Multimedia Retrieval Algorithmics

Remco C. Veltkamp

Department of Information and Computing Sciences, Utrecht University
Padualaan 14, 3584 CH, The Netherlands
Remco.Veltkamp@cs.uu.nl

Abstract. After text retrieval, the next waves in web searching and multimedia retrieval are the search for and delivery of images, music, video, and 3D scenes. Not only the perceptual and cognitive aspects, but also many of the algorithmic and performance aspects are still badly understood. One relevant issue is the design of dissimilarity measures (distance functions) that have desired properties. Another aspect is the development of algorithms that can compute or approximate these distances efficiently. Indexing data structures and search algorithms are necessary to make the search more efficient than sequential browsing through large collections. Apart from provable properties of individual algorithms, the experimental verification of the performance of a complete retrieval system is important to analyse merits and drawbacks of certain approaches, and to compare various techniques.

1 Introduction

Multimedia research has been going on since the nineteen-sixties, even if it was not called like that. A key aspect of multimedia research is interfacing: establishing a seamless interaction and communication between the user and the computer. In that respect it represents an important ingredient of current developments which are denoted by buzz phrases such as ubiquitous computing, ambient intelligence, context awareness, the disappearing computer, video at your fingertips, anything, anyone, anywhere, anytime. Multimedia retrieval is essential for coping with the problems of information overload, in production and content management, and in personalized usage. Indeed, the reason that email and web search engines have become so immensely popular are precisely that they cope with these issues with respect to text. However, if perceptually relevant multimedia methods, that guarantee performance, are not invented soon, there is no hope that similar problems are effectively solved with respect to images, music, video, and 3D models.

Since the first pictorial information systems in the early nineteen-eighties, research has come a long way in developing various methods to handle visual information by its content, as opposed to processing by keywords [1]. However, these content descriptions consist of low level color, texture, and shape features [2], and they often miss perceptual relevance. The methods for extracting and comparing these features are primarily heuristic, which, although they are

Jan van Leeuwen et al. (Eds.): SOFSEM 2007, LNCS 4362, pp. 138–154, 2007.

clever themselves, miss guaranteed properties. In contrast, an algorithmic approach is focused on provable properties, see section 2.

Looking at a particular multimedia framework as in figure 1, we see that those processes that are of an algorithmic nature are the extraction of features from the multimedia documents, the matching of the query features with the database features, the construction of the indexing data structure to speed up the searching, and the visualization of the resulting retrieved multimedia documents.

The big challenge in multimedia for the next years is the processing of information in a way that is perceptually and semantically relevant. Because of the need for personalized information access and searching, processing should be done in a manner that is guaranteed effective. Because the searching and filtering is performed on very large databases of multimedia information, it must be done with guaranteed efficiency also. The holy grail is not yet within reach. What makes this difficult is the gap between the high level semantic information and the low level features of current multimedia systems. For example, if one is looking for an image of the holy grail (such as figure 2) on the basis of image content features, one may query for a chalice shape and a star shape. However, simple edge detection yields a set of unconnected lines, not a star. Therefore, low level features will fail miserably for this purpose.

A concrete listing of research issues in multimedia is the following. Firstly, in order to arrive at semantic access, a necessary step is the identification of what is perceptually and cognitively important in the multimedia documents.

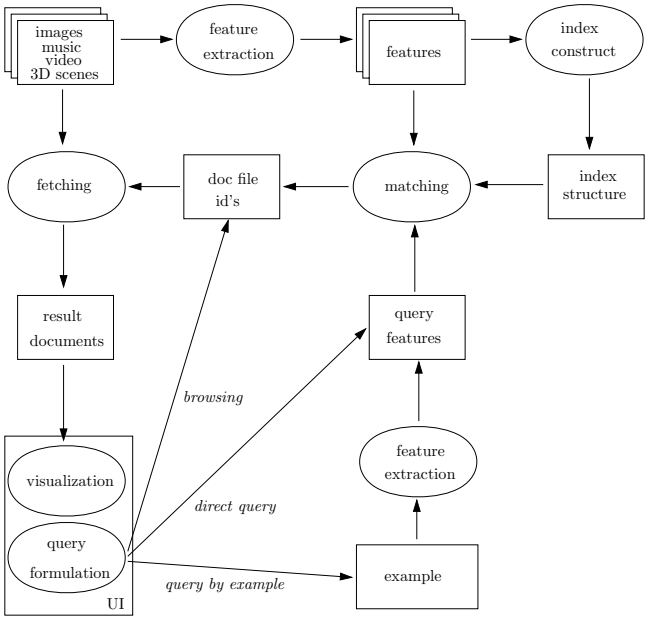


Fig. 1. Multimedia retrieval framework

Secondly, in order to cope with the data and information overload, it is becoming essential that effective and efficient searching techniques are developed. Indeed, not only company archives contain huge amounts of media. The success of mobile phone with sms (short message service) shows that as soon as consumer groups adopt devices like mobile phones with built-in digital cameras and mms (media message service) via broad band communication like GPRS or UMTS, massive amounts of images and video are produced and stored. Digital music and movies is already causing a very large amount of Internet traffic. The so-called fourth wave in multimedia (after images, video and music), consisting of 3D models and scenes, is showing more and more on the web. Together these media form an enormous amount of data, and it is essential to provide tools to match and filter, and to retrieve personalized information from it.

