Finding Motifs in Database of Shapes

Xiaopeng Xi Eamonn Keogh Li Wei Agenor Mafra-Neto

Computer Science & Engineering Department
University of California - Riverside
Riverside, CA 92521
{xxi,eamonn,wli}@cs.ucr.edu

ISCA Entomological Technologies
2060 Chicago Avenue
Riverside, CA 92507
isca@iscatech.com

Abstract

The problem of efficiently finding images that are similar to a target image has attracted much attention in the image processing community and is rightly considered an information retrieval task. However, the problem of finding structure and regularities in large image datasets is an area in which data mining is beginning to make fundamental contributions. In this work, we consider the new problem of discovering shape motifs, which are approximately repeated shapes within (or between) image collections. As we shall show, shape motifs can have applications in tasks as diverse as anthropology, law enforcement, and historical manuscript mining. Brute force discovery of shape motifs could be untenably slow, especially as many domains may require an expensive rotation invariant distance measure. We introduce an algorithm that is two to three orders of magnitude faster than brute force search, and demonstrate the utility of our approach with several real world datasets from diverse domains.

Keywords

shape, motif, time series, data mining

1. Introduction

The classic *information retrieval* task of efficiently locating images that are similar to a target image (i.e. query-by-content) has attracted much attention in the image processing community in the last decade [1][20][41]. However, the problem of finding structure and regularities in large image datasets is an area in which data mining is only just beginning to make contributions [30]. In this work, we consider a new image mining problem, the task of discovering approximately repeated shapes within an image/shape database. We call such repeated shapes *image motifs*.

To enhance the reader's intuition of image motifs, we begin with a simple concrete motivating example. Figure 1 shows a subset of a collection of petroglyphs.

Figure 1: Five abstract petroglyphs from southwestern United States (the images have been filtered to enhance contrast)

Petroglyphs are images that are carved or abraded into stone. The outer patina covered surface of the parent rock is removed to expose the usually lighter stone underneath. It has been estimated that there may be several million petroglyphs in North America alone [31][38]. These artifacts are a potential goldmine for anthropologists studying the spatiotemporal spread of cultures and peoples. While there has been an increasing effort to digitally document this valuable cultural resource, the sheer volume of data involved is a bottleneck to researchers. An important first step in exploring these massive image collections is to find repeated images or "motifs". Some petroglyphs motifs, such as images of bighorn sheep, are well known. However much less is known about the bewildering assortment of abstract images that abound. We have built a tool (explained in detail below) to allow rapid discovery of potential motifs in any collection of images. We applied this tool to a collection of 1,800 petroglyphs images, which includes the five images in Figure 1. The most promising motif is shown in Figure 2.

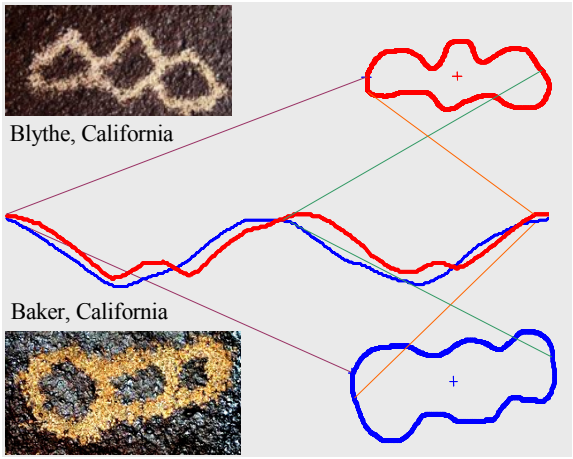

Figure 2: Two of the petroglyphs shown in Figure 1. To make the similarities of the two shapes clear, one is rotated and both shapes are mapped to one-dimensional representations. *Top*) From 15 miles west of Blythe, California. *Bottom*) From Cinder Cone Volcanic field, located 15 miles east of Baker, California

Remarkably, the dataset contains two examples of a shape consisting of three overlapping rings. While none of the anthropologists we showed this finding to could explain this (several tentatively suggested astronomical

significance[1]), they considered the finding interesting and novel.

While this simple example introduces and motivates the idea of image motifs, it also hints at the difficultly in finding them. The naïve brute force algorithm to find the closest matching pairs requires an all-to-all comparison of everything in the database. Furthermore, if, as in this case, we need to discover motifs with invariance to rotation, each comparison will require an expensive calculation, because most rotation invariant distance measures are quadratic. Many researchers have already noted (in the context of query-by-content) "*rotation is always something hard to handle compared with translation and scaling*" [24].

Most attempts to handle the rotation alignment problem work by aligning all the shapes to some cardinal orientation, typically the major axis. This approach may be useful for the limited domains in which there is a well-defined major axis, perhaps the indexing of long bones. However there is increasing recognition that the "…*major axis is sensitive to noise and unreliable*" [41]. For example a recent paper shows that under some circumstances, a single extra pixel can change the rotation by ± 90 degrees [43].

In this work, we introduce a linear time, rotation invariant algorithm to discover image motifs. While our algorithm is approximate, we will show with comprehensive experiments that it can find motifs with very high precision. Our approach works for most popular shape representations, for example, one-dimensional transforms of the original two-dimensional representations [1][3][5][12][32][41]. We will demonstrate the utility of image motifs in tasks as diverse as anthropology, crime prevention, and historical manuscript mining.

The rest of paper is organized as follows. In Section 2, we review related work and discuss some background material. In Section 3 and Section 4, we first give a generic framework for image motif discovery, and then introduce our techniques to speed up the search. Section 5 sees an extensive empirical evaluation. Finally Section 6 offers some conclusions and suggestions for future work.

2. Background and Related Work

2.1 Notation

Recall that in Figure 2 we emphasized the similarity between two shapes by comparing their one-dimensional representations. This is more than a visualization trick; this representation is at the heart of our approach. We first convert images into pseudo "time series" by measuring the distance from the centroid to all points on the shape boundary. Figure 3 offers a visual explanation.

[1] This is not as implausible as it first seems; just before this paper was submitted, astronomer John Barentine presented strong evidence that a petroglyph in Arizona records a supernova that occurred in 1006 AD.

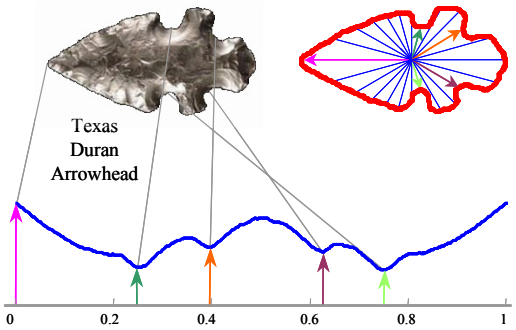

Figure 3: A visual explanation of how to convert a two-dimensional shape to a one-dimensional pseudo "time series"

Note that this 1-D representation of shape is only one of many proposed in the literature, however it does have the advantage of being simple and completely parameter free. Note that each 1-D representation is Z-normalized, removing the effects of scale or offset within the image (rotation invariance is considered below). At first glance, it may appear that this representation is too simple to really capture the true essence of a shape. However, a recent paper [20] compared this representation to state-of-the-art "sophisticated" representations on six diverse classification problems and found that it is at least as accurate, in spite (or perhaps, *because*) of its simplicity.

For brevity and simplicity we will refer to "time series" from now on, however the reader is aware that this representation can always be mapped back to the original shape. For concreteness, we begin with the definition of *time series*.

