Shape Matching: Similarity Measures and Algorithms

Remco C. Veltkamp

UU-CS-2001-03
January 2001

Shape Matching: Similarity Measures and Algorithms

Remco C. Veltkamp
Dept. Computing Science, Utrecht University
Padualaan 14, 3584 CH Utrecht, The Netherlands
Remco.Veltkamp@cs.uu.nl

January 25, 2001

Abstract

Shape matching is an important ingredient in shape retrieval, recognition and classification, alignment and registration, and approximation and simplification. This paper treats various aspects that are needed to solve shape matching problems: choosing the precise problem, selecting the properties of the similarity measure that are needed for the problem, choosing the specific similarity measure, and constructing the algorithm to compute the similarity. The focus is on methods that lie close to the field of computational geometry.

1 Introduction

There is no universal definition of what shape is. Impressions of shape can be conveyed by color or intensity patterns (texture), from which a geometrical representation can be derived. This is shown already in Plato's work Meno (380 BC) [41]. This is one of the so-called Socratic dialogues, where two persons discuss aspects of virtue; to honor and memorialize him, one person is called Socrates. In the dialogue, Socrates describes shape as follows (the word 'figure' is used for shape):

> "figure is the only existing thing that is found always following color".

This does not satisfy Meno, after which Socrates gives a definition in "terms employed in geometrical problems":

> "figure is limit of solid".

Here we also consider shape as something geometrical. We will use the term shape for a geometrical pattern, consisting of a set of points, curves, surfaces, solids, etc. This is commonly done, although 'shape' is sometimes used for a geometrical pattern modulo some transformation group, in particular similarity transformations (combinations of translations, rotations, and scalings).

Shape matching deals with transforming a shape, and measuring the resemblance with another one, using some similarity measure. So, shape similarity measures are an essential ingredient in shape matching. Although the term similarity is often used, dissimilarity corresponds to the notion of distance: small distance means small dissimilarity, and large similarity.

The algorithm to compute the similarity often depends on the precise measure, which depends on the required properties, which in turn depends on the particular matching problem for the application at hand. Therefore, section 2 is about the classification of matching problems, section 3 is about similarity measure properties, section 4 presents a number of specific similarity measures, and section 5 treats a few matching algorithms.

There are various ways to approach the shape matching problem. In this article we focus on methods that lie close to computational geometry, the subarea of algorithms design that deals with the design and analysis of algorithms for geometric problems involving objects like points, lines, polygons, and polyhedra. The standard approach taken in computational geometry is the development of exact, provably correct and efficient solutions to geometric problems. First some related work is mentioned in the next subsection.

1.1 Related work

Matching has been approached in a number of ways, including tree pruning [55], the generalized Hough transform [8] or pose clustering [51], geometric hashing [59], the alignment method [27], statistics [40], deformable templates [50], relaxation labeling [44], Fourier descriptors [35], wavelet transform [31], curvature scale space [36], and neural networks [21]. The following subsections treat a few methods in more detail. They are based on shape representations that depend on the global shape. Therefore, they are not robust against occlusion, and do not allow partial matching.

1.1.1 Moments

When a complete object in an image has been identified, it can be described by a set of moments $m_{p,q}$. The (p, q)-moment of an object $O \subseteq \mathbb{R}^2$ is given by

$$m_{p,q} = \int_{(x,y)\in O} x^p y^q \; dx \; dy.$$

For finite point sets the integral can be replaced by a summation. The infinite sequence of moments, $p, q = 0, 1, \ldots$, uniquely determines the shape, and vice versa. Variations such as Zernike moments are described in [32] and [12].

Based on such moments, a number of functions, moment invariants, can be defined that are invariant under certain transformations such as translation, scaling, and rotation. Using only a limited number of low order moment invariants, the less critical and noisy high order moments are discarded. A number of such moment invariants can be put into a feature vector, which can be used for matching.

Algebraic moments and other global object features such as area, circularity, eccentricity, compactness, major axis orientation, Euler number, concavity tree, shape numbers, can all be used for shape description [9], [42].

1.1.2 Modal matching

Rather than working with the area of a 2D object, the boundary can be used instead. Samples of the boundary can be described with Fourier descriptors, the coefficients of the discrete Fourier transform [56].

Another form of shape decomposition is the decomposition into an ordered set of eigenvectors, also called principal components. Again, the noisy high order components can be discarded, using only the most robust components. The idea is to consider n points on the boundary of an object, and to define a matrix D such that element D_{ij} determines how boundary points i and j of the object interact, typically involving the distance between points i and j.

The eigenvectors e_i of D, satisfying $D\,e_i = \lambda e_i$, $i = 1, \ldots, n$, are the *modes* of D, also called eigenshapes. To match two shapes, take the eigenvectors e_i of one object, and the eigenvectors e'_j of the other object, and compute a mismatch value $m(e_i, e'_j)$. For simplicity, let us assume that the

eigenvectors have the same length. For a fixed $i = i_0$, determine the value j_0 of j for which $m(e_{i_0}, e'_j)$ is minimal. If the value of i for which $m(e_i, e'_{j_0})$ is minimal is equal to i_0, then point i of one shape and point j of the other shape match. See for example [22] and [49] for variations on this basic technique of modal matching.

1.1.3 Curvature scale space

Another approach is the use of a scale space representation of the curvature of the contour of 2D objects. Let the contour C be parameterized by arc-length s: $C(s) = (x(s), y(s))$. The coordinate functions of C are convolved with a Gaussian kernel ϕ_σ of width σ:

$$x_\sigma(s) = \int x(s)\phi_\sigma(t-s)\,dt, \qquad \phi_\sigma(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{t^2}{2\sigma^2}},$$

and the same for $y(s)$. With increasing value of σ, the resulting contour gets smoother, see figure 1, and the number of zero crossings of the curvature along it decreases, until finally the contour is convex and the curvature is positive everywhere.

