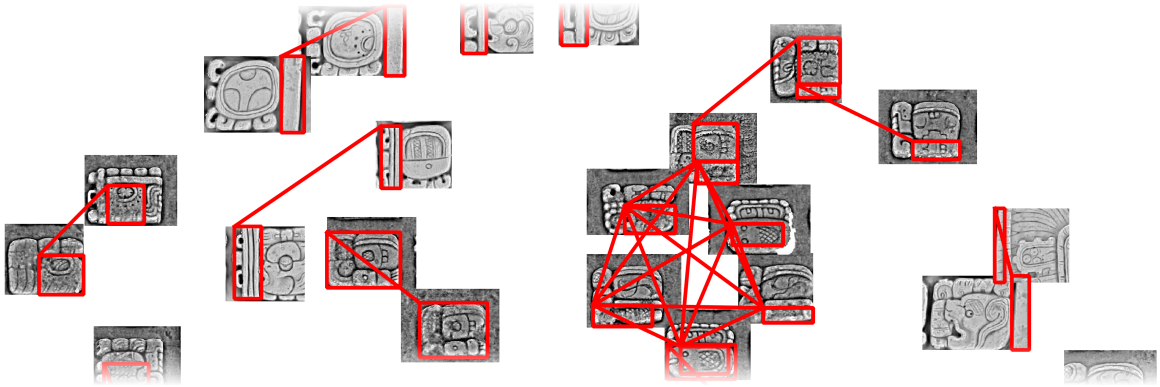
EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2018)
R. Sablatnig and M. Wimmer (Editors)

Visualizing Networks of Maya Glyphs by Clustering Subglyphs

B.Bogacz[1], F.Feldmann[1], C.Prager [2] and H.Mara[1]

[1]Heidelberg University, Interdisciplinary Center for Scientific Computing (IWR),
Forensic Computational Geometry Laboratory (FCGL), Germany
{bartosz.bogacz, felix.feldmann, hubert.mara}@iwr.uni-heidelberg.de
[2]Bonn University, Institute of Archaeology and Anthropology,
Department for the Anthropology of the Americas, Germany
cprager@uni-bonn.de



Abstract

Deciphering the Maya writing is an ongoing process that has already started in the early 19th century. Among the reasons why Maya hieroglyphic script and language are still undeciphered are inexpertly-created drawings of Maya writing systems resulting in a large number of misinterpretations concerning the contents of these glyphs. As a consequence, the decipherment of Maya writing systems has experienced several setbacks. Modern research in the domain of cultural heritage requires a maximum amount of precision in capturing and analyzing artifacts so that scholars can work on - preferably - unmodified data as much as possible. This work presents an approach to visualize similar Maya glyphs and parts thereof and enable discovering novel connections between glyphs based on a machine learning pipeline. The algorithm is demonstrated on 3D scans from sculptured monuments, which have been filtered using a Multiscale Integral Invariant Filter (MSII) and then projected as a 2D image. Maya glyphs are segmented from 2D images using projection profiles to generate a grid of columns and rows. Then, the glyphs themselves are segmented using the random walker approach, where background and foreground is separated based on the surface curvature of the original 3D surface. The retrieved subglyphs are first clustered by their sizes into a set of common sizes. For each glyph a feature vector based on Histogram of Gradients (HOG) is computed and used for a subsequent hierarchical clustering. The resultant clusters of glyph parts are used to discover and visualize connections between glyphs using a force directed network layout.

CCS Concepts

•Human-centered computing → Graph drawings; Information visualization; •Computing methodologies → Object identification; Cluster analysis; •Applied computing → Optical character recognition;

DOI: 10.2312/gch.20181346

Figure 1: *Panel 1 from the Maya site of* La Corona*. Acquired (with kind permission by* Instituto de Antropolog a e Historia de Guatemala*) using a structured light 3D scanner. Surface curvature computed using MSII, and visualization and rendering performed in GigaMesh (*`gigamesh.eu`*). This is one of the raster images of Maya panels used in our extraction and analysis pipeline.*