Definition 1. *Time Series*: A time series $T = (t_1,t_2,\ldots,t_n)$ is an ordered set of n real-valued variables. In our case the ordering is not temporal but spatial; it is defined by a clockwise sweep of the shape boundary.

Recall that we want to find approximately repeated images in an image database, which we formally define as *image matches*.

Definition 2. *Image Match*: Given two image time series T_1, T_2, and a threshold $\xi > 0$, if $D(T_1, T_2) < \xi$, then T_1 is a *match* of T_2.

Note that the distance between T_1 and T_2 can be measured by any of the common distance measures for time series, including Euclidean distance, Longest Common Subsequence, Dynamic Time Warping, etc. We will specify the distance function $D()$ in Section 4.

In some domains we may wish to exclude the possibility of certain items being matched together. For example, as illustrated in Figure 4, adjacent image frames in a video clip are usually very similar and are not interesting to us. We call such matches *trivial matches*.

Definition 3. *Image Trivial Match*: Given two adjacent image frames T_i, T_{i+1}, and a threshold $\varepsilon > 0$ ($\varepsilon < \xi$), if $D(T_i, T_{i+1}) < \varepsilon$, T_i trivial matches with T_{i+1}.

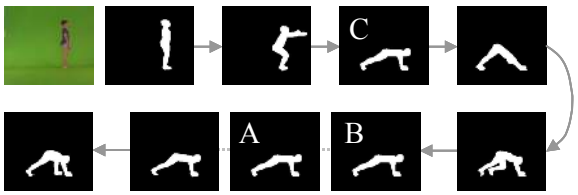

Figure 4: An illustration of a *trivial match*. The similarity between shapes **A** and **C** is interesting, because it suggests that the actor returned to a particular pose after a few minutes. In contrast, the similarity between shapes **A** and **B** is simply a result of the fact that they are adjacent frames

We are finally in a position to formally define *motifs* within an image dataset.

Definition 4. *Inner-class K-Motifs*: Given an image dataset $\Omega = \{T_i\}$, $i = 1 \ldots N$, and a threshold ξ, the most significant image motif in Ω (called *1st-Motif*) is the image T_j that has the highest count of non-trivial matches. The K^{th} most significant motif in Ω (called thereafter *K^{th}-Motif*) is the image T_k with the k^{th} highest count of non-trivial matches.

There is a simple generalization of this definition that can be very useful in some domains. Given two image datasets we may be interested in discovering if there are any shapes that occur in *both* datasets. Such an operation resembles a *join* over two image databases. Concrete examples of how this might be useful include:

- **Anthropology**: Given a set of petroglyphs (or arrowheads) from two regions or time periods, we may wish to find all examples that occur in both datasets. Such images may hint at cultural transfer [14] (cf. Figure 5).
- **Palaeography** (Study of old texts): Given a collection of shapes from an old manuscript and a set of modern images from the same domain, link all matching images. This linking can help annotate and give context to the older document (cf. Figure 11 and Figure 12).
- **Zoology:** Given a collection of shapes from two distinct taxonomic groups (i.e. Class, Order, Family, Genus etc), link all matching shapes. This linking may help identify organisms that look similar because of convergent evolution or mimicry (cf. Figure 10).
- **Law Enforcement:** Graffiti, which may be seen as an unwelcome successor to the petroglyphs discussed above, is the major source of intelligent for many law enforcement agencies [21]. An occurrence of a "tag" repeated in two distant locations may signal an attempt by a gang to take over a new territory [13].

We formalize these ideas with the definition of *inter-class motifs.*

Definition 5. *Inter-class K-Motifs*: Given two image datasets $\Omega = \{T_i\}$, $\Psi = \{T_j\}$, and a threshold ξ, the most significant image motif (called *1st-Motif*) is the image pair (T_p,T_q), $T_p \in \Omega$, $T_q \in \Psi$, which is the *image match* between these two image datasets with the shortest distance $D(T_p, T_q)$. The K^{th} most significant motif (called thereafter *K^{th}-Motif*) is the image pair (T_i,T_j), $T_i \in \Omega$, $T_j \in \Psi$, having the k^{th} shortest distance in all *image matches*.

2.2 Related Work

To the best of our knowledge, the discovery of image motifs is a new problem. However, in order to frame our contribution in its proper context, we will briefly consider related work and discuss their differences to our work.

It is important to recognize that image motif discovery is very different to the superficially similar sounding *replicate image* [7] or *near-duplicate image detection* [16] problems. In these research efforts, the problem is to detect copied images that are slightly altered by some transformations, e.g., changing exposure, contrast, color, saturation, cropping, or scaling. The typical application is detection of copyright violation or forged images [4][11][42].

These works usually first extract signatures invariant to transformation from images, then find replicates by comparing signatures. This body of work does not offer a solution to the task at hand, as we are interested in image motifs which ignore color and texture information, and consider *only* shape. For example in Figure 5, we are interested in automatically annotating centuries old documents [9] and finding evidence of cultural transfer between two locations. In both cases only *shapes* contain relevant information, colors and textures are not only irrelevant, but positively misleading.

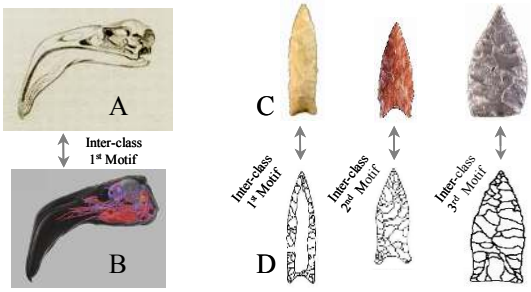

Figure 5: A visual explanation of why existing "near duplicate image detection" algorithms cannot be used for the task at hand. **A**) An 1839 lithograph by Cuvier of a flamingo's skull [9]. **B**) A 2006 X-ray CT scan of a flamingo's skull (lateral slice). **C**) A collection of arrowheads found in Texas. **D**) An anthropologist's field sketch of some arrowheads in southwestern United States

Image motif discovery must be robust to many distortions, especially *rotation*, which is generally agreed to be difficult to handle. A large number of papers achieve fast rotation invariant matching by extracting only rotation invariant features and indexing them with a feature vector [5]. This feature vector is often called the shapes "signature". There are literally dozens of rotation invariant features, including ratio of perimeter to area, fractal measures, elongatedness, circularity, min/max/mean curvature, entropy, perimeter of convex hull etc. In addition, many researchers have attempted to frame the shape-matching problem as a more familiar histogram-matching problem. For example in [29] the authors built a

histogram containing the distances between two randomly chosen points on the perimeter of the shapes in question. The approach seems to be attractive, for example it can trivially also handle 3D shapes. However it suffers from extremely poor precision. For example, it cannot differentiate between the shapes of the lowercase letters "**d**" and "**b**", or "**p**" and "**q**", since these pairs of shapes have identical histograms. In general, all these methods suffer from very poor discrimination ability [5]. Our experience with these methods suggests that they can be useful for making quick coarse discriminations, for example differentiating between skulls and arrowheads. However they could not make the fine distinctions to meaningfully match similar shapes of one class, for example arrowheads.