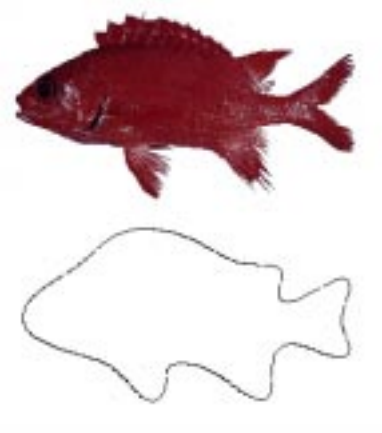

Figure 1: Contour smoothing, reducing curvature changes.

For continuously increasing σ, the positions of the curvature zero-crossings continuously move along the contour, until two such positions meet and annihilate. Matching of two objects can be done by matching points of annihilation in the (s, σ) plane [36]. Another way of reducing curvature changes is based on the turning angle function (see section 4.5) [34].

Matching with the curvature scale space is robust against slight affine distortion, as has been experimentally determined [37]. Be careful, however, to use this property for fish recognition, see section 3.

2 Matching problems

Shape matching is studied in various forms. Given two patterns and a dissimilarity measure, we can:

- (computation problem) compute the dissimilarity between the two patterns,
- (decision problem) for a given threshold, decide whether the dissimilarity is smaller than the threshold,

- (decision problem) for a given threshold, decide whether there exists a transformation such that the dissimilarity between the transformed pattern and the other pattern is smaller than the threshold,
- (optimization problem) find the transformation that minimizes the dissimilarity between the transformed pattern and the other pattern.

Sometimes the time complexities to solve these problems are rather high, so that it makes sense to devise approximation algorithms:

- (approximate optimization problem) find a transformation that gives a dissimilarity between the two patterns that is within a constant multiplicative factor from the minimum dissimilarity.

These problems play an important role in the following categories of applications.

Shape retrieval: search for all shapes in a typically large database of shapes that are similar to a query shape. Usually all shapes within a given distance from the query are determined (decision problem), or the first few shapes that have the smallest distance (computation problem). If the database is large, it may be infeasible to compute the similarity between the query and every database shape. An indexing structure can help to exclude large parts of the database from consideration at an early stage of the search, often using some form of triangle inequality property, see section 3.

Shape recognition and classification: determine whether a given shape matches a model sufficiently close (decision problem), or which of k class representatives is most similar (k computation problems).

Shape alignment and registration: transform one shape so that it best matches another shape (optimization problem), in whole or in part.

Shape approximation and simplification: construct a shape of fewer elements (points, segments, triangles, etc.), that is still similar to the original. There are many heuristics for approximating polygonal curves [45] and polyhedral surfaces [26]. Optimal methods construct an approximation with the fewest elements given a maximal dissimilarity, or with the smallest dissimilarity given the maximal number of elements. (Checking the former dissimilarity is a decision problem, the latter is a computation problem.)

3 Properties

In this section we list a number of properties. It can be desirable that a similarity measures has such properties. Whether or not specific properties are wanted will depend on the application at hand, sometimes a property will be useful, sometimes it will be undesirable. Some combinations of properties are contradictory, so that no distance function can be found satisfying them. A shape dissimilarity measure, or distance function, on a collection of shapes S is a function $d : S \times S \to \mathbb{R}$. In the properties listed below, it is understood that they must hold for all shapes A, B, or C in S.

Metric properties Nonnegativity means $d(A, B) \geq 0$. The identity property is that $d(A, A) = 0$. Uniqueness says that $d(A, B) = 0$ implies $A = B$. A strong form of the triangle inequality is $d(A, B)+d(A, C) \geq d(B, C)$. A distance function satisfying identity, uniqueness, and strong triangle inequality is called a metric. If a function satisfies only identity and strong triangle inequality, then

it is called a semimetric. Symmetry, $d(A, B) = d(B, A)$, follows from the strong triangle inequality and identity. Symmetry is not always wanted. Indeed, human perception does not always find that shape A is equally similar to B, as B is to A. In particular, a variant A of prototype B is often found more similar to B than vice versa [53].

A more frequently encountered formulation of the triangle inequality is the following: $d(A, B) + d(B, C) \geq d(A, C)$. Similarity measures for partial matching, giving a small distance $d(A, B)$ if a part of A matches a part of B, do not obey the triangle inequality. An illustration is given in figure 2: the distance from the man to the centaur is small, the distance from the centaur to the horse is small, but the distance from the man to the horse is large, so $d(man, centaur) + d(centaur, horse) > d(man, horse)$ does not hold. It therefore makes sense to formulate an even weaker form, the relaxed triangle inequality [18]: $c(d(A, B) + d(B, C)) \geq d(A, C)$, for some constant $c \geq 1$.

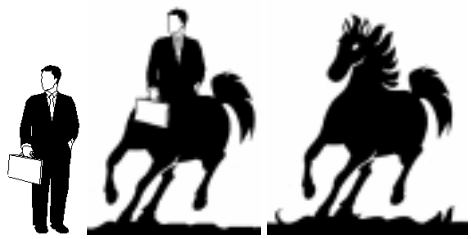

Figure 2: Under partial matching, the triangle inequality does not hold.

