

An efficient and effective similarity measure to enable data mining of petroglyphs

Qiang Zhu · Xiaoyue Wang · Eamonn Keogh · Sang-Hee Lee

Received: 24 February 2010 / Accepted: 20 August 2010 / Published online: 23 September 2010
© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract Rock art is an archaeological term for human-made markings on stone, including carved markings, known as petroglyphs, and painted markings, known as pictographs. It is believed that there are millions of petroglyphs in North America alone, and the study of this valued cultural resource has implications even beyond anthropology and history. Surprisingly, although image processing, information retrieval and data mining have had a large impact on many human endeavors, they have had essentially zero impact on the study of rock art. In this work we identify the reasons for this, and introduce a novel distance measure and algorithms which allow efficient and effective data mining of large collections of rock art.

Keywords Image processing · Similarity search · Cultural artifacts

Responsible editor: Geoffrey Webb.

Q. Zhu (✉) · X. Wang · E. Keogh
Department of Computer Science and Engineering, University of California Riverside,
Riverside, CA 92521, USA
e-mail: qzhu@cs.ucr.edu

X. Wang
e-mail: xwang@cs.ucr.edu

E. Keogh
e-mail: eamonn@cs.ucr.edu

S.-H. Lee
Department of Anthropology, University of California Riverside, Riverside,
CA 92521, USA
e-mail: sang-hee.lee@ucr.edu

1 Introduction

Rock art is an archaeological term for human-made markings on stone, including petroglyphs, *carvings* into stone surfaces and pictographs, *paintings* on stone. Figure 1 illustrates some examples of each, which hint at the extraordinary variability of rock art in terms of complexity.

Petroglyphs and pictographs are one of the earliest expressions of abstract thinking, and a true hallmark of humanity. They provide a rich body of information on several different dimensions, beyond their value as an aesthetic expression. Studies of rock art have implications beyond anthropology and history. For example, a recent study postulates the existence of a now-extinct Australian bat species based on extraordinarily detailed pictographs known to be at least 17,500 years old (Pettigrew et al. 2008); petroglyphs have been used in studies of climate change; and the changing inventories of species in the Dampier Archipelago from the Pleistocene to the early Holocene period have been reconstructed partly by petroglyph evidence (Aseyev 2008). However, in spite of these successes, progress in petroglyph research has been frustratingly slow.

A decade ago, Walt et al. summed up the state of petroglyph research by noting, “*Complete-site and cross-site research thus remains impossible, incomplete, or impressionistic*” (2006). Surprisingly, there has been little change in the intervening decade, yet in the same time frame we have seen significant advances in image processing and data mining. These advances have resulted in fielded applications in domains as diverse as medicine, entertainment, wildlife management, e-commerce, biometrics, zoology (Pan et al. 2006), etc. Nevertheless, these advances have had essentially zero impact on the analysis of petroglyphs and pictographs.

We believe that this is because the extraordinarily diverse and complex structure of rock art images defies most existing image matching algorithms. Most approaches are simply not suitable to capture the similarity of petroglyphs, and those that are, even in limited cases, do not scale to large collections we need to examine. In this work we introduce a novel distance measure for rock art, and show that it can correctly capture the *subjective* (and where available, *objective*) similarity between petroglyphs. We show how we can use this distance measure as a basis for several higher-level “data-mining” algorithms, for example finding repeated motifs, clustering, or simply enabling query-by-content. Our distance measure operates on binary features extracted from the data; however, this begs the question of how we can automatically extract high quality features from an unconstrained and extraordinarily diverse data source.



Fig. 1 A random selection of petroglyphs and pictographs, hinting at their incredible variability, complexity and beauty

Our solution is to use the human computation framework pioneered in the work of Luis von Ahn et al. (2003). In essence, we take the feature extraction steps which cannot (currently) be performed by algorithms and outsource them to humans in the form of a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), a puzzle design to tell humans and computers apart. As we shall show, this idea allows us to efficiently and accurately extract meaningful data from rock art.

The rest of this paper is organized as follows. Section 2 contains background information and a discussion of related work. In Sect. 3 we review the Generalized Hough Transform, and show how we can adapt it to produce a fast and robust distance measure for petroglyphs. We test our ideas with a comprehensive set of experiments in Sect. 4. In Sect. 5 we show how to frame the feature extraction step as a CAPTCHA, before offering conclusions and directions for future work in Sect. 6.

2 Background and related work

The earliest petroglyphs have traditionally been associated with the appearance of modern humans in Europe such as the famous example from the Lascaux Cave, France, and an early one from the Chauvet Cave, France which dates back to as early as 30,000 years ago (Valladas et al. 2001). Recent work has shown that the idea of expressing abstract motifs appears much earlier, 77,000 years ago in South Africa (Henshilwood et al. 2002). Given this long history, it is one of the most valuable sources of humanity that has persisted to the present time.

Beyond their value as an aesthetic expression, petroglyphs provide a rich source of information for researchers. Repeated motifs can be identified and traced through time and space, which in turn may shed light on the dynamic histories of human populations, patterns of their migrations and interactions, and even continuities to the present indigenous societies. However, the nature of petroglyphs poses an extremely difficult challenge. As in the case for any other artifacts of history, damage to petroglyphs is permanent and irreversible. However, unlike other artifacts that can be preserved and protected within the confines of a controlled environment in a museum, petroglyphs are mostly left in their natural settings, exposed to elements of nature that will erode them inevitably with time. There is an urgent need to identify petroglyphs and to archive them for humanity.

2.1 Background on rock art

As we shall show in Sect. 3, our algorithm assumes that the input images are (relatively) low-resolution bitmaps with a 1-bit color depth, one petroglyph per image. However, as Fig. 1 illustrates, obtaining such images may be non-trivial. With rare exceptions, petroglyphs do not lend themselves to automatic extraction with segmentation algorithms. For example, in the first and third images of Fig. 1, segmentation algorithms find the “edges” due to cracks in the rock to be more significant than the actual edges of the petroglyphs. Moreover, these images were chosen for this example for their high contrast and clarity; most petroglyphs would be even more challenging.

In spite of this, in the next two sections we show how we easily obtained tens of thousands of petroglyphs for this study, and how we plan to have at least one million examples in the very near future.

2.1.1 Human computation to process petroglyphs

The last five years have seen a flurry of research on *Human Computation* (also known as “crowdsourcing”), much of it leveraging off the pioneering work of Luis von Ahn at CMU (2003). The essence of human computation is to have computers do as much work as possible to solve a given problem, but to outsource certain critical steps to humans. These steps are ones which are difficult for computers, but simple for humans. One of the most famous examples is the *Google Image Labeler*, which is a program that allows the user to label random images to help improve the quality of Google’s image search results. Like many such efforts, human time is donated for free, because the task is embedded in a fun game, hence the recently coined term, Games with a Purpose, or GWAP (von Ahn 2006). We have exploited and extended some of these ideas to create a system called CAPTCHA-ROCK, which allows human volunteers to “help” computer algorithms segment and annotate petroglyphs (cf. Sect. 5).

Our similarity measure (which will be discussed in Sect. 3.5) for rock art, and thus all of the higher level data mining algorithms that build upon it, explicitly assumes that our human computation step can meaningfully extract the essential shape of the petroglyphs. It is not obvious that this is true. One might imagine that the subjective notions of the humans in the loop might dwarf any true object similarity that exists between the petroglyphs. We designed a detailed sanity check experiment to test this.

We found eight petroglyphs (examples **e** and **f** are technically *pictographs*) which can objectively group into four pairs. In the case of examples **a** through **f**, there is strong evidence based on co-location that they were created by the same hand.

We asked two volunteers, whom we denote SC and WY, to annotate the eight petroglyphs in our CAPTCHA-ROCK system. The volunteers were computer science graduate students, without any special knowledge or training in either anthropology or art. Neither of the volunteers met the other, or was allowed to see the other’s work. The annotation was conducted on a desktop with a mouse. Both volunteers annotated one sample petroglyph provided in our tool as the training before this experiment.

Figure 2 shows the eight original petroglyphs, and the transcribed skeletons produced by the two volunteers.

Note that there is significant variation between the two volunteers’ output. The most notable of these is in the two deer denoted **e** and **f**. User WY transcribed these as stick figures; however, user SC transcribed them as outlines. Another obvious difference is that user SC explicitly added a stroke for the tail in the bighorn sheep shown in **g** and **h**, something that user WY did not see as necessary.

In retrospect, we believe that we can reduce some of this subjectivity by providing a slightly longer and more detailed training session for users. However, we obviously need our similarity measure to be robust to such human factors. We can test to see if it is by hierarchically clustering all sixteen transcribed petroglyphs. We used the clustering distance measure that we will introduce in Sect. 3.5.

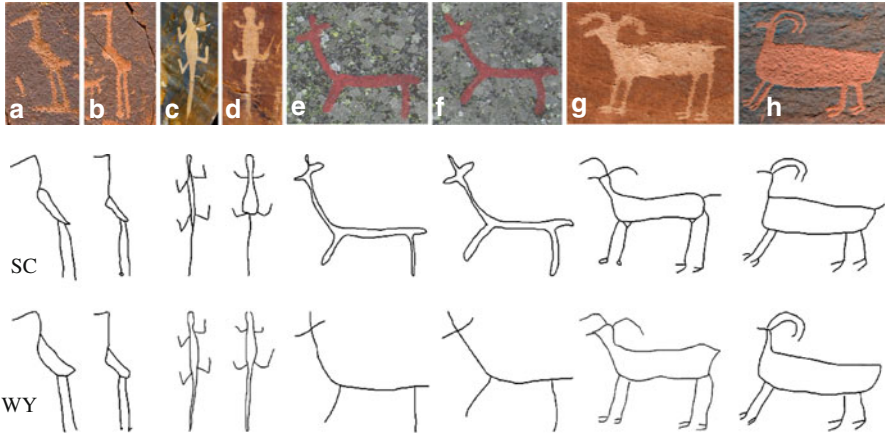


Fig. 2 First row eight original petroglyphs (a to h). Second row corresponding skeletons transcribed by the volunteer SC. Third row skeletons transcribed by the volunteer WY

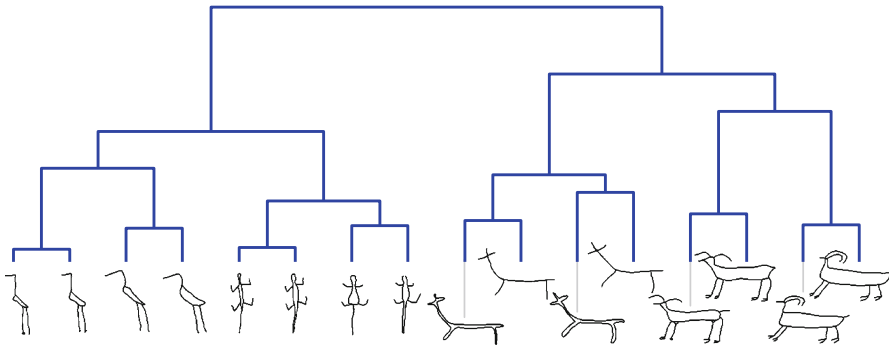


Fig. 3 A group-average linkage hierarchical clustering of all 16 transcribed petroglyphs

The results are very reassuring (Fig. 3). At the lowest branches of the dendrogram, each of the two transcribed versions of an original petroglyph is linked with each other. Likewise, at the next higher level of the dendrogram, both versions of the birds (lizards/deer/sheep) join together in a sub-tree before merging with the rest of the animals. This dendrogram shows that our distance measure is very robust to human variability and the CAPTCHA-ROCK system can capture the essential character of shapes.

2.1.2 Existing archives of petroglyphs

Beyond the examples captured by our human computation system, there are several other rich sources of rock art data to be mined. For example, anthropologists have been sketching petroglyphs for hundreds of years, and recent efforts to digitize historical manuscripts have made at least hundreds of books, each with at least a few thousand petroglyph images, freely available on the web. In Fig. 4 we show an example from the 1888 edition of a series of government reports (Powell 1888).

