ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-1173

Geometrical Feature Extraction for Cuneiforms

Ariella Richardson [1] and **Uzy Smilansky**[2 3]

Abstract. Cuneiform writing is one of the earliest methods of writing in human history. It is based on pressing a stylus (reed) on clay tablets, resulting in wedge marks (cuneiforms) which, when combined, provide meaningful symbols. Applying modern machine learning methods to the study of ancient cuneiform tablets is a fascinating task. In the present paper we describe a method for extracting geometrical features of the wedges imprinted by the stylus that is used for writing. We introduce two independent feature extraction methods to describe the wedges. The data for this study come from precise optical scans of three tablets, originating from different historical periods. We use these tablets to demonstrate the validity of our extracted features, and to demonstrate the accuracy of classifying the different tablets.

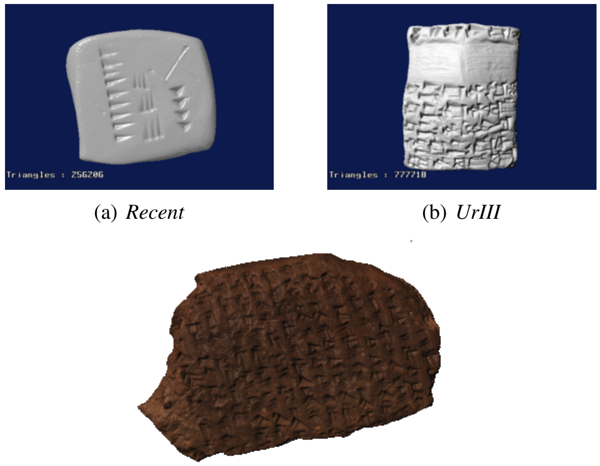
1 INTRODUCTION

Using machine learning for studying human writing has long been an active research area. Studies on modern handwriting, cover many diverse applications, such as character and word recognition, signature verification, and even the study of handwriting deficiencies [13, 15]. Machine learning is also used in the study of ancient writing, termed Paleography. Wolf et al. [19] study the The Cairo Genizah, a collection of texts discovered in the late 19th century. They apply machine learning methods for joining manuscript-fragments that are part of the same manuscript.

In this study we take another step backward in time and apply machine learning to the study of ancient Cuneiform tablets. Cuneiform tablets provide a means to study our ancestors' world, and to understand early civilizations with their complex social structures and sophisticated cultures. This form of writing served the Mesopotamian civilizations for thousands of years, and the earliest cultural and scientific achievements of mankind are recorded in this way. Examples of the type of tablets we study are shown in Figure 1, where the indentations are the cuneiform symbols we are attempting to analyze. Each character is formed by imprinting the stylus (the sharpened reed used as a writing tool) into wet clay, forming an imprint named a wedge. Figure 2 shows examples of single wedges cut from the tablets.

Some of the interesting questions to answer are: Did the shape of the stylus evolve over time? Does the same scribe always use a similar stylus? Are tablets written in certain areas similar to those written in others? Can one automatically join fragments of tablets? Are tablets from the same periods similar to each other?

In order to help answer these types of questions, we propose a feature extraction method for the cuneiform wedges. Our first contribution is describing the wedges, and therefore the stylus, as it is the inverse of the imprinted wedges. Our second contribution is showing how our feature extraction can be used by classification algorithms to classify different tablets.



(a) *Recent* (b) *UrIII*

(c) *Assyrian*

Figure 1. Scanned Clay Cuneiform Tablets.

Using scanners which are based on the structured light technique [17] a 3D scan of the cuneiform tablets can be made. These scans enable harnessing computer science in general and machine learning in particular to the study of cuneiform tablets. Previous applications enable visualization [3, 7], and attempt to perform Optical Character Recognition (OCR) [1, 10, 9, 2, 6].

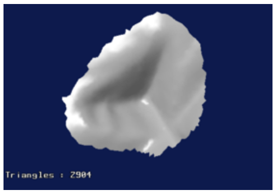
We focus on single wedges, and not on the cuneiform characters which are a combination of several wedges. This is analogous to using single pen strokes rather than letters when studying modern writing as suggested in [15]. We use the extracted wedge features for classification of cuneiform tablets.