Continuity properties It is often desirable that a similarity function has some continuity properties. The following four properties are about robustness, a form of continuity. Such properties are for example useful to be robust against the effects of discretization. Perturbation robustness: for each $\epsilon > 0$, there is an open set F of deformations sufficiently close to the identity, such that $d(f(A), A) < \epsilon$ for all $f \in F$. For example, it can be desirable that a distance function is robust against small affine distortion. Crack robustness: for each each $\epsilon > 0$, and each "crack" x in $bd(A)$, the boundary of A, an open neighborhood U of x exists such that for all B, $A - U = B - U$ implies $d(A, B) < \epsilon$. Blur robustness: for each $\epsilon > 0$, an open neighborhood U of $bd(A)$ exists, such that $d(A, B) < \epsilon$ for all B satisfying $B - U = A - U$ and $bd(A) \subseteq bd(B)$. Noise and occlusion robustness: for each $x \in \mathbb{R}^2 - A$, and each $\epsilon > 0$, an open neighborhood U of x exists such that for all B, $B - U = A - U$ implies $d(A, B) < \epsilon$.

Invariance A distance function d is invariant under a chosen group of transformations G if for all $g \in G$, $d(g(A), g(B)) = d(A, B)$. For object recognition, it is often desirable that the similarity measure is invariant under affine transformations, since this is a good approximation of weak perspective projections of points lying in or close to a plane. However, it depends on the application whether a large invariance group is wanted. For example, Sir d'Arcy Thompson [52] showed that the outlines of two hatchetfishes of different genus, *Argyropelecus olfersi* and *Sternoptyx diaphana*, can be transformed into each other by shear and scaling, see figure 3. So, the two fishes will be found to match the same model if the matching is invariant under affine transformations.

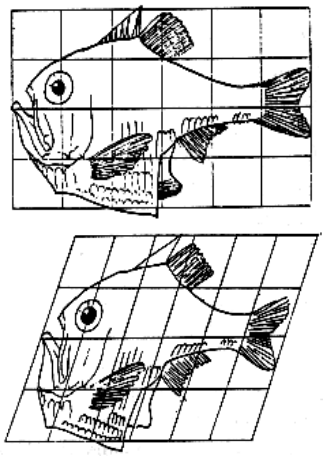

Figure 3: Two hatchetfishes of different genus: *Argyropelecus olfersi* and *Sternoptyx diaphana*. After [52].

4 Similarity Measures

4.1 Discrete Metric

Finding an affine invariant metric for patterns is not so difficult. Indeed, a metric that is invariant not only for affine transformations, but for general homeomorphisms is the discrete metric:

$$d(A,B) = \begin{cases} 0 & \text{if } A \text{ equals } B \\ 1 & \text{otherwise} \end{cases}$$

However, this metric lacks useful properties. For example, if a pattern A is only slightly distorted to form a pattern A', the discrete distance $d(A, A')$ is already maximal.

Under the discrete metric, computing the smallest $d(A, B)$ over all transformations in a transformation group G is equivalent to deciding whether there is some transformation g in G such that $g(A)$ equals B. This is known as exact congruence matching. For sets of n points in $\mathbb{R}^k$, the algorithms with the best known time complexity run in $\mathcal{O}(n \log n)$ time if G is translations, scaling, or homotheties (translation plus scaling), and $\mathcal{O}(n^{\lceil k/3 \rceil} \log n)$ time for rotations, isometries, and similarities [11].

4.2 L_p Distance, Minkowski Distance

Many similarity measures on shapes are based on the L_p distance between two points. For two points x, y in $\mathbb{R}^k$, the L_p distance is defined as $L_p(x,y) = (\sum_{i=0}^{k} |x_i - y_i|^p)^{1/p}$. This is also often called the Minkowski distance. For $p = 2$, this yields the Euclidean distance L_2. For $p = 1$, we get the Manhattan, city block, or taxicab distance L_1. For p approaching ∞, we get the max metric: $\max_i(|x_i - y_i|)$.

For all $p \geq 1$, the L_p distances are metrics. For $p < 1$ it is not a metric anymore, since the triangle inequality does not hold.

4.3 Bottleneck Distance

Let A and B be two point sets of size n, and $d(a, b)$ a distance between two points. The bottleneck distance $F(A, B)$ is the minimum over all $1-1$ correspondences f between A and B of the maximum distance $d(a, f(a))$. For the distance $d(a, b)$ between two points, an L_p distance could be chosen. An alternative is to compute an approximation $\tilde{F}$ to the real bottleneck distance F. An approximate matching between A and B with $\tilde{F}$ the furthest matched pair, such that $F < \tilde{F} < (1+\epsilon)F$, can be computed with a less complex algorithm [17].

The decision problem for translations, deciding whether there exists a translation ℓ such that $F(A+\ell, B) < \epsilon$ can also be solved, but takes considerably more time [17]. Because of the high degree in the computational complexity, it is interesting to look at approximations with a factor ϵ: $F(A+\ell, B) < (1+\epsilon)F(A+\ell^*, T)$, where ℓ^* is the optimal translation. Finding such a translation can be done in $\mathcal{O}(n^{2.5})$ time [48].

Variations on the bottleneck distance are the minimum weight distance, the most uniform distance, and the minimum deviation distance.

4.4 Hausdorff Distance

In many applications, for example stereo matching, not all points from A need to have a corresponding point in B, due to occlusion and noise. Typically, the two point sets are of different size, so that no one-to-one correspondence exists between all points. In that case, a dissimilarity measure that is often used is the Hausdorff distance. The Hausdorff distance is defined not only for finite point sets, it is defined on nonempty closed bounded subsets of any metric space.

The *directed* Hausdorff distance $\vec{h}(A, B)$ is defined as the lowest upper bound (supremum) over all points in A of the distances to B: $\vec{h}(A, B) = \sup_{a \in A} \inf_{b \in B} d(a, b)$, with $d(a, b)$ the underlying distance, for example the Euclidean distance (L_2). The Hausdorff distance $H(A, B)$ is the maximum of $\vec{h}(A, B)$ and $\vec{h}(B, A)$: $H(A, B) = \max\{\vec{d}(A, B), \vec{d}(B, A)\}$. For finite point sets, it can be computed using Voronoi diagrams in time $\mathcal{O}((m+n)\log(m+n))$ [2].