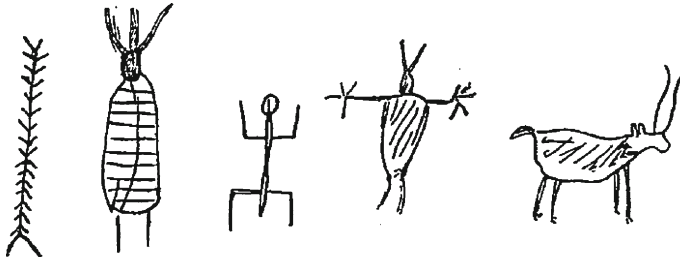


Fig. 4 An excerpt from an 1888 government report (Powell 1888). The original caption is “Petroglyph in Arizona”

Images of this type can be of particular interest because they may refer to petroglyphs which have long since been destroyed. Furthermore, although the petroglyphs in Fig. 4 predate photography, it is important to note that because petroglyphs often do not reproduce well in photographs, the practice of hand drawing or tracing petroglyphs is still used in modern anthropological texts.

2.2 Background on image processing

An understanding of *similarity* must be at the heart of any effort to analyze petroglyphs and other cultural artifacts. For example, an image of a horseman incised on a fossilized ostrich eggshell fragment was recently found among eolian deposits in the Gobi Desert, Mongolia (McDonald and Veth 2007). An obvious thing to do with such an image in order to place it in a cultural context is to ask if a *similar* image exists among the many petroglyphs in the region. Thus, we began this project with careful consideration of shape similarity.

In soliciting feedback and advice for early previews of this work from various researchers in the data mining and image processing community, the feedback obtained was almost always of the form “*Very nice, but have you considered using X*”, where *X* was Geometric Hashing (Wolfson and Rigoutsos 1997), Hausdorff Distance (Huttenlocher et al. 1993), Chamfer Matching (Borgefors 1998), Shape Contexts (Belongie et al. 2002), Fréchet Distance (Alt and Godau 1995), Skeleton Graphs (Bai and Latecki 2008), Zernike moments (Teague 1980), Earth Movers (Assent et al. 2008), etc. While we have considered (and in some cases experimented with; see (Zhu 2009) these distance measures, space limitations prohibit a detailed review and discussion of the pros and cons of each of them. Indeed, the preceding list is only a small subset of the hundreds of shape similarity measures in existence. (See Veltkamp 2001; Zhang and Lu 2004 and the references therein for an overview.) However, we argue that some of the unique properties of petroglyphs render most of them unsuitable for the task at hand. Consider the following difficulties illustrated by Fig. 5:

- A single atomic petroglyph may contain several disconnected parts. Thus, boundary based methods (Keogh et al. 2006) and graph based methods (Bai and Latecki 2008) cannot be applied, at least not directly (c.f. Fig. 14, which shows an example of a problem which would defeat boundary and graph based methods).

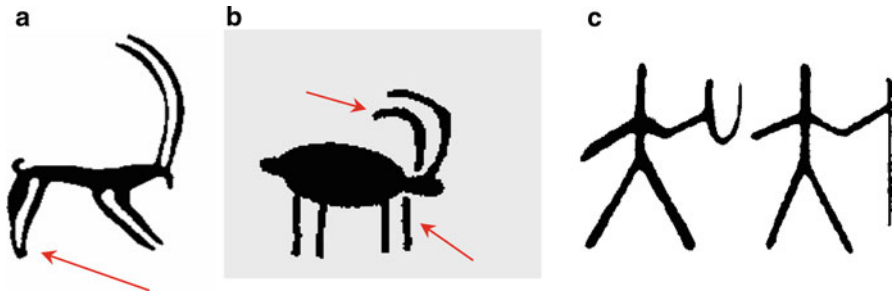


Fig. 5 **a** An Ibez petroglyph taken from [Takaki et al. \(2006\)](#) has its two rear hoofs fused. It is not clear if this is an artifact of scanning or the artist's intent, and it *does* make a critical difference to graph based methods. **b** This bighorn sheep from a classic work ([Grant et al. 1968](#)) has a disconnected leg and horn, which will greatly affect its representation for graph based methods. **c** Two petroglyphs from Easter Island are clearly distinct, yet identical in graph based representations

- Geometric hashing is a very useful technique for indexing large collections of shapes ([Wolfson and Rigoutsos 1997](#)). However, it is only well defined for machine parts and architectural drawings with many clearly defined right angles/intersections/circle centers, etc. It has not been shown to have utility for more general unconstrained shapes.
- There are many specialized distance measures which have been introduced for indexing music notation, Japanese kanji, mathematical symbols, pen-based computing, etc. At least some subsets of these look like at least some subsets of petroglyphs. However, it must be remembered that in these domains there are only a finite (and relatively small) number of possible classes, and we can at least imagine an idealized prototype for each class (i.e. a perfectly drawn square root sign). However, this is not the case for petroglyphs which do not generally fall into discrete classes, and cannot generally be seen as corrupted versions of an idealized template.

Instead of attempting an exhaustive discussion of why we have discounted existing shape distance measures, we will briefly review the positive reasons for why we chose the GHT measure.

- As we shall show, on real, but unlabeled anthropological datasets, the GHT produced subjectively correct answers (cf. Sect. 4.1). Furthermore, on *labeled* datasets which are very similar to petroglyphs, the GHT produces results which are competitive with state-of-the-art approaches.
- As we will demonstrate in this work, we are able to tightly lower bound the GHT, allowing for very efficient searches in large datasets. Moreover, we show that we can make a slight variant of the GHT obey the triangular inequality, thus allowing us to use off-the-shelf data mining algorithms, for example to find motifs ([Mueen et al. 2009](#)).
- The GHT makes essentially no assumptions about the data, and thus is defined for open/closed boundaries, for connected/disconnected shapes, etc. This is important because, as hinted at in Figs. 1, 2 and 3, petroglyphs are extraordinarily diverse.

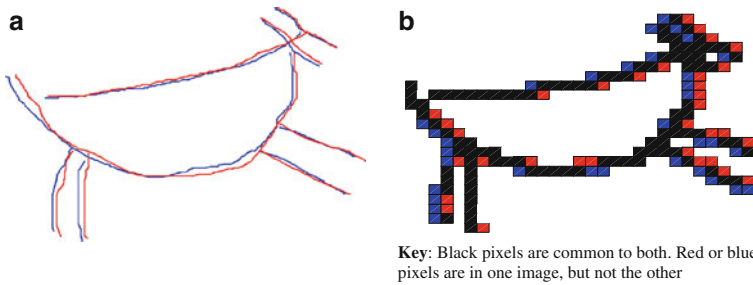


Fig. 6 **a** Two overlaid skeleton traces of the same image of a Bighorn sheep. **b** The same two images after downsampling

We are now in a position to give some intuition as to why we intend to do data mining on a relatively low resolution of the petroglyph images. Using our CAPTCHA-ROCK, we asked two individuals to trace a petroglyph of a bighorn sheep petroglyph found in Arizona; the resulting two skeletons are shown in Fig. 6a. The skeletons are on a bitmap of 340 by 250. Although the two images are very similar, less than 3.5% of the pixels from each image overlap. We can contrast this with the situation after converting the images to a down sampled representation, as shown in Fig. 6b. Here the images are transformed to a mere 30 by 23 grid representation. However, of the 130 pixels that form each image, 75.6% of the pixels are common to both.

In essence, the original image representation has spurious precision. This precision is unwarranted because, as we have shown in Fig. 3, there is some uncertainty introduced by the human element of the algorithm.¹ The quantizing produced in the down-sampling step also introduces some uncertainty, but this is completely dwarfed by original uncertainty. Furthermore, as we shall see, the lower resolution representation has several unique advantages which we can leverage off. In Sect. 4, we provide forceful empirical evidence that appropriate amounts of downsampling significantly improve accuracy in objective tests.

3 Generalized hough transform

We begin by reviewing the classic generalized Hough Transform algorithm and then introduce our modifications and extensions.

3.1 Classic generalized hough transform

The Hough transform (Hough 1966; Duda and Hart 1972) is a useful method for two-dimensional shape detection, but it is limited to analytic curves. It was generalized to *detect* arbitrary shapes in Ballard (1981) and Merlin and Farber (1975); however, these works did not explicitly encode a similarity measure.

¹ For those rare petroglyphs that can be processed without human intervention, there is uncertainty introduced by camera angle, focal length, etc.

We note that there are many variants of the Hough transform, and the notation in the literature is inconsistent. The particular variant of the algorithm we consider, and the notation we will use to describe it, is most similar to Merlin and Farber’s (1975), in which shapes are constituted of *edge points*. Edge points are simply the dark pixels in our one-bit representation of shapes. Suppose we have a candidate shape C defined as:

$$C_{[x,y]} = \begin{cases} 0 & \text{if } [x, y] \text{ is an edge point} \\ 1 & \text{otherwise} \end{cases}$$

and we want to find the *best fit* of a query shape Q defined in the same way as C . That is, given a reference point R in Q , to find the best point R' in C , if we put C onto Q (where only translation in the plane is allowed) and points R and R' coincide, then the number of matched edge points would be the maximal.

For clarity, we use a very simple example to illustrate the algorithm. Figure 7 shows a query shape Q and a candidate shape C . Note that the shapes can be disconnected, as in Q .

As shown in Fig. 8, the first step is to mark a reference point R in Q (usually the center of mass of all edge points) and rotate the edge points of Q around R by 180° (a and b of Fig. 8). We then draw vectors from R to each edge point (c of Fig. 8). These vectors form a “star-like” pattern which we will use to determine the best fit of Q in C .

To find the best alignment of Q to C , together with a numeric evaluation of their similarity, we do the following. The “star” vectors are superimposed on each edge point of C (as shown in Fig. 9a). An accumulator matrix A of the same dimensions as

Fig. 7 Toy examples of a query Q and a candidate match C . Each cell is a pixel, and the dark colors denote edge points of shapes

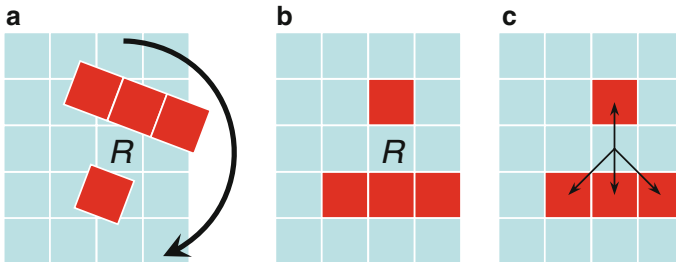
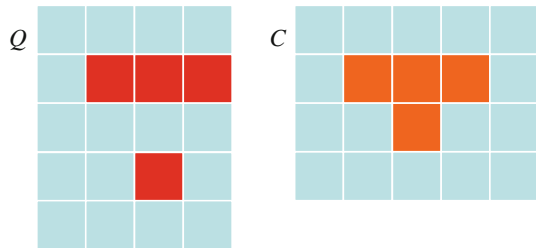


Fig. 8 a and b The shape Q is rotated 180° around center of mass R . c Four vectors of Q form a “star pattern”

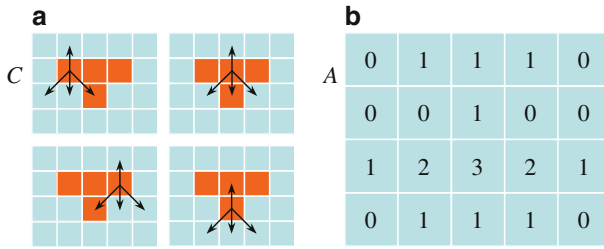


Fig. 9 Placement of vectors on each edge point of C (a) and the final accumulator A (b)

C is used to record the number of vector ends (i.e. the arrowheads) that fall into each cell (Fig. 9b shows the final accumulator).

The cell in A with the maximal value is the best point R' we want to find, and its value equals the maximal number of edge points that can be matched between Q and C . This is 3 in our example. Note that while R is the center of mass of Q by definition, point R' is not necessarily the center of mass of C .

Based on this maximal value, we can further obtain the *minimal unmatched edge points (MUE)* of Q . This is simply the number of edge points in Q minus the number of maximal matched points. This *MUE* can be used as a distance measure. In our toy example, with similar shapes, its value is 1. If Q were exactly the same as C , the *MUE* would be 0, meaning $D(Q, C) = 0$. As we shall later see, it can be useful to normalize and adjust this number before using it as a distance measure.

For concreteness we show the algorithm to compute the *minimal unmatched edge points* in Table 1.

If Q and C have $S \times S$ pixels, and we denote the number of edge points in Q and C by N_Q and N_C respectively, then the time complexity of this algorithm is $O(N_Q \times N_C + S^2)$.