Fig. 2. M. L. Kirk, "And Down the Long Beam Stole the Holy Grail", 1912

Thirdly, to make retrieval feasible from such large quantities, efficient searching methods must be invented. In particular, indexing data structures and algorithms must be designed that avoid the need to scan whole collections from front to back, but instead refer the user in a few steps to the right place in the collection.

Fourthly, any successful fully-fledged system needs to provide a combination of image, video, music, and 3D model handling with text capabilities. The integrated system engineering is far from trivial, and challenging in itself.

The algorithmic aspects of these items form an area of research, *multimedia algorithmics*, which requires a combination of theoretical algorithm design and application oriented experimentation.

In the next sections we discuss the field of multimedia algorithmics, the design of dissimilarity measures, the experimental evaluation of those, and indexing over large collections to speed up the retrieval.

2 Multimedia Algorithmics

Like all computer systems, all multimedia systems are built on algorithms. They are the crucial mechanisms for working with information in any representation: computing, deriving, deciding, checking, storing, searching, learning, managing, modeling, visualizing, comparing, optimizing, transforming, sending, protecting, etc. Any system in modern information and communication technology is an algorithmic system. When it comes to the design of algorithms for multimedia, this involves for example algorithms for extracting and grouping perceptually relevant patterns, computing the similarity, indexing and searching in large collections, and visualizing retrieval results in a way that is meaningful for relevance feedback. Research issues are the invention of new algorithms that solve problems in an efficient way, guaranteeing provable properties in a rigorous way, taking an axiomatic approach, basing derivations on first principles.

Apart from fundamental modeling, design and analysis of perceptually relevant algorithms, implementations and experimentation play a crucial role to show proof-of-concepts in practice. Implementation was characteristic of early work in algorithmics. Donald Knuth, one of the most influential researchers in early computer science, insisted on implementing every algorithm he designed, and on conducting rigorous analysis of the resulting code. Since then, appreciation has faded, but since a few years, the algorithms community has shown signs of returning to implementation and testing as an integral part of algorithm development [3].

The above-stated aspects are combined into a line of research that is rooted in the discipline of fundamental algorithm design, and applied to the domain of multimedia: multimedia algorithmics. The gap between the high level semantic information and the low level features of current multimedia systems makes it difficult to make a significant step forward. A challenging research agenda for the next years is to invent algorithms for multimedia along the following orthogonal axes:

1. The tasks in a typical multimedia framework that are of an algorithmic nature: perceptual feature extraction, pattern matching, indexing, and visualization.
2. The different media (images, music, video, 3D models and scenes) to which these task are applied.
3. The desired properties of algorithms that must be invented: robustness, invariance, and efficiency, etc.

Together, these aspects span a whole research space, as illustrated in figure 3.

3 Dissimilarity Measures

This section is about one of the aspects mentioned above, matching of patterns in various types of media. In particular we will look at geometric pattern matching, or shape matching. These geometric patterns could be shapes in images, musical patterns, the shape of 3D models and scenes, etc. Matching is the process of

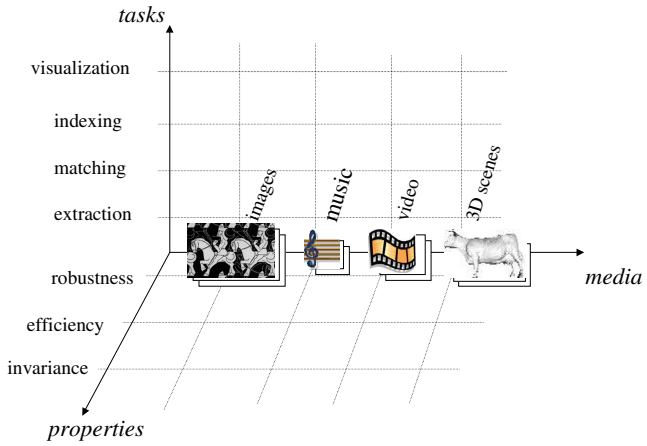


Fig. 3. Multimedia research space

computing the dissimilarity between two shapes, possibly minimized under some transformation group like translations and rotations. The design of a dissimilarity function that is suitable for a particular application, and the development of algorithms to compute that dissimilarity are important issues. We first discuss formal properties if dissimilarity measures, then we will look at the specific problem of matching polylines to a polygonal shape.

3.1 Properties

In this section we list a number of possible properties of similarity measures. Whether or not specific properties are desirable will depend on the particular application, sometimes a property will be useful, sometimes it will be undesirable. A shape dissimilarity measure, or distance function, on a collection of shapes S is a function $d : S \times S \to \mathbb{R}$. The following conditions apply to all the shapes A, B, or C in S.

1 (Nonnegativity) $d(A, B) \geq 0$.

2 (Identity) $d(A, A) = 0$ for all shapes A.

3 (Uniqueness) $d(A, B) = 0$ implies $A = B$.

4 (Strong triangle inequality) $d(A, B) + d(A, C) \geq d(B, C)$.

Nonnegativity (1) is implied by (2) and (4). A distance function satisfying (2), (3), and (4) is called a metric. If a function satisfies only (2) and (4), then it is called a semimetric. Symmetry (see below) follows from (4). A more common formulation of the triangle inequality is the following:

5 (Triangle inequality) $d(A, B) + d(B, C) \geq d(A, C)$.

Properties (2) and (5) do not imply symmetry.