There are a handful of papers that recognize that the above attempts at approximating rotation invariance are unsatisfactory for most domains/applications, and they achieve true rotation invariance by exhaustive brute force search, testing all possible rotations. This robustness comes at the expense of computational efficiency [1][2][3][12]. For example, paper [1] also matches shapes in the time series domain. While they note that most invariances are trivial to handle in this representation, they state "*rotation invariance can (only) be obtained by checking all possible circular shifts for the optimal diagonal path*." Similarly paper [37] notes "*in order to find the best matching result, we have to shift one curve* n *times, where* n *is the number of possible start points*.". Our application potentially suffers even more from the high computational complexity of true rotation invariant matching, because brute force motif discovery would require $O(|\Omega|^2)$ calls to the expensive rotation invariant comparison. As we shall see, our image motif discovery *does* use this brute force rotation alignment, but we are able to achieve enormous speedup by avoiding a large fraction of the expensive comparisons.

3. A Review of SAX

To avoid the high computational cost, our solution uses the idea of *hashing* to quickly locate potential motifs. However raw time series cannot be meaningfully hashed, because it is real-valued and high dimensional data. Thus the first step of our approach is to convert time series to symbolic representations. While there are at least 200 different symbolic representations of time series in the literature, the SAX (Symbolic Aggregate approXimation) representation is unique in that it supports both dimensionality reduction and lower bounding for Euclidean distance. In recent years, SAX has been widely used in anomaly detection [19], visualization [23][26], time series repeated pattern discovery [8][34], feature extraction [22], and many other data mining applications. In this section, we will briefly review the SAX representation, which is at the heart of our solution to the image motif discovery problem.

3.1 SAX Notation

A time series T of length n can be represented in a w-dimensional space by a vector $\overline{T} = \bar{t}_1,\ldots,\bar{t}_w$. The i^{th} element of $\overline{T}$ is calculated by the following equation:

$$\bar{t}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} t_j \qquad (1)$$

In other words, the time series is divided into w equal sized segments and the dimensionality of time series is decreased from n to w. The mean value of each segment is calculated and a vector of these values becomes the dimensionality-reduced representation. This simple representation, known as Piecewise Aggregate Approximation (PAA) [17], has been shown to rival more sophisticated dimensionality reduction techniques like Fourier transforms and wavelets [6] for the task of indexing and compressing time series [18].

Having transformed a time series into the PAA representation, we apply a further transformation to obtain a discrete representation. It is desirable to have a discretization technique that will produce symbols with equiprobability [8][19]. After performing extensive experiments on more than 100 datasets, we discovered that normalized time series have highly Gaussian distribution [25]. Based on this observation, we can simply determine the "breakpoints" that will produce equal-sized areas under a Gaussian curve.

Definition 6. *Breakpoints*: breakpoints are a sorted list of numbers $B = \beta_1,\ldots,\beta_{a-1}$ such that the area under a $N(0,1)$ Gaussian curve from β_i to $\beta_{i+1} = 1/a$ (β_0 and β_a are defined as $-\infty$ and ∞, respectively, a is the size of the alphabet).

These breakpoints may be determined by looking them up in a statistical table. For example Table 1 gives the breakpoints for values of a from 3 to 6.

Table 1: A lookup table that contains the breakpoints that divide a Gaussian distribution into an arbitrary number (from 3 to 6) of equiprobable regions

β_i \ a	3	4	5	6
β_1	-0.43	-0.67	-0.84	-0.97
β_2	0.43	0	-0.25	-0.43
β_3		0.67	0.25	0
β_4			0.84	0.43
β_5				0.97

It is important to note that the assumption of Gaussian distribution is not critical to our work, and deviations from this distribution will only affect the efficiency of our algorithms, not their correctness.

Once the breakpoints have been obtained we can discretize a time series in the following manner. We first obtain a PAA of the time series. All PAA coefficients that are below the smallest breakpoint are mapped to the symbol "**a**", all coefficients greater than or equal to the smallest

breakpoint and less than the second smallest breakpoint are mapped to symbol "**b**", etc. Figure 6 shows the idea.

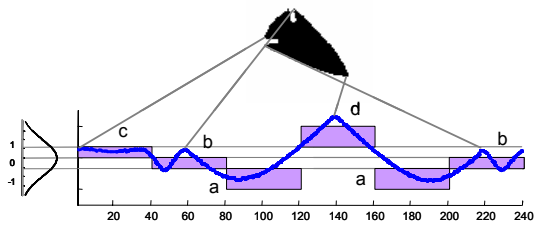

Figure 6: A time series (heavy blue line) is discretized by first obtaining a PAA approximation (shaded region) and then mapped to symbols (bold letters) using predetermined breakpoints. In this example, with n = 240, w = 6, and a = 4, the time series is mapped to the word **cbadab**

Note that in this example the four symbols, "**a**", "**b**", "**c**", and "d" are approximately equiprobable as we desired. We call the concatenation of symbols a *word.*

Definition 7. *Word*: A time series T of length n can be represented as a *word* $\hat{T} = \hat{t}_1,\ldots,\hat{t}_w$ as follows. Let α_i denotes the i^{th} element of the alphabet, i.e., α_1 = **a** and α_2 = **b**. Then the mapping from a PAA approximation $\bar{T}$ to a word $\hat{T}$ is obtained as follows:

$$\hat{t}_i = \alpha_j \quad \text{iff} \quad \beta_{j-1} \le \bar{t}_i < \beta_j \tag{2}$$

We have now completely defined our symbolic representation, then simply need to define an appropriated distance measure on it. By far the most common distance measure for time series is the Euclidean distance [18]. Given two time series T_1 and T_2 of the same length n, Eq. 3 defines their Euclidean distance.

$$ED(T_1, T_2) = \sqrt{\sum_{i=1}^{n} (t_{1i} - t_{2i})^2} \tag{3}$$

If we further transform the time series into the symbolic representation, we can define a MINDIST function that returns the lower bounding distance between the original time series of two words:

$$MINDIST(\hat{T}_1, \hat{T}_2) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} (dist(\hat{t}_{1i}, \hat{t}_{2i}))^2} \tag{4}$$

The function resembles Eq. 3 except for the multiplication by the square root of the compression rate, and the fact that the distance between individual points has been replaced by the sub-function *dist*(). The *dist*() function can be implemented using a table lookup as shown in Table 2.

Table 2: A lookup table used by the MINDIST function. This table is for an alphabet of size 4. The distance between two symbols can be read off by examining the corresponding row and column. For example *dist*(**a**,**b**) = 0 and *dist*(**a**,**c**) = 0.67

	a	**b**	**c**	**d**
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

The value in cell (r,c) for any lookup table can be calculated by the following expression.

$$cell_{r,c} = \begin{cases} 0, & if\ |r\text{-}c| \le 1 \\ \beta_{\max(r,c)-1} - \beta_{\min(r,c)}, & otherwise \end{cases} \tag{5}$$

For a given alphabet size a, the table needs only be calculated once, then stored for fast lookup.

4. Image Motif Discovery

Although SAX has proven to be a very effective method in finding motif subsequence from long time series [8][26][34], none of this work applies to the task of image matching/querying, given that it is hard to handle rotation invariance. In this section, we first show how to adapt SAX to handle shape matching with arbitrary rotations, and then show how to apply it in motif discovery problem.

Recall that the distance measure in *definition* 2 can be any common distance measures for time series. We use Euclidean distance in this work. If the shapes in question are rotationally aligned, Euclidean distance will reflect the intuitive similarity. However if the shapes are not rotationally aligned, the corresponding time series will also be misaligned. In this case, Euclidean distance can produce extremely poor results. To overcome this problem, we need the distance function to be rotation invariant. To achieve this, we need to hold one shape fixed, rotate the other, and record the minimum distance of all possible rotations. We accomplish this in the time series space by representing all rotations of a shape in a *rotation matrix*.

