \frac{\min_{F \in \mathcal{F}} \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}}{W}$. Constraints 2 and 4 forces all of A ’s weight to move to the positions of points in B . Constraint 3 ensures that this is done in a way that preserves the old percentages of weight in B . Next, we examine PTD’s properties.

4.1 Properties

Let us take a closer look at PTD’s definition. While measuring the $\text{PTD}(A, B)$ for any sets A and B , if we substitute the variables f_{ij} , $i, j = 1..n, m$ in its LP formulation, call it LP_1 with $f'_{ij} W$ ($W \neq 0$ is the total weight of A) we get the following LP problem: $\min_{F \in \mathcal{F}} \sum_{i=1}^m \sum_{j=1}^n f'_{ij} d_{ij}$, where $\mathcal{F}$ is defined by: (i) $f'_{ij} \geq 0$, (ii) $\sum_{j=1}^n f'_{ij} = w_i/W$, (iii) $\sum_{i=1}^m f'_{ij} = u_j/U$, (iv) $\sum_{i=1}^m \sum_{j=1}^n f'_{ij} = 1$.

It is clear that this new formulation, call it LP_2 , gives us the distance between the two sets of percentages of weights in A, B . Note that the total weights of the new sets are both equal to one. Since the substitution function $f_{ij} = f'_{ij} W, W \neq 0$ is bijective, LP_1 is equivalent to LP_2 . This means that we are working again on the space of equal total weight sets.

However, it’s obvious that more than one LP_1 problems can be equivalent to the same LP_2 problem i.e. any two weighted point sets of the same cardinality and positionally coincident, can have the same percentages of weight at the same positions although their corresponding individual weights are different.

We can now state the properties of PTD.

1. It obviously has the *identity* property.
2. It obeys the *triangle inequality*. This follows from the equivalence between the two LP formulations stated above.
3. It does not follow the *positivity* property since the distance between positionally coinciding sets with the same percentages of weights at the same positions is 0. However this is the only case in which the distance between two non-identical point sets is zero. The PTD will distinguish two sets B

and B' where the one came from the other by adding even only one point (see example in figure 1).

4. It has all the other properties that the EMD for equal total weight sets has.

It follows that the PTD is a pseudo-metric. Of course, by identifying sets with zero distance we can produce a metric on the resulting partition of the set $\mathcal{N}$ of generally unequal total weight sets.

5 Experimental Results

We have tested our pseudo-metric on two different image collections, one of logos and another one of fish contours. Our primary objective is to show that it can actually be used in shape-based object recognition; this together with its properties, namely the triangle inequality, would make it a good candidate for image retrieval applications as explained in the introduction of this paper. However, it was not our purpose to find and exhibit the best matching and retrieval procedure possible for logos and fish specifically; in the course of the next sections, we discuss also about how our experimental results can be improved.