Given two finit point sets A and B, the translation ℓ^* that minimizes the Hausdorff distance $d(A+\ell, B)$ can be determined in time $\mathcal{O}(mn(\log mn)^2)$ when the underlying metric is L_1 or L_∞ [14]. For other L_p metrics, $p = 2, 3, \ldots$ it can be computed in time $\mathcal{O}(mn(m+n)\alpha(mn)\log(m+n))$ [29]. ($\alpha(n)$ is the inverse Ackermann function, a very slowly increasing function.) This is done using the upper envelopes of Voronoi surfaces.

Computing the optimal rigid motion r (translation plus rotation), minimizing $H(r(A), B)$ can be done in $\mathcal{O}((m+n)^6 \log(mn))$ time [28]. This is done using dynamic Voronoi diagrams. Given a real value ϵ, deciding if there is a rigid motion such that $H(r(A), B) < \epsilon$ can be done in time $\mathcal{O}((m+n)m^2n^2 \log mn)$ [13].

Given the high complexities of these problems, it makes sense to look at approximations. Computing an approximate optimal Hausdorff distance under translation and rigid motion is discussed in [1].

The Hausdorff distance is very sensitive to noise: a single outlier can determine the distance value. For finite point sets, a similar measure that is not as sensitive is the *partial Hausdorff distance*. It is the maximum of the two directed partial Hausdorff distances: $H_k(A, B) = \max\{\vec{h}_k(A, B), \vec{h}_k(B, A)\}$, where the directed distances are defined as the k-th value in increasing order of the distance from a point in A to B: $\vec{h}_k(A, B) = k^{th}_{a \in A} \min_{b \in B} d(a, b)$. The partial Hausdorff distance is not a metric since it fails the triangle inequality. Deciding whether there is a translation plus scaling that brings the partial Hausdorff distance under a given threshold is done in [30] by means of a transformation space

subdivision scheme. The running time depends on the depth of subdivision of transformation space.

For pattern matching, the Hausdorff metric is often too sensitive to noise. For finite point sets, the partial Hausdorff distance is not that sensitive, but it is no metric. Alternatively, [7] observes that the Hausdorff distance of $A, B \subseteq X$, X having a finite number of elements, can be written as $H(A,B) = \sup_{x \in X} |d(x,A) - d(x,B)|$. The supremum is then replaced by an average: $\Delta^p(A,B) = (\frac{1}{|X|} \sum_{x \in X} |d(x,A) - d(x,B)|^p)^{1/p}$, where $d(x,A) = \inf_{a \in A} d(x,a)$, resulting in the ***p-th order mean Hausdorff distance***. This is a metric less sensitive to noise, but it is still not robust. It can also not be used for partial matching, since it depends on all points.

4.5 Turning Function Distance

The cumulative angle function, or turning function, $\Theta_A(s)$ of a polygon or polyline A gives the angle between the counterclockwise tangent and the x-axis as a function of the arc length s. $\Theta_A(s)$ keeps track of the turning that takes place, increasing with left hand turns, and decreasing with right hand turns, see figure 4.

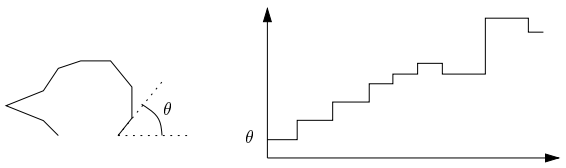

Figure 4: Polygonal curve and turning function.

Clearly, this function is invariant under translation of the polyline. Rotating a polyline over an angle θ results in a vertical shift of the function with an amount θ. For polygons and polylines, the turning function is a piecewise constant function, increasing or decreasing at the vertices, and constant between two consecutive vertices.

In [6] the turning angle function is used to match polygons. First the size of the polygons are scaled so that they have equal perimeter. The L_p metric on function spaces, applied to Θ_A and Θ_B, gives a dissimilarity measure on A and B: $d(A,B) = \left(\int |\Theta_A(s) - \Theta_B(s)|^p \, ds \right)^{1/p}$, see figure 5.

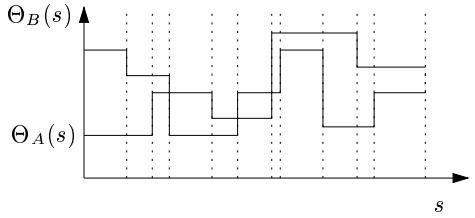

Figure 5: Rectangles enclosed by $\Theta_A(s)$, $\Theta_B(s)$, and dotted lines are used for evaluation of dissimilarity.

In [58], for the purpose of retrieving hieroglyphic shapes, polyline curves do not have the same length, so that partial matching can be performed. In that case the starting point of the shorter one is moved along the longer one, considering only the turning function where the arc lengths overlap. This is a variation of the algorithms for matching closed polygons with respect to the turning function, which can be done in $\mathcal{O}(mn \log(mn))$ time [6].

Partial matching under scaling, in addition to translation and rotation, is more involved. It can be done in time $\mathcal{O}(m^2 n^2)$, see [15].

4.6 Fréchet Distance

The Hausdorff distance is often not appropriate to measure the dissimilarity between curves. For all points on A, the distance to the closest point on B may be small, but if we walk forward along curves A and B simultaneously, and measure the distance between corresponding points, the maximum of these distances may be larger. This is what is called the Fréchet distance. More formally, let A and B be two parameterized curves $A(\alpha(t))$ and $B(\beta(t))$, and let their parameterizations α and β be continuous functions of the same parameter $t \in [0, 1]$, such that $\alpha(0) = \beta(0) = 0$, and $\alpha(1) = \beta(1) = 1$. The Fréchet distance is the minimum over all monotone increasing parameterizations $\alpha(t)$ and $\beta(t)$ of the maximal distance $d(A(\alpha(t)), B(\beta(t)))$, $t \in [0, 1]$.