Definition 8. *Rotation Matrix*: Given a time series T of length n, all its possible rotations (i.e. circular shifts) constitute a rotation matrix RT of size n by n.

$$RT = \begin{pmatrix} t_1, t_2, \ldots, t_{n-1}, t_n \\ t_2, \ldots, t_{n-1}, t_n, t_1 \\ . \\ . \\ t_n, t_1, t_2, \ldots, t_{n-1} \end{pmatrix} \tag{6}$$

Each row of the matrix is simply a time series shifted (rotated) by one from its neighbors. For notational convenience, we denote the i^{th} row as T^i, which allows us to denote the rotation matrix in the more compact form of $RT = \{T^1, T^2, \ldots, T^n\}$.

Note that we do not need to actually build the full matrix if space is premium, however doing this simplifies the notation and allows some optimizations [20].

As we have already seen in Figure 1 and Figure 2 (and as we shall see again in Figure 11 and Figure 12), we cannot generally expect images be perfectly aligned. We therefore define the *Rotation invariant Euclidean Distance* between two time series.

Definition 9. *Rotation invariant Euclidean Distance*: Given two time series T_1 and T_2 of length n, the rotation invariant Euclidean distance between them is defined as

$$RED(T_1, T_2) = \min_{1 \le j \le n} ED(T_1, T_2^j) \tag{7}$$

The rotation invariant Euclidean distance provides an intuitive measure of the distance between two shapes, at

the expense of efficiency. The time complexity to compare two time series of length n is $O(n^2)$. Note that this rotation invariant Euclidean distance is denoted as "$D(T_k, T_i)$" in *definition* 4 and 5.

4.1 Min-error SAX

As illustrated in Figure 6 we can convert any time series into a SAX word. The conversion of time series into SAX is at the heart of dozens of research efforts [8][19][22][26][34] and a well-understood process. However in the special case that the time series comes from a shape, we are offered a unique chance to improve the quality of approximation with no space overhead. Recall that, as illustrated in Figure 3, we convert shapes into time series with a simple "unwinding" process. Note that the starting point for this process is completely arbitrary. This observation allows an optimization, because it may happen that some of the arbitrary starting points will lead to better SAX approximations.

For example, assume we have two arrow images A and B, where B is simply A being rotated by 15 degrees. Their time series and corresponding SAX representations are shown in Figure 7.

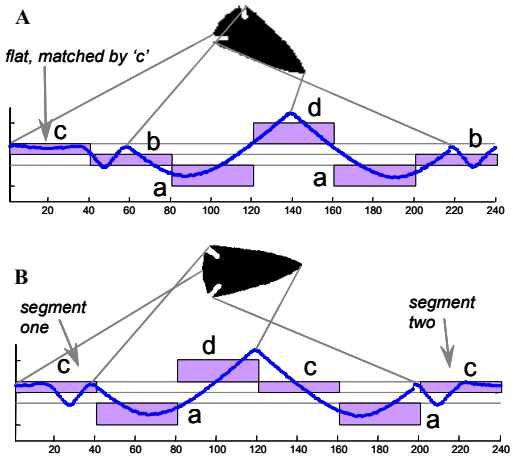

Figure 7: An arrowhead image with different rotations. *Top*) The first SAX symbol **c** approximates the first 40 data points perfectly. *Bottom*) However the same plateau time series is divided into two parts (the first and last segments of time series)

At the first sight, they look similar. But note that for in the top version of the arrowhead, the first symbol **c** matches perfectly with a plateau in the time series, while in the bottom version of the arrowhead, this plateau segment spreads across two segments (the first and the last segments). Intuitively, we may expect that the SAX word **cbadab** gives better approximation than bottom one **cadcac**. In fact, this is the case; the reconstruction errors are 106.35 and 144.65 respectively (see Appendix A for definition of SAX reconstruction error). Based on this observation, every time we convert a shape time series into a SAX word, we test all possible circular shifts of the time series and choose the one that has the smallest reconstruction error. We apply this optimization throughout the paper.

4.2 Random Projection Motif Discovery

The image motif discovery problem lands itself to a simple brute force solution. We simply need to compare each shape in Ω to every other shape using rotation invariant Euclidean distance, and record all those shapes that are within threshold ξ of each other. This can be trivially achieved with a pair of nested loops. The problem with this solution is its high time complexity $O(|\Omega|^2 n^2)$, which is clearly intractable for large datasets. Note that $O(n^2)$ is the time for a single rotation invariant comparison. There are some optimizations for rotation invariant comparison to reduce its complexity close to linear for most datasets [20]. It is the quadratic dependence on $|\Omega|$ that makes the brute force algorithm untenable for larger datasets.

We propose a motif discovery algorithm which reduces the number of rotation invariant comparisons as much as possible. The intuition of our solution is that two similar shapes are likely to have similar SAX representations (for the moment ignores the problem of rotation invariance). Actually this observation is at the heart of dozens of research efforts [8][22][25][26].

Our algorithm takes advantage of techniques that can efficiently find approximately repeated patterns in *discrete strings* [36]. The work of Tompa and Buhler and follow-up work by many researchers show that approximately repeated patterns can be found by hashing randomly "masked" versions of the strings in question. Information about which strings collide with others can then be used to prune the search space. Here "masked" simply means that one or more positions in the strings are ignored during the hashing process. The idea is that two words might be similar, but differ in just a few locations, as in a***b***ca and a***a***ca. By randomly masking and therefore ignoring some positions, the algorithm has a chance to ignore the "misspelled" position and discover the similarities. A surprising fact is that only a small constant number of iterations of masking and hashing are needed to find all motifs with high probability [36].

This solution, known as *random projection*, requires two modifications before we can apply it to image motif discovery. First, we need to do some modification to make it be able to find rotation invariant similarities between time series or circular shifts of SAX words. Second, unlike the usage of random projection on DNA strings, we are not finished after discovering motifs in SAX words. We must check the raw time series pointed by the SAX words to make sure they are true motifs.

As the first modification, for each SAX word $\hat{T}$ corresponding to an image, we add every possible circular shift of it to the list of words to be hashed. We call this list the rotation matrix RT. So that if two images T_i and T_j are similar, but are rotated differently, they may still be similar under some circular shifts. For example in Figure 8, the i^{th} shape in the arrowheads datasets maps to the SAX word $\hat{T}$ = **bacb**, so we add **bacb**, **acbb**, **cbba**, and **bbac** to the rotation matrix.

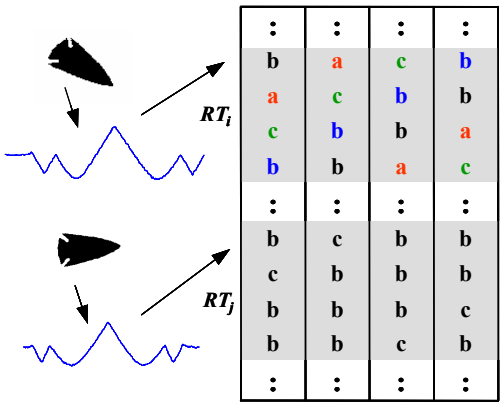

Figure 8: An example of representing an image with rotation variant SAX words. Because the length of SAX string is 4, every image time series has four possible rotations

The redundancy of having all possible circular shifts may appear to hurt the space complexity, but recall that a SAX word only requires $w\lceil a \rceil$ bits. With all possible circular shifts this becomes $w^2\lceil a \rceil$ bits per original shape. This is still much smaller than the raw time series, and completely inconsequent compared to the raw images.