Table 1 The *minimal unmatched edge points (MUE)* from Q to C

Procedure $[MUE] = \text{Classic_GHT}(Q, C)$

- 1 $(R_x, R_y) \leftarrow$ center of mass of Q ;
- 2 **foreach** edge point (x, y) in Q
- 3 $x \leftarrow 2 \times R_x - x; V_x \leftarrow x - R_x$;
- 4 $y \leftarrow 2 \times R_y - y; V_y \leftarrow y - R_y$;
- 5 add (V_x, V_y) to the set *Vectors*;
- 6 **endfor**
- 7 Initialize a matrix A with the same size of C to 0;
- 8 **foreach** edge point (x, y) in C
- 9 **foreach** vector (V_x, V_y) in *Vectors*
- 10 $A(x + V_x, y + V_y) ++$;
- 11 **endfor**
- 12 **endfor**
- 13 $MUE \leftarrow$ number of edge points of $Q - \max(A)$;

3.2 A new cell incrementation strategy

The classic GHT algorithm can be seen as a cell value incrementation process of the accumulator (as reflected line 8–12 in Table 1), and we need to wait for all of the incrementation to finish before we can obtain the value for any particular cell. Here we propose a new cell value incrementation strategy which allows obtaining the cell values one by one. This will allow us, for the first time, to use a *lower bounding* strategy for the GHT (Zhu et al. 2009).

Instead of superimposing vectors on edge points and increasing the value of the corresponding cell, we reverse this process by checking all positions that are possible to increase the value of one particular cell. To achieve this, we need to reverse the direction of the vectors.

Figure 10 shows this simple idea (using the same example as in the last section): first we draw vectors from R to each edge point of Q , but without rotating Q (a); if we want to calculate the value of a particular cell, say, the one at the third row and second column, we superimpose all vectors onto that cell (b). Then we check every cell with a vector falling into it: if this is also an edge point, we increase the cell value by 1 (because it is guaranteed, when using the classic GHT, that one vector superimposed on this edge point will fall into the target cell). Finally, after checking four cells, we obtain the value 2 for this cell.

It is obvious that our new cell value incrementation strategy is equivalent to the classic one. However, this strategy has one advantage in that it allows for the implementation of the cell incrementation process in parallel, which avoids nesting-for-loops in the classic GHT (line 8–12 in Table 1). In this paper, however, we are not going to discuss this. We will instead utilize the nice property “obtaining cell value one by one” as a base to explore a lower bound of minimal unmatched edge points in the next two sections.

3.3 The intuition behind lower bounding

As noted above, the time complexity of the GHT is quite high, and this limits its applicability for larger datasets. The classic data mining solution to the problem of time

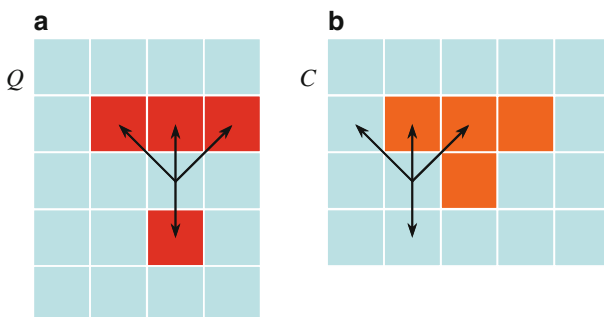


Fig. 10 Four vectors of Q (a) and placement of vectors on one cell of C (b)

consuming distance measures is to find an efficiently computable tight *lower bound* to the distance measure, and to use this bound to cheaply prune off unpromising candidates (Keogh et al. 2006).

We are now in a position to show the first known lower bound of the GHT-based distance. Our idea is based on extracting one-dimensional “signatures” from the two-dimensional query and candidate images. While we extract signatures from both the rows and columns, for ease of exposition we begin by showing just the column signature, which we denote as $SigCx$.

For a candidate shape C with m rows and n columns, we have:

$$SigCx = \left\{ \sum_{i=1}^m C_{[1,i]}, \sum_{i=1}^m C_{[2,i]}, \dots, \sum_{i=1}^m C_{[n,i]} \right\}$$

In other words, we are simply counting all of the edge points in each column of C . For example, the truncated-corner square shape shown on the Fig. 11. *right* has $SigCx = \{0,0,0,3,2,2,2,3,0,0,0\}$.

We can extract these signatures as part of the preprocessing of the images, and store them in an index. At query time, we can use an identical technique to extract a signature, $SigQx$, from the query image Q . As shown in Fig. 12 the only difference is that we truncate any leading or trailing 0’s from the $SigQx$ signature.

As it happens, the *MUE* distance in this case is 4, a number we can compute using the algorithm in the previous section. However, we can compute a lower bound to this value by looking at *just* the respective signatures.

We can obtain the intuition behind the lower bound by imagining that Q “wants” to match perfectly to C , with no missing edge points. As we place “star” vectors to one cell on the center column of C , if Q “wants” all vectors to fall into edge points of C , a necessary, but not sufficient, condition for this to happen is that the number of vectors falling into each column is less than or equal to the number of edge points in that column. This is equivalent to checking whether each value in a $SigQx$ cell is less than or equal to the corresponding cell in $SigCx$ (as shown in Fig. 12).

Referring to Fig. 12, we can see that in the slot $SigQx_1$ we need two edge points, and the corresponding slot in $SigCx_i$ actually has three. There is no penalty for $SigCx$ having a surfeit of edge points. In the next slot $SigQx_2$ we need two edge points, and the corresponding slot in $SigCx_{i+1}$ has the two required edge points.

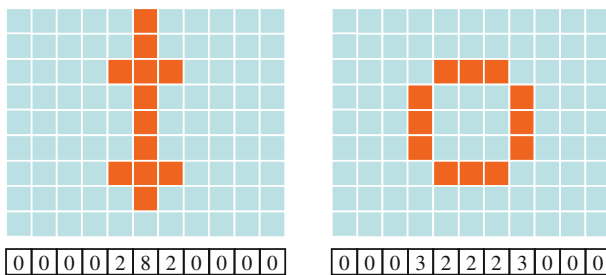


Fig. 11 We can extract “signatures” from shapes by summing up the number of edge points in each column

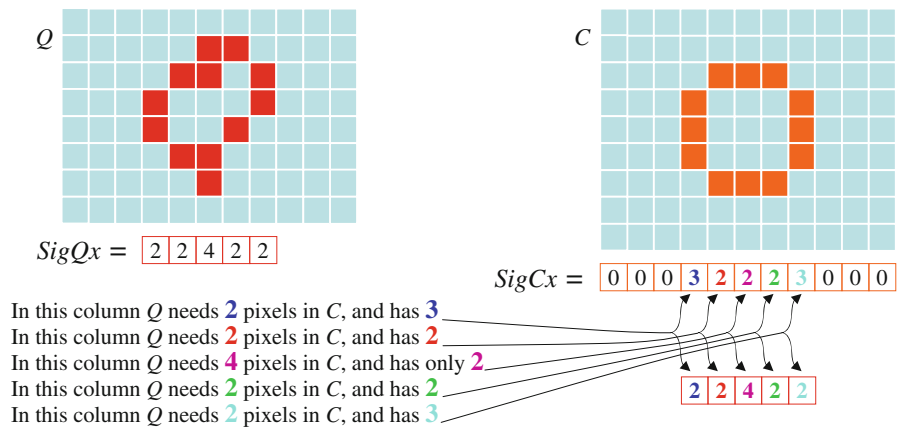


Fig. 12 (left) A query image Q has its signature $SigQx$ extracted. (right) By noting how many edge points it needs C to have at each column, and how many edge points the column as C actually has, we can derive a lower bound of $D(Q, C)$

However, in the slot $SigQx_3$ we need four pixels, but the corresponding slot in $SigCx_{i+1}$ has only two pixels. Thus, we are guaranteed that no matter how the pixels are arranged, this column will contribute at least two to the number of missed edge points in the accumulator. As we continue, we find that neither of the two remaining slots contributes to the lower bound, because in each case there are at least enough pixels in $SigCx$ to satisfy $SigQx$. Thus, we can say that in this alignment, the lower bound $LB(SigQx, SigCx_{[4:8]}) = 2$.

Note that this lower bound is only for the particular alignment shown in Fig. 12; if we had shifted $SigQx$ one to the left, the lower bound would be 12, and if we had shifted $SigQx$ one to the right, the lower bound would also be 12. If we test all alignments, we must choose the *smallest* value discovered as the true lower bound for the columns, which we denote as $LB(SigQx, SigCx) = 2$.

Finally, as hinted at above, we can do the same thing for the rows, using $SigQy$ and $SigCy$. The final global lower bound to $D(Q, C)$ is then simply the *larger* of the two individual lower bounds.

3.4 A formal description of the lower bound

We expand the intuition presented in the last section to introduce a formal description of the lower bound. We again begin by considering the lower bound for just the columns. The algorithm is formalized in Table 2, which takes in a query shape Q and the column signature of candidate shape C . As described in the previous section, to obtain $LB(SigQx, SigCx)$, we need to shift $SigQx$ from the left to the right of $SigCx$ by aligning the center of mass of $SigQx$ to each cell of $SigCx$ (lines 5, 7 and 8 of Table 2). In each alignment, we calculate the lower bound for each column of C . Note that when some cells of $SigQx$ shift out of $SigCx$, the edge points in these cells cannot find points in C to match them and then all contribute to the number of missed points

Table 2 Algorithm to calculate the column lower bound of GHT by giving the query shape Q and column signature of candidate shape C

```

Procedure  $[LBx] = LB\_GHT(Q, SigCx)$ 
1  $SigQx \leftarrow$  column signature of  $Q$ ;
2  $LBx \leftarrow$  number of edge points in  $Q$ ;
3  $Rx \leftarrow$  center of mass of  $SigQx$ ;
4  $left \leftarrow Rx - 1$ ;
5 for  $i \leftarrow 1$ :  $length(SigCx)$ 
6    $mised \leftarrow 0$ ;
7   for  $j \leftarrow 1$ :  $length(SigQx)$ 
8      $k \leftarrow (i - left) + (j - 1)$ ;
9     if  $k < 1 || k > length(SigCx)$ 
10       $mised \leftarrow mised + SigQx[j]$ ;
11     else
12        $delta \leftarrow SigQx[j] - SigCx[k]$ ;
13       if  $delta > 0$ 
14          $mised \leftarrow mised + delta$ ;
15       endif
16     endif
17     if  $mised > LBx$ 
18       break;
19     endif
20   endfor
21   if  $mised < LBx$ 
22      $LBx \leftarrow mised$ ;
23   endif
24 endfor

```

(line 9–10 of Table 2). Finally, $LB(SigQx, SigCx)$ is the minimal value of all of these lower bounds (reflected in line 21–23 of Table 2).

One important optimization we use here is *early abandoning*. When calculating the lower bound for a column, if the number of missed points exceeds the current best (*smallest*) lower bound, we can stop calculations and shift to the next position (line 17–19 of Table 2). For a better pruning, we can align $SigQx$ and $SigCx$ by their centers of mass first, and then shift stepwise to two sides (omitted in Table 2 for brevity).

In summary, we have:

$$\begin{aligned}
 & LB(SigQx, SigCx) \\
 &= \min_{i=1}^{length(SigCx)} LB(SigQx, SigCx[i - left : i - left + length(SigQx) - 1])
 \end{aligned}$$

To get the final lower bound, we simply run the algorithm in Table 2 again, this time with $SigCy$ instead of $SigCx$, and with all column operators changed to row operations.

After then calculating $LB(\text{Sig}Qy, \text{Sig}Cy)$, the final lower bound $LB(Q, C)$, is simply $\max[LB(\text{Sig}Qx, \text{Sig}Cx), LB(\text{Sig}Qy, \text{Sig}Cy)]$.

The time complexity of our lower bound algorithm is $O(S^2)$. Note that it is independent of the number of edge points in images. As we shall show in Sect. 4.4, a similarity search using the lower bound achieves a one to two order of magnitude speed-up.

3.5 Variants on the basic distance measure

While the *MUE* is in itself a useful distance measure, it is helpful to consider slight variations of it to enable higher-level data mining algorithms. Note that in every case, we can still use the lower bound technique to speed up the high-level data mining algorithms. Below we consider three useful variants, and in the next section we empirically evaluate them.

