Ownership Protection of Shape Datasets with Geodesic Distance Preservation

Michail Vlachos [†] Claudio Lucchese [⋆] Deepak Rajan [†] Philip S. Yu [‡]
[†] IBM T.J. Watson Research Center
[‡] University of Illinois, Chicago
[⋆] University of Venice

ABSTRACT

Protection of one's intellectual property is a topic with important technological and legal facets. The significance of this issue is amplified nowadays due to the ease of data dissemination through the internet. Here, we provide technological mechanisms for establishing the ownership of a dataset consisting of multiple objects. The objects that we consider in this work are shapes (i.e., two dimensional contours), which abound in disciplines such as medicine, biology, anthropology and natural sciences. The protection of the dataset is achieved through means of embedding of an imperceptible ownership 'seal', that imparts only minute visual distortions. This seal needs to be embedded in the proper data space so that its removal or destruction is particularly difficult. Our technique is robust to many common transformations, such as data rotation, translation, scaling, noise addition and resampling. In addition to that, the proposed scheme also guarantees that important distances between the dataset shapes/objects are not distorted. We achieve this by preserving the geodesic distances between the dataset objects. Geodesic distances capture a significant part of the dataset structure, and their usefulness is recognized in many machine learning, visualization and clustering algorithms. Therefore, if a practitioner uses the protected dataset as input to a variety of mining, machine learning, or database operations, the output will be the same as on the original dataset. We illustrate and validate the applicability of our methods on image shapes extracted from anthropological and natural science data.

1. INTRODUCTION

Data are frequently outsourced by companies to mining firms, for the purpose of extracting and distilling useful information. Data owners, nonetheless, need also maintain the principal rights over the datasets they share, which in many cases have been obtained after expensive and laborious procedures. This work presents such a protection mechanism, that can provide convincing evidence about the legal ownership of a shared dataset, without compromising the dataset usability, under a wide set of mining and database operations or machine learning tasks. We achieve that by guaranteeing that the relationship between the dataset objects remains undistorted.

In this work we focus on databases of shapes. As a shape we consider the 2-dimensional representation of a singular object. Usually the shape is extracted from an image of the object, as part of a feature extraction procedure. For example, given an image of a leaf we can extract its perimeter and store it as a two dimensional sequence. The color or texture of the leaf are not important for pattern matching algorithms and thus can be discarded. Therefore, a shape is essentially a *compressed* representation of an object's image, allowing for fast similarity search, classification and clustering on the dataset. Shape datasets are quite common nowadays [23, 24] and have significant applications in a multitude of fields. For example:

1. In **biometrics** and in various surveillance applications, face contours can be utilized as important features for recognizing a person's identity.
2. Similarly, in **medical imaging**, databases of tumor shapes, can be contrasted with MRI and X-Ray images for identifying significant pathologies [15, 11, 2].
3. In **anthropology** and evolutionary sciences cranial characteristics (e.g., skulls) are important morphometric features that can useful in establishing evolutionary paths [22].
4. Finally, in many **natural sciences**, comparison of shapes plays an important role in establishing accurate taxonomies. For example in ichthyology, study of the fish shape can indicate its species, or in forestry science comparison of leaf contours is extensively utilized for classifying the affinities between tree species [19, 7].

The embedding of the ownership seal in the dataset will be achieved through means of watermarking; however this work approaches watermarking from a novel perspective, by additionally preserving the mining and visualization capacity of a dataset. The proposed scheme accepts as input a collection of shapes (2-dimensional sequences), and embeds a secret key in all of them, with the purpose of satisfying all the following objectives:

- Provide an ownership determination mechanism for the whole dataset.
- Introduce only imperceptible visual distortions for each shape in the dataset.
- Be robust to common data transformations.

EDBT'08, March 25–30, 2008, Nantes, France.
EBDT 2008, Nantes, France.

- Provide the facility to tune properly the data protection scheme, so that important distances between the dataset objects are retained, therefore search and mining operations on the watermarked dataset are still meaningful.

The proposed approach slightly distorts the shapes but attempts to preserve as well as possible the object distances. Our problem formulation is inherently distance preserving, since a large subset of database, mining and learning algorithms are distance based (e.g. Nearest Neighbor (NN) search, NN-classification, outlier detection). One question that immediately arises is which distances are most important and therefore should be preserved. Preserving all pairwise distances can be not only prohibitive computationally, but may actually be impossible in practice, too. Therefore, here we propose to preserve the *geodesic* distances between the dataset objects. Geodesic distances are preserved by maintaining the original minimum-spanning-tree (MST) of object distances. Such an approach, in addition to preserving the local dataset structure (nearest neighbors), will also preserve the semi-global dataset structure. The importance of geodesic distances is attested in many research efforts, with applications that range from data visualization and dimensionality reduction (e.g. ISOMAP [20]), to phylogeny construction [16] and data clustering [8]. In this work, we will also demonstrate the usefulness of geodesic distance preservation using additional visualization and clustering applications.

Our contributions are: 1) present a robust rights protection scheme for shapes (and sequences in general) based on watermarking principles, 2) show how to embed the ownership key in a space that is invariant to common shape transformations, 3) delineate algorithms that provide geodesic distance preservation and show extensions on hierarchical clustering algorithms, such as dendrograms, 4) demonstrate applicability of the algorithms for mining, clustering and visualization techniques on a variety of real datasets.

1.1 Methodology and Difference from Previous Work

Our rights protection scheme extends traditional watermarking techniques. There has been a proliferation of watermarking research on multimedia datasets [5]. However, multimedia watermarking is concerned with watermarking a single object and not a collection of objects, therefore not focusing on the maintenance of the inter-relationship between objects, which is one of the primary objectives of this work.

The embedded seal or watermark essentially adds noise on the original data, which has also been the topic of discussion in many works that consider privacy preserving datamining [13, 10, 14, 3]. However, these techniques typically do not work directly on the actual perturbed data (like our technique), but attempt to reconstruct the original data distribution using the known noise distribution that has been added on the dataset [1, 17]. Privacy preservation can also be achieved through limited dataset view, for example, by horizontal or vertical distribution of the data to different sites [21, 9, 25]. In our setting, the dataset cannot be dissected in portions, but is being distributed as a whole.

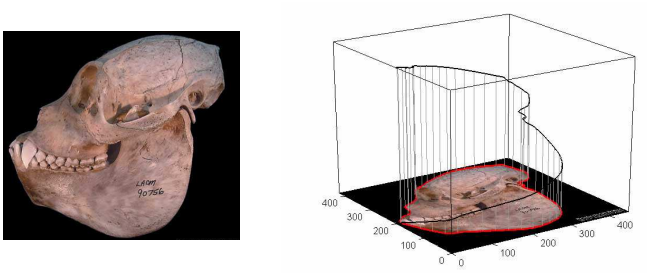
Watermarking in data streams has also been presented in [18] by Sion, et al. That work examines watermarking on a *single* numeric sequence and does not consider a collection of them, with the purpose of maintaining their original pairwise relationships. Additionally, [18] does not examine issues important for this work, such as resilience to geometric data transformations.

In general, our setting presents additional challenges compared to traditional watermarking or privacy preservation techniques, in the sense that not only do we work on the perturbed data itself, but more importantly, we provide additional *guarantees*, such as geodesic distance preservation. Finally, since our problem is inherently distance based, our current problem formulation and solution can easily be tailored to work on an extensive gamut of mining and machine-learning algorithms, many of which are strongly dependent on the use of a distance function (clustering, outlier detection, etc).