After getting all possible circular shifted SAX words for each image time series, we start random projection. As in [36], several randomly chosen columns are masked off, and the rest columns are hashed into the buckets. At the same time, a collision matrix is maintained to keep record of collisions. Because similar shapes have high possibility to be hashed to the same bucket, after many times of random projections, these similar pairs will have larger values in collision matrix. Figure 9 illustrates the random projection process.

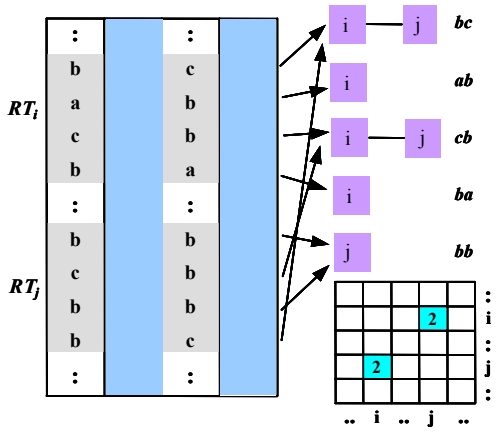

Figure 9: Random projection performed on SAX words. The mask size is 2 and the (randomly chosen) mask is {2, 4}. Columns 2 and 4 are masked off and the substrings at column {1, 3} are hashed to buckets. The value in collision matrix at the bottom right records the number of collisions between arrowheads T_i and T_j after one projection

In order to give the algorithm a high probability to ignore "misspelled" positions, we need to perform several iterations of random projections. A natural question is *when to stop*. The simplest stopping criterion is user interruption. We can treat the random projection process as an anytime algorithm, letting user interrupt the execution at any time and retrieve the best-so-far result. Another stopping criterion will be keeping random projection until collision matrix requires more than linear space. In this case, the number of iterations can be O(|Ω|). But in practice generally it is significantly smaller than O(|Ω|). According to our experiments, 20 to 100 iterations are enough to catch similar images. So we hardcoded number of iterations to 30 for experiments in this paper.

During the random projection, we change mask size dynamically. Initially mask size is set to zero, which means at the beginning all SAX words are compared in full length. Then in each iteration, the mask size increases by 1. The iteration repeats until the user issues an interruption or the predefined number of iterations is reached. After projection, if some cells have values that are significantly larger than the average in collision matrix, we treat them as motif candidates. We then calculate the rotation invariant Euclidean distance between the original time series of these candidates. Thanks to the lower bounding property of SAX representation, the last step can be conducted very efficiently. If $MINDIST(\hat{t}_i, \hat{t}_j) \leq \xi$, we only need to check $\hat{t}_k$ if and only if $MINDIST(\hat{t}_i, \hat{t}_k) \leq \xi$ or $MINDIST(\hat{t}_j, \hat{t}_k) \leq \xi$. We will show in Section 5 that our algorithm is very effective in catching image motif candidates during projection step and locating true motifs by examining these candidates. Note that we only consider the image pairs that have the largest collision value as candidates. As we will show in Section 5, the number of these ties is less than 0.1% of total number of pairs $|\Omega|^2$, and it is enough to give us high precision of true motifs. Table 3 outlines our motif discovery algorithm, where Ω is the image dataset, K is the number of motifs to be mined, and ξ is distance threshold for image motifs.

Table 3: Motif Discovery Algorithm

```
1   Function {K-motifs} = Motif-discover(Ω, K, ξ, i )
2   {T̂_i} = SAX(Ω) ;          //convert image time series to SAX
3   generate RT (T̂_i) ;   // rotation invariance matrix in fig. 8
4   K-moifs = Ø ;
5   iteration = 0;
6   M = zeros;       // initialize collision matrix as zero matrix
7   while iteration  ≤ i  and user_not_interrupt
8      Random_Projection(RT) ;
9      Update(M) ;                  // update collision matrix M
10     iteration = iteration + 1;
11  end;
12  Sort(M) ;
13  k = 0;
14  for each (p,q) in M that has the largest value
15     if {p,q} ∩ {k-motif} != Ø and RED(T_p, T_q) < ξ
16        add p, q to {k-motif}
17     else if k < K
18        k = k + 1;
19        add p, q to {k-motif}
20     end;
21  end;
```

4.3 Time and Space Complexity

Motif discovery is generally computationally expensive, which in worst case needs $O(N^2)$ time, where N is the size of dataset. In this subsection, we will show that our motif discovery algorithm requires only linear space and time.

We first look at space complexity. Assume we have N image time series of length n, with corresponding SAX words of length m. As illustrated in Figure 8, the rotation matrix RT has $m*N$ rows and m columns. Note that although the length of time series varies from one hundred to several thousands, its SAX word length is much shorter, usually from 10 to 100 based on our experiments. Furthermore, each SAX word only needs $m*log_2\boldsymbol{a}$ bits ($\boldsymbol{a}$ is the alphabet size, usually from 3 to 5), so the actual size of RT is in linear space, and much less than original size of dataset. In addition, collision matrix M is implemented as sparse matrix, which takes up much smaller size compared to full matrix. Although in the worst situation, the matrix will be filled with $i*|RT|$ non-zero values (i is the number of iterations), from our experiments and also as pointed in [8], i is usually a small value from 20 to 100.

The most time-consuming part of our algorithm is the random projection with collision recording process. Its time complexity is $O(i*|RT|)$, which is linear.

5. Experimental Evaluation

In this section, we demonstrate the utility of image motifs and provide a detailed study of the effectiveness and efficiency of our algorithm.

5.1 Mining Butterfly Images

There is an increasing interest in using computers to aid in the study of zoology, particularly in *morphometrics,* the study of organism shape and form [33]. This is especially true in entomology because entomologists are challenged by the extraordinary number of insect species, with more than 925,000 species described — more than all other animal groups combined. Even if we were to limit our attention to just the order of Lepidoptera (butterflies and moths), we must deal with more than 20,000 species.

To demonstrate the potential utility of motifs in entomological *morphometrics*, we performed a simple experiment. The experiment was contrived in that we had a strong suspicion as to the final result, however it at least hints at the utility of our ideas.

We chose to work with an extraordinarily diverse group with about 5,000 members, the Nymphalidae, one of the five families of butterflies. Within Nymphalidae there are 12 subfamilies, including Danainae and Limenitidinae. We collected several hundred examples of each group and performed motif join. The 1st Inter-class motif is shown in Figure 10.

The fact that the Inter-class 1st-Motif pair is not only similar in shape, but in color and pattern is at first surprising, given the extraordinary variation that exists within both subfamilies. However this convergence in physical appearance is *not* a coincidence, but rather an example of Müllerian mimicry. Müllerian mimicry is a result of the evolutionary pressure for toxic species mimic each other to display similar warning signals (aposematism) because predators that better associate these signs with unprofitability have higher survival rates than those that do not. Mulerian mimicry drives the evolution and establishment large regional mimetic rings often seen in tropical habitats, made up from the summation of tens of mimicry rings, each containing dozens of species, most belonging to Nymphalidae butterflies, but a few species belonging to other butterfly families (e.g., Papilionidae, Pieridae, Arctiidae and others) [15].