[4] considers the computation of the Fréchet distance for the special case of polylines. Deciding whether the Fréchet distance is smaller than a given constant, can be done in time $\mathcal{O}(mn)$. Based on this result, and the 'parametric search' technique, it is derived that the computation of the Fréchet distance can be done in time $\mathcal{O}(mn \log(mn))$. Although the algorithm has low asymptotic complexity, it is not really practical. The parametric search technique used here makes use of a sorting network with very high constants in the running time. A simpler sorting algorithm leads to an asymptotic running time of $\mathcal{O}(mn(\log mn)^3)$. Still, the parametric search is not easy to implement. A simpler algorithm, which runs in time $\mathcal{O}(mn(m + n) \log(mn))$ is given in [20].

A variation of the Fréchet distance is obtained by dropping the monotonicity condition of the parameterization. The resulting Fréchet distance $d(A, B)$ is a semimetric: zero distance need not mean that the objects are the same. Another variation is to consider partial matching: finding the part of one curve to which the other has the smallest Fréchet distance.

Parameterized contours are curves where the starting point and ending point are the same. However, the starting and ending point could as well lie somewhere else on the contour, without changing the shape of the contour curve. For convex contours, the Fréchet distance is equal to the Hausdorff distance [4].

4.7 Nonlinear elastic matching distance

Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two finite sets of ordered contour points, and let f be a correspondence between all points in A and all points in B such that there are no $a_1 < a_2$, with $f(a_1) > f(a_2)$. The stretch $s(a_i, b_j)$ of $(a_i, f(a_i) = b_j)$ is 1 if either $f(a_{i-1}) = b_j$ or $f(a_i) = b_{j-1}$, or 0 otherwise. The nonlinear elastic matching distance $NEM(A, B)$ is the minimum over all correspondences f of $\sum s(a_i, b_j) + d(a_i, b_j)$, with $d(a_i, b_j)$ the difference between the tangent angles at a_i and b_j. It can be computed using dynamic programming [16]. This measure is not a metric, since it does not obey the triangle inequality.

The *relaxed nonlinear elastic matching distance* NEM_r is a variation of NEM, where the stretch $s(a_i, b_j)$ of $(a_i, f(a_i) = b_j)$ is r (rather than 1) if either $f(a_{i-1}) = b_j$ or $f(a_i) = b_{j-1}$, or 0 otherwise, where $r \geq 1$ is a chosen constant. The resulting distance is not a metric, but it does obey the relaxed triangle inequality [18].

4.8 Reflection Distance

The *reflection metric* [24] is an affine-invariant metric that is defined on finite unions of curves in the plane or surfaces in space. They are converted into real-valued functions on the plane. Then, these functions are compared using integration, resulting in a similarity measure for the corresponding patterns.

The functions are formed as follows, for each finite union of curves A. For each $x \in \mathbb{R}^2$, the *visibility star* V_A^x is defined as the union of open line segments connecting points of A that are visible from x: $V_A^x = \bigcup\{\overline{xa} \mid a \in A \text{ and } A \cap \overline{xa} = \varnothing\}$. The *reflection star* R_A^x is defined by intersecting V_A^x with its reflection in x: $R_A^x = \{x + v \in \mathbb{R}^2 \mid x - v \in V_A^x \text{ and } x + v \in V_A^x\}$, see figure 6. The

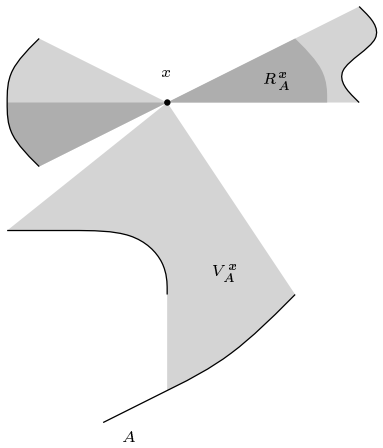

Figure 6: Reflection star at x (dark grey).

function $\rho_A : \mathbb{R}^2 \to \mathbb{R}$ is the area of the reflection star in each point: $\rho_A(x) = \text{area}(R_A^x)$. Observe that for points x outside the convex hull of A, this area is always zero. The reflection metric between patterns A and B defines a normalized difference of the corresponding functions ρ_A and ρ_B:

$$d(A, B) = \frac{\int_{\mathbb{R}^2} |\rho_A(x) - \rho_B(x)| \, dx}{\int_{\mathbb{R}^2} \max(\rho_A(x), \rho_B(x)) \, dx}.$$

From the definition follows that the reflection metric is invariant under all affine transformations. In contrast to the Fréchet distance, this metric is defined also for patterns consisting of multiple curves. In addition, the reflection metric is deformation, blur, crack, and noise occlusion robust. Computing the reflection metric by explicitly constructing the overlay of two visibility graphs results in a high time complexity, $\mathcal{O}(c(m + n))$, with c the complexity of the overlay, which is $\mathcal{O}((m + n)^4)$ [23].

4.9 Area of Symmetric Difference, Template Metric

For two compact sets A and B, the area of symmetric difference, also called template metric, is defined as $area((A - B) \cup (B - A))$. Unlike the area of overlap, this measure is a metric.

Translating convex polygons so that their centroids coincide also gives an approximate solution for the symmetric difference, which is at most 11/3 of the optimal solution under translations [3]. This also holds for a set of transformations F other than translations, if the following holds: the centroid of A, $c(A)$, is equivariant under the transformations, i.e. $c(f(A)) = f(c(A))$ for all f in F, and F is closed under composition with translation.