1.2 Overview

First we will present how to embed the ownership seal in each shape of the dataset using watermarking principles. The watermark will be embedded in the frequency domain, and in particular, only in the magnitude of the Fourier descriptors in order to provide robustness against rotational or other attacks. We will also demonstrate how to detect the watermark using an advanced correlational filter which operates in a slightly modified domain, and which provides better detectability than commonly used correlation detectors. The distortion of each shape with respect to the watermark embedding power will also be provided in a closed form solution.

Later on, we will revisit a visualization technique that is based on the spanning-tree, in order to show that the spanning tree before and after the watermarking is not modified. We will provide an algorithm that finds the *maximum* watermark embedding power which does not distort the spanning-tree (and hence the geodesic distances). We will also provide an algorithm that preserves the dendrogram structure after the watermarking, thus making our contribution applicable to clustering as well.



Red Howler Monkey Male
(Alouatta seniculus seniculus)

Conversion of skull shape into a
two-dimensional sequence

Figure 1: Shape perimeter converted into a 2D sequence

Our findings are demonstrated on image contour data from anthropology and the natural sciences. We demonstrate that typical mining tasks in these disciplines, including taxonomy categorization and construction of phylogenies, are not affected by our rights protection scheme. We treat images of shapes as 2-dimensional sequences by extracting the shape perimeter and sequencing adjacent peripheral points. An example of this procedure is shown in Figure 1, where the skull shape of a Red Howler monkey is extracted.

In general, the scope of this work is not only to pro-

vide a technique for convincingly claiming ownership on a dataset, but to *guarantee* that by working directly on the watermarked data, the output of a wide class of data mining, learning and visualization algorithms will remain the same. This property is very beneficial, because it provides a protected but still usable copy of the dataset. Additional benefits of the watermarking approach can also be realized when different distributions of the dataset are marked with different keys. This provides a mechanism for conclusively indicating the source of a leak, by identifying which key was originally embedded in the dataset.

2. RIGHTS PROTECTION THROUGH WATERMARKING

In this section we explain how to embed a secret watermark in a collection of shapes (or contours), which will serve as an ownership seal for the whole dataset. Each shape contour is stored and treated as a 2-dimensional sequence of values (or 2D time-series). For embedding the watermark we will use a spread spectrum approach [5]. This will distribute the power of the watermark across multiple frequencies and over a number of dataset objects, making its removal particularly difficult. The watermark embedded in the dataset will satisfy the following desirable properties:

1. **Imperceptible**; there is no apparent visual distortion on each shape of the dataset.

2. **Detectable**; the correlation distribution of the watermarked data with the correct key, should be sufficiently distinct than the distribution with a random key, so as to allow the conclusive determination of the watermark presence.

3. **Preservation of geodesic distances**; the power of the watermark is tuned in such a way, so that the spanning-tree of all the objects does not change after the watermarking.

4. **Robust** to malicious attacks; our technique works by embedding the watermark in a space that is invariant to common transformations such as translation, scaling or rotation. We assume that a malicious attacker can apply various data transformations in an attempt to remove the embedded watermark, to the extent that the data usability is not hindered. For example, global rotation of all shapes will not change their relative distance.

In the following sections we explain how the above requirements are satisfied by the proposed watermarking scheme.

2.1 Embedding the watermark

We consider each shape as a vector of complex numbers $x = \{x_1, \ldots, x_n\}$, where $x_k = a_k + b_k i$ (i is the imaginary unit with i^2=-1), and the real and imaginary parts, a_k and b_k, respectively, describe the coordinates of the k-th point of shape x. Each of the 2D sequences describes the coordinates of the shape perimeter (as shown in Fig. 1).

In each shape contour we will embed a *watermark*, which is a *secret information* that will be hidden inside each sequence. The watermark is encoded in a vector $W \in \{-1, 1, 0\}^n$, that is, taking 3 distinct values and having the same length as x. The embedding of the watermark consists of a composition function that, given x and W, returns a modified sequence which is *similar* to x and *encloses* W. In what follows we explain more clearly the meaning of *similar* and describe how to discover the *enclosed* watermark.

In order to provide better resilience against malicious attacks, we will not embed the watermark in the original *space domain* but into the *frequency domain*, instead. Every sequence x will thus be represented by a set of Fourier descriptors $X = \{X_1, \ldots, X_n\}$, where n is the number of points in x, as well as the number of its frequency components. The mapping from one domain to the other is described by the well known normalized discrete Fourier transform $ft(x)$ and its inverse $ift(X)$.

Every coefficient X_j can be expressed in terms of its *magnitude* ρ_j and *phase* ϕ_j, that is, $X_j = \rho_j e^{\phi_j i}$. We use a multiplicative embedding of the watermark only on the magnitudes, but we retain the original phases.

DEFINITION 1 (MULTIPLICATIVE FOURIER EMBEDDING). *For a sequence $x \in \mathbb{C}^n$ and a watermark $W \in \mathbb{R}^n$, the* multiplicative Fourier embedding *generates a watermarked sequence $\widehat{x}$ by replacing the magnitudes of each Fourier descriptor of x with a watermarked magnitude $\widehat{\rho_j}$:*

$$\widehat{\rho_j} = \rho_j \cdot (1 + pW_j)$$

where power $0 \leq p \leq 1$ *specifies the intensity of the watermark.*

Using the modified magnitudes $\widehat{\rho_j}$ and the original phases ϕ_j, we go back from the frequency domain to the space domain and reconstruct the watermarked sequence using the inverse discrete Fourier transform. An overview of the described methodology is provided in Figure 2.

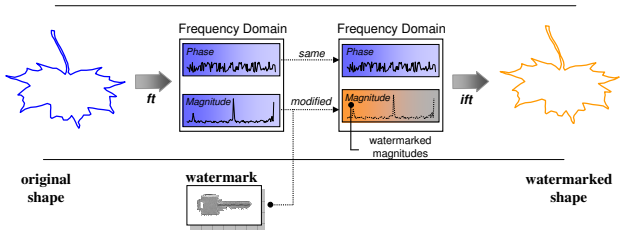


Figure 2: Overview of watermarking technique.

Recall that in each frequency component of the sequence we will embed an element of the secret watermark $W \in \{-1, 1, 0\}^n$. The multiplicative watermark is beneficial because larger magnitudes are allowed to hold (mask) larger embedding powers. By construction, W will contain $l/2$ values of -1 and and $l/2$ values of +1, where $l < n$, and therefore $\sum W_j = 0$. Only those l elements of $W_j \neq 0$ encode the owner signature, independently of the sequence length n.

The choice of which Fourier descriptors (frequencies) will be used to embed the watermark, i.e. $W_j \neq 0$, may affect the effectiveness of the detection process against malicious attacks. Indeed, the magnitudes ρ_j typically have a very skewed distribution, where most of the energy is concentrated in very few frequency descriptors. We embed the watermark in the coefficients with the highest energy, since these describe the essence of the shape, making more difficult the watermark removal without inducing significant distortion in the shape. In Fig. 3 we can see the reconstruction of a shape from a dataset of skulls, when being approximated using 2 to 16 coefficients with the highest energy.