Our feature extraction is based on viewing the wedge as a three dimensional corner generated by the intersection of three planes (tetrahedron), as illustrated in Figure 3, and is inspired by archaeological studies [12, 11].

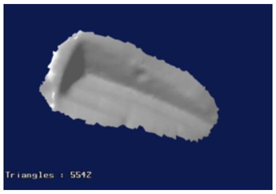
We use three cuneiform tablets to demonstrate our feature extraction. We will show the stylus features extracted for each of the tablets, and show that we obtain a precise measure. We also show how using our features for the classification of tablets results in high accuracy.

The paper is structured as follows: Section 2 discusses some related work. Section 3 introduces our feature extraction method. Sections 4 and 5 describe the data we used and our experimental results. We conclude in Section 6.

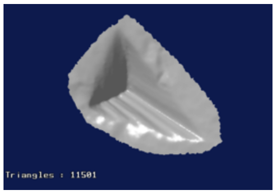
[1] Jerusalem College of Technology, Jerusalem, Israel.
[2] Department of Physics of complex Systems, The Weizmann Institute of Science, Rehovot, Israel.
[3] The computerized Archaeology laboratory, Institute of Archeology, The Hebrew University, Jerusalem Israel.



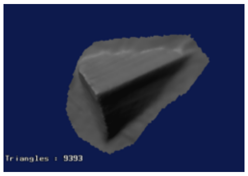
(a) wedge from *UrIII* tablet



(b) wedge from *UrIII* tablet



(c) wedge from *recent* tablet



(d) wedge from *recent* tablet (external view)

Figure 2. Single wedges

2 BACKGROUND AND RELATED WORK

The data retrieved from the tablet scan is in the form of a 3D mesh. Using 3D scans of cuneiform tablets is not new, and has been studied using a variety of methods. One possible way to use the 3D scans is to extract the 2D information held in the scan and use image processing methods developed for 2D data such as the work of Malzbender et al. [8] who use Polynomial Texture Maps for tablet visualization. However, since the cuneiform symbols are inherently 3D, the depth of the symbols, the position of the tablet and many other features are of importance when reading the tablets. This motivates investigation of the 3D data.

Most research performed on 3D scans of cuneiform tablets involves visualizing the tablet such as [8, 3, 7]. The aim of the visualization is to provide a means to read the tablets without the need to physically have the tablet. As tablets are often kept in museums, and may be harmed by handling, this is in itself an important task.

Visualization of the tablets is not the only way computer science can contribute to the study of cuneiform tablets. An example to an important and interesting task involves extracting the cuneiform characters automatically. Hilal et al. [1] attempt to address this task by using 2D pictures of **hand written** copies of tablets. They use intensity curves for the drawn images to differentiate between characters.

Classification of cuneiform tablets was performed in the ECML/PKDD Discovery Challenge 3 [5]. In this challenge clay tablets were translated by specialists from cuneiform symbols to a set of ASCII characters. The task involved classifying these symbols in order to identify and date documents. There is also a Unicode character set that is specific to cuneiform. However, methods such as these require manual copying by a trained archaeologist for performing the transcription of the characters from the tablet. This is a slow task requiring the work of a trained archaeologist, which could benefit from the application of automated methods on 3D scanned data.

Using 3D scanning for feature extraction, that is later used as input for classification algorithms has been performed by Richardson et al. [16]. Their work was performed on a lithic artifacts. As the features for cuneiforms are inherently different to those of lithic artifacts, this work cannot be transfered directly to our problem. However, their success indicates that computerized analysis is beneficial to archaeological studies, and feature extraction is a beneficial step.

Active research on character extraction from 3D scans is being performed by Mara [9] who also introduced GigaMesh [10] and by Gerfrid, Muller and Fisseler [2, 6]. The approach used by Mara et al. [10] is based on capturing the volume of character that is below the surface. The volume is found using the intersection of multiple spheres with the area below the tablet surface. This study provides an enhanced visual 3D picture of the tablet focusing on enhancing the characters. In other work [9], Mara shows how the characters that are found can be used as a basis for OCR.

Most similar to our work is a research project that is currently being conducted by Gerfrid, Muller and Fisseler [2, 6]. Gerfrid et al. propose a "top-down" method. They use the full tablet, extract single wedges from the tablet and then analyze them. They model the cuneiform wedges as tetrahedrons (as we do), find the surface that best fits the tablet, and use it to extract the wedges from the tablet. Once the wedges are found, their features are estimated. The wedge features are extracted by segmenting each wedge into three parts using the mesh cell normals, and finding the normals for each face. They then derive the wedge edges from the faces.