4.10 Transport Distance, Earth Mover's Distance

Given two weighted point patterns $A = \{(A_1, w(A_1)), \ldots, (A_m, w(A_m))\}$ and $B = \{(B_1, w(B_1)), \ldots, (B_n, w(B_n))\}$, where A_i and B_i are subsets of $\mathbb{R}^2$, with associates weights $w(A_i), w(B_i)$. The

transport distance between A and B is the minimum amount of work needed to transform A into B. (The notion of work as in physics: the amount of energy needed to displace a mass.) This is a discrete form of what is also called the Monge-Kantorovich distance. The transport distance is used for various purposes, and goes under different names. Monge first stated it in 1781 to describe the most efficient way to transport piles of soil. It is called the Augean Stable by David Mumford, after the story in Greek mythology about the moving of piles of dirt from the huge cattle stables of king Augeas by Hercules [5]. The name Earth Mover's Distance is coined by Jorge Stolfi. The transport distance has been used in heat transform problems [43], object contour matching [19], and color-based image retrieval [46].

Let f_{ij} be the flow from location A_i to B_j, F the matrix of elements f_{ij}, and d_{ij} the 'ground distance' between A_i and B_j. Then the transport distance is

$$\frac{\min_F \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}},$$

under the following conditions: $f_{ij} \geq 0$ for all i and j, $\sum_{i=1}^{m} f_{ij} \leq w(B_j)$, $\sum_{j=1}^{n} f_{ij} \leq w(A_i)$, and $\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min\{\sum_{i=1}^{m} w(A_i), \sum_{j=1}^{n} w(B_j)\}$. If the total weights of the two sets are equal, and the ground distance is metric, then the transport distance is a metric. It can be computed by linear programming in polynomial time. The often used simplex algorithm can give exponential time complexity in the worst case, but often performs linear in practice.

5 Algorithms

In the previous section, algorithms were mentioned along with the description of the measure, when the algorithm is specific for that measure. This section mentions a few algorithms that are more general.

5.1 Voting schemes

Geometric hashing [33, 59] is a method that determines if there is a transformed subset of the query point set that matches a subset of a target point set. The method first constructs a single hash table for all target point sets together. It is described here for 2D. Each point is represented as $e_0 + \kappa(e_1 - e_0) + \lambda(e_2 - e_0)$, for some fixed choice of points e_0, e_1, e_2, and the (κ, λ)-plane is quantized into a two-dimensional table, mapping each real coordinate pair (κ, λ) to an integer index pair (k, ℓ).

Let there be N target point sets B_i. For each target point set, the following is done. For each three non-collinear points e_0, e_1, e_2 from the point set, express the other points as $e_0+\kappa(e_1-e_0)+\lambda(e_2-e_0)$, and append the tuple (i, e_0, e_1, e_2) to entry (k, ℓ). If there are $\mathcal{O}(m)$ points in each target point set, the construction of the hash table is of complexity $\mathcal{O}(Nm^4)$.

Now, given a query point set A, choose three noncollinear points e'_0, e'_1, e'_2 from the point set, and express each other point as $e'_0 + \kappa(e'_1 - e'_0) + \lambda(e'_2 - e'_0)$, and tally a vote for each tuple (i, e_0, e_1, e_2) in entry (k, ℓ) of the table. The tuple (i, e_0, e_1, e_2) that receives most votes indicates the target point set T_i containing the query point set. The affine transformation that maps (e'_0, e'_1, e'_2) to the winner (e_0, e_1, e_2) is assumed to be the transformation between the two shapes. The complexity of matching a single query set of n points is $\mathcal{O}(n)$. There are several variations of this basic method, such as balancing the hashing table, or avoiding taking all possible $\mathcal{O}(n^3)$ 3-tuples.

The generalized Hough transform [8], or pose clustering [51], is also a voting scheme. Here, affine transformations are represented by six coefficients. The quantized transformation space is represented

as a six-dimensional table. Now for each triplet of points in one set, and each triplet of points from the other set, compute the transformation between the two triples, and tally a vote in the corresponding entry of the table. Again the winner entry determines the matching transformation. The complexity of matching a single query set is $\mathcal{O}(Nm^3n^3)$.

In the alignment method [27, 54], for each triplet of points from the query set, and each triplet from the target set, we compute the transformation between them. With each such transformation, all the other points from the target set are transformed. If they match with query points, the transformation receives a vote, and if the number of votes is above a chosen threshold, the transformation is assumed to be the matching transformation between the query and the target. The complexity of matching a single query set is $\mathcal{O}(Nm^4n^3)$.

Variations of these methods also work for geometric features other than points, such as segments, or points with normal vectors [10], and for other transformations than affine transformations. A comparison between geometric hashing, pose clustering, and the alignment method is made in [60]. Other voting schemes exist, for example taking a probabilistic approach [39].

5.2 Subdivision Schemes

As mentioned above, deciding whether there is a translation plus scaling that brings the partial Hausdorff distance under a given threshold is done in [30] by a progressive subdivision of transformation space. The subdivision of transformation space is generalized to a general 'geometric branch and bound' framework in [25]. Here the optimal transformation is approximated to any desired accuracy. The matching can be done with respect to any transformation group G, for example similarity (translation, rotation, and scaling), or affine transformations (translation, rotation, scaling, and shear).

The algorithm uses a lower bound $\lambda(C, A, B)$ for the similarity $d(g(A), B)$ over $g \in C \subseteq G$, where C is a set of transformations represented by a rectangular cell in parameter space. The algorithm starts with a cell C that contains all possible minima, which is inserted in a priority queue, using the lower bound $\lambda(C, A, B)$ as a key. In each iteration, the cell having a minimal associated value of λ is extracted from the priority queue. If the size of the corresponding cell is so small that it achieves the desired accuracy, some transformation in that cell is reported as a (pseudo-)minimum. Otherwise the cell is split in two, the lower bounds of its sub-cells will be evaluated, and subsequently inserted into the priority queue.