Driven by these considerations, we have chosen to embed the watermark in those coefficients having, on average,

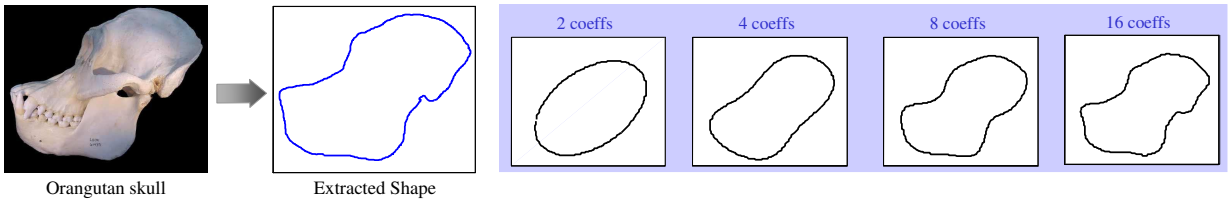


Figure 3: Shape reconstruction for different number of Fourier coefficients that contain the highest energy.

the largest magnitudes. Some randomization process can also be inserted in this stage, so as to ensure additional robustness. However, no portion of the watermark will be embedded on the first Fourier descriptor X_1 (the DC component), since it captures the center of gravity of the shape x ($X_1 = \sum x_j/\sqrt{n}$), and it is therefore easily susceptible to attacks. For example, a simple translation will change the center of gravity of x (and the DC component) without affecting its shape, but it will erase this part of the watermark.

Let $\mu_j(\mathcal{D})$ be the average value of the j-th magnitude across the dataset $\mathcal{D}$, then, the watermark W is formally defined as follows:

$$W_j = \begin{cases} 0 & \text{if } j = 1 \quad \text{(DC component)} \\ \{-1, 1\} & \text{if } \mu_j(\mathcal{D}) \text{is among the } l \text{ largest } \mu_{i\neq 0}(\mathcal{D}) \\ 0 & \text{otherwise} \end{cases}$$

with $\sum W_j = 0$.

Symbol	Description
$\mathcal{D}$	original dataset of shapes
$\widehat{\mathcal{D}}$	watermarked dataset
x	sequence in space-domain
X	sequence in frequency domain
n	number of points in a sequence
$X_j = \rho_j e^{\phi_j i}$	Fourier descriptor as a function of its magnitude and phase
p	embedding power
$\widehat{X_j} = \widehat{\rho_j} e^{\widehat{\phi_j} i}$	Watermarked Fourier descriptor as a function of its watermarked magnitude and phase
$\mu_j(\mathcal{D})$	mean of ρ_j across the shape perimeters in $\mathcal{D}$
l	number of non-zero elements of watermark
χ	correlation
$\widehat{\mathcal{D}}_p(x, y)$	distance between two shapes x, y when watermarked using power p

Table 1: Notation used in the paper

Table 1 summarizes the notation used throughout the paper.

Resilience of Frequency Embedding: By construction the embedded watermark is resistant to:

1) Rotations; by rotating all objects by the same amount, all their points change, but the original distances between objects remain the same. This can be readily realized by observing that if R is a given rotation matrix and x,y any two shape sequences in the dataset, then $\|R(x-y)\| = \|x-y\|$, since the rows and columns of R are orthonormal and $RR^T = 1$. So this is an attack that can easily be performed with the purpose of destroying all the original values (and possibly the secret watermark) but without affecting any of the relative object positions. Rotations affect only the phase in the frequency domain and not the magnitudes. Since our watermark is embedded in the magnitude, it will be invariant to similar attacks.

2) Translations; if one globally translates all the objects by the same amount (e.g. 10 points to the left) all the original shape values will change, but the relative positions of all objects will remain the same. Global translations only affect the first frequency component (the DC), where we don't embed any part of the watermark, therefore our scheme is also robust to translation attacks.

In the experimental section, in addition to geometric transformations, we also show the resilience of our scheme to other attacks, including noise addition and resampling.

2.2 Error introduced by the watermark.

Altering each shape in order to embed a watermark adds some noise in the dataset. We measure this noise as the relative error ϵ introduced in a given sequence x.

Let $\|\cdot\|$ be L_2 norm of a vector, then due to Parseval's theorem, and after some algebraic manipulations, it is easy to see that:

$$\begin{aligned} \|x - \widehat{x}\| &= \|X - \widehat{X}\| = \ldots = \\ &= \sqrt{\|\rho - \widehat{\rho}\|^2 + 2\sum_j \rho_j \widehat{\rho}_j [1 - \cos(\phi_j - \widehat{\phi}_j)]} \\ &= \|\rho - \widehat{\rho}\| \qquad (since\ \phi_j = \widehat{\phi}_j\ ) \\ &= \|\rho - \rho(1 + pW)\| \\ &= p\ \|\rho W\| \end{aligned}$$

and therefore:$\epsilon(x, \widehat{x}) = \frac{1}{\|x\|}\ \|x - \widehat{x}\| = p\ \frac{\|\rho W\|}{\|x\|}$

This means that the watermark embedding introduces an error which is proportional to the embedding power and to the norm of those descriptors for which $W_j \neq 0$. Given this immediate relationship between power and error, we will often refer to the Euclidean error introduced by the watermark to quantify the amount of power used during the embedding process.

In Figure 4 we can see the result of the Fourier embedding for a watermark of length $l = 64$ using different embedding powers. In this dataset, an error of $\epsilon = 1\%$ is already visible, and therefore the user can set this as a potential upper bound on the watermark embedding power.

2.3 Watermark Detection

The detection step attempts to measure the correlation χ between the watermarked magnitudes $\widehat{\rho}$ and the watermark W. Given a watermarked shape $\widehat{x}$ and a watermark W, the larger the correlation between the two, the higher the probability that W was the actual embedded watermark. Since we spread the watermark across all objects of the dataset, we will instead measure the correlation between W and the

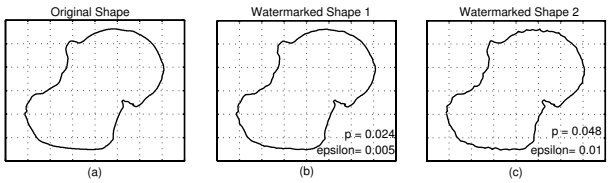


Figure 4: Watermark embedding in one of the shapes from the `skulls` dataset.

average magnitudes $\mu(\widehat{\mathcal{D}})$, where $\mu_j(\widehat{\mathcal{D}})$ is the average value of $\widehat{\rho}_j$ for every trajectory $\widehat{x} \in \widehat{\mathcal{D}}$.

Unfortunately, directly measuring the correlation between these two would not have been very effective. Since our goal is to minimize distortion (that is, use a small embedding power), then every set of magnitudes $\widehat{\rho}$ is dominated by the original level of average magnitudes $\mu(\mathcal{D})$, which, in a sense, behave like a background noise, masking the embedded watermark pW we want to discover.

To overcome this difficulty, we record $\mu(\mathcal{D})$ during the embedding process and remove its bias before the detection takes place. The correlation between $\widehat{\mathcal{D}}$ and W is thus defined as follows:

DEFINITION 2 (WATERMARK DETECTION). *Let $\widehat{\mathcal{D}}$ be a watermarked dataset and let W be the actual watermark. The correlation between $\widehat{\mathcal{D}}$ and W given the average magnitudes in the original dataset $\mu(\mathcal{D})$ is:*

$$\chi(W, \widehat{\mathcal{D}}) = \left(\frac{\mu(\widehat{\mathcal{D}})}{\mu(\mathcal{D})} - 1 \right) \times W$$

where the above division is *element-wise* and the multiplication is an inner product.

This improved scheme can be thought of as an inversion of the multiplicative embedding, and allows for a highly effective detection of the watermark. It is easy to see that the correct watermark will have maximum correlation $(pW) \times W$, while any other watermark $W' \neq W$ will have a smaller correlation $(pW') \times W$.

The cost we have to pay for such an effective detection process is to store, together with the watermark W (length l), the vector $\mu(\mathcal{D})$ (also length l). Therefore, the full $2l$ values can be considered as the new key used in the watermarking. The additional l values impose a very small additional storage cost and in practice lead to enhanced security. In fact, consider that it is almost impossible for a malicious third party to measure $\chi(W, \widehat{\mathcal{D}})$ since $\mu(\mathcal{D})$ is unknown, and therefore very difficult to be discovered, in an attempt to remove the embedded watermark W.