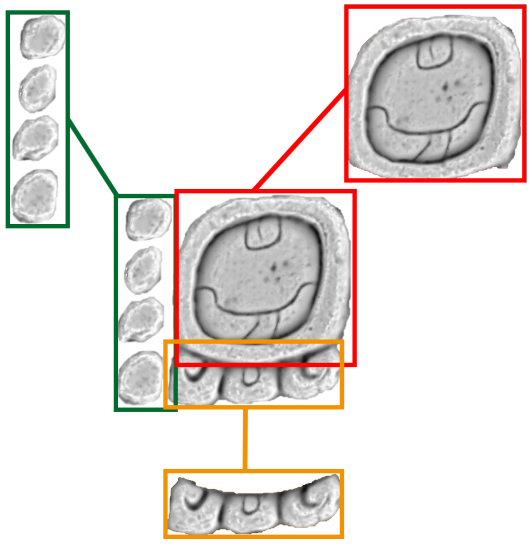


Figure 2: *The date symbol* KAN*, T506, with its annotated subglyphs. Due to their varying sizes, the subglyphs will be assigned to different clusters of similar image size. Subsequent visual clustering of the subglyphs is then performed independently for each cluster of image sizes.*

1. Introduction

Creating tools to simplify the analysis and the encoding of ancient historical materials such as the writing system of the Maya culture, or in any other domain of digital humanities, e.g. the retrieving of cuneiform characters [MK13] is vital for the recognition and discovery of patterns.

The interpretation and understanding of ancient Maya inscriptions requires the identification of the basic individual glyphs of their writing system. Currently this identification process is performed manually, using the printed catalogue by Thompson and Stuart [TS62], which contains all known deciphered and undeciphered glyphs.

This work develops a machine learning pipeline for an automated Maya glyph segmentation from the projected 3D scans and a subsequent clustering of the segmented glyps and their subglyphs. As a feature descriptor we use *Histogram of Oriented Gradients (HoG)*. The main idea of the descriptor is that the appearance and the shape of an object in an image can be represented by the local changes in intensity and edges. With this information we perform a hierarchical clustering to visualize similar glyphs and subglyphs on the images.

Retrieving Maya glyphs using Shape Descriptors has been addressed by researchers from the Idiap (Institut Dalle Molle d'intelligence artificielle perceptive) Research Institute in Switzerland. In [RRPOGP09] Maya glyphs have been retrieved with the *Shape Context* method by Belongie [BMP02]. The authors used binarized images that were drawings, human interpretations of the original glyph inscriptions. Each point of the glyph's shape is represented in an histogram characterized by its angle and distance from the root. By comparing the similarity of the histograms similar glyphs have been retrieved. Furthermore in [RRPOGP11] a new descriptor has been introduced *Histogram of Oriented Shape Context (HOOSC)*, which extends the *Shape Context* descriptor by Belongie with the distribution of the orientation of the *Shape Context* descriptor similar to *HOG* [DT05].

In [RRMM17] Rangel et al. present an approach to estimate a dimensionality reduction technique and visualize a corpus of Maya glyphs in 2D space using a pre-trained VGGnet [SZ15], a convolutional neural network architecture. By using the result of the dimensionality reduction as their ground-truth, the authors avoid dealing with the inherent challenges of data-sparsity in the analysis of Maya script. Further, the resulting visualization focuses on a high-level abstract overview, where each glyph is represented as a point. In our work, a-priori learning of the dataset is not necessary, allowing us to analyze a small corpus of glyphs where correspondences and shared similarities between glyphs are directly presented and highlighted.

2. Data Sets & Pre-processing

The data set consists of two 3D scans of Maya monuments, taken by Christian Prager during the exhibition "Maya. Das Rätsel der Königsstätte" in Speyer, Germany [SG16] (Project Text Database and Dictionary of Classic Mayan, University of Bonn). The glyphs in this data set were carved into stone. Consequently fewer people worked on the glyphs and there are less differences in the shapes of glyphs. Contrary to manually created tracings the glyphs are an exact depiction of the original inscriptions.

The La Corona panel dating 692 A.D. (Figure 1) is in its original size 55.5cm $\times$ 42.0cm in its dimensions. The panel assembles 65 glyph blocks, which have an average of 3.8cm $\times$ 4.3cm. The created 3D scan has been exported as an 2D image with a resolution