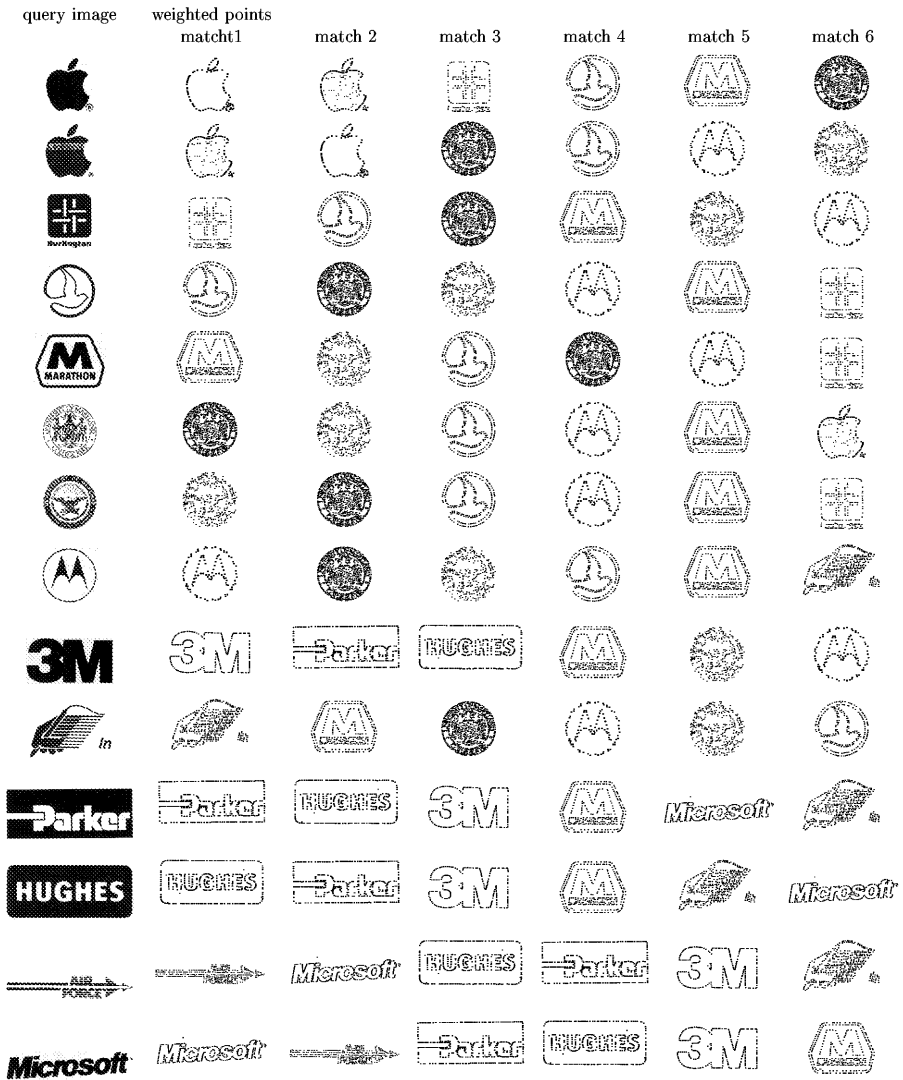
In both cases the original images are reduced to feature weighted point sets. The weight of a point represents its importance in the image: its edge strength value in the first collection and a curvature measure value in the second. In order to calculate the PTD between two sets, each set is first normalized by its total weight as the LP_2 formulation suggests. The distance computation is based on the EMD publicly available code [9]. This code implements a special version of the simplex method, see section 3.2.

5.1 Logos

The UMD-Logo-Database [13] contains 105 grey-scale images that are black and white scanned versions of logos. We have selected only 14 logos, good representatives of the whole database, to test the PTD. These can be seen in the first column of figure 2.

The weighted point set for each logo is constructed as follows. First, we identify edges using a Canny edge detector [23] and corners using the SUSAN corner detector [22] and [26]. Then an intermediate weighted point set is constructed by selecting only edge points that are also corners, keeping the intensity in the edge image, rather than in the original image, as the point weight. The higher the intensity the more important the point. In this way, we keep the most dominant in the logo's shape points reducing the initial number of edge points from some thousands to some hundreds - in most cases less than 500. This makes the computation of the PTD between two sets possible in a reasonable amount of time. For example the PTD between two sets of around 300 points each, takes no more than 20 seconds on a typical Pentium III, 500MHz with 128MB of memory.

Finally, the bounding box of the logo in each image is first centered to the origin and then scaled to a unit area rectangle. Taking into account the purpose



tion precedes the normalization step, outliers may be prevented from wrongly determining the bounding box of each logo's.

We have computed the distance for every pair of the selected logos. Each line in figure 2 shows the initial logo image, its corresponding weighted point set and best 6 matches. The width of the dots in the weighted point sets is relative to the values of the weight. This can be easily seen in the apple in the second row where points in its interior have lower weight than points on its outline. As an indication of the actual distance values, here are the values for the Apple logo in the first row, from left to right: 0.0, 0.087, 0.146, 0.147, 0.152 and 0.157. Interestingly enough, the worst match for both Apple logos is the Microsoft logo (this is not shown here due to limited space) with a distance value of 0.3958 and 0.3956 respectively.