2.3.1 Detection process

In between the embedding and the detection process, the dataset may be subject to attacks/transformation by a malicious attacker. The correlation values between the correct watermark and incorrect watermarks will thus be distorted. For this reason the detection process is probabilistic in nature.

We say that watermarking is successful if the correlation of the watermarked dataset with the correct key is consistently larger than the correlation with any other key, irrespective of the embedding key. In particular, we will say that that W has been embedded in $\widehat{\mathcal{D}}$ if $\chi(W, \widehat{\mathcal{D}}) \geq \omega$, for a given

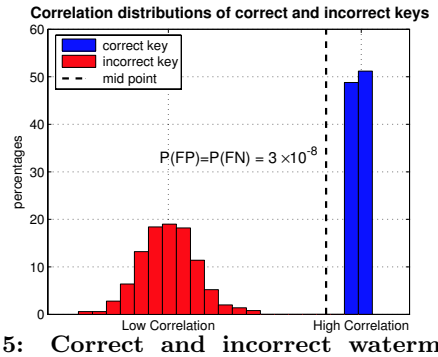


Figure 5: Correct and incorrect watermark empirical correlation distributions. Dataset = `skulls`, $p = p_{.5}$.

threshold ω. If the correlation with a wrong key W' becomes larger than the one with the correct key W we have a false positive. If this is the case, the owner of W' may claim the ownership of the dataset against owner of W.

The best choice of ω is the correlation value that separates best between correct and incorrect key correlations. The value of ω is set empirically as follows. Given a dataset watermarked with a random watermark W and then transformed by a malicious attacker, we measure its correlation with W and with other 500 distinct incorrect watermarks W'. We repeat this experiment for 500 different Ws and this results to two probability distributions. We denote with α the empirical correlation distribution of the correct key and with β the wrong key's empirical correlation distribution. The more these distributions are separated the better we can detect the presence of the watermark (see Figure 5).

If the correlation of a wrong watermark is larger than ω we have a *false positive*. A similar argument holds for the case of *false negatives*. Giving the same importance to both cases, we choose the value of ω to be exactly in the middle of the two distributions when properly normalized, i.e. ω is such that:

$$\frac{\text{mean}(\alpha) - \omega}{\text{std}(\alpha)} = \frac{\omega - \text{mean}(\beta)}{\text{std}(\beta)}$$

Since α and β exhibit strongly Gaussian distributions (see Fig.5), we can measure the false positive FP and negative probabilities (FN) as follows:

$$P(FP) = \frac{1}{\text{std}(\alpha)\sqrt{2\pi}} \int_{\omega}^{+\infty} \exp\left(-\left(\frac{\chi - \text{mean}(\alpha)}{\text{std}(\alpha)\sqrt{2}} \right)^2 \right) d\chi$$

and symmetrically for $P(FN)$.

That is, we choose ω as the probability point such that $P(FP) = P(FN)$. In the experimental section we will show that it is possible to maintain low false positive/negative rates in presence of many adversarial attacks.

3. GEODESIC DISTANCES - MINIMUM SPANNING TREE

Now we describe how to properly tune the watermark embedding power p so that the geodesic distances between the dataset objects are not distorted. The importance of geodesic, or minimum path, distances is well recognized because of their usefulness in a variety of data analysis tasks, including applications such as logistics [6], data clustering [8], visualization [12] and phylogeny construction in biological applications [16]. Recently a wide class of dimensionality

reduction/visualization techniques that exploit geodesic distances have gained large popularity, because they have successfully been applied to uncover hidden structures in high-dimensional datasets. ISOMAP [20] is probably one of the most well known such techniques, which uses the minimum-spanning-tree of distances between objects in order to better approximate the dataset structure, discover non-linearities and estimate more accurate the intrinsic data dimensionality.

In general, preservation of the minimum-spanning-tree (MST) is useful because: 1) The Nearest-Neighbor (NN) of every object will be retained, hence the local dataset structure will not change. Note, that many mining algorithms utilize the nearest-neighbors, such as NN-search or NN-classification. 2) The MST will also preserve important global dataset characteristics, therefore the relationship between distant objects (and clusters) will not be lost.

In order to show that the geodesic (or spanning-tree) distances before and after the watermarking will remain the same, we will make use of a visualization technique that is based on the minimum-spanning-tree. This visualization technique uses the spanning tree in order to visualize the relationships between shapes/objects on 2-dimensions. Our rights protection scheme does not distort the MST therefore the mapping will remain the same. Since our approach *guarantees* MST preservation, it can be exploited by any technique that utilizes the spanning-tree or the 1-Nearest-Neighbor, which the MST also preserves. Therefore, dimensionality reduction/compression techniques such as ISOMAP can also be used to visualize the watermarked shapes, with guarantees on the visualization outcome.

Based upon the techniques for geodesic distance preservation, we will also show how our methodology can be extended to preserve dendrograms, which are commonly employed in anthropological or natural sciences for discovering clusters and relationships between the examined objects/species.

In the sections that follow, we will revisit a mapping / visualization technique that relies on the MST, which will help us illustrate the minute differences on the spanning-tree that are introduced by our right protection/watermarking technique. We will explain what constraints need to be embedded in the watermarking scheme in order to guarantee MST-preservation. Additional constraints will also be incorporated for guaranteeing dendrogram preservation. Finally, we evaluate the outcome of our methodology, before and after the incorporation of the watermark.

3.1 Minimum-Spanning-Tree Mapping

Here, we describe succinctly a mapping technique proposed by Lee, et al. [12], which utilizes the Minimum-Spanning-Tree (MST) and a triangulation method for displaying relationship of objects on 2-dimensions. The technique preserves 2 distances per object on the two-dimensional space. The first distance preserved is the distance to the nearest neighbor of every object. The second distance can either be different for every object (e.g. its 2NN), or it can be the distance to a reference point. The latter option creates a powerful visualization technique which allows not only the Nearest Neighbors to be preserved (local structure), but additionally preserves distances towards a single reference point, giving the option for global data view with respect to that object. For visualizing the relationship between shapes, we will, later on, adapt the reference point (or pivot) technique.

The computation of the MST requires $O(VlogE)$ time for V vertices and E edges using Kruskal's algorithm. Once the MST is calculated the mapping on the 2D space can commence from any point/object that the user designates and the MST tree is mapped either in a breadth or depth-first-search manner. In this work we utilize a BFS mapping. Let us see how the mapping works with a running example.

Suppose the first two points (A and B) of the MST are already mapped, as shown in Fig. 6 (a). Let us assume that the second distance preserved per object is the distance with respect to a reference point which in our case is the first point. The third point is mapped at the intersection of circles centered at the reference points. The circles are centered at A and B with radii of $D(A, C)$ -the distance between points A and C- and $D(B, C)$, respectively. Due to the triangle inequality, the circles either intersect at 2 positions or are tangent. Any position on the circles' intersection will retain the original distances towards the two reference points. The position of point C is shown in Fig. 6 (a). The fourth point is mapped at the intersection of circles centered at A and C (Fig. 6 (b)) and the fifth point is mapped similarly (Fig. 6 (c)). The process continues until all the points of the MST at positioned on the 2D plane and the final result is shown in Fig. 6 (d).