Similarity measures for partial matching, giving a small distance $d(A, B)$ if a part of A matches a part of B, in general do not obey the triangle inequality. A counterexample is the following: the distance from a man to a centaur is small, the distance from a centaur to a horse is small, but the distance from a man to a horse is large, so $d(man, centaur) + d(centaur, horse) \geq d(man, horse)$ does not hold. It therefore makes sense to formulate an even weaker form:

6 (Relaxed triangle inequality) $c(d(A, B) + d(B, C)) \geq d(A, C)$, for some constant $c \geq 1$.

7 (Symmetry) $d(A, B) = d(B, A)$.

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.

8 (Invariance) 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.

The following properties are about robustness, a form of continuity. They state that a small change in the shapes lead to small changes in the dissimilarity value. For shapes defined in $\mathbb{R}^2$ we can require that an arbitrary small change in shape leads to an arbitrary small in distance, but for shapes in $\mathbb{Z}^2$ (raster images), the smallest change in distance value can be some fixed value larger than zero. We therefore speak of an 'attainable $\epsilon > 0$'.

9 (Deformation robustness) For each attainable $\epsilon > 0$, there is an open set F of homeomorphisms sufficiently close to the identity, such that $d(f(A), A) < \epsilon$ for all $f \in F$.

10 (Noise robustness) For shapes in $\mathbb{R}^2$, noise is an extra region anywhere in the plane, and robustness can be defined as: for each $x \in (\mathbb{R}^2 - A)$, and each attainable $\epsilon > 0$, an open neighborhood U of x exists such that for all B, $B - U = A - U$ implies $d(A, B) < \epsilon$. When we consider contours, we interpret noise as an extra region attached to any location on the contour, and define robustness similarly.

3.2 Multiple Polyline to Polygon Matching

There is evidence that, for the task of object recognition, the human visual system uses a part-based representation. Biederman [4], for example, suggested that objects are segmented at regions of deep concavity into an arrangement of simple geometric components. For the retrieval of polygonal shapes, we have therefore developed an algorithm to search for the best matching polygon, given one or more query parts. This dissimilarity measure models partial matching, is translation and rotation invariant, and deformation robust.

Let P_1 be a polyline, and let $P_1(s)$ be the point on P_1 at distance s along the polyline from its beginning. The *turning-angle function* Θ_1 of a polyline P_1 measures the angle of the counterclockwise tangent at $P_1(s)$ with respect to a reference orientation as a function of s. It is a piecewise constant function, with jumps corresponding to the vertices of P_1. The domain of the function is $[0, \ell_1]$, where ℓ_1 is the length of P_1. Rotating P_1 by an angle θ corresponds to shifting Θ_1 over a distance θ in the vertical direction.

The turning-angle function Θ_P of a polygon P is defined in the same way, except that the distance s is measured by going counterclockwise around the polygon from an arbitarily chosen *reference point*. Since P is a closed polyline, we can keep going around the polygon, and the domain of Θ_P can thus be extended to the entire real line, where $\Theta_P(s + \ell_P) = \Theta_P(s) + 2\pi$. Moving the location of the reference point over a distance s along the boundary of P corresponds to shifting Θ_P horizontally over a distance s.

To measure the mismatch between P_1 and the part of P starting at $P(t)$, we align $P_1(0)$ with $P(t)$ by shifting the turning-angle function of P over a distance t and computing the L_2-distance between the two turning-angle functions, minimized over all possible rotations θ (that is: vertical shiftings of the turning functions). The squared mismatch between P_1 and P, as a function of t, is thus given by:

$$d_1(t) := \min_{\theta \in \mathbb{R}} \int_0^{\ell_1} (\Theta_P(s + t) - \Theta_1(s) + \theta)^2 ds. \tag{1}$$

An ordered set of k polylines $\{P_1, P_2, \ldots, P_k\}$ can be represented by concatenating the turning-angle functions of the individual polylines. Thus we get a function $\Theta_{PL} : [0, \ell_k] \to \mathbb{R}$, where ℓ_j is the cumulative length of polylines P_1 through P_j. For $1 \leq j \leq k$ and $\ell_{j-1} \leq s \leq \ell_j$ we have $\Theta_{PL}(s) := \Theta_j(s - \ell_{j-1})$, so that each polyline P_j is represented by the section of Θ_{PL} on the domain $[\ell_{j-1}, \ell_j]$. The squared mismatch between P_j and P (shifted by t) is now given by:

$$d_j(t) := \min_{\theta \in \mathbb{R}} \int_{\ell_{j-1}}^{\ell_j} (\Theta_P(s + t) - \Theta_{PL}(s) + \theta)^2 ds. \tag{2}$$

We now express the mismatch between the set of polylines $\{P_1, P_2, \ldots, P_k\}$ and P as the square root of the sum of squared mismatches between each polyline and P, minimized over all valid shiftings:

$$\mathsf{d}(P_1, \ldots, P_k; P) := \min_{\text{valid shiftings } t_1 \ldots t_k} \left(\sum_{j=1}^{k} d_j(t_j) \right)^{1/2}. \tag{3}$$

It remains to define what the valid shiftings are. To keep the polylines disjoint (except possibly at their endpoints) and in counterclockwise order around the polygon, each polyline has to be shifted at least as far as the previous one, that is: $t_{j-1} \leq t_j$ for all $1 < j \leq k$. Furthermore, to make sure that P_k does not wrap around the polygon beyond the starting point of P_1, we have to require that $\ell_k + t_k \leq t_1 + \ell_P$ (see figure 4).