Query-by-Content: In the simple examples we have considered thus far, we have implicitly assumed that the number of edge points in Q and C was the same. While *MUE* is surprisingly robust to small deviations from this assumption (say, less than a factor of two differences) it is clear that it has a bias. In particular, images that have relatively numerous edge points simply tend to be somewhat similar to everything. Since any large collection of images will invariably contain a few of these “rich” images, they can distort the results of any nearest neighbor searches. To mitigate this problem we define the nearest neighbor distance from Q to C as:

$$D_{nn}(Q, C) = \begin{cases} \frac{1}{N_Q - MUE(Q, C)} \sqrt{N_C / N_Q} & \text{if } N_C > N_Q \\ \frac{1}{N_Q - MUE(Q, C)} & \text{otherwise} \end{cases}$$

Note that we do not use *MUE* directly, but the inverse of “ $N_Q - MUE$ ” (i.e. *maximal matched edge points*). The term $\sqrt{N_C / N_Q}$ is an explicit penalty for the problem $N_C \gg N_Q$. Note that we can still use the lower bound of *MUE* to lower bound $D_{nn}(Q, C)$.

Clustering: The D_{nn} measure is perfect for similarity searching, which requires *one-to-all* matching. However, clustering requires *all-to-all* matching. In this case, with all things being equal, the D_{nn} measure would be biased into claiming that two images with many edge points are more similar than two images with few edge points. We can use $D_{\text{clustering}}(Q, C)$ to compensate for this:

$$D_{\text{clustering}}(Q, C) = \sqrt{N_Q \times N_C} \times [D_{nn}(Q, C) + D_{nn}(C, Q)]$$

Finding Motifs: Many data mining algorithms explicitly require a distance measure that obeys the triangular inequality. As a concrete example, we recently introduced an efficient and exact algorithm for finding motifs (approximately repeated patterns) (Mueen et al. 2009), which makes no assumptions about the data or distance measure, other than the triangular inequality. We can modify *MUE* to obtain such a distance with:

$$D_{\text{motifs}}(Q, C) = (N_Q + N_C) / 2 - (N_Q - MUE(Q, C))$$

Proof that D_{motifs} obeys the triangular inequality

Here we introduce a new notation QC_m , which is the set containing the maximal number of edge points that match between Q and C , and $|QC_m|$ is the size of the set.

Now we can rewrite the motif distance measure as:

$$D_{\text{motifs}}(Q, C) = (N_Q + N_C)/2 - |QC_m|$$

Given any three images A , B and C , we're going to prove:

$$D_{\text{motifs}}(A, B) + D_{\text{motifs}}(B, C) \geq D_{\text{motifs}}(A, C)$$

By the distance definition we have:

$$\begin{aligned} [(N_A + N_B)/2 - |ABm|] + [(N_B + N_C)/2 \\ - |BCm|] \geq [(N_A + N_C)/2 - |ACm|] \end{aligned}$$

That is:

$$N_B \geq |ABm| + |BCm| - |ACm| \quad (*)$$

To prove this, we use the following two lemmas.

Lemma 1

$$|ABm \cap BCm| \leq |ACm|$$

The proof is trivial, because:

$$ABm \cap BCm \subseteq A, \quad ABm \cap BCm \subseteq C$$

And AC_m contains maximal matched edge points between A and C , so:

$$ABm \cap BCm \subseteq ACm$$

Then we obtain:

$$|ABm \cap BCm| \leq |ACm|$$

Lemma 2 $|ABm \cap BCm| \geq |ABm| + |BCm| - NB$

$$\begin{aligned} |ABm \cap BCm| &= NB - |\overline{ABm \cap BCm}| \\ &= NB - |\overline{ABm} \cup \overline{BCm}| \\ &\geq NB - |\overline{ABm} + \overline{BCm}| \\ &\geq NB - [(NB - |ABm|) + (NB - |BCm|)] \\ &\geq |ABm| + |BCm| - NB \end{aligned}$$

Combine Lemma 1 and 2, we can easily obtain the inequality (*).

4 Experimental results

We have designed all experiments such that they are not only reproducible, but *easily* reproducible. To this end, we have built a webpage (Zhu 2009) which contains all datasets and code used in this work, together with spreadsheets which contain the raw numbers displayed in all of the figures. The webpage also contains many additional experiments which we did not include for brevity; however, we note that this paper is completely self-contained. All of the experiments are performed on a computer with an Intel i7-920 processor and 6.0GB of DDR3 memory.

4.1 Evaluation of utility

We begin with simple sanity checks. We took a collection of petroglyphs from the Southwest USA and extracted fourteen images that would reasonably be grouped into seven pairs. Figure 13 shows the clustering obtained by our distance measure.

Not only does the measure correctly group the seven pairs, but the higher level structure of the dendrogram correctly groups the images into Bighorn Sheep/Anthropomorphs/Atlatls.² Note that due to the thinning preprocessing step, the measure seems invariant to the hollow/solid nature of the Atlatls.

In the 1920's Dr. Stephen Chauvet noticed that many of the petroglyphs discovered on Easter Island showed humans in poses very similar to petroglyphs created by the Harappa culture (in what is now modern-day Pakistan). He noted these similarities in his 1935 text (Chauvet 1935), which inspired a flurry of speculation about the origin of the Easter Island peoples.³ It is natural to ask if our proposed distance measure could have “noticed” this similarity. This is a *very* difficult challenge for a distance measure, because the Harappa culture used stick-figures, whereas the Easter Island petroglyphs used highly stylized outlines. Nevertheless, as we can see in Fig. 14, our method can capture the intuitive similarities.

4.2 Finding motifs in large collections of petroglyphs

We tested whether our distance measure could potentially find meaningful motifs. We first scanned pages 33–132 (81–87 were skipped) of the book (Smith and Turner 1975) which contains collections of Indian rock art in southern California, then cut them into 2,852 images to make one petroglyph per image.

With 2,852 images in the database, there are 4,065,526 possible pairs that *might* be motifs. Figure 15 shows a histogram of all pair-wise distances of these pairs. The

² An Atlatl is a spear-throwing device.

³ DNA analyses now show that this speculation was wrong; the Easter Island people are descended from Polynesians.

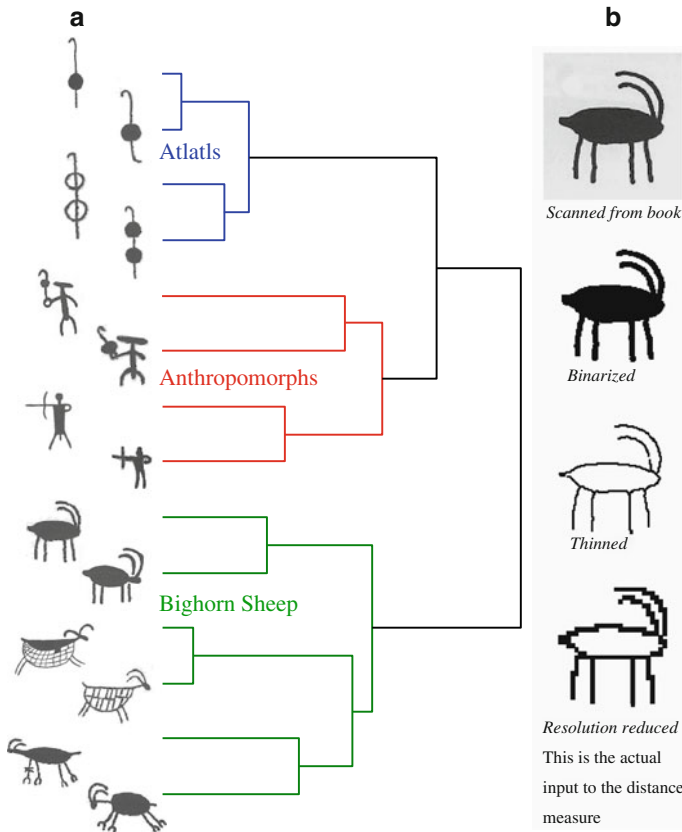
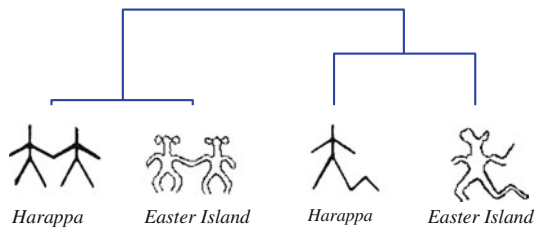


Fig. 13 a: A group-average linkage hierarchical clustering of typical Southwestern USA petroglyphs, with the $D_{\text{clustering}}$ measure. b: While the dendrogram to the left shows the full resolution images for clarity, the images input to the distance measure have been binarized, thinned and scaled to fit into a 30 by 30 bounding rectangle

Fig. 14 The GHT distance is able to find the intuitive similarity between pairs of anthropomorphic figures, in spite of the different styles of representations



mean distance between two objects is approximately 120, and we can see visually that we can expect few if any pairs to be closer than 40, a value we denote the *motif cutoff*.

In Fig. 15 we also show five representative examples of motif pairs which have a distance to each other of less than the cutoff. A total of 52 pairs, or 0.00128% of the possible candidates, passed this test. Note that these motifs are subjectively very similar. Furthermore, some of the examples show useful invariances. For example,

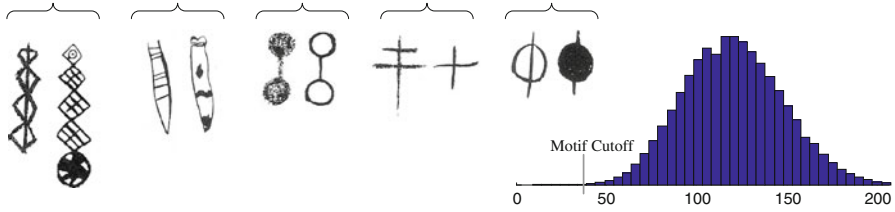


Fig. 15 (left) Five representative motif pairs from the top 52 motifs. (right) Histogram of the pair-wise distance D_{motifs} in 2,852 petroglyphs

some are invariant to solid/filled boundaries and the “cross” example is invariant to an extra crossbeam.

More interestingly, some of these motifs are in fact known (or at least, heavily speculated) to be true meaningful motifs. For example, Patterson and others have claimed that the dumbbell shape was used by both Hopi and Navajo tribes to indicate “a conference took place here” (1992).

Because of the triangular inequality of D_{motifs} , we can combine it with the algorithm recently published in Mueen et al. (2009) to efficiently find a pair of images whose distance is the smallest in a given dataset. For this petroglyph dataset, when using brute force, it takes nearly 14 h to discover the top motifs, but it takes less than 18 min with our algorithm. Given that it can take months or years to collect a significant number of petroglyphs, it might be argued that there is little effective difference between minutes and hours. However, we are currently building an interactive tool for anthropologists that will allow queries such as “find all motifs within 10 miles of this location” and “find all motifs within 100 miles of this location, with tentative dates post 1100 AD”. If we can support answering such queries in tens of minutes, a researcher can interact with the system and frame/modify hypotheses in a single session. A system that takes tens of hours would be frustratingly slow to use. Furthermore, as we will show in Sect. 4.4, brute-force motif discovery in very large datasets (containing more than one million objects) is truly intractable.

4.3 Evaluation of accuracy

Beyond the 2,852 petroglyphs considered in the last section, there are currently no large collections of objectively labeled petroglyphs (although we will plan to use the human computation tool discussed in Sect. 2.1.1 to create a public database of one million petroglyphs by 2011). Therefore, in this section we will test four publicly available datasets that are very similar to (at least some kinds of) petroglyphs. With these experiments we intend to show:

- Competitive or superior accuracy for query-by-content compared to some state-of-the-art algorithms.
- Relative insensitivity to the amount of downsampling, and that we can learn a good value from training that. These two facts show that our method does not rely on careful parameter tuning.

- As claimed in Fig. 6, very high resolution imagery *hinders* rather than *helps* accuracy.

The first dataset is the NicIcon dataset (Niels et al. 2008), which contains 24,441 images from the 14 categories shown in Fig. 16. Thirty-three participants were asked to sketch these icons in different sizes (small, medium and large) and a digital tablet was used to record the data (spatial, time and pressure coordinates). Note that contrary to the original intention for the data and subsequent algorithms, our algorithm *only* considers the shape (offline data), and completely ignores pen speed and pressure information (online data).

We did both writer dependent (WD) and writer independent (WI) tests, in both cases, randomly choosing 60% of the data as the training set and the rest as the testing set, the same division as used the original paper (Niels et al. 2008).

The original offline data is in 234×234 pixels, and keep in mind that we need to first set the amount of downsampling (the only user-specified parameter of our algorithm). Using the training data alone, we tested nine resolutions from 5×5 to 80×80 for both WD and WI tests by the simple one-nearest-neighbor classifier. Figure 17 shows the results.

This plot suggests that the sampling rate is not critical. The error rate only increased significantly when the resolution was reduced to 5×5 , which is clearly highly under-sampled for any non-trivial dataset.