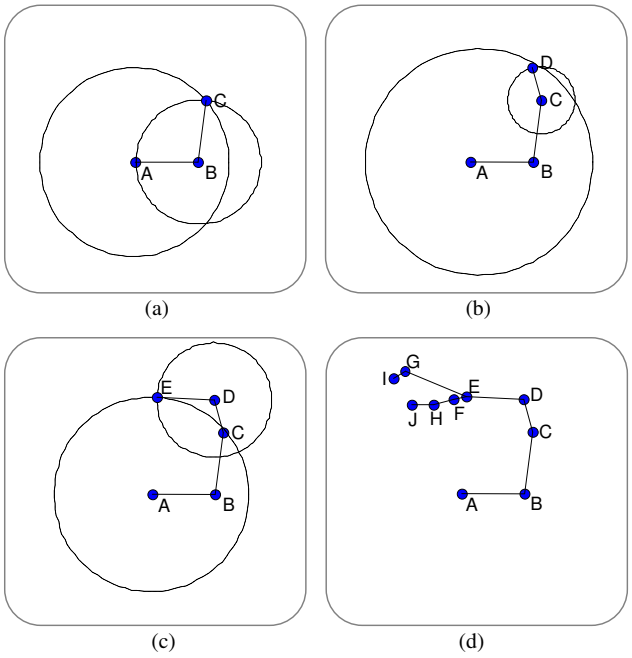


Figure 6: Mapping through MST and triangulation

We will utilize the above visualization technique to demonstrate that the geodesic distances before and after the watermarking remain virtually the same, and that the spanning-tree is not distorted. Of course, the results for any MST-based algorithm will not be altered, similarly.

4. MST-PRESERVING WATERMARKING

The focus of this section is to guarantee that the spanning-tree for a set of objects is identical before and after watermarking. We achieve this by discovering the appropriate

watermark embedding power, which ensures preservation of the original Minimum-Spanning-Tree. To achieve this goal it is necessary to preserve the edges of the minimum-spanning-tree.

We call this task *MST-Preserving (MST-P)* watermarking. Below we formally define the generalized MST-Preservation problem, which also allows for an error tolerance in the Spanning-Tree preservation:

MST-P Watermarking Problem. *Given dataset $\mathcal{D}$, minimum threshold (p_{min}) and maximum threshold (p_{max}), find the largest p, $p_{min} \leq p \leq p_{max}$, such that after the watermark embedding, at most a fraction τ among the edges of the Euclidean minimum spanning tree do not match the MST of the original dataset.*

Therefore, we would like to discover the largest power p, $p_{min} \leq p \leq p_{max}$ that guarantees MST-preservation, since larger energies of the embedded watermark provide better detection and resilience to attacks. We dictate an upper bound p_{max} on the embedding power in order to guarantee minimization of visual distortion for the shapes after the watermark embedding. A proper p_{max} value can be determined empirically by observing at which power level the shapes are distorted (Fig. 7). This process can, also, be easily automated by examining how the gradient of the watermarked shape changes (i.e., how smooth it is before and after watermarking), and stopping when the cumulative gradient exceeds a certain threshold.

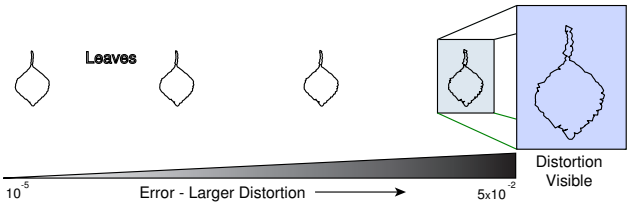


Figure 7: Visual distortions for different watermark embedding powers, for the `leaves` dataset.

Setting the minimum power p_{min} is a way of providing assurance for high detectability for the watermark and additionally limits the search space for the best power for MST-preservation. In our experiments we set $p_{max} = p_{10^{-2}}$, i.e. the power that introduces at most a 1% relative error. Also, for simplicity, minimum power was set to $p_{min} = 0$.

Finally, one can allow for a percentage of edges of the spanning tree to be different, either for the purpose of embedding an even stronger watermark, or for guaranteeing that the algorithm will return a result within the designated watermark power embedding range. For all our experiments we set τ to zero, effectively enforcing full maintenance of the MST. This constraint was always satisfied for our datasets.

Before explaining our technique for MST-preservation it is useful to derive a closed form formula of the distance between two watermarked shapes $\widehat{x}$ and $\widehat{y}$ as a function of the embedding power p:

$$\begin{aligned}\widehat{D}_p^2(x,y) &= \|\widehat{x}-\widehat{y}\|^2 = \|\widehat{X}-\widehat{Y}\|^2 \\ &= \|(1+pW)\times X-(1+pW)\times Y\|^2 \\ &= \|(1+pW)\times (X-Y)\|^2 \\ &= \|W^2\times (X-Y)\|^2\ p^2 + \\ &\quad 2\|W\times (X-Y)\|^2\ p + \\ &\quad D^2(x,y)\end{aligned}$$

We use this parameterized distance function to calculate the largest power between p_{min} and p_{max} that ensures the required preservation property.

4.1 MST-preservation

In order to preserve the minimum spanning tree we must preserve its fundamental property. Let $T(\mathcal{D}, E)$ be a minimum spanning tree, where $\mathcal{D}$ are the nodes, and E is the set of $|\mathcal{D}|-1$ edges composing the tree. If we remove an edge $e(x,y) \in E$, we split the original tree in two connected components U_e and V_e. Since T is the *minimum* spanning tree, such edge $e(x,y)$ has the property of being the *shortest* edge that may link U_e with V_e. If we denote with $D(e)$ the distance $D(x,y)$, for every edge $e(x,y) \in E$ it holds that:

$$D(e) \leq D(u,v) \qquad \forall u \in U_e, \forall v \in V_e \tag{1}$$

Recall that we want to guarantee that the Euclidean minimum spanning tree T_p after having embedded a watermark with power p is still the same as the original T. This implies that for each edge $e \in E$ the above property still holds after the watermarking, i.e.:

$$\widehat{D}_p(e) \leq \widehat{D}_p(u,v) \qquad \forall u \in U_e, \forall v \in V_e \tag{2}$$

where $\widehat{D}_p$ is the distance between the two objects in the watermarked database.

Given that we are able to express the distance between two objects as a function of the embedding power p, the MST-P problem requires to find the largest p such that at most $\tau \cdot |E|$ edges of the original MST do not satisfy the above inequality. In the worst case, this would require $O(|D|^3)$ inequalities.

We saw earlier that distance $\widehat{D}_p$, or equivalently $\widehat{D}_p^2$, can be expressed as a quadratic function of the embedding power p. The MST-P problem can thus be solved by finding the solutions of a system of quadratic inequalities, for which we provide Algorithm 1.

Algorithm 1 MST-preservation.

```
1: T(|D|, E) = find Euclidean MST of D
2: for all e ∈ E do
3:    good_powers(e) = [p_min, p_max]
4:    for all u ∈ U_e do
5:       for all v ∈ V_e do
6:          bad_powers = solve(D̂_p(e) > D̂_p(u,v))
7:          good_powers(e) = good_powers(e) \ bad_powers
8:       end for
9:    end for
10: end for
11: p = max{p : |{e : p ∉ good_powers(e)}| < τ · |D|}
```

Let $solve(\langle EXPR \rangle)$ return all powers $p \in [p_{min}, p_{max}]$ which satisfy inequality $EXPR$.

Complexity: Since function *solve* can be computed in constant time, and the MST can be obtained in $O(|\mathcal{D}| \log |\mathcal{D}|)$ time, the worst-case complexity of this algorithm is $O(|\mathcal{D}|^3)$. Notice that this is an offline computation cost before the data are released (and therefore not a prohibitive cost), and represents only a small additional overhead for providing assurances about the dataset ownership.