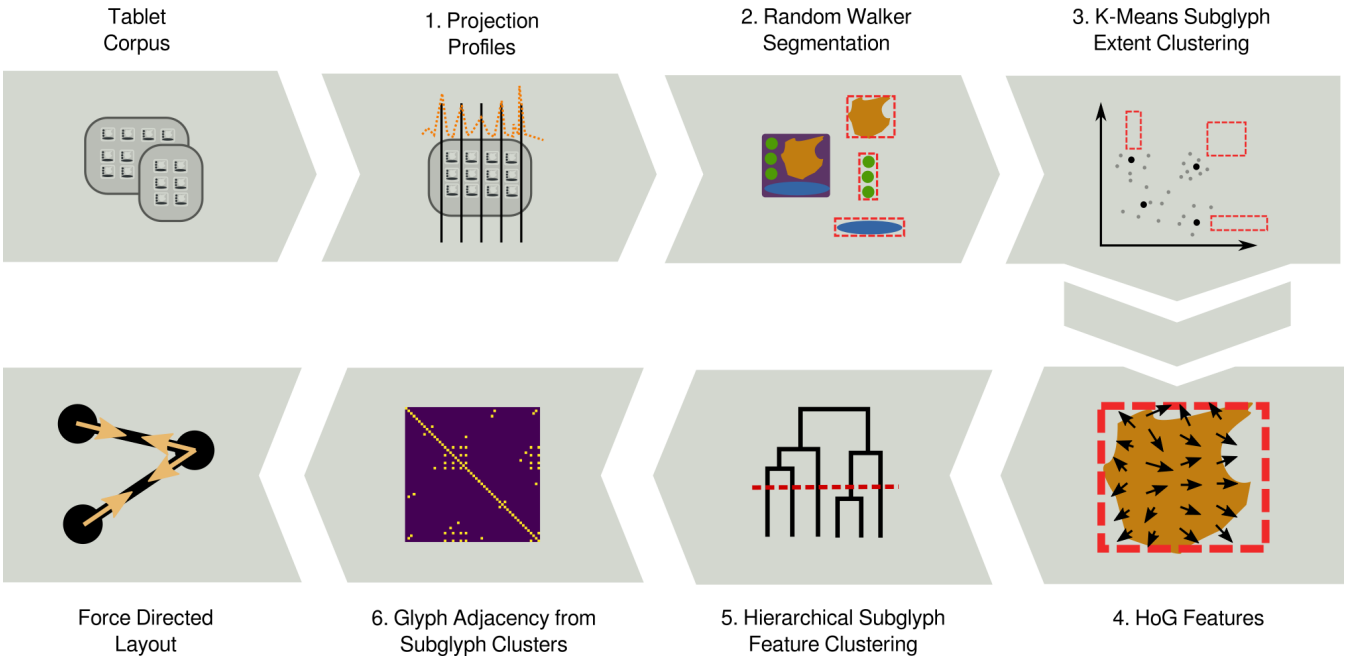


Figure 3: *Processing pipeline of our segmentation, clustering and visualization approach.*

of 6555×4966 pixles. Next, Cancuen, Panel 1 (767 A.D.), which is with 141.7cm × 95.1cm larger than the Corona panels and situates 160 glyph blocks with an average size of 5.8cm × 4.1cm. The image data has a resolution of 4000 × 2027 pixels.

Both data sets benefit from a clear structured ordering of the Maya glyphs on the panels, where the glyph blocks are arranged in a matrix like representation. Typically, Maya glpyh-blocks are composed of several signs, which are assembled around a larger sign, usually called main sign. In Figure 2 the glyph main glyph *KAN*(red) is surrounded by the numbering symbol to the left (green) and a date symbol on the lower side (yellow). In this work we were interested in segmenting the glyph blocks from the panels and subsequently its signs.

After 3D acquisition, we compute surface features using the *Multi-Scale Integral Invariant* method [MKJB10] using the *GigaMesh* framework. For each vertex on the surface of the mesh, the MSII approach computes the ratio of bounded volumes below and above a set of virtual spheres of increasing size. The resulting feature vector describes the surface curvature in terms of convexity and concaveness. Each element of the feature vector represents one of the multiple scales of the spheres used to intersect the surface. This allows us to robustly identify and label grooves and plateaus on the surface of Maya panels. Since glyphs and, to a lesser degree, subglyphs can be modeled as plateaus surrounded and separated by grooves, we rely on segmentation within the MSII feature space.

3. Data Processing Pipeline

In this work we distinguish between glyph blocks and signs. Glyphs blocks in Maya hieroglyphic script are usually rectangular, often quadratic, well separated symbols used to signify concepts. Signs are distinct and recognizable patterns and depictions inside these glyphs blocks, some of which are repeated in similar form across very different glyphs blocks. However, in this work, due to the automated approach of segmentation the extracted pattern do not always represent proper signs. We therefore refer to these extracted patterns as subglyphs as they originate from an original glyph block. Thus, the main two concepts used in this work are glyph blocks, shortened to *glyphs*, and patterns extracted from these, the *subglyphs*.