In contrast we propose a "bottom-up" method. We perform this study under the assumption that the wedges have already been cut from tablet, and focus on extracting precise features for each wedge. We propose two sets of features describing the wedge imprinted by the stylus. Both sets of features describe the angles between the wedge edges. However we derive one set of features from the face area of the wedge, and the other set from the curved edge areas, resulting in two independent feature sets. Our proposed extracted features provide a precise robust description of the wedges on a tablet.

3 FEATURE EXTRACTION METHOD

Our feature extraction algorithm uses the digitized 3-D model of the surface as input. The focus of our study was to extract features for a single wedge, which is a single imprint of the stylus. The wedges we use are cut from the scan of the clay tablet manually, as shown in Figure 2. In order to explain the various parts of our feature extraction method, we will use an example wedge shown in Figure 3. This single wedge is viewed from the outside.

A wedge can be viewed as a triangular pyramid with three triangular faces emerging from a common vertex. Thus, the angles between the edges of the three triangles (α, β, γ in Figure 3) uniquely define the geometry of the wedge, and thus the geometry of the stylus that imprinted the wedge.

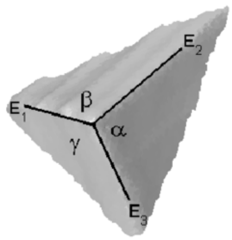


Figure 3. A scanned model of a wedge. The edges E_1, E_2, E_3 we expect to obtain, and the angles α, β, γ between them.

The first stage of the wedge analysis consists of segregating the domains where the edges and faces appear on the measured model (coloured in red and blue , respectively, in Figure 4). The pyramid geometry , namely, the angles α, β, γ on the pyramid, is now extracted in two independent ways, which are based on different sets of points on the surface. One method makes use of the edge domains, and the other uses the face domains. We combine local measures such as the local curvature of the mesh, together with global features such as the plane or line passing through large areas of the mesh.

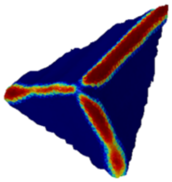


Figure 4. A wedge separated into high (red) and low (blue) curvature areas.

3.1 Preprocessing

We proceed to describe how the geometrical features are extracted from the 3D scan. The 3D scan is a triangular mesh composed of a list of triangular cells and vertices. The triangular cells are often termed faces, however in order to differentiate between the mesh faces to the wedge faces, we will refer to them as mesh cells. There is no information on which parts of the mesh compose each face, or the edges and tip of the wedge, therefore we propose a method to find them. First we will describe some preprocessing we perform on the wedge mesh, and then the two feature extraction methods.

3.1.1 Cap

We first restrict the area used in the analysis to avoid the vicinity of the wedge boundaries where the surface shows curving, or deformations typical for elongated wedges. This is done by using cells that are within a sphere centered at the deepest point of the wedge. We term these cells the *Cap* cells. The radius of the sphere is determined by the depth of the wedge. The sphere can be seen in Figure 5 as the blue area, while the grey areas on the wedge are not included in the analysis. The restriction of the wedge area to the *Cap* is beneficial to our study since we are focusing on finding the common features of the various wedges on a specific tablet, in order to model the stylus or classify different tablets. If we were to study the differences between different wedges, for example in order to perform OCR, then this stage would be eliminated.

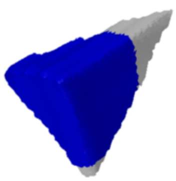


Figure 5. The cap (blue) area of a wedge.

3.1.2 Curvature measures

We calculate the curvature measures as proposed by Cohen-Steiner and Morvan [4]. The curvature measures calculated for each cell are: the direction of minimum/maximum curvature ($Umin$/$Umax$), minimum/maximum curvature ($Cmin$/$Cmax$) , the Gaussian curvature ($Cgauss$) and the normal to the surface of the cell ($Norm$).

3.2 Edge Features

The algorithm which analyzes the edge domains (coloured red in Figure 4) finds the local edge direction using the direction of the minimum surface curvature. The resulting directions are combined to define the best fitting lines which emerge from a single tip point. These lines are the edges of the wedge, and the angles $\alpha_E, \beta_E, \gamma_E$ between these edges are our first set of features.