4.2 Extending to Dendrogram Preservation

Now, we discuss how the techniques introduced in Section 4.1 can be extended for clustering techniques, and in specific for dendrogram preservation. Dendrograms are instantiations of hierarchical clustering schemes. One such

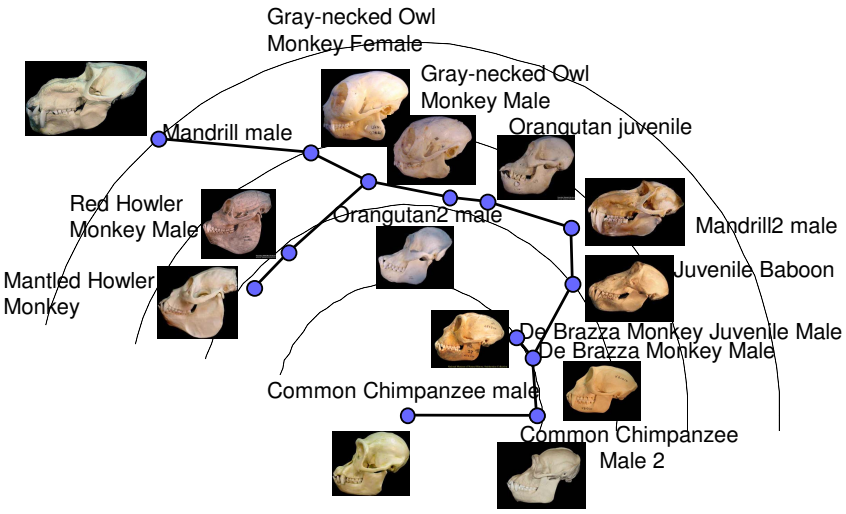


Figure 8: `Skulls` data and MST visualization

approach is often referred to as Divisive Hierarchical single linkage clustering (DHC). In this scheme, all the nodes of the graph are first clustered into a single group. Then, considering the various edges in the MST in decreasing order, the nodes are successively broken into clusters until the necessary number of clusters is obtained. We define the Dendrogram-preserving (DHC-P) watermarking problem as follows:

DHC-P Watermarking Problem. *Given dataset $\mathcal{D}$, minimum threshold (p_{min}) and maximum threshold (p_{max}), find the largest p, $p_{min} \leq p \leq p_{max}$, such that after the watermark embedding, the dendrogram obtained is the same as the dendrogram on the original dataset.*

To guarantee that the dataset (after watermarking) is DHC-preserving, we ensure that it is MST-preserving, and that the order of the edges in the MST (sorted by distance) remains the same. For the rest of this discussion, we treat the edges in the MST as a sorted (decreasing distance order) array. In this notion, we represent the i^{th} largest edge in E by $E(i)$. Thus to ensure that the watermarked dataset is DHC-preserving, we need to guarantee that the ordering of the edges in E does not change after watermarking, i.e., each edge should still be larger than all the edges that are shorter than it in the original dataset.

$$\widehat{D}_p(E(i)) \geq \widehat{D}_p(E(j)) \qquad \forall i, j \leq |E|, i < j \tag{3}$$

Also, consider that usually only a certain number of clusters $k << |\mathcal{D}|$ are of interest to the user. Therefore, for each of the $k-1$ larger edges we enforce the order constraint with an additional set of inequalities. In other words, inequality 3 needs to be satisfied only for $i, j \leq k$. For the other edges we only need to ensure that

$$\widehat{D}_p(E(k)) \geq \widehat{D}_p(E(j)) \qquad \forall k < j \leq |E|$$

The relative ordering amongst the remaining $|E| - k$ edges need not be maintained.

Built on top of Algorithm 1, we introduce a *DHC-preserving* algorithm which aims at finding the largest power that will preserve k clusters generated via DHC. After the MST-preservation is run in order to get a baseline set of feasible watermark embedding powers, a new iteration is performed which processes only a subset of the minimum spanning tree edges, and ensures that the necessary orderings are maintained. The resulting algorithm is depicted as Algorithm 2. Note that, the algorithm is slightly over-constrained, since, in order to get the same set of k clusters, it is not needed to preserve the whole MST.

Algorithm 2 DHC-preservation.

k = *number of desired clusters*
$T(|\mathcal{D}|, E)$ = *find Euclidean MST of* $\mathcal{D}$
good_powers(E) = *run MST preservation algorithm*
for all $i \in \{1, \ldots, k-1\}$ **do**
  bad_powers = $solve(\widehat{D}_p(E(i+1)) > \widehat{D}_p(E(i)))$
  good_powers($E(i)$) = good_powers(e) \ bad_powers
end for
for all $j \in \{k+1, \ldots, |E|\}$ **do**
  bad_powers = $solve(\widehat{D}_p(E(j)) > \widehat{D}_p(E(k)))$
  good_powers($E(j)$) = good_powers(e) \ bad_powers
end for
$p = \max\{p : |\{e : p \notin \text{good\_powers}(e)\}| < \tau \cdot |\mathcal{D}|\}$

Let $solve(\langle EXPR \rangle)$ return all powers $p \in [p_{min}, p_{max}]$ which satisfy inequality $EXPR$.

Complexity: Since the additional constraints may require to solve at most $O(|\mathcal{D}|)$ inequalities, the global complexity of the algorithm is still dominated by the MST-preservation, and it is therefore $O(|\mathcal{D}|^3)$.

5. EXPERIMENTAL EVALUATION

We evaluate our algorithms, by depicting that geodesic distances are not distorted through visualization and clustering applications. We also examine the robustness of our scheme under various attacks on the watermarked shapes. For our experiments we utilize 3 shape datasets; more information on them is provided in Table 2.

dataset	# points per shape	# shapes	# classes
skulls	1500	16	7
leaves	128	1125	15
fish	256	247	10

Table 2: Characteristics of the datasets

5.1 Usefulness of MST-preservation

In this section we demonstrate the usefulness of retaining the geodesic distances. We will also show that our rights protection technique does not change the geodesic distances between the watermarked shapes. We embed the secret watermark in each shape and then we use the mapping technique presented in section 3.1 in order to visualize the relationship between the shapes. For this example we use a simple Euclidean distance to evaluate the similarity between the shapes, even though more complex measures could be plugged into our algorithm as well.

We first demonstrate the meaningfulness of results using the spanning-tree visualization on watermarked shapes of skulls. The result is shown in Fig. 8. Note that on the figure we plot the original image from which every shape is derived, simply for presentational purposes; however we clarify that the rights protection scheme works on the dataset of contours of those shapes. We utilize the MST mapping of section 3.1 and select the chimpanzee as the pivot point for the visualization. We can notice that the results projected by the spanning-tree visualization are in consensus with the current views on primate evolution [4].

In our second example, we demonstrate another spanning-tree mapping, illustrating the MST before and after the shapes are watermarked. For this example we utilize shapes from the `leaves` dataset and the resulting MST mapping is shown in Fig. 9. We fill-in the different leaf shapes with diverse colors in order to differentiate the leaves that belong to the different tree species. In this example we contrast the MST on the original shapes (black lines) against the MST on the watermarked shapes (orange line). We can observe that the two spanning trees (and therefore the geodesic distances) for any practical purpose are almost identical. Therefore, our algorithm *accurately* chose the watermark embedding power, so as not to distort the geodesic distances, and hence the spanning-tree.