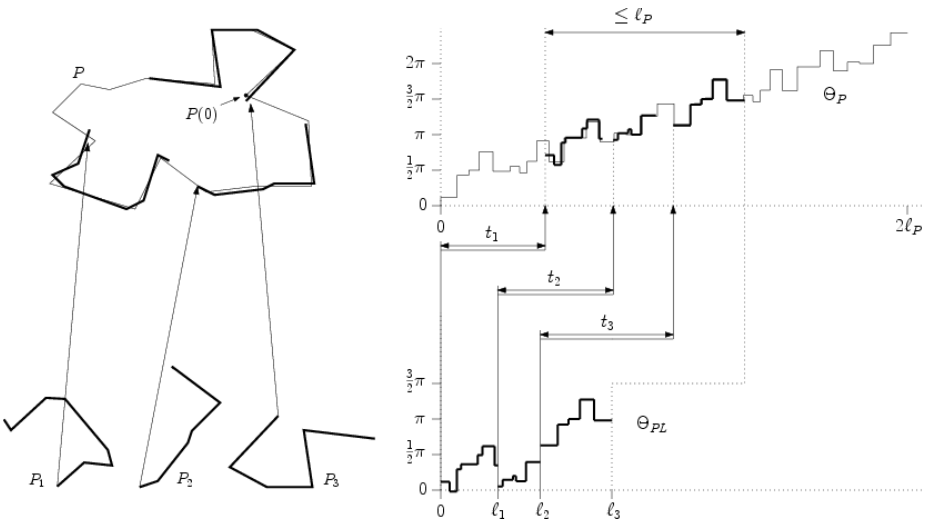


Fig. 4. To match polylines $P_1, \ldots, P_3$ to polygon P, we shift the turning functions of the polylines over the turning function of the polygon. To maintain the order of the polylines around the polygon, we need to guarantee $t_1 \leq t_2 \leq t_3$ and $\ell_3 + t_3 \leq t_1 + \ell_P$.

In [5] we show that the optimal placement and the distance value can be computed in $O(km^2n^2)$ time with a straightforward dynamic programming algorithm, and in $O(kmn \log(mn))$ time and space with a novel fast algorithm.

4 Experimental Evaluation

In order to compare different dissimilarity measures, we can look at the formal properties they have, such as listed in section 3.1. Another way is to evaluate how well they perform in practice on a specific task. One way to make such comparisons is on the basis of a chosen ground truth. The Motion Picture Expert Group (MPEG), a working group of ISO/IEC (see `http://www.chiariglione.org/mpeg/`) has defined the MPEG-7 standard for description and search of audio and visual content. The data set created by the MPEG-7 committee for evaluation of shape similarity measures [6,7] offers an excellent possibility for objective experimental comparison of the existing approaches evaluated based on the retrieval rate. The shapes were restricted to simple pre-segmented shapes defined by their outer closed contours. The goal of the MPEG-7 Core Experiment CE-Shape-1 was to evaluate the performance of 2D shape descriptors under change of a view point with respect to objects, non-rigid object motion, and noise. In addition, the descriptors should be scale and rotation invariant.

The test set consists of 70 different classes of shapes, each class containing 20 similar objects, usually (heavily) distorted versions of a single base shape. The whole data set therefore consists of 1400 shapes. For example, each row in figure 5 shows four shapes from the same class.

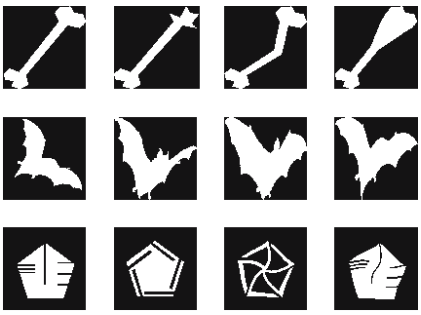


Fig. 5. Example images from the MPEG-7 Core Experiment CE-Shape-1 part B

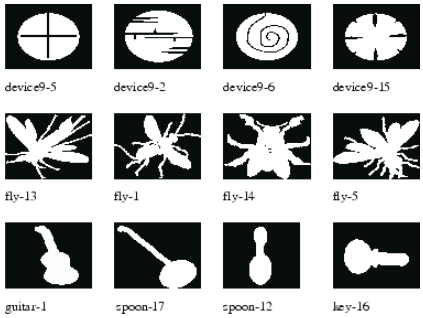


Fig. 6. Images with the same name prefix belong to the same class

We focus our attention on the performance evaluation of shape descriptors in experiments established in Part B of the MPEG-7 CE-Shape-1 data set [6]. Each image was used as a query, and the retrieval rate is expressed by the so called Bull's Eye score: the fraction of images that belong to the same class in the top 40 matches.

Strong shape variations within the same classes make that no shape similarity measure achieves a 100% retrieval rate. E.g., see the third row in figure 5 and the first and the second rows in figure 6. The third row shows spoons that are more similar to shapes in different classes than to themselves.

A region-based and a contour-based shape similarity method are part of the MPEG-7 standard. The contour-based method is the Curvature Scale Space (CSS) method [8]. This technique matches two shapes based on their CSS-image, which is constructed by iteratively convolving the contour with a Gaussian smoothing kernel, until the shape is completely convex. When at a certain iteration a curvature zero-crossing disappears due to the convolution process, a peak is created in the CSS-image. Two shapes are now matched by comparing the peaks in their CSS-images.