For almost all logos the weights are within a very small range. Discarding the weights, thus setting them all to 1, can produce bad matches only for those sets that have many points with low weight. For example a query with the second in the query column Apple logo, gives as a best match, after itself of course, the rounded logo with the bird inside (fourth in the query image column); the other Apple logo, first in the column, comes second. This is due to the fact that more weight in the interior of the query Apple logo has to travel towards the outline of the first Apple logo. In general, the use of the edge strength as the weight value will really pay off if the test database consists of noisy e.g blurred images, provided that our feature detectors are good enough to avoid most of the noise. Then, the higher the intensity of a point in the edge image, the more certain we are about its participation in the object's shape.

query points matcht1	match 2	match 3	match 4	match 5	match 6
					

Fig. 3. Top six matches of the Apple logo using the EMD.

Last, we have compared our PTD results with those of EMD: for the second in the query column Apple logo, the best six matches using EMD are shown in figure 3. The reason behind this is that the weighted point that corresponds to the Apple logo has almost a hundred points while the one of the University of Maryland has almost a thousand and much more total weight. When they are matched almost every point in the Apple's set will coincide or be very close to a point in the Maryland's set and thus little weight is moved.

5.2 Fish Contours

Our second test data set comes from the SQUID database [7] which contains 1100 images of contours of marine animals, mostly fish. Each one of them is thinned to a one pixel width contour which is actually a simple polygon with its defining points ordered, in our case, clockwise.

Preprocessing. In order for our computations to be independent of orientation, position and scaling the following preprocessing steps are applied to the polygons. First, each polygon's center of mass is translated to the origin. Then the polygon is rotated around the origin in such a way that its major principal axis, identified with the PCA method [15], coincides with the x-axis. Next, each polygon is re-translated so that the center of its bounding box lies on the origin. The reason behind this translation lies in our choice of corners as feature points, see next section; since they probably lie on the bounding box, we would like them to play a bigger role in the translational normalization. Finally the polygon is scaled so that its bounding box has unit area.

Corner Extraction. Corners are points with high curvature and therefore significant for a polygon's shape. As already mentioned in the beginning of section 5, we would like to select points with high curvature, and construct a weighted point set with the weights set to the actual curvature value. However, discrete curvature estimation is a difficult problem by itself. Several algorithms have been proposed, each one with its advantages and drawbacks [27].

We used a simple, fast and easy to implement corner detection algorithm by Rosenfeld and Weszka [20], [24]. The algorithm computes the value of a curvature measure for each point and then retains only points of local maxima as corners. A smoothing parameter determines the region of support, the curvature measure value for each point and, as a result, the number of output corners. We set this smoothing parameter to 0.045 for all input polygons. This gives us a small number of corners between 15 and 30 in most cases.

Here we should re-emphasize the fact that this curvature measure was used as a way to obtain weighted point sets and demonstrate the use of the proposed pseudo-metric and is not of the best interest of the application itself; we applied it on one particular collection where all the contours have been generated in the same way and on the same scale. Moreover, while computing the curvature measure, the same smoothing parameter was used for all the contours. Further discussion on the effects of this feature extraction choice on the matching results can be found in the next section.

Distance Computation and Results. We computed the distance for all pairs of all fish polygons. We should mention here that although the PCA method gives the two principal axis, it lacks any information about their direction, resulting thus in a two way ambiguity for each of them. In our case this means that there are two configurations resulting in the same set of principal axis. This has been

observed by other researchers as well [14]. We deal with this problem as follows. Let A, B be two normalized weighted point sets and $-B$ a rotated by 180 degrees around the origin version of B . Then: $PTD(A, B) = \min\{PTD(A, B), PTD(A, -B)\}$. Since the sets are small the computation of the distance takes no more than 0.2 seconds on a machine with the same configuration as the one used for the logo collection.