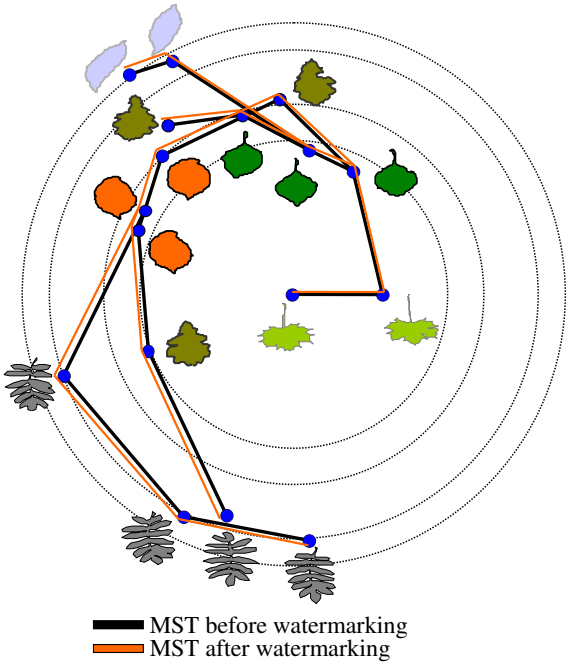


Figure 9: MST preservation on the `Leaves` dataset

Finally, in Fig.10 we demonstrate the visualization of watermarked shapes using the dendrogram preservation algorithm. We note that the dendrogram on the watermarked shapes is *identical* to the one based on the original shapes (which is omitted for brevity). One can also extract meaningful information on the dendrogram of the watermarked shapes, which captures all the information that the original dendrogram conveys. Similar species are still grouped together and therefore the mining capacity of the dataset is not lost.

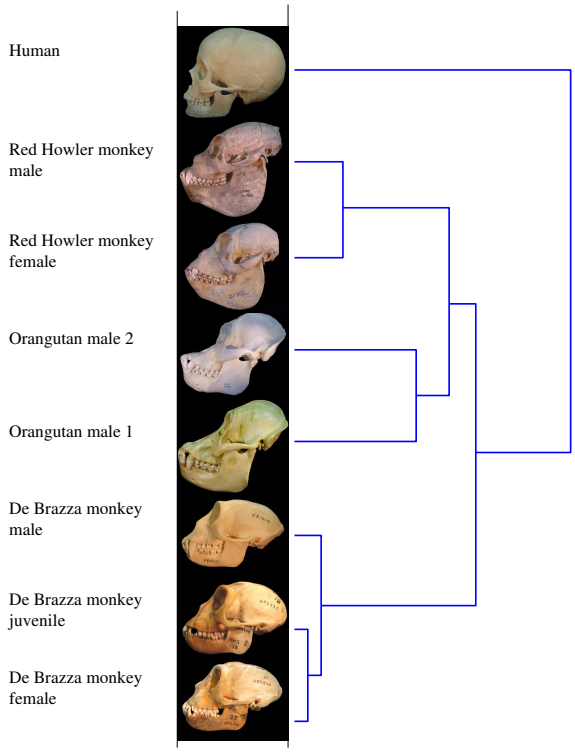


Figure 10: Watermarked Shapes and Dendrogram Preservation

5.2 Resilience to attacks

After we have determined the best watermark embedding power and having shown that geodesic distances are not distorted by the presence of the watermark, we test the watermark detectability under various adversary attacks. Any recipient of the dataset can perform a number of operations, with either malicious or innocuous intentions, which can potentially diminish the detectability of the embedded watermark. We assume that any of the possible transformations can only be performed up to a certain intensity degree, so as not to destroy the actual value and usability of the dataset (i.e., a shape cannot be completely distorted). In each test the embedding power p utilized, is the maximum power that preserves the original spanning-tree of all shapes in the dataset.

We examine the effectiveness of our watermarking methodology under four types of attacks.

■ **Geometric transformations**, such as global translation or rotation of the objects, do not distort a shape (or change the relative position of the objects), but may destroy a watermark if it is not embedded in the proper space.

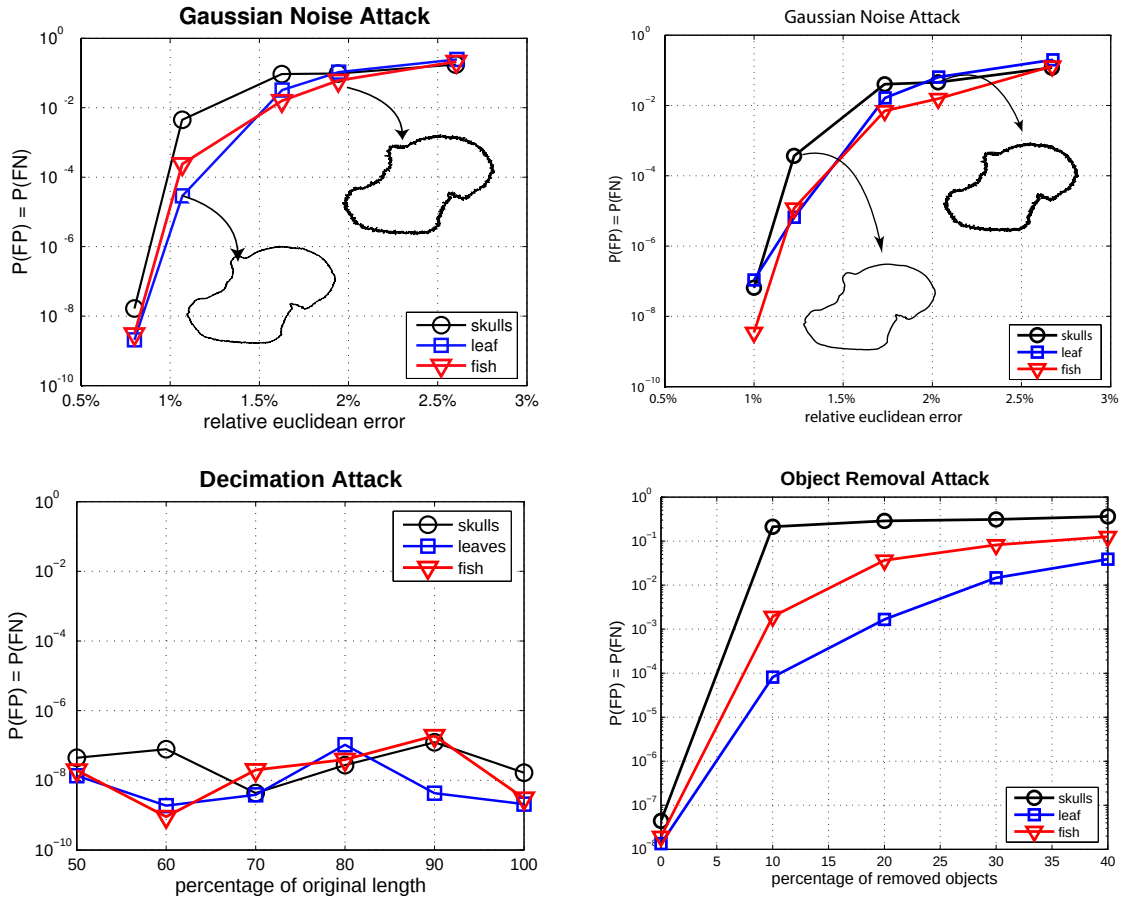


Figure 11: (a) Gaussian Noise in space; notice that the rightmost shape is very distorted, (b) Gaussian Noise in Frequency (c) Decimation and (d) Data reduction attacks.

We construct such attacks by watermarking the shapes and then applying random translations, scalings and rotations on each of the objects (same geometric transformation on each one of them -otherwise the relationship between the objects may change). The experiment is repeated 100 times and in Table 3 we report the average false positive/negative rates after each geometric transformations.