The multiple polyline to polygon matching algorithm of section 3.2 has been implemented in C++ and is evaluated in a part-based shape retrieval application (see `http://give-lab.cs.uu.nl/Matching/Mtam/`) with the Core Experiment CE-Shape-1 part B test set. We compared our matching to the CSS method, as well as to matching the global contours with turning angle functions (GTA) with respect to the Bulls Eye score. These experimental results indicate that for those classes with a low performance of the CSS matching, our approach consistently performs better. See figure 7 for two examples. The interactive selection of part to query with, makes a comparison on all images from the test set infeasible, but a rigorous experimental evaluation is given in [9]. The running time for a single query on the MPEG-7 test set of 1400 images is typically about one second on a 2 GHz PC.

CSS/GTA Query Image	Part-based Query Parts	Bull's Eye Score		
		CSS	GTA	MPP
 beetle-20		10	30	65
 ray-3		15	15	70

Fig. 7. A comparison of the Curvature Scale Space (CSS), the Global Turning Angle function (GTA), and our Multiple Polyline to Polygon (MPP) matching

In order to compare the performance of various similarity measures, we built the framework SIDESTEP – Shape-based Image Delivery Statistics Evaluation Project, `http://give-lab.cs.uu.nl/sidestep/`. Performance measures such as the number of true/false positives, true/false negative, specificity, precision, recall, negative predicted value, relative error, k-th tier, total performance, and Bull's Eye score can be evaluated for a single query, over a whole class, or over a whole collection, see figure 8.

In [10] we have compared many dissimilarity measures on the basis of their formal properties, as well as on their performance in terms of the Bull's Eye score on the MPEG-7 test collection. The difference between the Bull's Eye scores of these dissimilarity measures as reported in the literature, and the performances of the reimplement methods in SIDESTEP is significant. Our conjecture is that this is caused by the following. Firstly, several methods are not trivial to implement, and are inherently complex. Secondly, the description in the literature is often not sufficiently detailed to allow a straightforward implementation. Thirdly, fine tuning and engineering has a large impact on the performance for a specific data set. It would be good for the scientific community if the reported test results are made reproducible and verifiable by publishing data sets and software along with the articles.

The MPEG-7 test set provides a strict classification, which is not always available. The ground truth developed in [11] was used at the "1st Annual Music Information Retrieval Evaluation eXchange" (MIREX) 2005 for comparing various methods for measuring melodic similarity for notated music. This ground truth does not give one single correct order of matches for every query. One reason is that limited numbers of experts do not allow statistically significant differences in ranks for every single item. Also, for some alternative ways of altering a melody, human experts simply do not agree on which one changes the melody more. See figure 9 for an example. In cases like this, even increasing the number of experts might not always avoid situations where the ground truth

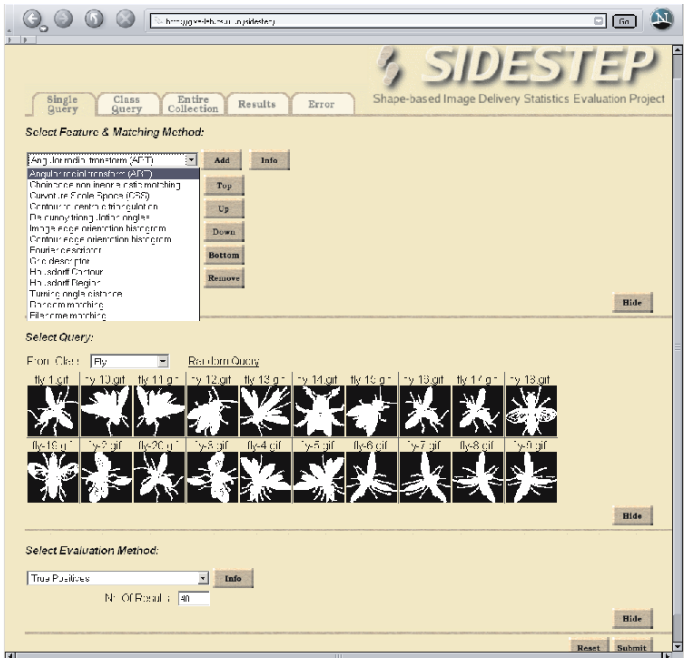


Fig. 8. SIDESTEP interface

contains only *groups* of matches whose correct order is reliably known, while the correct order of matches within the groups is not known. Here, the 31 experts we asked do not agree on whether the second or the third piece is more similar to the query. The third piece is shorter, but otherwise identical to the query, while the second one contains more musical material from the query, but two ties are missing.

In [11] we proposed a measure (called "average dynamic recall") that measures, at any point in the result list, the recall among the documents that the user should have seen so far. Unlike Kekäläinen's and Järvelin's measures [12], this measure only requires a partially ordered result list as ground truth, but no similarity scores, and it works without a binary relevance scale. It does not have any parameters that can be chosen arbitrarily, and it is easy to interpret.