In figure 4 the five best matches for example polygon queries are shown. The results are organized in pairs of rows where the first row shows the normalized polygon and the second the normalized weighted point set. Their orientation is the one of the two that gave the minimum distance according to the previous paragraph. The distance values, from left to right, for the shark query are as follows: 0.0, 0.082, 0.091, 0.102, 0.109. In contrast with the point sets in figure 2, the weights here are spread over the whole range of values in $(0, 1)$; this is shown by dots of variable width. Thus discarding the weights, influences the results; the same, more or less, set of shapes is then retrieved but more similar shapes can come after less similar ones.

The corner extraction method used here imposes a trade off between the number of output corner points and the relevance of the curvature measure value to the actual curvature value. Adjusting the smoothing parameter to give out more corners will result in a weight distribution quite different from the actual curvature distribution. Less corners, however, enclose less information on the polygon's shape.

Figure 5 shows a counter-intuitive example. Here, the first column corresponds to the query and best match and the second to the second best match. Note for example that the two spikes close to the 'belly' of the second fish are matched with point's on the 'belly' of the first fish although the latter does not have spikes. Note also that the belly of the second fish is represented in its weighted point set only by one point of low weight. A bad corner detector and curvature measure can give point sets with points in close positions which correspond to dissimilar shapes. A way to prevent bad matches, is to incorporate a better curvature measure together with a way of selecting a set of points that contains almost all points of relatively high curvature and is big enough to outline the polygon's shape.

We have examined many other query results through a simple demo that can be also found in our web page [16]. Although shape similarity is a highly subjective matter, generally argued similar shapes are given as a result for most queries. Overall the results are satisfactory and indicate that our pseudo-metric is promising.

Moreover, the web site demonstrates two ways of retrieval: the full distance matrix approach and the vantage (reference) shapes method, described in section 1. In the first all the distances are precomputed. This approach is of course fast, and gives all best matches that the user is interested in without false positives. A query with a new image can be dealt with by using linear search, where the distances from the new image to all the images in the database are computed. In our case, taking into account the average size of a weighted point set