The previous algorithm is simple, and tests show that it has a typical running time of a few minutes for translation, translation plus scaling, rigid transformations and sometimes even for affine transformation. Since transformation space is split up recursively, the algorithm is also expected to work well in applications in which the minima lie in small clusters. A speed-up can be achieved by combining the progressive subdivision with alignment [38].

6 Concluding remarks

We have discussed a number of shape similarity properties. More possibly useful properties are formulated in [57]. It is a challenging research task to construct similarity measure with a chosen set of properties. We can use a number of constructions to achieve some properties, such as remapping, normalization, going from semi-metric to metric, defining semi-metrics on orbits, extension of pattern space with the empty set, vantageing, and imbedding patterns in a function space, see [57].

A difficult problem is partial matching. Applications where this plays a vital role is the registration of scanning data sets from multiple views, reverse engineering, and shape database retrieval. The

difficulty is that the distance measure must be suitable for partial matching. The dissimilarity must be small when two shapes contain similar regions, and the measure should not penalize for regions that do not match. Also, the number of local minima of the distance can be large. For example, even for rigid motions in 2D, the lower bound on the worst case number of minima of the Hausdorff distance is $\Omega(n^5)$ [47]. So, for large values of n, the time complexity is prohibitively high if all local minima should be evaluated. Finding a good approximate transformation is essential. After that, numerical optimization techniques can perhaps be used to find the optimum. This cannot be done right from the start, because such techniques get easily stuck in local minima.

Another approach to partial matching is to first decompose the shapes into smaller parts, and do the matching with the parts. For example, retrieving shapes similar to the centaur from figure 2 with partial matching, should yield both the man and the horse. If these shape are decomposed into a buste and a body, than matching is much easier. The advantages of taking apart buste and body was already recognized by Xenophon in his work Cyropaedia (370s BC) [61]:

> "Indeed, my state will be better than being grown together in one piece. [...] so what else shall I be than a centaur that can be taken apart and put together."

References

[1] O. Aichholzer, H. Alt, and G. Rote. Matching shapes with a reference point. In *International Journal of Computational Geometry and Applications*, volume 7, pages 349–363, August 1997.

[2] H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, pages 251–265, 1995.

[3] H. Alt, U. Fuchs, G. Rote, and G. Weber. Matching convex shapes with respect to the symmetric difference. In *Algorithms ESA '96, Proceedings of the 4th Annual European Symposium on Algorithms, Barcelona, Spain, September '96*, pages 320–333. LNCS 1136, Springer, 1996.

[4] H. Alt and M. Godeau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, pages 75–91, 1995.

[5] Apollodorus. *Library, 100-200 AD*. Sir James George Frazer (ed.), Harvard University Press, 1921, ISBN 0674991354, 0674991362. See also `http://www.perseus.tufts.edu/cgi-bin/ptext?lookup=Apollod.+2.5.5`.

[6] E. Arkin, P. Chew, D. Huttenlocher, K. Kedem, and J. Mitchel. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209–215, 1991.

[7] A. J. Baddeley. An error metric for binary images. In *Robust Computer Vision: Quality of Vision Algorithms, Proc. of the Int. Workshop on Robust Computer Vision, Bonn, 1992*, pages 59–78. Wichmann, 1992.

[8] D. H. Ballard. Generalized Hough transform to detect arbitrary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):111–122, 1981.

[9] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982.

[10] G. Barequet and M. Sharir. Partial surface matching by using directed footprints. In *Proceedings of the 12th Annual Symposium on Computational Geometry*, pages 409–410, 1996.

[11] P. Brass and C. Knauer. Testing the congruence of d-dimensional point sets. In *Proceedings 16th Annual ACM Symposium on Computational Geometry*, pages 310–314, 2000.

[12] C. C. Chen. Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683–686, 1993.

[13] L. P. Chew, M. T. Goodrich, D. P. Huttenlocher, K. Kedem, J. M. Kleinberg, and D. Kravets. Geometric pattern matching under Euclidean motion. *Computational Geometry, Theory and Applications*, 7:113–124, 1997.

[14] P. Chew and K. Kedem. Improvements on approximate pattern matching. In *3rd Scandinavian Workshop on Algorithm Theory*, Lecture Notes in Computer Science 621, pages 318–325. Springer, 1992.