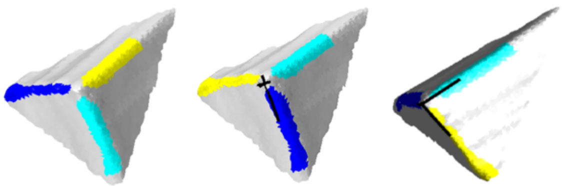
The algorithm $ExtractEdgeFeatures$ is described in Algo. 1. The input to the algorithm $ExtractEdgeFeatures$ is the mesh of the *Cap* , with curvature measures and a threshold for excluding the tip area. The output is the angles between the edge vectors E_i, $i = 1, 2, 3$ which define the wedge edges. These angles are denoted by $\alpha_E, \beta_E, \gamma_E$.

Algorithm 1 ExtractEdgeFeatures
Input: $Cap, Cmax(Cap), Cmin(Cap), Umin(Cap), t$
Output: $\alpha_E, \beta_E, \gamma_E, tip_E$

$M = Cmax(Cap) > mean(Cmax(Cap))\&Cmin(Cap) < t$
$[M_1, M_2, M_3] = Kmeans(Umin(M), 3)$
for all $i = 1$ to 3 **do**
  $E_i =$ vector closest to $Umin(M_i)$
$tip_E =$ point with $max(Cgauss)$
$\alpha_E = angleBetween(E_2, E_3)$
$\beta_E = angleBetween(E_3, E_1)$
$\gamma_E = angleBetween(E_1, E_2)$

The cells used for the *edge features* are selected from the *Cap* cells. We use cells on the *Cap* that have a maximum curvature ($Cmax$) that is **above** average, this selects cells that are on the curved areas of the wedge. We exclude the cells that are in the tip area by removing cell with a low minimum curvature ($Cmin$).

Since we wish to extract three edges defined by these cells, we must segregate the cells into three sets, each corresponding to a single edge. This is done by clustering the direction of the minimum curvature $Umin$ into 3 clusters, using K-Means. The cells that are used in this stage, clustered into the 3 clusters appear in Figure 6(a). The output clusters of mesh cells are labeled M_1, M_2, M_3.



(a) (b) (c)

Figure 6. (a) Wedge with areas of high curvature on *Cap* selected for the edge analysis. Each color correlates to a cluster. Displays the extracted edges as shown from the outside (b) and from the inside (c) of the wedge.

For each of the 3 clusters, we now obtain a vector that represents the edge belonging to this cluster. The edge direction is obtained using the direction of the minimum curvature ($Umin$), by finding the average direction for all the cells in this cluster. The tip of the wedge is estimated to be the point with the highest Gaussian curvature ($Cgauss$), and the edges are lines that come out of the tip in the edge direction. The stylus is not a perfectly straight tetrahedron, and the estimated edges cut through the wedge. Figure 6(b) shows the extracted edges as seen from the outside of the wedge, and Figure 6(c) shows the edges as viewed from the inside of the wedge.

3.3 Face Features

The second extraction algorithm uses the face area (coloured blue in Figure 4) and finds the best fitting planar faces and tip point. From the intersection of these planes we derive our second set of edges and thus the angles $\alpha_F, \beta_F, \gamma_F$ between the wedge edges, and creates our second independent feature set.

The algorithm $ExtractFaceFeatures$ is described in Algo. 2. The input to the algorithm $ExtractFaceFeatures$ is the mesh of the *Cap*, with curvature measures and a threshold for selecting curvature. The output is the angles between the edge vectors E_i, $i = 1, 2, 3$ which define the wedge edges. These angles are denoted by $\alpha_F, \beta_F, \gamma_F$.