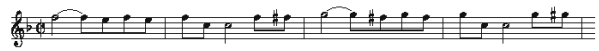
Consider a result list

$$\langle R_1, R_2, \ldots \rangle$$

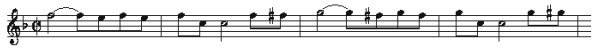
and a ground truth of g groups of items

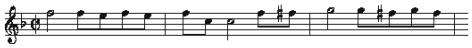
$$\langle (G_1^1, G_2^1, \ldots, G_{m_1}^1), (G_1^2, \ldots, G_{m_2}^2), \ldots, (G_1^g, \ldots, G_{m_g}^g) \rangle$$

(with m_i denoting the number of members of group i) where we know that $rank(G_j^i) < rank(G_l^k)$ if and only if $i < k$, but we do not know whether $rank(G_j^i) < rank(G_p^i)$ for any i (unless $j = p$). We propose to calculate the result quality as follows. Let $n = \sum_{i=1}^{g} m_i$ be the number of matches in the



Query: Peter von Winter (1754-1825): Domus Israel speravit, RISM A/II signature: 600.054.278

1. 
Peter von Winter: Domus Israel speravit, 600.054.278

2. 
Peter von Winter : Domus Israel speravit, 600.055.822

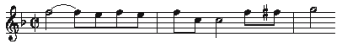
3. 
Anonymus: Offertories, 450.040.980

Fig. 9. Ground truth for Winter: "Domus Israel speravit"

ground truth and c the number of the group that contains the ith item in the ground truth ($\sum_{v=1}^{c} m_v \geq i \wedge \sum_{v=1}^{c-1} m_v < i$). Then we can define r_i, the recall after the item R_i, as:

$$r_i = \frac{\#\{R_w | w \leq i \wedge \exists j,k : j \leq c \wedge R_w = G_k^j\}}{i}.$$

The result quality is then defined as:

$$ADR = \frac{1}{n} \sum_{i=1}^{n} r_i.$$

This measure was used at the MIREX 2005 and 2006 competitions for symbolic melodic similarity, and the 3D shape retrieval contest (SHREC) 2006.

5 Indexing

Proximity searching in multimedia databases has gained more and more interest over the years. In particular searching in dissimilarity spaces (rather than extracting a feature vector for each database object) is an increasing area of research. With growing multimedia databases indexing has become a necessity.

Vantage indexing works as follows: given a multimedia database A and a distance measure $d : A \times A \rightarrow \mathbb{R}$, select from the database a set of m objects $A^* = \{A_1^*, ...A_m^*\}$, the so called vantage objects. Compute the distance from each database object A_i to each vantage object, thus creating a point $p_i = (x_1, ...x_m)$, such that $x_j = d(A_i, A_j^*)$. Each database object corresponds to a point in the m-dimensional vantage space.

A query on the database now translates to a range-search or a nearest-neighbor search in this m-dimensional vantage space: compute the distance from

the query object q to each vantage object (i.e. position q in the vantage space) and retrieve all objects within a certain range around q (in the case of a range query), or retrieve the k nearest neighbors to q (in case of a nearest neighbor query). The distance measure used on the points in vantage space is L_∞.

Vleugels and Veltkamp show [13] that as long as the triangle inequality holds for the distance measure d defined on the database objects, recall (ratio of number of relevant retrieved objects to the total number of relevant objects in the whole data base) is 100%, meaning that there are no false negatives. However, false positives are not excluded from the querying results, so precision (ratio of number of relevant retrieved objects to the total number of retrieved objects) is not necessarily 100%. We claim that by choosing the right vantage objects, precision can increase significantly.

The retrieval performance of a vantage index can improve significantly with a proper choice of vantage objects. This improvement is measured in terms of false positives, as defined below. Let δ be the distance measure in vantage space.

Definition 1. Return set *Given $\epsilon > 0$ and query A_q, object A_i is included in the return set of A_q if and only if $\delta(A_q, A_i) \leq \epsilon$.*

Definition 2. False positive *A_p is a false positive for query A_q if $\delta(A_q, A_p) \leq \epsilon$ and $d(A_q, A_p) > \epsilon$.*

We present a new technique for selecting vantage objects that is based on two criteria which address the number of false positives in the retrieval results directly. The first criterion (spacing) concerns the relevance of a single vantage object, the second criterion (correlation) deals with the redundancy of a vantage object with respect to the other vantage objects. We call this method Spacing-based Selection.

The main idea is to keep the number of objects that are returned for a query A_q and range ϵ low. Since false negatives are not possible under the condition that the triangle inequality holds for d, minimization of the number of false positives is achieved by spreading out the database along the vantage space as much as possible. False positives are, intuitively speaking, pushed out of the returned sets.

5.1 Spacing

In this section we will define a criterion for the relevance of a single vantage object V_j. A priori the query object A_q is unknown, so the distance $d(A_q, V_j)$ between a certain query A_q and vantage object V_j is unknown. The size of the range query (ϵ) is unknown beforehand as well. Optimal performance (achieved by small return sets given a query A_q and range ϵ) should therefore be scored over all possible queries and all possible ranges ϵ.

This is achieved by avoiding clusters on the vantage axis belonging to V_j. Our first criterion therefore concerns the *spacing* between objects on a single vantage axis, which is defined as follows:

Definition 3. *The spacing between two consecutive objects A_i and A_{i+1} on the vantage axis of V_j is $d(A_{i+1}, V_j) - d(A_i, V_j)$.*

Let μ be the average spacing. Then the variance of spacing is given by $\frac{1}{n-1}\sum_{i=1}^{n-1}((d(A_{i+1},V_j)-d(A_i,V_j))-\mu)^2$. To ensure that the database objects are evenly spread in vantage space, the variance of spacing has to be as small as possible. A vantage object with a small variance of spacing has a high discriminative power over the database, and is said to be a relevant vantage object.