We then repeated the experiment on the whole dataset with the downsampling size of 20×20 pixels, which achieved the best error rate in our previous test, and obtained an error rate 4.78% for WD and 8.46% for WI. The dataset creators tested on the online data using three classifiers (Niels et al. 2008): the multilayered perceptron, the linear multi-class SVM classifier and a Dynamic Time Warping Based (DTWB) algorithm.

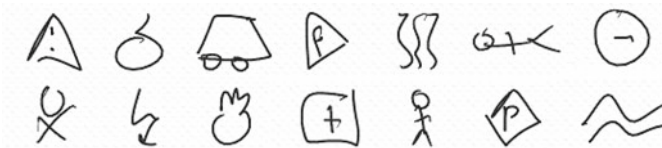


Fig. 16 Examples of 14 categories from NicIcon dataset

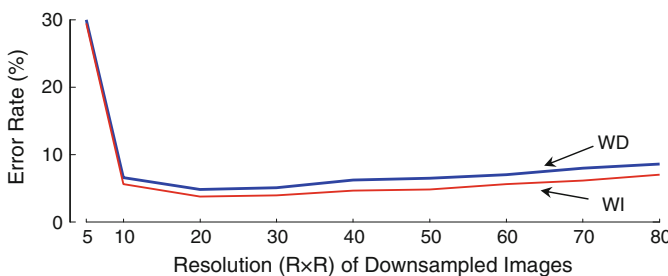


Fig. 17 Error rate versus Resolution. Downsampling test on training set of the WD and WI tests in nine resolutions. Error rate makes little difference once the resolution is larger than 10×10 . Note WI is better than WD here, because the training set of WI contains only 60% of writers while that of WD contains all

The reported error rate for WD varies from 1.94% to 15.61% and 5.3% to 20.01% for WI. Only the DTWB is better than our method, but recall that the DTWB had access to information about the pen speed, pen pressure, and the direction in which the lines were drawn, all of which were unknown to our algorithm. While the original authors do not measure time for classification, each comparison with the DTWB measure requires DTW calculations to be performed a number of times which are quadratic in the number of line strokes (i.e., the number of pen-ups) in each image, which is clearly very expensive.

We also tested without any downsampling, and the error rate increased dramatically: 31.75% for WD and 35.75% for WI, even worse than the ultra-low resolution of 5×5 . This verifies our analysis in Sect. 2.2.

Another petroglyph-like dataset is introduced by Khosravi and Kabir (2007). This is a very large dataset of handwritten Farsi digits extracted from about 11,942 registration forms. They obtained 102,352 binary images of Farsi digits, and chose 60,000 for training and 20,000 for testing (see samples in Fig. 18).

The size of images in the Farsi dataset is smaller than in the NicIcon dataset: the minimum bounding rectangle (MBR) of the largest digits is 54×64 pixels. We also did a downsampling test first on the training set from 5×5 to 30×30 pixels, using a one-nearest-neighbor classifier. The results are shown in Fig. 19.

With the resolution of 20×20 (the best one in the downsampling test), we obtained an error rate of 4.54% for the whole dataset. Borji et al. (2008) performed extensive empirical tests on this dataset, testing multiple algorithms, 3-NN, ANN, SVM_{polynomial}, SVM_{linear} and SVM_{rbf}, each with four parameter choices (two choices of filters *times* two numbers of orientations). Of the twenty reported error rates, the mean was 8.69%, and only four combinations beat our approach, with a best performance of 2.36%.

Fig. 18 Sample digits from Farsi dataset. *Note:* numbers 2, 3 and 4 are very similar (3 and 4 in the third row are even impossible for humans to distinguish); some digits have different styles (4 and 6); some digits are in bad quality (7, 8 and 9 in the third row)

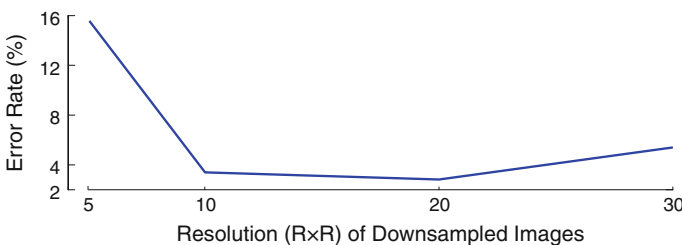
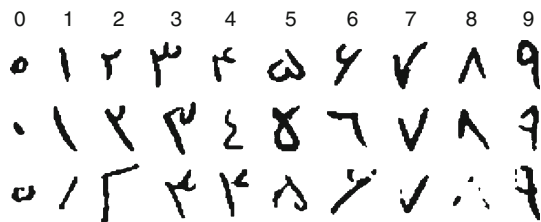


Fig. 19 Error rate versus Resolution. Downsampling test on the training set of Farsi dataset in four resolutions. Note that the error rate varies little when the resolution is greater than 10×10

However, it is important to note that in addition to the two explicit parameter choices, there are at least four other parameters set “in the background” here.

We performed the third test on an old handwritten musical symbol dataset (Fornés et al. 2008). These symbols were extracted from a collection of old music scores (17th–19th centuries). Figure 20 shows samples of 2128 clefs from 24 different authors, illustrating the large deformations and variations among different writing styles.

The authors of (Fornés et al. 2008) first extracted some features from each symbol and used a Dynamic Time Warping Based (DTWB) algorithm as the classifier. To achieve the rotation invariance, features for all considered orientations need to be extracted and compared. Because such a DTW measure is time consuming, they chose one representative (the data sample with the minimum mean distance to the rest of the samples from the same class) for each class (two for Alto clefs due to their huge variability as shown in the *third* row of Fig. 20), so only four comparisons were required to determine the class for one input symbol.

We repeated the test in the same way, and this time we skipped the step of *setting the downsampling size* and simply used the resolution 20×20 (which achieved the best results in the previous two datasets). Four representative symbols for three classes obtained by the GHT measure are shown in Fig. 21.

In Table 3, we compare the classification accuracy of four distance measures. Zernike moments (number of moments = 7) and ART descriptors (radial order = 2, angular

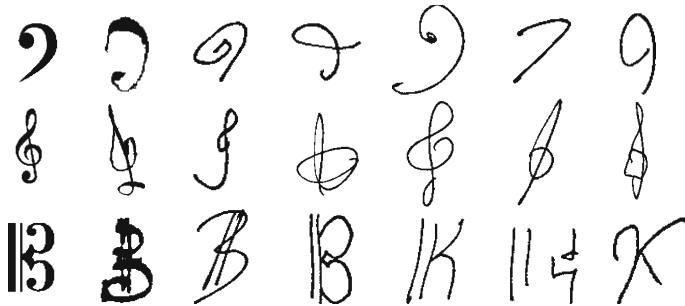


Fig. 20 Sample clef symbols from the old handwritten musical symbol dataset. *First row* printed Bass clef followed by six samples; *second row* printed Treble clef followed by six samples; *third row* printed Alto clef followed by six samples, in which the last three are very different from the first three

Fig. 21 Representative clefs for Bass, Treble and Alto (the last two)



Table 3 Comparison of the classification accuracy on 2128 clef symbols

Method	Zernike moment	ART	DTWB	GHT
Accuracy (%)	65.07	72.74	95.81	87.12

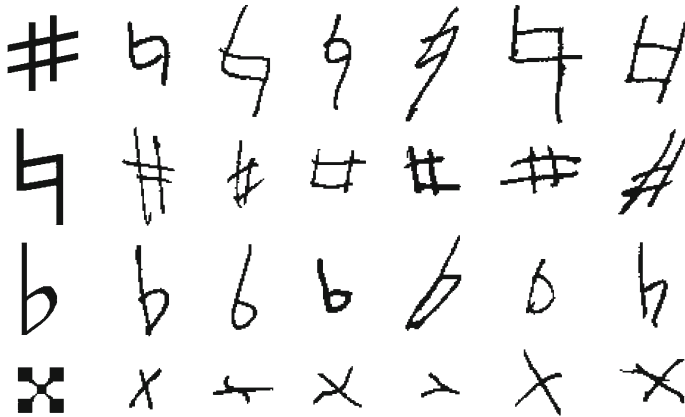


Fig. 22 Sample symbols from 1970 accidentals. In each row, a printed accidental symbol is followed by six samples in the corresponding class. Four classes are Sharp, Natural, Flat, Double Sharp (from *top* to *bottom*)

Table 4 Comparison of the classification accuracy on 2128 clefs and 1970 accidentals

Method	Zernike moment	ART	DTWB	GHT
Accuracy (%)	43.97	52.26	89.55	62.29

order = 11) are also from (Fornés et al. 2008), both of which are robust and invariant to scale and rotation.

Our GHT distance measure is competitive and only worse than the DTWB method, but we should note that we didn't do any parameter tuning, while the number of features may be an important parameter to the accuracy of DTWB. In addition to this, with the tight lower bound we have on the GHT distance, it is fast enough to do a classic leave-one-out one-nearest-neighbor classification and a 99.58% accuracy was achieved.

An extension of this experiment is to add 1970 accidentals in four classes, which were drawn by eight different authors (shown in Fig. 22). One may find some sharps and naturals are very similar and can be easily misclassified.

In this test, we have eight representatives (four clefs and four accidentals) for seven classes, and eight comparisons are required to determine the class. The classification accuracy is compared in Table 4.

Again, our method is only worse than DTWB. When we tested by the leave-one-out 1NN classifier, the accuracy was increased to 98.49% with an acceptable increment in time.

The final dataset we tested is an architectural symbol dataset (Sanchez et al. 2004) consisting of 7414 samples in 50 classes (the printed version is shown in Fig. 23) by 21 users. Each user drew 25 symbols and over 11 samples per symbol.

Mas et al. (2005, 2006) (the dataset creators) tried syntactic approaches to recognize these symbols by an adjacency grammar, in which each symbol was described in terms of the primitives that form it and the relations among these primitives. The test was

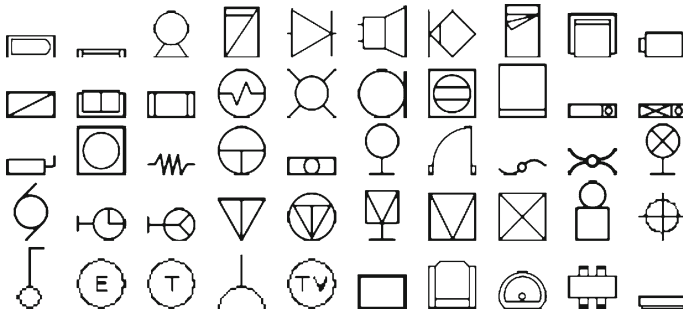


Fig. 23 Fifty classes (printed version) of architectural symbol dataset

Table 5 Comparison of the classification accuracy on architectural symbol dataset

Method	Zernike moment	DTW-cyclic	DTW-full	GHT
Accuracy on first 5 classes (%)	62	61	97	86.45
Accuracy on all 50 classes (%)	26	38	87	53.33

performed on a subset of ten symbols, and an accuracy of 87.7% was obtained in (Mas et al. 2005). Then they further constructed an incremental parser in (Mas et al. 2006) to improve the performance to 91.78%. However, its main drawback is the complexity. The time complexity of the incremental parser is $O(n^3)$, where n is the larger number between the productions of the grammar and the number of primitives entered into the system. The performance of the parser also relies heavily on the specification of the grammatical rules, and the set of constraints that has been defined. In comparison, our efficiently computed GHT measure doesn't depend on the number of classes or the relation between them, and it achieved a much better result of 99.29%.

The only experiment we found on the whole architectural symbol dataset was done by Fornés (2009). She compared the classification accuracy on the first five classes and all 50 classes by three methods, one of which (DTW-full) is the same as the one performed on the old handwritten musical symbol dataset we have discussed above. We repeated both tests by the GHT measure, again without any parameter tuning. All results are shown in Table 5.

The DTW-cyclic is a variation of DTW-full, in which only one feature vector is extracted from each orientation and so the computational cost is reduced. Our method is only worse than DTW-full, and the lower time complexity enables a leave-one-out 1NN classifier. When this was used, we achieved 100% on the first five classes and 96.18% on all 50 classes.

Having shown that low resolution images can produce high accuracy in our domain and the relative insensitivity of the downsampling size, we have fixed the resolution to 30×30 pixels in all remaining experiments in this paper.