Algorithm 2 ExtractFaceFeatures
Input: $Cap, Cmax(Cap), Cmin(Cap), Norm(Cap), t$
Output: $\alpha_F, \beta_F, \gamma_F, tip_F$

```
M = Cmax(Cap) < mean(Cmax(Cap))&Cmin(Cap) < t
[M_1, M_2, M_3] = Kmeans(Norm(M), 3)
for all i = 1 to 3 do
    N_i = vector closest to Norm(M_i)
repeat
    tip_F = getTipPointFaces(N_1, N_2, N_3, M_1, M_2, M_3)
    for all i = 1 to 3 do
        N_i = getNormalFromTip(tip_F, Norm(M_1),
        Norm(M_2), Norm(M_3))
until convergence
F_1 = N_2 × N_3
F_2 = N_3 × N_1
F_3 = N_1 × N_2
α_F = angleBetween(F_2, F_3)
β_F = angleBetween(F_3, F_1)
γ_F = angleBetween(F_1, F_2)
```

As for the *edge features*, we use the *Cap* cells of the wedge. However for the *face features* we use the cells on the *Cap* that have a maximum curvature ($Cmax$) that is **below** average. We again exclude the cells that are in the tip area by removing cell with a low minimum curvature ($Cmin$).

We cluster the cells into three clusters, one for each wedge face. This is done by using K-Means on the cell normals ($Norm$), $K = 3$. The output clusters of mesh cells are labeled M_1, M_2, M_3. The low curvature area used in this part of the analysis appears in Figure 7(a). Each color represents a cluster found using K-means.

Once we have 3 clusters of cells, we find the vector that best represents the cell normals, this is our first guess for the normal to the wedge face (and is also the method developed independently by [6]). As the normals do not always provide a precise representation of the wedge faces, we improve the analysis further. We iteratively generate the wedge tip by fitting a plane through the cluster cells with the estimated normal, and then refining the normal by fitting planes to the cells using the estimated tip. When the process converges, we have the estimated tip, along with three wedge face normals. The cross product of these normals provides us with 3 edges, and the angles between them are the *face features*. The edges extracted are shown in Figure 7(b).

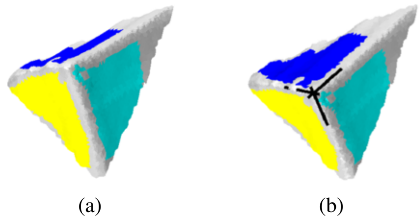


Figure 7. (a) Wedge with areas of low curvature on *Cap* selected for the face analysis. Each color correlates to a cluster. (b) displays the extracted edges as shown from the outside of the wedge.

3.4 Curvature Features

Curvature measures are also descriptive of the wedges, and are natural candidates for wedge features. We use the average and standard deviation of the curvature measures $Cmin$,$Cmax$,$Cgauss$ described in Section 3.1.2. We calculate the curvature measures over the face area (coloured blue in Figure 4), and label this feature set $curvF$, and separately over the edge area (coloured red in Figure 4) and label this feature set as $curvE$.

4 DATA SETS

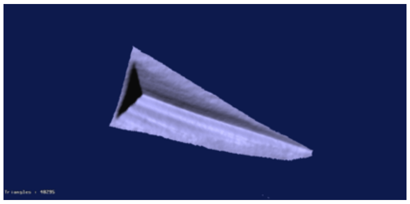
4.1 Tablets

There are three tablets used in this study. The first is a *recent* tablet prepared especially for the present study by Prof J Marzahn, see Figure 1(a). The second is the *UrIII* tablet from the archaeological collection of the Hebrew University - Jerusalem, shown in Figure 1(b), and the third an *Assyrian* tablet displayed in Figure 1(c) from the Heidelberg Academy of Sciences and Humanities (Forschungsstelle Literarische Keilschrifttexte aus Assur).

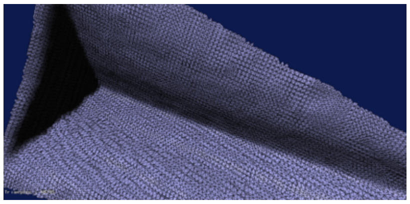
From each tablet we manually extracted single wedges. There are 25 wedges in the *recent* set , 20 wedges in the *UrIII* set, and 8 in the *Assyrian* set. The wedges are of a variety of orientations and shapes and sizes.

4.2 Scanning

We use scanners which are based on the structured light technique [17]. They are available commercially, and can achieve high spatial accuracy, exceeding 0.05mm. This level of precision is sufficient to resolve various features which appear on the wedge surfaces, the most prominent are the imprints of micro-tubes or capillary lines of the plant (separated by approximately .65mm) along the reed sections (Figure 8(a)), as well as faults which are due to the granularity of the clay, damages and cracks. The output data are stored as digitized triangulated surfaces as shown in Figure 8(b). The tablets and the stylus tips were scanned at the Computerized Archaeology Laboratories, The Hebrew University, Jerusalem, and by the commercial firm TrigonArt, Berlin.