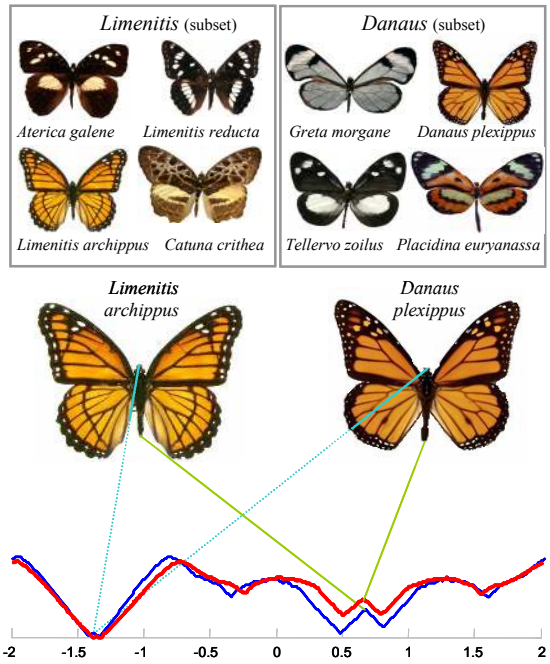

Figure 10: *Top*) Some examples from two subfamilies of Nymphalidae, Limenitidinae and Danainae. *Bottom*) The Inter-class 1st-Motif pair is not only similar in shape, but in color and pattern, a fact which can be explained by Müllerian mimicry

5.2 Annotating Historical Manuscripts

In this experiment, we demonstrate one potential application of inter-class K-Motifs, mining historical texts. The need for algorithms to automatically index and annotate old manuscripts has been brought to the forefront by Google's announcement of a long term plan to digitize tens of millions of old texts in the next decade [35]. While the bulk of the old volumes will contain nothing but text, we can expect millions of images will also be digitized and benefit from enhancement of annotation.

We consider a classic text, British Desmidiaceae, vol. 2 (1905) by the father and son team, West & West [39]. This is a fundamental work on desmids (single-celled freshwater green algae). The book was published when microscopy was a mature science, but before microscopic photography was possible. It contains color and monochrome drawings of exceptional quality.

Approximately 1,150 taxa are described in the five volumes.

The modern reader is impressed by the quality of the illustrations, and stunned by the diversity of algae shapes. However they cannot help but curious if the alien looking illustrations are faithful reproductions of reality or fanciful imaginings[2]. To test this we used our algorithm to find *Inter-class K-Motifs* between two image datasets, Ω is the set of pages from the text in question, and Ψ are the results of a Google image query for "*Desmidiaceae, Micrasterias, Closterium, Euastrum*" (keywords used in the original text). Figure 11 shows one page from the text, and three of the linked images from the web.

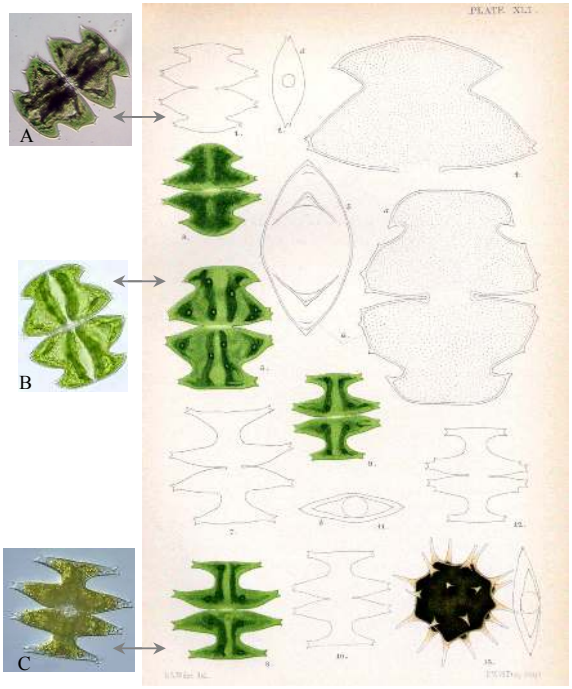

Figure 11: *Right*) Plate 41 from the classic text, British Desmidiaceae, vol. 2 (1905) by West & West. *Left*) After finding the *Inter-class K-Motifs*, individual figures have been linked to images returned by a Google image query. Only three linked images are shown for clarity

Note that the algorithm only considered the shape information, however the color and texture similarity of many of the matches, for example "B" in Figure 11, strongly suggests that the results are not spurious. In Figure 12 we give a visual intuition as to why two shapes are considered so similar in the time series representation.

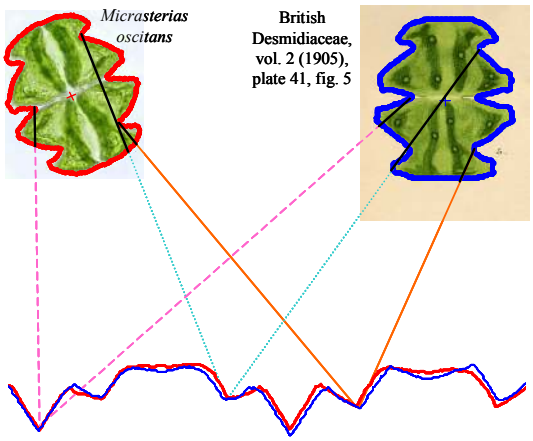

Figure 12: A visual explanation of why two shapes from Figure 11 were linked as Inter-class Motifs. The real image was taken by Fabio Rindi and David John (who retain the copyright). It shows a *Micrasterias oscitans* found in a bog pool in Galway, Ireland on 22nd of Sep 2005

In Figure 13 we show another example on a page featuring drawings of the genus *Closterium*.

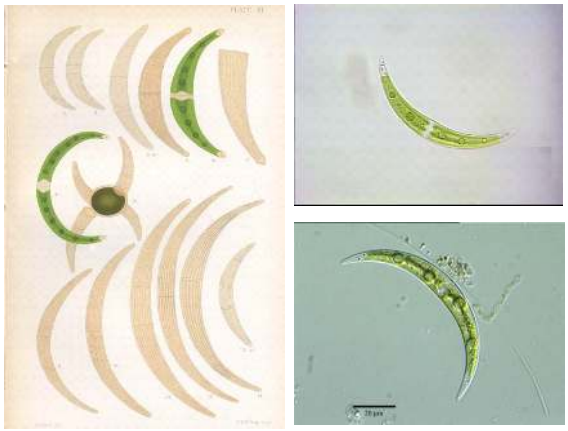

Figure 13: *Left*) Plate 11 from British Desmidiaceae, vol. 2 (1905) by West & West. *Right*) After performing a shape-motif-join, individual figures have been linked to images returned by a Google image query '*Closterium*'

Note that in all these examples, the need for rotation invariance is apparent.

5.3 Efficiency of Motif Discovery Algorithm

In the previous subsections, we have shown that our motif discovery algorithm is very effective in finding image motifs. In this subsection, we will further demonstrate that our approach is not only effective but also efficient, which allows us to discover motifs in linear time with high precision. All the datasets used here are freely available at our website [40].

We test on six image datasets, including SQUID[3] [27], mpeg-7 shapes[4], yoga, chicken [28], Swedish-leaf, and MNIST. SQUID contains 1,100 different sea animal

[2] Note that contemporary publications using the microscopes astronomical analogue, the telescope, had "discovered" and detailed complex systems of canals on Mars [10].