	$P(FP) = P(FN)$ after			
dataset	no attack	translation	+scaling	+rotation
skulls	7.4E-9	1.4E-9	6.6E-9	1.2E-7
fish	5.8E-9	3.1E-9	3.7E-9	1.0E-9
leaves	5.9E-9	6.1E-9	1.1E-8	6.8E-8

Table 3: Geometric attacks

Geometric attacks can potentially be harmful for certain watermarking schemes, such as approaches that change the least significant bits of an object [1, 18]. For our methodology the high detectability of the watermark is guaranteed by the properties of the Fourier descriptors. In fact, for each of the datasets, false positive/negative rates are in the area of 10^{-9} and the minute changes are typically attributed to rounding errors.

■ **Noise addition** is a more critical attack because it can potentially destroy the embedded watermark. For this attack, we translate all points of each shape using a vector whose coordinates are drawn by a normal distribution with mean 0 and variances that result in .5%−3% relative Euclidean error compared to the original shape. In Figure 11(a) we plot the results on the watermark detectability and on the same figure we overlay the distortion caused on the shape by the addition of noise. From the figure it is apparent that in order to erase the watermark an attacker would have to introduce a large error completely distorting the visual appearance of the shapes, rendering the dataset useless.

An adversary may also add Gaussian noise in the frequency domain, which is where the watermark is embedded. The results for this attack are depicted in Fig. 11(b). Again, large amount of noise would need to be added which would destroy the dataset usability.

■ **Decimation:** On this attack each dataset shape is represented by a smaller set of points that best approximate the original shape contour. A shorter sequence is obtained by sampling equidistant points from the spline associated with the original sequence. Decimation is a significant attack, because even though it does not change significantly the shape, it allows the adversary to generate a new sequence which has no points in common to the original shape. In our tests (see Figure 11(c)), even when using half-length sequences, watermark detection is not affected.

■ **Object removal** from the dataset; in this type of attack, we are interested in evaluating watermark detection in

the presence of dataset amputations. Our detection technique relies partially on the fact that we can spread the watermark over all the shapes sequences in the dataset. Therefore, as more objects are removed from the dataset, detectability is reduced, but it is still kept at high levels (Fig. 11(d)). The `skulls` is the most affected dataset, because it consists of smallest number of objects (only 16).

■ **Double watermarking**; finally one can consider the situation where an attacker attempts to add one's own watermark and claim ownership of the dataset. For this type of attack the legitimate owner can simply present the original dataset which contains neither watermark (which of course the attacker cannot present), effectively resolving the ownership problem. Notice, this is the single attack that requires the existence of the original dataset.

To summarize, with these experiments we have shown that the detectability of the embedded watermark is not hindered at all by geometric transformations. Additionally, a malicious adversary would have to destroy the usability of the dataset (distort the shapes significantly) in an effort to erase the hidden watermark.

6. CONCLUSIONS

We have presented the first ownership protection mechanism for shapes with geodesic distance preservation, providing guarantees on the outcome for a wide class of mining algorithms. We have shown that the embedded ownership seal imparts a minimal visual distortion on object shapes and is very robust under a variety of attacks. Finally, our findings are verified and visualized empirically for anthropological and natural science data.

There are several possible avenues for extending and improving this work. For example, the runtime of the rudimentary and, rather, costly algorithm for the MST preservation, can be significantly improved by exploiting the triangle inequality, thus pruning a great amount of unnecessary distance calculations.

7. REFERENCES

[1] R. Agrawal and J. Kiernan. Watermarking Relational Databases. In *Proc. of VLDB*, 2002.

[2] C. E. Brodley, A. C. Kak, J. G. Dy, C. R. Shyu, A. Aisen, and L. Broderick. Content-based retrieval from medical image databases: A synergy of human interaction, machine learning and computer vision. In *Proceedings of the The Sixteenth National Conference on Artificial Intelligence, pp. 760-767*, 1999.

[3] K. Chen and L. Liu. Privacy Preserving Data Classification with Rotation Perturbation. In *ICDM*, pages 589–592, 2005.

[4] Z. Cheng and et al. A genome-wide comparison of recent chimpanzee and human segmental duplications. In *Nature 437, 88-93*, 2005.

[5] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure Spread Spectrum Watermarking for Multimedia. In *IEEE Transactions on Image Processing*, 1995.

[6] P. Das, N. Chakraborti, and P. Chaudhuri. Spherical Minimax Location Problem . In *Journal Computational Optimization and Applications, Vol. 18, No 3*, pages 311–326, 2001.

[7] A. Gandhi. Content-Based Image Retrieval: Plant Species Identification. In *Master thesis, Oregon State University*, 2002.

[8] V. P. George Economou and A. Ifantis. Geodesic Distance and MST-Based Image Segmentation. In *EUSIPCO*, 2004.

[9] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright. A New Privacy-Preserving Distributed k-Clustering Algorithm. In *SIAM SDM*, 2006.

[10] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the Privacy Preserving Properties of Random Data Perturbation Techniques. In *ICDM*, 2003.

[11] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas. Fast and effective retrieval of medical tumor shapes. In *IEEE Transactions on Knowledge and Data Engineering, vol. 10, pp. 889-904*, 1998.

[12] R. Lee, J. Slagle, and H. Blum. A Triangulation Method for the Sequential Mapping of Points from N-Space to Two-Space. In *IEEE Transactions on Computers*, 1977.

[13] L. Liu, M. Kantarcioglu, and B. Thuraisingham. The applicability of the perturbation model-based privacy preserving data mining for real-world data. In *ICDM International Workshop on Privacy Aspects of Data-Mining*, 2006.

[14] S. Oliveira and O. Zaiane. Privacy preserving clustering by data transformation. In *SBBD*, 2003.

[15] E. Petrakis and C. Faloutsos. Similarity Searching in Medical Image Databases. In *IEEE Transactions on Knowledge and Data Engineering, vol. 9, pp. 435-447*, 1997.

[16] S. J. Shyu, Y. T. Tsai, and R. Lee. The Minimal Spanning Tree Preservation Approaches for DNA Multiple Sequence Alignment and Evolutionary Tree Construction. In *Journal of Combinatorial Optimization 8*, pages 453–468, 2004.

[17] R. Sion, M. Atallah, and S. Prabhakar. Rights Protection for Relational Data. In *IEEE TKDE, Vol 16, No 6*, 2004.

[18] R. Sion, M. J. Atallah, and S. Prabhakar. Rights Protection for Discrete Numeric Streams. In *IEEE Trans. Knowl. Data Eng. 18(5): 699-714*, 2006.

[19] O. Soderkvist. Computer Vision Classification of Leaves from Swedish Trees. In *Master thesis, Linkoping University, Sweden*, 2001.

[20] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science v.290 no.5500*, pages 2319–2323, 2000.

[21] J. Vaidya and C. Clifton. Privacy-Preserving K-Means Clustering over Vertically Partitioned Data. In *SIGKDD*, 2003.

[22] L. Wei, E. Keogh, X. Xi, and S.-H. Lee. Supporting Anthropological Research with Efficient Rotation Invariant Shape Similarity Measure. In *Journal of the Royal Society Interface*, 2007.

[23] X. Xi, E. Keogh, L. Wei, and A. Mafra-Neto. Finding Motifs in Database of Shapes. In *Proc. of SDM*, 2007.

[24] D. Yankov and E. Keogh. Manifold Clustering of Shapes. In *Proc. of ICDM*, 2006.

[25] H. Yu, J. Vaidya, and X. Jiang. Privacy-Preserving SVM Classification on Vertically Partitioned Data. In *PAKDD*, pages 647–656, 2006.