[15] S. D. Cohen and L. J. Guibas. Partial matching of planar polylines under similarity transformations. In *Proceedings of the 8th Annual Symposium on Discrete Algorithms*, pages 777–786, 1997.
[16] G. Cortelazzo, G. A. Mian, G. Vezzi, and P. Zamperoni. Trademark shapes description by string-matching techniques. *Pattern Recognition*, 27:1005–1018, 1994.
[17] A. Efrat and A. Itai. Improvements on bottleneck matching and related problems using geometry. *Proceedings of the 12th Symposium on Computational Geometry*, pages 301–310, 1996.
[18] R. Fagin and L. Stockmeyer. Relaxing the triangle inequality in pattern matching. *Int. Journal of Computer Vision*, 28(3):219–231, 1998.
[19] D. Fry. *Shape Recognition using Metrics on the Space of Shapes*. PhD thesis, Harvard University, Department of Mathematics, 1993.
[20] M. Godau. A natural metric for curves - computing the distance for polygonal chains and approximation algorithms. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS)*, Lecture Notes in Computer Science 480, pages 127–136. Springer, 1991.
[21] S. Gold. *Matching and Learning Structural and Spatial Representations with Neural Networks*. PhD thesis, Yale University, 1995.
[22] B. Günsel and A. M. Tekalp. Shape similarity matching for query-by-example. *Pattern Recognition*, 31(7):931–944, 1998.
[23] M. Hagedoorn, M. Overmars, and R. C. Veltkamp. A new visibility partition for affine pattern matching. In *Discrete Geometry for Computer Imagery: 9th International Conference Proceedings DGCI 2000*, LNCS 1953, pages 358–370. Springer, 2000.
[24] M. Hagedoorn and R. C. Veltkamp. Metric pattern spaces. Technical Report UU-CS-1999-03, Utrecht University, 1999.
[25] M. Hagedoorn and R. C. Veltkamp. Reliable and efficient pattern matching using an affine invariant metric. *International Journal of Computer Vision*, 31(2/3):203–225, 1999.
[26] P. S. Heckbert and M. Garland. Survey of polygonal surface simplification algorithms. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1997. Multiresolution Surface Modeling Course SIGGRAPH '97.
[27] D. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proceedings of the International Conference on Computer Vision, London*, pages 102–111, 1987.
[28] D. P. Huttenlocher, K. Kedem, and J. M. Kleinberg. On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under Euclidean motion in the plane. In *Proceedings of the 8th Annuual ACM Symposium on Computational Geometry*, pages 110–120, 1992.
[29] D. P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of Voronoi surfaces and its applications. *Discrete and Computational Geometry*, 9:267–291, 1993.
[30] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machinen Intelligence*, 15:850–863, 1993.
[31] C. Jacobs, A. Finkelstein, and D. Salesin. Fast multiresolution image querying. In *Computer Graphics Proceedings SIGGRAPH*, pages 277–286, 1995.
[32] A. Khotanzad and Y. H. Hong. Invariant image recongnition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.
[33] Y. Lamdan and H. J. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *2nd Inter. Conf. on Comput. Vision*, pages 238–249, 1988.
[34] L. J. Latecki and R. Lakämper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73(3):441–454, 1999.
[35] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.
[36] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *Image Databases and Multi-Media Search, proceedings of the First International Workshop IDB-MMS'96, Amsterdam, The Netherlands*, pages 35–42, 1996.
[37] F. Moktarian and S. Abassi. Retrieval of similar shapes under affine transform. In *Visual Information and Information Systems, Proceedings VISUAL'99*, pages 566–574. LNCS 1614, Springer, 1999.
[38] D. M. Mount, N. S. Netanyahu, and J. LeMoigne. Efficient algorithms for robust point pattern matching and applications to image registration. *Pattern Recognition*, 32:17–38, 1999.
[39] C. F. Olson. Efficient pose clustering using a randomized algorithm. *International Journal of Computer Vision*, 23(2):131–147, 1997.

[40] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3d models with shape distributions. In this proceedings, 2001.

[41] Plato. *Meno, 380 BC*. W. Lamb (ed.), Plato in Twelve Volumes, vol. 3. Harvard University Press, 1967. ISBN 0674991842. See also `http://www.perseus.tufts.edu/cgi-bin/ptext?lookup=Plat.+Meno+70a`.

[42] R. J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphics Models and Image Processing*, 54(5):438–460, 1992.

[43] S. Rachev. The Monge-Kantorovich mass transference problem and its stochastical applications. *Theory of Probability and Applications*, 29:647–676, 1985.

[44] S. Ranade and A. Rosenfeld. Point pattern matching by relaxation. *Pattern Recognition*, 12:269–275, 1980.

[45] P. L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 19(5):659–666, 1997.

[46] Y. Rubner, C. Tomassi, and L. Guibas. A metric for distributions with applications to image databases. In *Proc. of the IEEE Int. Conf. on Comp. Vision, Bombay, India*, pages 59–66, 1998.

[47] W. J. Rucklidge. Lower bounds for the complexity of the graph of the Hausdorff distance as a function of transformation. *Discrete & Computational Geometry*, 16:135–153, September 1996.

[48] S. Schirra. Approximate decision algorithms for approximate congruence. *Information Processing Letters*, 43:29–34, 1992.

[49] S. Sclaroff. Deformable prototypes for encoding shape categories in image databases. *Pattern Recognition*, 30(4):627–641, 1997.

[50] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6), June 1995.

[51] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40(3):361–387, 1987.

[52] D. W. Thomsom. Morphology and mathematics. *Transactions of the Royal Society of Edinburgh*, Volume L, Part IV, No. 27:857–895, 1915.

[53] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[54] S. Ullman. *High-Level Vision*. MIT Press, 1996.

[55] S. Umeyama. Parameterized point pattern matching and its application to recognition of object families. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):136–144, 1993.

[56] P. J. van Otterloo. *A Contour-Oriented Approach to Shape Analysis*. Hemel Hampstead, Prentice Hall, 1992.

[57] R. C. Veltkamp and M. Hagedoorn. Shape similarities, properties, and constructions. In *Advances in Visual Information Systems, Proceedings of the 4th International Conference, VISUAL 2000, Lyon, France, November 2000*, LNCS 1929, pages 467–476. Springer, 2000.

[58] J. Vleugels and R. C. Veltkamp. Efficient image retrieval through vantage objects. In D. P. Huijsmans and A. W. M. Smeulders, editors, *Visual Information and Information Systems – Proceedings of the Third International Conference VISUAL'99, Amsterdam, The Netherlands, June 1999*, LNCS 1614, pages 575–584. Springer, 1999.

[59] H. Wolfson and I. Rigoutsos. Geometric hashing: an overview. *IEEE Computational Science & Engineering*, pages 10–21, October-December 1997.

[60] H. J. Wolfson. Model-based object recognition by geometric hashing. In *Proceedings of the 1st European Conference on Computer Vision*, Lecture Notes in Computer Science 427, pages 526–536. Springer, 1990.

[61] Xenophon. *Cyropaedia, 370s BC*. W. Miller (ed.), Xenophon in Seven Volumes, vol. 5,6, Harvard University Press, 1983, 1979. ISBN 0674990579, 0674990587. See also `http://www.perseus.tufts.edu/cgi-bin/ptext?lookup=Xen.+Cyrop.+4.3.1`.