[3] www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html

[4] www.cis.temple.edu/~latecki/research.html#shape

images. Mpeg-7 shapes dataset consists of 1,400 different shapes of animals, insects, crafts etc. Yoga dataset is generated from video sequences of male and female performing yoga actions. Chicken dataset has images of chicken legs, breasts etc. with different rotations. Swedish leaf dataset has 15 species of leaves. MNIST contains 10,000 instances of handwriting number '0' to '9'. There are several reasons why we choose these datasets to test. Firstly, all these datasets contain rotated shapes. We can verify that our motif discovery algorithm is able to locate similar shapes with different rotations. Secondly, each of these datasets has very similar shapes, which guarantees that they contain image motifs. Finally, these diverse datasets include different kinds of images, such as marine animals, human actions sequences, and Arabian numbers etc. Figure 14 shows example images from these six datasets.

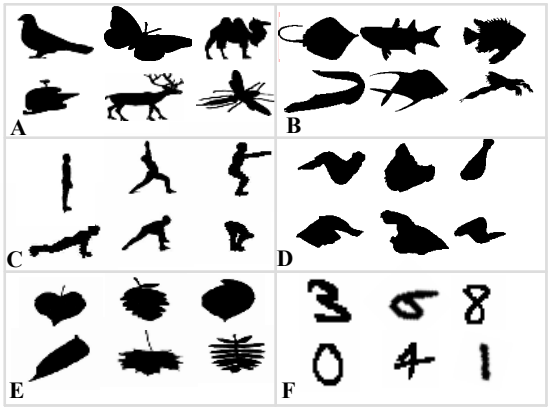

Figure 14: Examples of shapes from six datasets, A - mpeg-7 shapes, B - SQUID, C - yoga, D - chicken, E - Swedish leaves, F - MNIST

We randomly select 1,000 instances from each dataset. Chicken dataset has only 446 images, so we make 1,000 instances by rotating them with random angles. These 6,000 images are converted into time series. Because the lengths of these time series vary from 128 to 3,280, we unify their lengths to 1,024.

We compare three strategies for motif discovery: *brute force* method, brute force with *early abandon*, and our *motif discovery* described in Section 4. Brute force method performs an exhaustive search, computing rotation invariant Euclidean distance for each pair of images. Suppose we have N image time series of length n, then *brute force* requires N^2 rotation invariant comparisons. Brute force with *early abandon* prunes distance computation by the threshold (best-so-far minimum distance in the computation). We randomly select 500, 1000, 2000 and 4000 instances from all 6000 instances, execute three methods ten times to get the average results. Both the number of rotation invariant Euclidean distance computations and the running time of the three strategies are given in Figure 15.

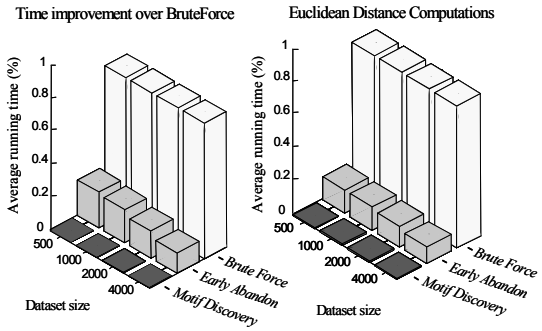

Figure 15: Compared to *brute force* method, only 0.076% distance computations are needed by our *motif discovery* algorithm. The running time is 2 to 3 orders of magnitude shorter

We can see that the *motif discovery* algorithm prunes more than 99.99% computations of the *brute force* method, and only takes about 3% to 7% computation of the *early abandon* method. For running time, we record time spent on *motif discovery* from three parts: converting time series to SAX, random projection, and searching true motif in candidate time series. Although the first part can be done offline, we still include it in execution time because it is nearly constant. The time spent on the second part is almost constant, since in practice we simply set length of SAX word m to 20 and the iteration number i to 30 (user can use different parameter settings in a certain range, but according to our experiments, it will not affect much of the accuracy in motif discovery). Actually the most time-consuming part is in phase three, finding true motifs from candidates. Notice that as shown in the right of Figure 15, SAX projection pruned more than 99.99% computations, indicating that the third part is also very efficient. Overall, our *motif discovery* algorithm is 2 to 3 orders of magnitude faster than *brute force* method, which is clearly shown in the left of Figure 15.

In addition to efficiency, the *motif discovery* is very effective in finding true motif images. We compare the motifs found by our algorithm with those found by *brute force* method, which guarantees to catch all true motif images. Table 4 shows that our method achieves very high accuracy (the ratio that number of true motifs found by our method over the number of true motifs found by *brute force* method).

Table 4: Accuracies of *motif discovery* algorithm. Because the number of motifs is averaged over ten times run, we record them as real values. ξ is the distance threshold given in *definition* 2

Dataset size		500	1000	2000	4000
		$\xi = 1.0$	$\xi = 1.0$	$\xi = 0.5$	$\xi = 0.3$
Number of motifs	*Brute force*	5.9	16.4	18.3	17.2
	Motif discovery	5.2	15.8	18.2	17.2
Accuracy (%) (*motif discovery*/*brute force*)		85.14	95.31	99.83	100

6. Conclusions

We have introduced the new problem of finding approximately repeated shapes in large image databases. Although the brute force approach needs quadratic time, we propose a novel algorithm that uses random projection to identify potential image motifs efficiently. Experimental results show that our approach can efficiently find image motifs with high precision.

Ongoing work includes collaboration with anthropologists on a detailed study of projectile-point cultural artifact transfer, and an application to the study on convergent evolution in the order Coleoptera (beetles). We are also considering combing the shape information currently used with (appropriately weighted) information about color and texture to find image motifs.

Acknowledgments: We would like to thank Dr. Leslie Quintero and Dr. Sang-Hee Lee of the Department of Anthropology, UC Riverside, and Dr. David Marshall and the other members of the DIADIST project based in Cardiff University.

REFERENCE

[1] Adamek, T. and O'Connor, N.E. A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Circuits and Systems for Video Technology*, 14(5): 742-753, 2004.

[2] Adamek, T. and O'Connor, N.E. Efficient contour-based shape representation and matching. *Multimedia Information Retrieval 2003*: 138-143.

[3] Attalla, E. and Siy, P. Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching. *Pattern Recognition*, 38(12): 2229-2241, 2005.

[4] Berrani, S., Amsaleg, L., and Gros, P. Robust content-based image searches for copyright protection. *In Proceedings of ACM Workshop on Multimedia Databases*, 2003.

[5] Cardone, A., Gupta, S.K., and Karnik, M. A survey of shape similarity assessment algorithms for product design and manufacturing applications. *ASME Journal of Computing and Information Science in Engineering*, 3(2): 109-118, 2003.

[6] Chan, K. and Fu, A. W. Efficient time series matching by wavelets. *In Proceedings of the 15th IEEE International Conference on Data Engineering (ICDE'99)*, pp. 126-133, 1999.

[7] Chang, E., Wang, J., Li, C., and Wiederhold, G. RIME: A replicated image detector for the world-wide web. *In Proceedings of SPIE*, 1998.

[8] Chiu, B., Keogh, E., and Lonardi, S. Probabilistic discovery of time series motifs. *In Proceedings of 9th International Conference on Knowledge Discovery and Data Mining (SIGKDD'03)*, pp 493-498, 2004.

[9] Cuviers, G. Le Règne Animal distribue d'après son Organisation pour servir de Base à l'Histoire Naturelle des Animeaux et d'Introduction à l'Anatomie combarée. Printed by Paul Renouard, published by Fortin, Masson et Cie, Libraires, in Paris/France, 1839.