(a) Full wedge surface note striations imprinted by the reed micro-tubes

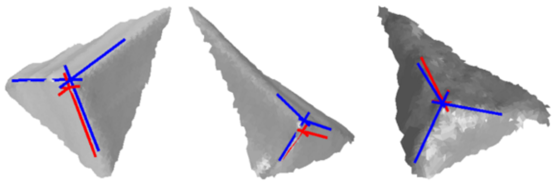


(b) Close-up of wedge surface, with mesh

Figure 8. Digitized images of single wedge.

5 EXPERIMENTAL RESULTS

For each of the wedges in each of the tablets we extracted both the *face features* and the *edge features*. Examples of wedges and their extracted features appear in Figure 9. The the *face features* are visible in blue. The *edge features* in red, are partially visible (some parts of the edges are not visible, since they are inside the wedge).



(a) (b) (c)

Figure 9. Wedges with face features (blue), and edge features (red).

5.1 Wedge Feature Evaluation

The first result describes the features of the stylus that is estimated to have been used for each tablet. The stylus is estimated by averaging the wedge features that we extracted for all the wedges on each tablet. The *recent* tablet was scanned at two resolutions referred to as low and high, respectively.

As the edges are the intersections of the wedge faces, the two sets of features are easily obtainable from each other. If we were to obtain a perfect scan, and if the stylus were shaped like a perfect geometrical tetrahedron, these features would be expected to coincide. However, since the stylus is not perfect, the clay is granular and moreover, the scanning resolution is always finite, the features do not coincide. Note also that the faces are not always perfect planes: indeed Figure 8(a) shows clearly a set of parallel deformations on one of the faces. These originate from the reed tubular structure.

The mean values and standard deviations for features in all data sets are summarized in Table 1. The results for the *recent* tablet indicate that under favorable conditions our method can achieve a margin of uncertainty of 1 for the extracted angles. The margin of uncertainty for the *UrIII* and the *Assyrian* data are higher than the one seen in the *recent* tablet data. This might be due to various possible causes lower scan resolution, more rounding of the edges and deformations caused by the proximity of other wedges. Comparing the three sets of data, one can see significant differences between the geometries, showing that the *UrIII* stylus is less sharp than the *recent* stylus. The *Assyrian* stylus has even larger angles, meaning that it is even wider than the others. We wish to mention that in an early stage of our study we investigated the use of the face normals alone to find the edges (as in [6]), and found a much larger margin of uncertainty in our results. This lead us to the more precise method we presented.

Table 1. Wedge angles

	Feature	α	β	γ
recent (high)	face	88.7 ± 0.6	92.4 ± 0.6	96.5 ± 0.8
	edge	89.6 ± 1.0	88.2 ± 0.6	93.0 ± 1.4
recent (low)	face	90.9 ± 0.5	92.8 ± 0.4	97.6 ± 0.7
	edge	88.3 ± 0.7	87.2 ± 1.0	89.8 ± 1.1
URIII	face	92.0 ± 1.1	98.6 ± 1.1	102.9 ± 1.4
	edge	95.1 ± 1.8	98.1 ± 1.5	95.7 ± 1.3
Assyriyan	face	89.1 ± 2.5	100.2 ± 3.3	117.6 ± 2.1
	edge	104.6 ± 5.2	102 ± 4.9	105.7 ± 5.6

5.2 Tablet Classification

The second question we address is whether the extracted features can be used to differentiate between the tablets. We use three types of features the *edge features* and *face features* that we defined in Sections 3.3 and 3.2 and the *curv features* described in Section 3.4.

For each wedge we form a feature vector using either the *edge features*, the *face features*, the *curv features* or combinations of these features (see Table 2 for details). Each feature vector is labeled with the tablet name. We perform pairwise classification of wedges from the three tablets. We use Decision Trees (DT) and Support Vector Machine (SVM) as the classification algorithms. We used the default Matlab DT (*classregtree*) and the default Matlab SVM (*svmtrain*) with a Gaussian radial basis function kernel with a scaling factor of 1 (as proposed in [18]), and Sequential Minimal Optimization [14]. The pairwise classification accuracy was measured using random 10-fold cross validation (for the *Assyrian* set we used 8-folds). Each test was repeated 50 times.