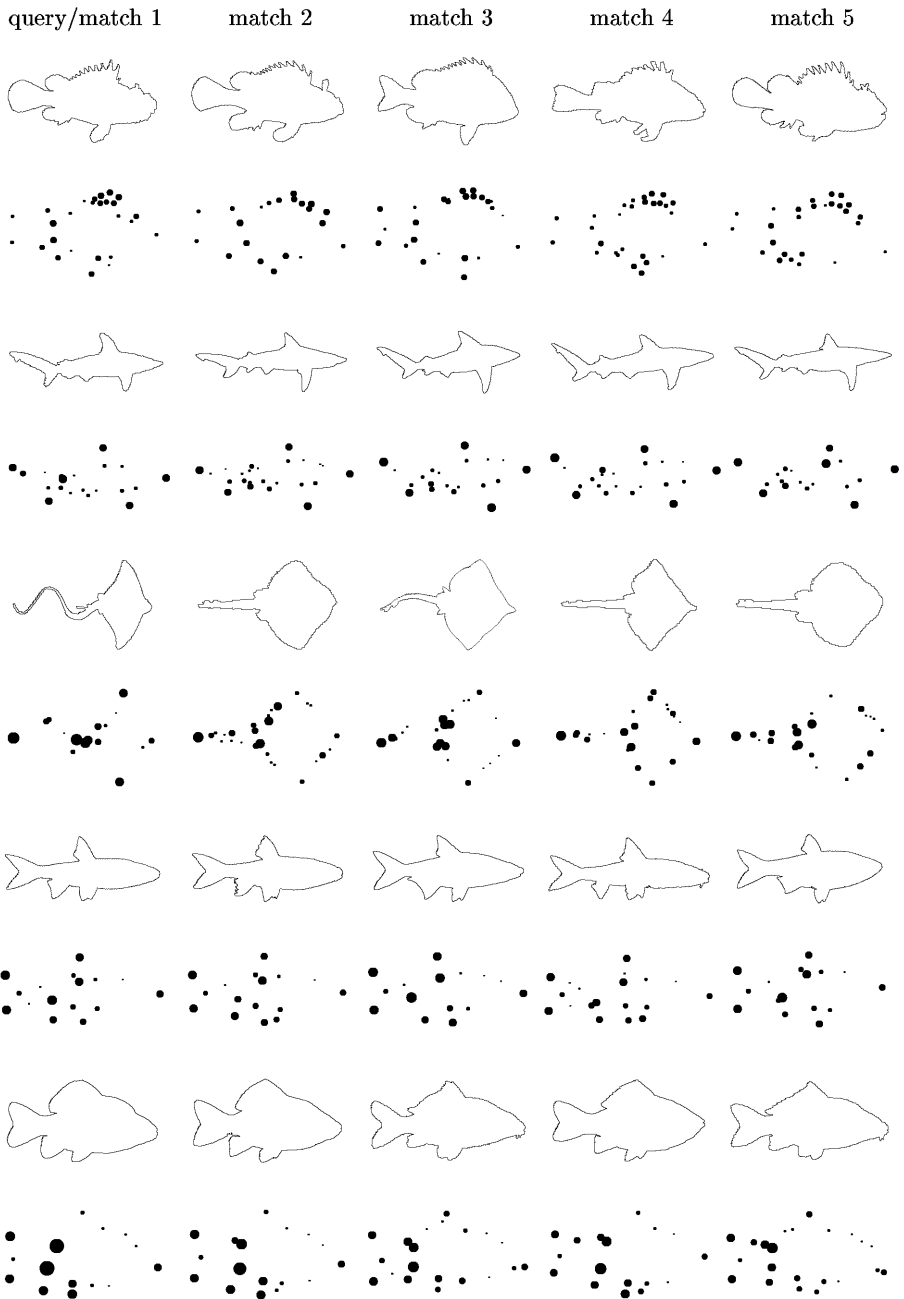


Fig. 4. Top five matches for selected queries of fish polygons.

query/match 1	match 2
	
	

Fig. 5. A counter-intuitive match.

corresponding to a fish contour, this method will give a query time of about three to four minutes; clearly, the linear search method is not at all suitable for larger collections. An alternative to the latter is the vantage objects method which, taking advantage of the triangle inequality, is both scalable and fast. Given a query image and a positive constant ϵ , all images in the database that are within distance ϵ from the query are returned, together with some false positives. In our case, using for example ten vantage objects, the query time is not more than two seconds, on the average.

6 Concluding Remarks

We have presented a pseudo-metric for weighted point sets, based on weight transportation. Our distance is an improvement over previous approaches in that it obeys triangle inequality, respects scaling of the ground distance, is invariant under rigid motion and does not identify arbitrarily different point sets.

We show that PTD can be used in shape based matching and, through triangle inequality, in efficient shape retrieval by testing it on two collections with different characteristics. For the company logo collection, the weights are set to an intensity value indicating edge strength; for the fish contour database the weights are set to a curvature measure value. We also show the importance of good feature extraction in the matching results giving directions on how to improve them as well.

Finally, we should mention that in order to take into account transformation factors, a different approach is to minimize the proposed pseudo-metric under all transformations in a given transformation group. An algorithm for minimizing the EMD was given in [6]. Although this approach increases the computation time considerably and can be trapped in local minima, it is a better approach for applications that require partial matching when the EMD on unequal total weight sets is used [5]. A more general and difficult to solve partial matching problem can be formulated as follows. Given two sets A, B , $\epsilon \geq 0$, $k \geq 0$ and a

transformation group G find subsets A' of A and B' of B , with $|A'| \geq k$, and $g \in G$ such that $PTD(g(A'), B') \leq \epsilon$. This gives us some challenging future research directions.