4.4 Evaluation of speed and scalability

As noted in Sect. 2, while we currently have only thousands of petroglyphs, we expect to shortly have on the order of a million. Therefore, we tested our algorithm dataset

containing more than one million objects. To make this possible, we made our own synthetic petroglyph dataset. We first obtained the twenty-two petroglyphs (samples are shown in the top row of Fig. 24). Then ten volunteers were asked to duplicate the petroglyphs by drawing them with an HP pavilion tx2510us tablet PC. A total of 250 petroglyphs were created in this way as our basic dataset (samples are shown in the second row of Fig. 24). We then applied a random second-order *Polynomial Transformation* to each image in the basic dataset to make [39 79 159 319 639 1,279 2,559 5,119] distorted copies of each (as shown in the third row of Fig. 24). With this basic dataset, we finally created eight datasets from the size of 10,000 to 1,280,000.

We first did a leave-one-out one-nearest-neighbor test. For each dataset, we randomly picked an image as the testing sample, removed it from the dataset and found its nearest neighbor using our lower bound based algorithm. We repeated this process ten times; Fig. 25 shows the result.

We can see that the range between the maximal and minimal time is relatively small. When viewed on a normal scale plot (see Zhu (2009)), we can see that the average running time is linear to the size of the dataset. While this is a test of *scalability*, we note in passing that the *accuracy* of this 22-class problem is 100% for all experiments.

It is natural to ask how much of the effectiveness of the search can be attributed to our lower bound. We measured the pruning rate as

$$\text{pruning rate} = 1 - \frac{\text{number of GHT calculations, lower bound search}}{\text{number of GHT calculations, brute force search}}$$

for each of the 10 runs; the result is shown in Fig. 26.

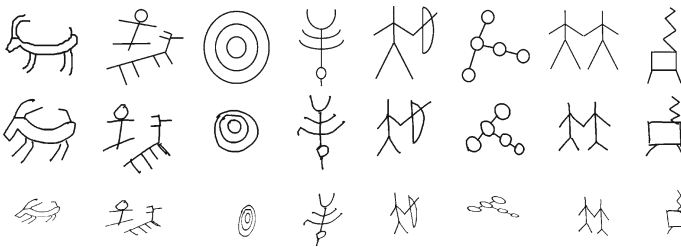


Fig. 24 The synthetic petroglyphs dataset. *First row* samples of petroglyph templates; *second row* sample petroglyphs of the basic “human-copied” dataset; *third row* samples of distorted petroglyphs. Note that for each template, we have copies in different scales, translations, orientations and non-linear distortions

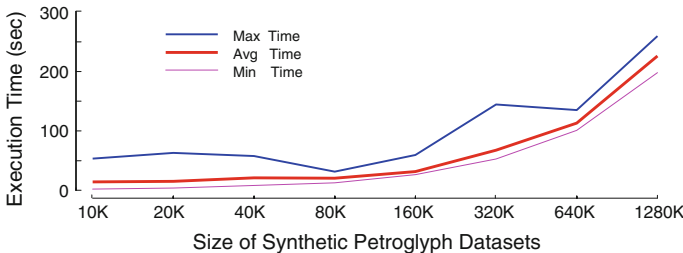


Fig. 25 Time taken for the 1-NN query on eight synthetic petroglyph datasets. For each dataset, the maximal, average and minimal time of ten runs are reported. Note log scale is used in x axis

The results show that the pruning is extremely effective, particularly for larger datasets. The average prune rate exceeds 99.0% when examining 80,000 objects, and even the *minimal* prune rate is more than 96.9% at that point.

We then did the same experiment using the brute force algorithm. Figure 27 compares the percentage of execution time for our lower bound algorithm relative to the brute force algorithm. Notice that for the largest dataset, our lower bound time is only 2% of the brute force one.

In addition to query-by-content, we also tested our ability to find motifs. Compared to the tests in Sect. 4.2, our synthetic petroglyph datasets are much larger. Figure 28 shows the running time of finding motifs in these datasets.

A brute force algorithm to find motifs requires time quadratic in the size of the dataset. But from a normal scale plot (see Zhu (2009)), we find that our algorithm scales linearly. This is because we only need to calculate a tiny fraction of the exact distance between two images: even for the smallest dataset with 10,000 objects, we can prune 99.84% of the calculations, and by the time we consider 1,280,000 images, we can prune more than 99.99% of the calculations. In Fig. 29 we show the explicit speed-up over the brute force search. Even for the smallest dataset, our algorithm is 712 times faster and by the time we see the largest dataset, our algorithm is more than 100,000 times faster.

While these results show that we can make the otherwise intractable task of finding motifs in large datasets tenable, they do not consider the *effectiveness*. Normally motif discovery cannot be evaluated directly in terms of accuracy, since we assume unlabeled

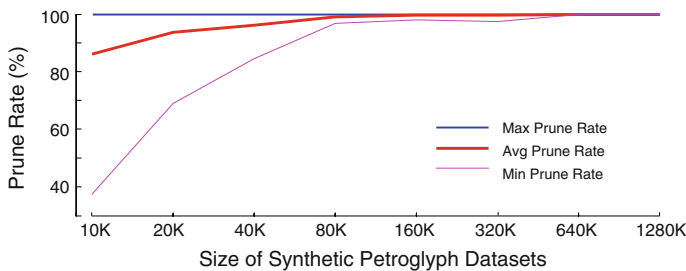


Fig. 26 Pruning rate of our lower bound algorithm on eight synthetic petroglyph datasets. For each dataset, maximal, average and minimal rates are reported. Note log scale is used in x axis

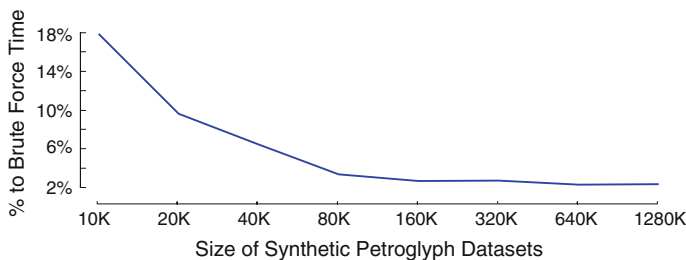


Fig. 27 Percentage of execution time for our lower bound algorithm relative to the brute force algorithm. Note log scale is used in x axis

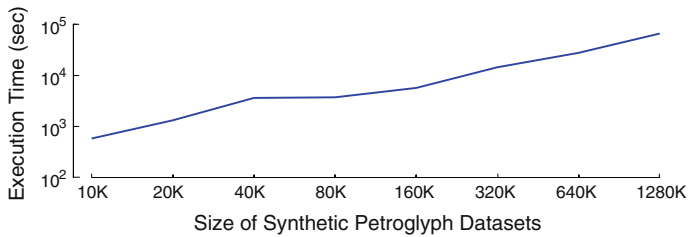


Fig. 28 Time of finding motifs in eight synthetic petroglyph datasets. Note log–log scale

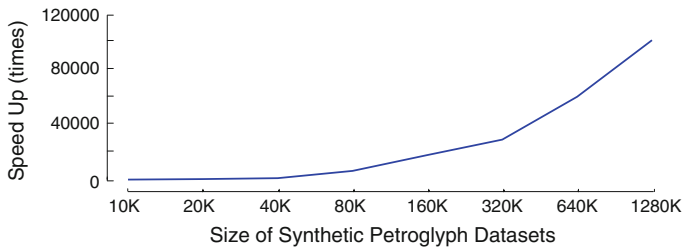


Fig. 29 Speed-up of our lower bound algorithm against brute force algorithm of finding motifs in increasingly large petroglyph datasets. For the brute force algorithm, we only ran it on the 10,000 datasets and extrapolated other values. Note log scale is used in x axis

data. However, since we actually know the labels in this case, we can measure the accuracy. For example, when testing the dataset with 80,000 petroglyph images (from 22 classes) over 100 runs on random sets of 80,000 objects (taken from a pool of 1280 K), we found that on 99 occasions the labels agreed.

In Sect. 5.4, we will further investigate the utility of our distance measure for supporting the similarity search in rock art databases.

5 Obtaining massive quantities of rock art data

The results in the previous section hint at the fact that if we could only extract the “skeletonized” rock art data, a wealth of opportunities for anthropological data mining would open up. However, as we shall demonstrate, and as the reader may already be convinced of, in the vast majority of cases, the extraction of meaningful data from photographs of rock art is likely to be beyond the capabilities of image segmentation algorithms for a long time. With this in mind, we proposed to extract useful information from unconstrained images of rock art by turning the problem into a CAPTCHA (von Ahn et al. 2003). CAPTCHAs are tests given by a machine to ensure that a response is generated by a human, not a computer. The most familiar instantiation of them is a sequence of distorted letters that the user must reproduce. Figure 30 shows two examples.

These CAPTCHAs operate on inherently discrete data (*text*, albeit distorted) and expect discrete responses (keystrokes); we can therefore use equality tests to decide if the test was passed, i.e. equals, (‘28iVW’, ‘28iVW’)?

In contrast, our method will consider inherently real-valued data (photographs of rock art) and expects real-valued responses (mouse movements). We cannot expect to test for equality. This creates a significant challenge which we have overcome by using the distance measure introduced in Sect. 3.5 to test if a tracing of a petroglyph is close enough to a real pattern to indicate human intelligence. We begin by demonstrating that we cannot solve the extraction problem with current algorithms.

5.1 Do petroglyphs allow automatic feature extraction?

A fundamental assumption that motivates the introduction of CAPTCHAs for annotating petroglyphs is that there is no automatic segmentation algorithm that can robustly segment rock art. To demonstrate this, we conducted a simple experiment on what is probably one of the most amiable images imaginable, the famous petroglyphs of Alta, Norway.

We took one image of a reindeer as shown in Fig. 31a, and tried segmenting it with six different methods: the Sobel method, Prewitt method, Roberts method, Laplacian of Gaussian method, Zero-cross method and Canny method (Ziou and Tabbone 1998). In each case we spent fifteen minutes adjusting the parameters to achieve the best (subjectively) feature extraction. The best result, using the Prewitt method is shown in Fig. 31b.

Note that while our efforts have paid off in that we have captured much of the animal in question, we are missing a large section of the rump. What is worse, we have many spurious lines corresponding to cracks in the rock. Of course, it is possible that a more sophisticated algorithm could be tuned to do a better job; however, this tuned version is unlikely to generalize to other petroglyphs. Furthermore, it is worth

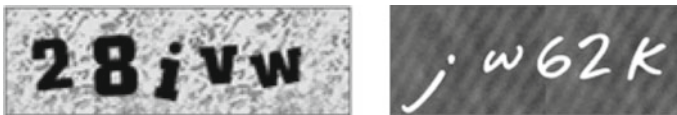


Fig. 30 Two examples of CAPTCHAs. In order to solve the CAPTCHA and get access to the next webpage (in this case, offering a free email account) the user must type in 28iVW and jw62K respectively

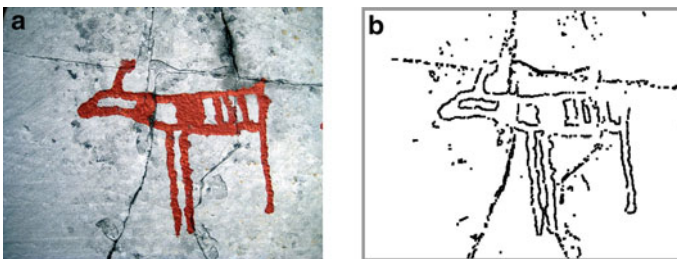


Fig. 31 **a** Reindeer rock art from Norway, dating to 4200–500BC. The rock carvings have been retouched in bright red by researchers, making them extremely high contrast. **b** a segmentation of the image using the Prewitt method, carefully tuned

restating that this example is among the highest contrast, cleanest examples of rock art.

5.2 CAPTCHA-ROCK

While our ultimate goal is to introduce a method that will allow us to capture data from real photographs of petroglyphs, for ease of exposition we will begin discussing the problem as if our only intention were to produce an image-based CAPTCHA with *artificial* data.

5.2.1 A simple image-based Stickman CAPTCHA

It is simple to write a program to produce random instances of a “stick figure”; Figure 32 shows four examples:

To ensure each stickman is unique (with very high probability), we have parameterized the code. The following features are parameterized:

- The head size and aspect ratio
- The length of the humerus, forearm, femur, tibia and foot
- The angles of knees, elbows, ankle and torso (these may be asymmetric)

There are other elements of a human stick figure that we could represent and parameterize, but this simple model is sufficient for our purposes.