5.2 Correlation

It is not sufficient to just select relevant vantage objects, they also should be non-redundant. A low variance of spacing does not guarantee that the database is well spread out in vantage space, since the vantage axes might be strongly correlated.

Therefore, we compute all linear correlation coefficients for all pairs of vantage objects and make sure these coefficients do not exceed a certain threshold. Experiments show that on the MPEG-7 shape images set pairwise correlation is sufficient and that higher order correlations are not an issue.

5.3 Algorithm

Spacing-based Selection selects a set of vantage objects according to the criteria defined above with a randomized incremental algorithm. The key idea is to add the database objects one by one to the index while inspecting the variance of spacing and correlation properties of the vantage objects after each object has been added. As soon as either the variance of spacing of one object or the correlation of a pair of objects exceeds a certain threshold, a vantage object is replaced by a randomly chosen new vantage object. These repair steps are typically necessary only at early stages of execution of the algorithm, thus keeping the amount of work that has to be redone small. For details, see the algorithm in figure 10.

The complexity of our algorithm is expressed in terms of distance calculations, since these are by far the most expensive part of the process. The running time complexity is then $O(\sum_{i=0}^{n} P_i \times i + (1-P_i)\times k)$ where k is the (in our case constant) number of vantage objects and P_i is the chance that, at iteration i, a vantage object has to be replaced by a new one. This chance depends on the choice for ϵ_{spac} and ϵ_{corr}. There is a clear trade-off here: the stricter these threshold values are, the better the selected vantage objects will perform but also the higher the chance a vantage object has to be replaced, resulting in a longer running time. If we only look at spacing and set ϵ_{spac} such that, for instance, P_i is $(\log n)/i$, the running time would be $O(n\log n)$ since k is a small constant (8 in our experiments).

5.4 Experimental Evaluation

We implemented our algorithm and tested it on MPEG-7 test set CE-Shape-1 part B, and the distance measure used to calculate the distance between two of these shape images is the Curvature Scale Space (CSS), discussed in section 4. To justify our criteria, we manually selected four sets of eight vantage objects that

Input: Database A with objects $A_1, ..., A_n$, $d(A, A) \rightarrow \mathbb{R}$, thresholds ϵ_{corr} and ϵ_{spac}
Output: Vantage Index with Vantage objects $V_1, V_2, ..., V_m$

```
select initial V_1, V_2, ..., V_m randomly
for All objects A_i do in random order
    for All objects V_j do
        compute d(A_i, V_j)
        add A_i to index
        if var(Spacing)(V_j) > ε_spac then
            remove V_j
            select new vantage object randomly
    if for any pair p(V_k, V_l), Corr(V_k, V_l)> ε_corr then
        remove p's worst spaced object
        select new vantage object randomly
```

Fig. 10. Spacing-based Selection

either satisfy both criteria (weakest correlation and lowest variance of spacing: *weak-low*), none (strongest correlation and highest variance of spacing: *strong-high*) or a *strong-low* or *weak-high* combination.

The performance of these four sets of vantage objects was evaluated by querying with all 1400 objects. The number of nearest neighbors that was retrieved for each query object varied from 1 to 200. The distance of the furthest nearest neighbor functioned as ϵ, which was used to calculate the number of false positives among these nearest neighbors, see Definition 2. For each vantage index, and all k-NN queries, $k = 1, ..., 200$, an average ratio of false positives in result was calculated over all 1400 queries. The results are displayed in figure 11, together with some typical runs of our algorithm, the "MaxMin" approach [13] and the "loss-based" approach [14].

These results show that both criteria need to be satisfied in order to achieve good performance (only the set called *weak-low* scores less than 50% false positives for all sizes of nearest neighbor query). Furthermore, it shows that our algorithm can actually select a set of vantage objects in which these criteria are satisfied, since false positive ratios are low for these sets. For more details, see [15].

6 Concluding Remarks

Motivated by the need for perceptually relevant multimedia algorithmics, we looked at properties of shape dissimilarity measures, showed a framework for the experimental performance evaluation of dissimilarities (SIDESTEP), and introduced a new performance measure (Average Dynamic Recall). Because in human perception the parts of objects play an important role, we developed a dissimilarity measure for multiple polyline to polygon matching, and designed an efficient algorithms to compute it. We then introduced a way to decrease the number of false positive retrievals by selecting vantage objects for indexing on the basis of an objective function that has a direct relation with the number of false positives, rather than by a heuristic.