[10] Evans, J. E. and Maunder, E. W. Experiments as to the Actuality of the 'Canals' observed on Mars. *MNRAS*, 63 (1903) 488.

[11] Fridrich, J., Soukal, D., and Lukas, J. Detection of copy-move forgery in digital images. *In Digital Forensic Research Workshop*, 2003.

[12] Gdalyahu, Y. and Weinshall, D. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'99)*, 21(12): 1312-1328, Dec. 1999.

[13] Gutierrez, A. www.graffititracker.net, 2006.

[14] Hall, D, A and Wisner, G. Texas site suggests link with European Upper Paleolithic. *Mammoth Trumpet (Journal of the Center for the Study of the First Americans)* Volume 15, Number 1, 2000.

[15] Joron, M. Mimicry. In Resh V. and Cardé, RT (eds) *Encyclopedia of Insects*. Elsevier Science, San Diego CA, pp 714-726, 2003.

[16] Ke, Y., Sukthankar, R., and Huston, L. Efficient near-duplicate detection and sub-image retrieval. *ACM Multimedia Conference*, 2004.

[17] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems*, pp 263-286, 2000.

[18] Keogh, E. and Kasetty, S. On the need for time series data mining benchmarks: a survey and empirical demonstration. *In Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 102-111, 2002.

[19] Keogh, E., Lonardi, S., and Ratanamahatana, C.A. Towards parameter-free data mining. *In Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[20] Keogh, E., Wei, L., Xi, X., Lee, S.H., and Vlachos, M. LB_Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures. *In Proceedings of Very Large Databases(VLDB'06)*, 2006, to appear.

[21] Kephart, T. Graffiti as Intelligence for Law Enforcement. *Western Society of Criminology 31st Annual Conference*, February 19- 22, 2004.

[22] Kitaguchi, S. Extracting Feature based on Motif from a Chronic Hepatitis Dataset. *In Proceedings of 18th Annual Conference of the Japanese Society for Artificial Intelligence (JSAI'04)*, 2004.

[23] Kumar, N., Lolla N., Keogh E., Lonardi, S., Ratanamahatana, C., and Wei, L. Time-series bitmaps: a practical visualization tool for working with large time series datasets. *SIAM Data Mining Conference,* 2005.

[24] Li, D. and Simske, S. Shape retrieval based on distance ratio distributions. *HP Tech Report. HPL-2002-251*.

[25] Lin, J., Keogh, E., Lonardi, S., and Chiu, B. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. *In Proceeding of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003.

[26] Lin, J., Keogh, E., Lonardi, S., Lankford, J.P., and Nystrom, D.M. Visually Mining and Monitoring Massive Time Series. *In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 460-469, 2004.

[27] Mokhtarian, F. and Bober, M. Curvature Scale Space Representation: Theory, Applications and MPEG-7 Standardization. *Kluwer Academic*, 2003.

[28] Mollineda, R. A., Vidal, E., and Casacuberta, F. Cyclic Sequence Alignments: Approximate Versus Optimal Techniques. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 16(3): 291-299, 2002.

[29] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. Shape Distributions. *ACM Transactions on Graphics*, 21(4): 807-832, October, 2002.

[30] Pan, J.Y., Balan, A., Xing, P., Traina, A.J.M., and Faloutsos, C. Automatic Mining of Fruit Fly Embryo Images. *SIGKDD*, 2006, to appear.

[31] Pope, G. A. Weathering of petroglyphs: direct assessment and implications for dating methods. *Antiquity*, 74(2000): 833–843.

[32] Ratanamahatana, C.A. and Keogh, E. Three myths about Dynamic Time Warping Data Mining. *In Proceedings of SIAM International Conference on Data Mining (SDM '05)*, Newport Beach, CA, 2005.

[33] Richtsmeier J, DeLeon V, Lele S. The Promise of Geometric Morphometrics. American Journal of Physical Anthropology. vol:119 iss:s35 pp:63 -91, 2002.

[34] Rombo, S. and Terracina, G. Discovering Representative Models in Large Time Series Databases. *In Proceedings of the 6th International Conference on Flexible Query Answering Systems*, pp 84-97, 2004.

[35] Said, C. Revolutionary chapter: Google's ambitious book-scanning plan seen as key shift in paper-based culture. San Francisco Chronicle. December 20, 2004.

[36] Tompa, M. & Buhler, J. Finding motifs using random projections. *In proceedings of the 5th Int'l Conference on Computational Molecular Biology*. Montreal, Canada, Apr 22-25. pp 67-74, 2001.

[37] Wang, Z., Chi, Z., Feng, D., and Wang, Q. Leaf Image Retrieval with Shape Features. In *Proceedings of the 4th International Conference on Advances in Visual Information Systems*, pp 477- 487, 2000.

[38] Watchman, A. A universal standard for reporting the ages of petroglyphs and rock paintings. *In M. Strecker and P. Bahn (eds), Dating and the earliest known rock art*, pp. 1-3. Oxbow Books, Oxford, 1999.

[39] West, W. and West, G.S. A Monograph of the British Desmidiaceae. Vols.I–V. *The Ray Society*, London, 1904–1922.

[40] www.cs.ucr.edu/~xxi/SDM07.

[41] Zhang, D. and Lu, G. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1-19, 2004.

[42] Zhang, D. and Chang, S. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. ACM Multimedia 2004: 877-884.

[43] Zunic, J., Rosin, p., and Kopanja, L. Shape Orientability. ACCV (2) 2006: pp 11-20.

Appendix A: SAX Reconstruction Error

In the main text we made reference to SAX reconstruction error. While the term "reconstruction error" is well defined for other representations such as wavelets and Fourier approximations, it is not generally used for symbolic representations of discrete data. Here we show that we can quantitively measure the reconstruction error of SAX representation.

By converting time series to SAX words, we reduce the dimensionality of time series. Clearly some information is lost during the conversion. To measure how well the SAX representation approximates the original time series, we define SAX reconstruction error. The SAX reconstruction error is the sum of the distance between each data point in the time series and the middle line of the SAX symbol that the data point maps, or more formally as:

$$error = \sqrt{\sum_{i=1}^{n} (t_i - \beta_{2\alpha_i - 1})^2} \qquad (8)$$

where t_i is the i^{th} data point of the time series, α_i is the SAX symbol that t_i maps to, and $\beta_{2\alpha_i - 1}$ is the value that divides the region of SAX symbol α_i into two equiprobable parts. For example, in Figure 16, a time series is converted into SAX word **cadcac**. The alphabet size is four. According to Table 1, the breakpoints are (-0.67, 0, 0.67), as shown by the left Y-axis of Figure 16. The values that divide each symbol region to two equiprobable parts are (-1.15, -0.32, 0.32, 1.15), shown in the right Y-axis of Figure 16. These can again be looked up from Table 1 since dividing each region (the shaded area in Figure 16) to two parts is equivalent to doubling the alphabet size to eight.

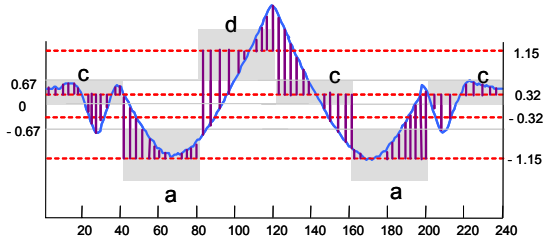

Figure 16: A visual illustration of SAX reconstruction error. The reconstruction error is calculated as the sum of the distance between each data point and the middle line (the dot line) of the SAX symbol that the data point maps to