For reasons that we will see shortly, it is useful to ask what the average distance is between two randomly created figures under the GHT distance measure discussed in the previous section. To calculate this, we generated 1,000 pairs of stickmen and calculated the distance between each pair, summarizing the results in a histogram in Fig. 33.

If we instead produce random stickmen, and ask humans to *trace* their outline on the screen with the mouse pointer (as in Fig. 34), we might expect the distances between the generated and traced outlines to be generally smaller.

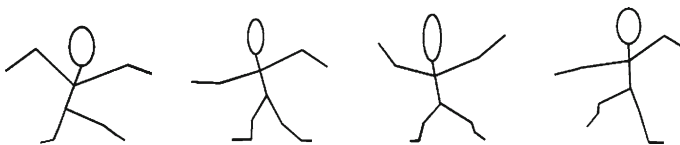


Fig. 32 Four examples of a parameterized Stickman

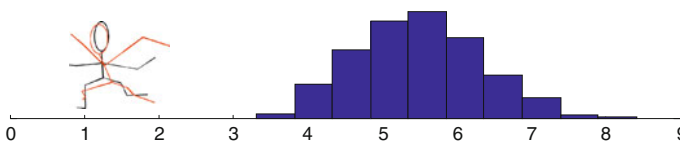


Fig. 33 (left) A pair of randomly generated stickmen. (right) The distribution of GHT distances between 1,000 pairs of randomly generated stickmen

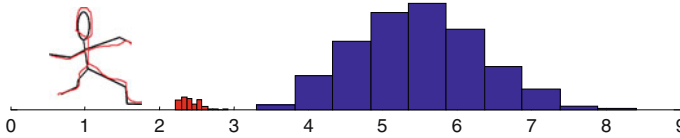


Fig. 34 (left) A randomly generated stickman in *black* and a human tracing of it in *red*. (right) The distribution of GHT distances between randomly generated stickmen and human tracings of them are shown with a finer bucket size (in *red*), because there is less data. The distribution of GHT distances between two randomly generated stickmen is shown for context (in *blue*)

To verify this, we generated 20 stickmen and asked volunteers to trace them. How well a person can trace the stickmen depends on their dexterity, input device, screen size, etc. Given these variations, we asked three volunteers to trace each stickman on their own personal machines. Figure 34 also shows the distribution of these distances.

It is easy to see that we could use these results to create a simple stickman CAPTCHA. We could produce a stickman, and ask the user to trace it. If a human traces the stickman, we can be near certain that the distance to the template will be less than 3 (from Fig. 33). For simplicity, here we assume that the attacker has the code to produce the stickmen, and simply sends a random stickman as his guess. If that is so, his guess will almost certainly be greater than 3 (from Fig. 33/Fig. 34) and we can reject his attempt. Of course the attacker could use an image processing algorithm to produce a “customized guess”, and we could counter by imbedding the stickman in a field of distracters and distortions; however, as we shall see, an even better idea is to find “stickmen” in rock art, the subject of the next section.

5.2.2 The CAPTCHA-ROCK system: extracting data from petroglyphs

Motivated by our experiences with the *stickmen* CAPTCHA, we can now ask: is it possible to design a CAPTCHA system which provides high security (serves as a CAPTCHA) while collecting useful information about rock art (serves as a Human Computation tool)? The reCAPTCHA (2008) proposed by Luis von Ahn is the ideal paradigm to follow. In this system, which is designed to transcribe degraded text from scanned books and newspapers, each test gives the user two words to recognize, one of which is a “control word”, whose answer is *known*, and the other of which is an *unknown* word. If the user can correctly type the “control word”, the reCAPTCHA assumes that the inputs come from a human and the answer for the “unknown word” is correct (or at least plausible). Once an “unknown word” receives enough “votes” from a same answer, it can become a “control word”. This system has already been used to transcribe several hundred million degraded words, which OCR systems failed to parse.

We can use the same idea to build our CAPTCHA-ROCK system. An example is shown in Fig. 35, in which the user is asked to trace both petroglyphs correctly to pass the CAPTCHA.



Fig. 35 A CAPTCHA-ROCK consists of two rock art images: one control image and one unknown image. Note that users do not know which one is which

There are still three extra problems/questions we must solve to make this work:

- How should we build the initial “control image” set?

To frustrate robots that break challenges by simple random guesses, the set of control words in reCAPTCHA contains more than 100,000 items. Do we also need such a large “control image” set? We believe that the answer is no. We can frustrate an attacker that attempts to simply memorize the entire control image set (with solution tracings) by performing simple scale and translation operations to images. This means that even if the attacker’s algorithm correctly detects *which* image is the control image (perhaps by color), the relative location of the petroglyph within the image may have changed just enough so that even if the attacker sends the right trace, it will be in the wrong location or at the wrong scale.

As we shall show in the next section, our initial experiments show that even for a control set containing only 143 images, by using these simple scale and translation operations, CAPTCHA-ROCK has a pass rate of 0.020% for attackers, even after the attacker has been given a careful tracing for each image in the control set.

- Is one trace (i.e one “solution”) per image enough?

Assume two histograms in Fig. 34 have an overlap. To assure the low pass rate for robots, we have to set a smaller threshold. In this case a false negative occurs: some legitimate attempts by humans would be denied.

Our solution is to store multiple traces for one image. When a tentative solution is submitted by a user, we compare it to all traces and pass it if there is *at least* one distance below the threshold. Although human traces for the same image vary, by comparing to more than one “interpretation”, the possibility of finding a close enough match increases. Note that this will not affect the pass rate for robot significantly. If a random trace is far from one trace, it is also far from other traces of the same image.

Based on our experiments which will be presented in Sect. 5.3, three traces per image improve the human pass rate, without helping the attackers.

- When can we promote an “unknown image” to a “control image”?

Once we have recorded three traces for an “unknown image”, we promote it into the control set. Note that as a control image, the CAPTCHA-ROCK system will obtain more traces for the image. What should we do with these additional traces? We could

ignore them, we could add them to the original three traces, or we could temporarily merge the new offering with the other three, expunging the one that has the furthest average distance from the others. We leave these considerations for future work.

5.3 Quantitative evaluation of the CAPTCHA-ROCK system

In this section, we intend to show that:

- Our CAPTCHA-ROCK is very easy for humans to solve (they have a high pass rate) and hard for robots (they have a low pass rate).
- Storing multiple traces for each image helps increase the pass rate for humans, while not affecting the pass rate for robots significantly.
- A small “control image” set is sufficient, at least to bootstrap the system.

We randomly chose 143 images from our rock art image database, and had four volunteers draw traces for each image. The volunteers worked completely independently of each other. We call this initial trace dataset *Trace_ini*.

For reasons we will see shortly, we performed two rounds of rescaling and 2-dimensional translation to each trace in *Trace_ini*. In the first round, we rescaled each trace image to 10–50% of its original size, and translated it in 2 dimensions by plus/minus 0–3 times of the size in the X and Y axis independently. We call this new dataset *Trace_robot*. In the second round, every 4 traces of the same image in *Trace_ini* were performed by a same rescale and transition, and this dataset is called *Trace_human*.

As we assume that there is no automatic algorithm for extracting rock art data, we need to come up with an attack model. We make the pessimistic assumption that the attacker has our entire 143 image database, together with a human trace for each image. These seems to suggest that if the attacker simply submits a random tracing, he would have a one in 143 chance of passing the test, but recall that the images have been rescaled/translated before being presented as a CAPTCHA. Such distortion means that even if the attacker happens to send the correct trace, it will probably not line up with the stored template, and will fail the test.

We first tested the system with one randomly chosen (of four possibilities) trace from *Trace_human* for the “control image”. To model the attacks from robots, each time we picked one trace from *Trace_robot* (but not those from the same person of the challenge trace). There were thus 3×143 tries for each challenge. As the human input, we picked the other three traces of the “control image” in *Trace_human*.

Using a threshold of seven, only 41 of 245,388 robot tests could pass, a pass rate of 0.014%; whereas 1,632 of 1,716 human tests passed, a pass rate of 94.99%.

Then we tested with three traces for each image. As noted in Sect. 5.2.2, each input from the user was compared to three traces of the “control image”, and if its distance to *any one* of them is below the threshold, the user passed the test. Each time we picked three traces of a same image from *Trace_human* as traces of the “control image”. To model the attacks from robots, each time we picked one trace from the fourth person in *Trace_robot*. Thus, there were 143 tries for each challenge. As the human input we picked the remaining trace of the “control image” in *Trace_human*.

Using the same threshold of 7, only 22 of 81,796 robot tests could pass, a pass rate of 0.020%; while only 5 of 572 human tests could not pass, a pass rate of 99.13%.

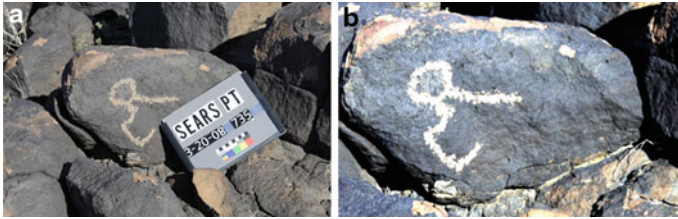


Fig. 36 **a** A petroglyph from Utah that has been indexed in our database. **b** An image of the same petroglyph found on Flickr.com. Could this image be used as a query to retrieve the anthropologist’s annotated version in our database?

Although our pass rate for robots is slightly larger than the generally accepted figure of 0.01% (Chellapilla et al. 2005), note that all results are based on the initial control image set, with only 143 images. We expect the robot rate to decrease with more data, while the human rate should stay almost constant. Further recall that these results assume the pessimistic and unrealistic assumption that the attacker has traces for the entire database.

5.4 Supporting similarity search

The major goal of this work is to produce a dataset that will enable research by anthropologists. However, a minor goal is to produce a tool for non-specialists to query a database of petroglyphs. This tool could be used to support tourism Dickinson, and to encourage an appreciation of indigenous people’s cultural achievements.

We envision the following scenario: *A hiker on a trail spots a petroglyph, and wants to know if it is known, and if so, what anthropologists and/or tribal historians have said about it. She photographs the petroglyph on her iPhone, traces the outline, and submits the query...*

In order for this query to return the correct answer, our system must have several invariances. Some are trivial, as we are operating on a binary representation of the data, color and contrast invariance is automatically achieved.⁴ However, as shown in Fig. 36, there need to be at least somewhat invariant in size, angle of view, etc.

To test the feasibility of this scenario we obtained several examples of images of petroglyphs that we know are in our database (referred to the “control image” set in Sect. 5.2.2), but which were taken on a different day, by a different person, with a different camera etc. To normalize our expectations we also obtained photographs of petroglyphs that are known *not* to be in our database.

We had volunteers trace these petroglyph images. Note that in each case, these volunteers had not seen the data in the database, and were not familiar with our project.

Recall from the previous section that each petroglyph in our data collection had been traced by four independent volunteers. This means that for each petroglyph we had four models we could use to index it. We could also choose to have only one model for each petroglyph instead, by either averaging all four, or choosing the most

⁴ For faint petroglyphs changing the contrast/color balance can enhance the petroglyphs visibility (Landon and Seales 2006; Mark and Billo 2002).

typical one. However, here we kept all four models, both for simplicity, and because (as we shall see) it is instructive.

Our small dataset in this preliminary experiment does not warrant calculating precision/recall or similar statistics. Instead, we show typical results, and archive *all* results at (Zhu 2010). Figure 37 shows an example of a query using a (different photograph of) petroglyph that is in our database. The results are quite promising. Note that the query was taken from an image that was not as tightly cropped, and the user issuing the query (rightly or wrongly) traced a hook-like appendage on the left leg of the figure. Furthermore, note that among the four tracings in the database there is significant disagreement. For example, one individual did not trace the head as a circle. In spite of this, query-by-content is clearly successful in this example, as the first four matches are correct (the maximum possible).

In Fig. 38, we see two more queries for which the relevant petroglyph (traced from a different photograph) is known to be in the database. For the “wheel” the 1st, 2nd and 4th matches are correct, and the two others are at least plausible. For the bighorn sheep petroglyph, the first four matches are correct (the maximum possible), and the 5th match is also plausible.