Classification results can be seen in Table 2. For the classification of the *recent* set vs. the *UrIII* set, the best accuracy of 99% was obtained using $curvE$ with DT. The *UrIII* and *Assyrian* classification was most accurate, with an accuracy of 90% when using either the *edge features* with SVM or the *face features* combined with $curvF$ and using DT or when combining all features with DT. Classifying the *recent* vs. *Assyrian* sets was 100% accurate for a variety of methods. When averaging the different methods over the 3 datasets, we found that SVM with the combined *edge and face features*, or DT with either the *face features* combined with $curvF$ or combining all features, has the best classification results (with an average accuracy of 95%). It is interesting to note that although we found that the *face features* had a lower margin of uncertainty than the *edge features* (Section 5.1), the *face features* do not necessarily provide better classification results than the *edge features*. Overall the classification accuracy rates are high, indicating the features we extracted are useful for tablet classification.

Table 2. Classification Results

Feature	Algo	*recent* vs. *UrIII*	*recent* vs. *Ass*	*UrIII* vs. *Ass*	Average
face	DT	0.85	1	0.89	0.92
face	SVM	0.88	1	0.89	0.92
edge	DT	0.81	0.77	0.88	0.82
edge	SVM)	0.87	1	0.9	0.92
edge& face	DT	0.86	1	0.86	0.90
edge& face	SVM	0.97	1	0.87	0.95
curvF	DT	0.94	0.99	0.63	0.85
curvF	SVM	0.97	0.97	0.88	0.94
curvE	DT	0.99	1	0.77	0.92
curvE	SVM	0.96	0.9	0.72	0.86
curvF & curvE	DT	0.93	0.99	0.61	0.84
curvF & curvE	SVM	0.9	0.92	0.75	0.86
face & curvF	DT	0.94	1	0.9	0.95
face& curvF	SVM	0.93	0.95	0.88	0.92
edge& curvE	DT	0.99	1	0.76	0.92
edge & curvE	SVM	0.9	0.95	0.74	0.86
all	DT	0.94	1	0.9	0.95
all	SVM	0.73	0.89	0.67	0.76

6 CONCLUSION

This study was motivated by archaeological questions on cuneiform tablets, such as: What is the exact shape of the stylus used for imprinting cuneiform tablets? How can one classify tablets ? We used 3D scans of the cuneiform tablets, and cut out single wedges that are imprinted by a single impression of the stylus.

Our study presents a method to extract geometrical features from these cuneiform wedges. We introduced two types of features that can be extracted from the wedge. One feature set extracts information from the high curvature (edge) area of the wedge, and the other uses the low curvature (face) areas. These two feature sets are independent, as they different areas of the scanned wedge.

We used three scanned cuneiform tablets to demonstrate the accuracy of the extracted features, and showed that the stylus geometry can be measured to within the accuracy of 1-2 degrees.

We demonstrated how these features, and other curvature features can be used to classify the tablets. Our experimental results show that the features we propose can be used successfully to classify the different tablets we use (with an average accuracy of 90%). This will enable applying our method to the classification task of other tablets, in archaeological study.

There are several possible extensions of this work. In this study the wedges we used were manually cut from the full scan of the tablet. In order to increase efficiency we should integrate an automated method to extract the wedges such as the one proposed by Fisseler et al. [6]. Another interesting extension, is to classify wedges based on other measures such as their orientation, depth, size to enable further study of the differences between various scribes. It would also be interesting to combine several wedges to create the cuneiform symbols, and use them for OCR.

ACKNOWLEDGEMENTS

The support from the Braginsky Center for the Interface between Science and the Humanities at the Weizmann Institute is gratefully acknowledged. We would like to thank Professor Joachim Marzahn for introducing us to this problem, for many discussions and for providing the *recent* tablet. For instructive discussions, and the use of their clay tablets we are indebted to Professor Stefan Maul (*Assyrian Tablet*) and Dr Uri Gabbay (*UrIII Tablet*).