Acknowledgements. We thank Geert J. Giezeman for his help in building the demo, and Mirela Tanase for the contour thinning code and useful discussions. This work was partially supported by STW, project Shape Matching Environment, number UIF.5055.

References

1. Julio Barros, James French, Worthy Martin, Patrick Kelly, and Mike Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In *Proceedings of SPIE, volume 2670, Storage and Retrieval for Still Image and Video Databases IV*, pages 392–403, 1996.
2. L. Boxer. On hausdorff-like metrics for fuzzy sets. *Pattern Recognition Letters*, 18, 1997.
3. Peter Braß. On the non-existence of hausdorff-like metrics for fuzzy sets. Technical Report (B 00-02), Freie Universität Berlin, Fachbereich Mathematik und Informatik, 2000.
4. B.B. Chaudhuri and A. Rosenfeld. On a metric distance between fuzzy sets. *Pattern Recognition Letters*, 17, 1996.
5. Scott Cohen. *Finding Color and Shape Patterns in Images*. PhD thesis, Stanford University, Department of Computer Science, 1999.
6. Scott Cohen and Leonidas Guibas. The earth mover's distance under transformation sets. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, pages 173–187, September 1999.
7. SQUID database. <http://www.ee.surrey.ac.uk/research/vssp/imagedb/demo.html>.
8. Jiu-Liu Fan. Note on hausdorff-like metrics for fuzzy sets. *Pattern Recognition Letters*, 19, 1998.
9. Code for the Earth Movers Distance(EMD). <http://vision.stanford.edu/~rubner/emd/default.htm>.
10. David Spotts Fry. *Shape Recognition using Metric on the Space of Shapes*. PhD thesis, Harvard University, Division of Applied Sciences, 1993.
11. Michiel Hagerdoorn. *Pattern matching using similarity measures*. PhD thesis, Universiteit Utrecht, Institute of Information and Computer Science, 2000.
12. S. Hillier and Gerald J. Lieberman. *Introduction to Mathematical Programming*. McGraw-Hill, 1990.
13. UMD logo database. <http://documents.cfar.umd.edu/resources/database/umdlogo.html>.
14. M. Novotni and R. Klein. A geometric approach to 3d object comparison. In *International Conference on Shape Modelling and Applications*, pages 167–175, 2001.
15. M. Petrou and P. Bosedgianni. *Image Processing: The Fundamentals*. John Wiley, 1999.
16. Fish polygon retrieval demo. <http://give-lab.cs.uu.nl/matching/ptd>.
17. Svetlozar T. Rachev and Ludger Rüschendorf. *Mass Transportation Problems*, volume I: Theory. Springer, 1998.

18. Svetlozar T. Rachev and Ludger Rüschendorf. *Mass Transportation Problems*, volume II: Applications. Springer, 1998.
19. Thomas L. Magnanti Ravindra K. Ahuja and James B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, 1993.
20. Azriel Rosenfeld and Joan S. Weszka. An improved method of angle detection on digital curves. *IEEE Transactions on Computers*, 24(9):940–941, 1975.
21. Yossi Rubner. *Perceptual Metrics for Image Database Navigation*. PhD thesis, Stanford University, Department of Computer Science, 1999.
22. S.M. Smith and J. M. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, May 1997.
23. Canny Edge Detection Software.
<http://www.computing.edu.au/~geoff/ftp.html>.
24. C.-H. Teh and R.T. Chin. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):859– 872, 1989.
25. Jules Vleugels and Remco C. Veltkamp. Efficient image retrieval through vantage objects. *Pattern Recognition*, 35(1):69–80, 2002.
26. SUSAN's web page. <http://www.fmrib.ox.ac.uk/~steve/susan/index.html>.
27. Marcel Worring and Arnold W. M. Smeulders. Digital curvature estimation. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 58(3):366–382, 1993.