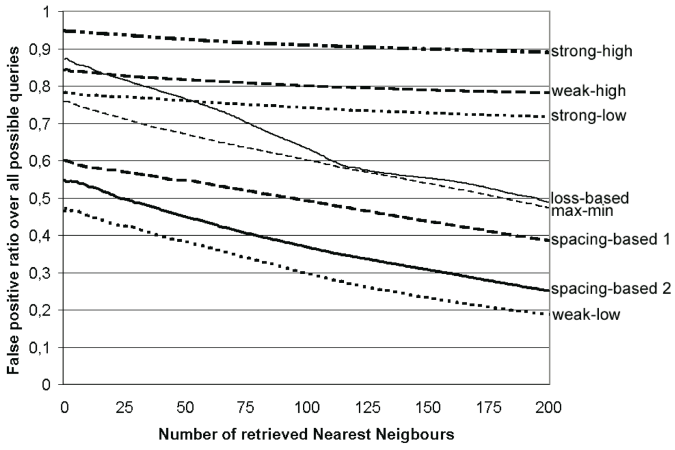


Fig. 11. MPEG-7: false positive ratios

This paper primarily shows examples in the domain of image retrieval, but we have taken a similar approach to music retrieval. As a dissimilarity measure we have designed the Proportional Transportation Distance [16], a normalized version of the Earth Mover's Distance [17]. It satisfies the triangle inequality, which makes it suitable for indexing with the vantage method. Indeed, we have used it in combination with the vantage indexing method in our music retrieval systems Muugle (`http://give-lab.cs.uu.nl/muugle`) [18] and Orpheus (`http://give-lab.cs.uu.nl/orpheus/`). The vantage indexing made it possible to identify anonymous incipits (beginnings of pieces, for example twenty notes long) from the RISM A/II collection [19] consisting of about 480,000 incipits [20]. All 80,000 anonymous incipits were compared to the remaining 400,000 ones, giving a total of 32,000,000,000 comparisons. Should a single comparison take 1 ms, this would have taken about 370 days. The vantage indexing made it possible to do this within a day on a 1 GHz PC. A total of 17,895 incipits were identified.

Acknowledgment. I want to thank all persons I have worked with on multimedia retrieval, and with whom the results reported here are obtained. In particular I thank Martijn Bosma, Panos Giannopoulos, Herman Haverkort, Reinier van Leuken, Mirela Tanase, Rainer Typke, and Frans Wiering. This research was supported by the FP6 IST projects 511572-2 PROFI and 506766 AIM@SHAPE.

References

1. Smeulders, A.W., Worring, M., Santini, S., Gupta, A., and Jain, R.: Content-Based Image Retrieval at the End of the Early Years. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** 12 (2000) 1349–1380
2. Veltkamp, R.C. and Tanase, M.: A Survey of Content-Based Image Retrieval Systems. In Marques, O., Furht, B., (eds): Content-Based Image and Video Retrieval, Kluwer (2002) 47–101

3. Moret, B.: Towards a Discipline of Experimental Algorithmics. In Goldwasser, M., Johnson, D., McGeoch, C., (eds): Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges. DIMACS Monographs 59, American Mathematical Society (2002) 197–213
4. Biederman, I.: Recognition-by-Components: A Theory of Human Image Understanding. Psychological Review **94** 2 (1987) 115–147
5. Tanase, M., Veltkamp, R.C., and Haverkort, H.: Multiple Polyline to Polygon Matching. In: Proceedings 16th Annual Symposium on Algorithms and Computation (ISAAC), LNCS **3827** (2005) 60–70
6. Bober, M., Kim, J.D., Kim, H.K., Kim, Y.S., Kim, W.Y., and Muller, K.: Summary of the Results in Shape Descriptor Core Experiment, iso/iec jtc1/sc29/wg11/mpeg99/m4869 (1999)
7. Latecki, L.J., Lakaemper, R., and Eckhardt, U.: Shape Descriptors for Non-Rigid Shapes with a Single Closed Contour. In: Proc. Conference on Computer Vision and Pattern Recognition (CVPR) (2000) 424–429
8. Mokhtarian, F., Abbasi, S., and Kittler, J.: Efficient and Robust Retrieval by Shape Content through Curvature Scale Space. In: Proceedings of IDB-MMS'96 (1996) 35–42
9. Tanase, M.: Shape Deomposition and Retrieval. PhD Thesis, Utrecht University, Department of Computer Science (2005)
10. Veltkamp, R.C. and Latecki, L.J.: Properties and Performances of Shape Similarity Measures. In: Batagelj et al. (eds.), Data Science and Classification, Proceedings of the IFCS06 Conference, Spinger (2006) 47–56
11. Typke, R., Veltkamp, R.C., and Wiering, F.: A Measure for Evaluating Retrieval Techniques Based on Partially Ordered Ground Truth Lists. In: Proceedings International Conference on Multimedia & Expo (ICME) (2006)
12. Järvelin, K. and Kekäläinen, J.: Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information Systems **20** 4 (2002) 422–446
13. Vleugels, J. and Veltkamp, R.C.: Efficient Image Retrieval through Vantage Objects. Pattern Recognition (2002) 69–80
14. Henning, C. and Latecki, L.J.: The Choice of Vantage Objects for Image Retrieval. Pattern Recognition (2003) 2187–2196
15. van Leuken, R.H., Veltkamp, R.C., and Typke, R.: Selecting Vantage Objects for Similarity Indexing. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR) (2006)
16. Giannopoulos, P. and Veltkamp, R.C.: A Pseudo-Metric for Weighted Point Sets. In: Proceedings European Conference on Computer Vision (ECCV 2002), Springer, LNCS **2352** (2002) 715–730
17. Rubner, Y.: Perceptual Metrics for Image Database Navigation. PhD Thesis, Stanford University, Department of Computer Science (1999)
18. Bosma, M., Veltkamp, R.C., and Wiering, F.: Muugle: A Music Retrieval Experimentation Framework. In: Proceedings of the 9th International Conferncen on Music Perception and Cognition (2006) 1297–1303
19. Répertoire International des Sources Musicales (RISM): Serie A/II, manuscrits musicaux après 1600. K. G. Saur Verlag, München, Germany (2002)
20. Typke, R., Giannopoulos, P., Veltkamp, R.C., Wiering, F., and van Oostrum, R.: Using Transportation Distances for Measuring Melodic Similarity. In: Proceedings of the Fourth International Conference on Music Information Retrieval (ISMIR 2003) (2003) 107–114