REFERENCES

[1] Hilal M. Yousif Al-Bayatti, Abdul Munim Rahma, and Haithem Alani, 'Cuneiform symbols recognition using intensity curves', *Int. Arab J. Inf. Technol.*, **3**(3), 237–241, (2006).

[2] Michele Cammarosano, Gerfrid G.W. Mller, Denis Fisseler, and Frank Weichert, 'Schriftmetrologie des keils: Dreidimensionale analyse vonkeileindrcken und handschriften', in *Die Welt des Orients*, (2014).

[3] J. Cohen, D. Duncan, D. Snyder, J. Cooper, S. Kumar, D. Hahn, Y. Chen, B. Purnomo, and J. Graettinger, 'iClay: digitizing cuneiform', in *Proceedings of the Fifth International Symposium on Virtual Reality, Archaeology, and Cultural Heritage*, pp. 135–143, (2004).

[4] David Cohen-Steiner and Jean-Marie Morvan, 'Restricted delaunay triangulations and normal cycle', in *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, SCG '03, pp. 312–321, New York, NY, USA, (2003). ACM.

[5] Dimitris Diamantis Dimitrios Mavroeidis and Michalis Vazirgiannis, 'Using semi-supervised learning for mining sumerian administrative documents in the kingdom of the iii dynasty of ur', in *Proceedings of the ECML/PKDD2007 Discovery Challenge*, pp. 76–82, (2007).

[6] Denis Fisseler, Frank Weichert, Gerfrid G.W. Mller, and Michele Cammarosano, 'Towards an interactive and automated script feature analysis of 3d scanned cuneiform tablets', in *Scientific Computing and Cultural Heritage*, Contributions in Mathematical and Computational Sciences, Springer Berlin Heidelberg, (2013).

[7] Hendrik Hameeuw and Gert Willems, in *Akkadica*, (2013).

[8] Tom Malzbender, Dan Gelb, and Hans Wolters, 'Polynomial texture maps', in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pp. 519–528, New York, NY, USA, (2001). ACM.

[9] Hubert Mara, *Multi-Scale Integral Invariants for Robust Character Extraction from Irregular Polygon Mesh Data*, Ph.D. dissertation, Ruprecht-Karls-University, Heidelberg, 2012.

[10] Hubert Mara, Susanne Krömker, Stefan Jakob, and Bernd Breuckmann, 'Gigamesh and gilgamesh: –3D multiscale integral invariant cuneiform character extraction', in *Proceedings of the 11th International conference on Virtual Reality, Archaeology and Cultural Heritage*, VAST'10, pp. 131–138, Aire-la-Ville, Switzerland, Switzerland, (2010). Eurographics Association.

[11] Joachim Marzahn and Tobias Krmer, in *Schreiben mit Keilschrift*, (2013).

[12] Leopold Messerschmidt, 'Zur technik des tontafelschreibens', in *Orientalistische Literatur-Zeitung*, (1906).

[13] Rejean Plamondon and Sargur N. Srihari, 'Online and off-line handwriting recognition: a comprehensive survey', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(1), 63–84, (Jan 2000).

[14] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines, 1998.

[15] Ariella Richardson, Sarit Kraus, Patrice L. Weiss, and Sara Rosenblum, 'COACH - cumulative online algorithm for classification of handwriting deficiencies', in *IAAI'08 Proceedings of the 20th national conference on Innovative applications of artificial intelligence*, pp. 1725–1730, (2008).

[16] Eitan Richardson, Leore Grosman, Uzy Smilansky, and Michael Werman, 'Extracting scar and ridge features from 3d-scanned lithic artifacts', in *Proc. of the The Computer Applications and Quantitative Methods in Archaeology*, volume to appear, (2012).

[17] Joaquim Salvi, Jordi Pags, and Joan Batlle, 'Pattern codification strategies in structured light systems', *Pattern Recognition*, **37**(4), 827 – 849, (2004). Agent Based Computer Vision.

[18] Chih wei Hsu, Chih chung Chang, and Chih jen Lin. A practical guide to support vector classification, 2010.

[19] Lior Wolf, Rotem Littman, Naama Mayer, Tanya German, Nachum Dershowitz, Roni Shweka, and Yaacov Choueka, 'Identifying join candidates in the cairo genizah', *Int. J. Comput. Vision*, **94**(1), 118–135, (August 2011).