Finally, we consider the more difficult case, queries for which we know the relevant petroglyph is *not* in the database. Here the judgment of quality is subjective. Note also

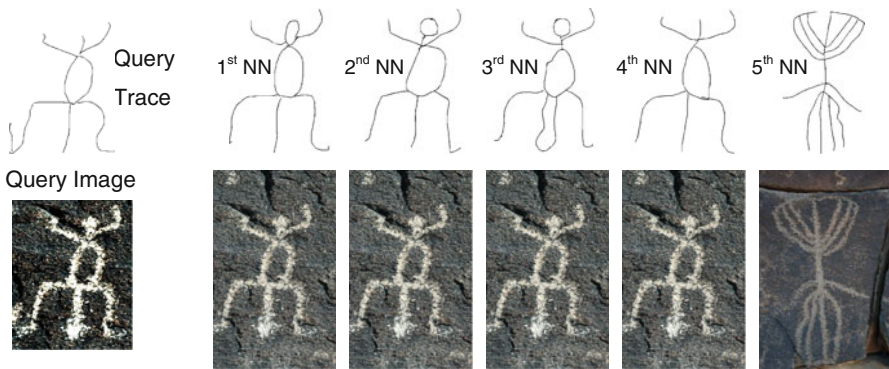


Fig. 37 (left) A query petroglyph that happens to be in our database and its tracing. (right) The five nearest neighbors to the query; the first four all refer to the same image, the correct target

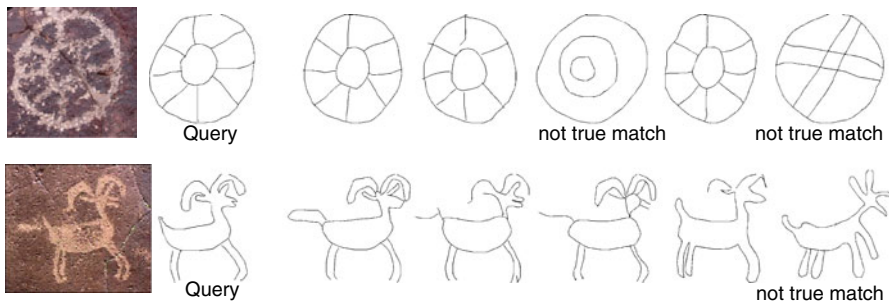


Fig. 38 An abstract (top) and animal petroglyph (bottom) which had been traced and issued as queries to our database. The list of the five nearest neighbors to each are shown left to right

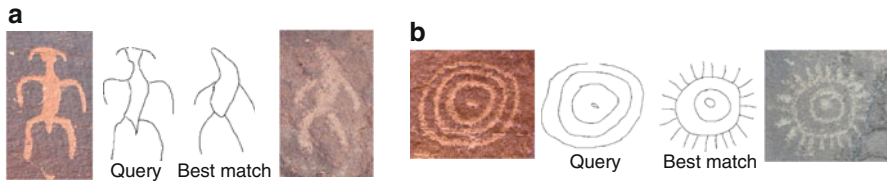


Fig. 39 **a** An anthropomorph is used as a query, and it retrieves another stylized human figure with similar limbs, **b** a petroglyph of concentric circles retrieves a “sunburst”

that we might expect to do better and better at this case as the database grows larger and larger. In Fig. 39 we show two queries and their best matches. In both cases the returned answers are reasonable.

6 Conclusions and future work

In this work we consider, *for the first time*, the problem of mining large collections of rock art. We introduce an explicit framing of the GHT algorithm as a similarity measure, and show that by lower bounding the measure we can effectively mine large data archives.

We also introduce the CAPTCHA-ROCK system, which demonstrates *for the first time* a method to crowdsource the annotation and feature extraction of rock art images. We note several limitations of the CAPTCHA-ROCK system. We could become victims of our own success, given that there are *only* tens of millions of rock images in existence, but there is a need for tens of millions of CAPTCHAs per day. So even if we capture only a small fraction of the CAPTCHA market, we may run out of rock art images. We believe that we may be able to bypass this issue by generating synthetic rock art images, in a spirit similar to the stickmen shown in Sect. 5.2.1. In addition, a significant fraction of petroglyph images may not be amenable to our system. Our CAPTCHA is not usable for blind users, and our system may be difficult to use on small screens such as iPhones. However, even if our framework does not succeed as a CAPTCHA, the basic ideas work as a framework to allow volunteers to help annotate data. We tentatively estimate that if every graduate student in anthropology in the US were to donate just one hour a month to the project, all of the world’s rock art could be processed in just a few years.

For the more general problem of creating distance measures for rock art, we also note some limitations of our current work. Our method does not allow rotation invariance or support partial shape matching. Both of these issues are seen in real rock art databases, and are thus the focus of our current research efforts.

Acknowledgements This work was funded by NSF 0803410 and NSF 0808770. Field work for this project was funded by a National Geographic Society/Waitt Grant. We would like to thank the many donors of datasets, particularly Dr. Robert Mark and Evelyn Billo of www.rupestrian.com, Taryn T. Rampley of UCR and the Document Analysis Group of the Computer Vision Center in Universitat Autònoma de Barcelona.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Alt H, Godau M (1995) Computing the Fréchet distance between two polygonal curves. *Int J Comput Geom Appl* 5(1–2):75–91
- Aseyev IV (2008) Horseman image on an ostrich eggshell fragment. *Archaeol Ethnol Anthropol Eurasia* 34(2):96–99
- Assent I, Wichterich M, Meisen T, Seidl T (2008) Efficient similarity search using the earth mover's distance for large multimedia databases. In: *Proceedings of IEEE ICDE conference*, pp 307–316
- Bai X, Latecki LJ (2008) Path similarity skeleton graph matching. *IEEE Trans Pattern Anal Mach Intell* 30(7):1282–1292
- Ballard DH (1981) Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognit* 13:111–122
- Belongie S, Karanikas H, Tjortjjs C (2002) Shape matching and object recognition using shape contexts. *IEEE Trans Pattern Anal Mach Intell* 24:509–522
- Borgefors G (1998) Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Trans Pattern Anal Mach Intell* 10(6):849–865
- Borji A, Hamidi M, Mahmoudi F (2008) Robust handwritten character recognition with features inspired by visual ventral stream. *Neural Process Lett* 28(2):97–111
- Chauvet S-C (1935) *L'île de Pâques et ses Mystères* (Easter Island and its Mysteries). Éditions Tel, Paris
- Chellapilla K, Larson K, Simard P, Czerwinski M (2005) Designing human friendly human interaction proofs (HIPs). In: *Proceedings of ACM CHI*, pp 711–720
- Dickinson EA (under review) Evoking the Sacred: Commercial Appropriations of Nature in “The Petroglyphs”. *West J Commun*
- Duda RO, Hart PE (1972) Use of the Hough transform to detect lines and curves in pictures. *Commun ACM* 15(1):11–15
- Fornés A, Lladós J, Sanchez G (2008) Old handwritten musical symbol classification by a dynamic time warping based method. in *graphics recognition: recent advances and new opportunities*. In: Liu W, Lladós J, Ogier JM (eds) *Lecture notes in computer science*, vol 5046. Springer-Verlag, Berlin, Heidelberg, pp 51–60
- Fornés A (2009) *Writer identification by a combination of graphical features in the framework of old handwritten music scores*. PhD dissertation, Universitat Autònoma de Barcelona
- Grant C, Baird J, Pringle JK (1968) *Rock drawings of the Coso range*. Maturango Museum, China Lake, CA
- Henshilwood CS, d'Errico F, Yates R, Jacobs Z, Tribolo C, Duller GAT, Mercier N, Sealy JC, Valladas H, Watts I, Wintle AG (2002) Emergence of modern human behavior: middle Stone Age engravings from South Africa. *Science* 295:1278–1280
- Hough PVC (1966) Method and mean for recognizing complex pattern. US Patent 3,069,654
- Huttenlocher D, Klanderman G, Kl G, Rucklidge W (1993) Comparing images using the hausdorff distance. *IEEE Trans Pattern Anal Mach Intell* 15:850–863
- Keogh E, Wei L, Xi X, Lee SH, Vlachos M (2006) LB_Keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. In: *VLDB*
- Khosravi H, Kabir E (2007) Introducing a very large dataset of handwritten Farsi digits and a study on their varieties. *Pattern Recognit Lett* 28(10):1133–1141
- Landon GV, Seales WB (2006) Petroglyph digitization: enabling cultural heri-tage scholarship, machine vision and applications. 17(6):361–371
- Mark RK, Billo E (2002) Application of digital image enhancement in rock art recording. *Am Indian Rock Art* 28:121–128
- Mas J, Sanchez G, Lladós J (2005) An adjacency grammar to recognize symbols and gestures in a digital pen framework. *Lecture Notes in Computer Science*, vol 3523. Springer, Berlin/Heidelberg, pp 115–122
- Mas J, Sanchez G, Lladós J (2006) An incremental parser to recognize diagram symbols and gestures represented by adjacency grammars. In: Liu W, Lladós J (eds) *Graphics recognition: ten year review and perspectives*. *Lecture notes in computer science*, vol 3926. Springer-Verlag, Berlin/Heidelberg, pp 252–263
- McDonald JJ, Veth PM (2007) Pilbara and Western desert rock art: style graphics in arid landscapes. In: *Rock art in the frame of cultural heritage of humankind*. *Proceedings of the XXII Valcamonica symposium*, pp 327–334

- Merlin PM, Farber DJ (1975) A parallel mechanism for detecting curves in pictures. *IEEE Trans Comput* C24:96–98
- Muen A, Keogh E, Zhu Q, Cash S, Westover B (2009) Exact discovery of time series motifs. In: SDM
- Niels R, Willems D, Vuurpijl L (2008) The NicIcon database of handwritten icons. In: 11th international conference on frontiers in handwriting recognition (ICFHR)
- Pan J, Balan A, Xing Eric P, Traina A, Faloutsos C (2006) Automatic mining of fruit fly embryo images. In: KDD
- Patterson A (1992) Rock art symbols of the greater southwest. Johnson Books, Boulder, CO
- Pettigrew J, Nugent M, McPhee A, Wallman J (2008) An unexpected, stripe-faced flying fox in ice age rock art of Australia's Kimberley. *J Antiq*
- Powell JW (ed) (1888) Annual report of the bureau of American ethnology to the secretary of the smithsonian institution. Bureau of American Ethnology, Washington, DC
- Sanchez G, Valveny E, Llados J, Mas Romeu J, Lozano N (2004) A platform to extract knowledge from graphic documents. application to an architectural sketch understanding scenario. In: Dengel A, Marinai S (eds) Document analysis systems VI. Lecture notes in computer science, vol. 3163, Florence—Italy. Springer-Verlag, Heidelberg, pp 389/400
- Smith Gerald A, Turner Wilson G (1975) Indian rock art of southern California, with selected petroglyph catalog. San Bernardino County Museum Association
- Takaki R, Toriwaki J, Mizuno S, Izuhara R, Khudjanazarov M, Reutova M (2006) Shape analysis of petroglyphs in central Asia. *Forma* 21(3):243–258
- Teague MR (1980) Image analysis via the general theory of moments. *J Opt Soc Am* 70:920–930
- Valladas H, Clottes J, Geneste J-M, Garcia MA, Arnold M, Cachier H, Tisnérat-Laborde N (2001) Palaeolithic paintings: evolution of prehistoric cave art. *Nature* 413:479
- Veltkamp RC (2001) Shape matching: similarity measures and algorithms. In: International conference on shape modeling and applications
- von Ahn L, Blum M, Hopper N, Langford J (2003) CAPTCHA: using hard AI problems for security. *Advances in cryptology. Lecture Notes in Computer Science*, Springer, Heidelberg, pp 294–311
- von Ahn L, Maurer B, McMillen C, Abraham D, Blum M (2008) reCAPTCHA: human-based character recognition via web security measures. *Science* 321:1465–1468
- von Ahn L (2006) Games with a purpose. *Computer* 39(6):92–94
- Walt H, David B, Brayer J, Musello C (2006) The international rock art database project. www.cs.unm.edu/~brayer/rock/waltet.html
- Wolfson HJ, Rigoutsos I (1997) Geometric hashing: an overview. *IEEE Comput Sci Eng* 4(4):10–21
- Zhang D, Lu G (2004) Review of shape representation and description techniques. *Pattern Recognit* 37(1):1–19
- Zhu Q (2009) Petroglyph webpage. <http://www.cs.ucr.edu/~qzhu/petro.html>
- Zhu Q (2010) CAPTCHA rock webpage. http://www.cs.ucr.edu/~qzhu/CAPTCHA_Rock.html
- Zhu Q, Wang X, Keogh E, Lee SH (2009) Augmenting the generalized hough transform to enable the mining of petroglyphs. In: SIGKDD, pp 1057–1066
- Ziou D, Tabbone S (1998) Edge detection techniques an overview. *Int J Pattern Recognit Image Anal* 8(4):537–559