Our contributions in this work are as follows: (i) exploiting the surface curvature of Maya panels by pre-procssing with MSII for subsequent image processing (ii) automatically identifying, segmenting, and clustering repeating subglyphs in a small corpus of Maya panels, and (iii) visualizing the hereby created network of glyphs sharing visually similar subglyphs.

We develop a multiple staged segmentation and clustering pipeline to approach the challenge of identifying and segmenting subglyphs and visualize the resulting glyph network using a force directed graph layout. Our pipeline is composed of the six following stages: (1) The segmentation of glyphs from Maya panels, (2) the segmentation of subglyphs from the glyphs, (3) the clustering of subglyph shapes, (4) the feature transform of the subglyphs, (5) subsequent hierarchical clustering thereof, and (6) the graph computation and layout. This process is illustrated in Figure 3.

4. Glyph Segmentation

We segment glyphs from Maya panels by employing a projection profile approach to detect horizontal and vertical lines, as shown by Santos et al. in [DCRC09] for Latin script. The pixel intensities of the raster image of the 2D projection of the MSII filtered

panel are summed along its vertical and horizontal axis. Then, a minimum filter is applied on this signal to detect extremal points. Extremal points are located where the minimum filter and the signal are equal. These are columns or rows with the most whitespace on the panel, they contain the lowest pixel intensities and identify the absence of glyphs.

We compute a grid given the horizontal and vertical extremal points that is overlaid on the panel and from which glyphs are extracted. Figure 4 depicts the extracted grid lines and the projection profiles of a Maya panel. Glyphs extracted from the grid are rescaled to an uniform size to allow comparisons across different panels.

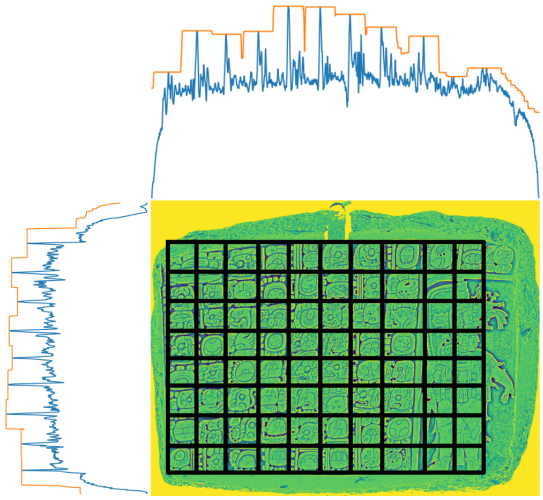


Figure 4: *Grid lines (black) resulting from vertical and horizontal profile projections (blue) and minimum filter response (orange) of a MSII filtered and rendered panel raster image.*

5. Subglyph Segmentation

While glyph segmentation proceeds by detecting vertical and horizontal whitespace on panels to extract a regular grid, subglyph segmentation requires a more flexible approach, the arrangement and shapes of subglyphs in a glyph are varied.

We employ the random walker segmentation approach by Grady in [Gra06]. The random walker segmentation algorithm models a raster image as a graph with pixels as nodes and edges connecting to the 8-neighborhood of pixels, where the edge weights are given by the pixel similarity of two pixels. Then, from a set of labeled points random walks are computed. The label of an arbitrary pixel is given by the first random walk reaching it from the initial set of labeled pixels.

The initial set of labels can be either given a-priori or, as in our case, computed from the image itself. On MSII filtered Maya panels the pixel values correspond to the surface curvature of the panel. Therefore, we segment subglyphs by thresholding the pixel values, that is, selecting and separating regions based on surface curvature. Concave regions, i.e. gaps separating subglyphs, have low pixels values. We define those to be the background. Subglyph faces are often convex and have high pixel values, we define those as the foreground.

To mark those regions for the random walker segmentation, we apply two thresholds on the extracted glyph images, a background threshold and a foreground threshold. The values of the MSII feature vectors have been averaged and rescaled to a range of $[0,1]$. We experimented with different threshold values and determined $t_1 < 0.5$ for the background and $t_2 > 0.7$ for the foreground to give the best segmentation results. The process of the thresholding, marking and segmentation is shown in Figure 5.

6. Subglyph Clustering

As the segmented regions of subglyphs have various extents, we normalize the extracted raster images to uniform size. However, the extents of the segmented subglyphs are semantically relevant. Comparing a very wide subglyph to a very narrow subglyph makes no sense, both images would have to be significantly distorted to extract feature vector of the same length.

Therefore, we discretize the subglyph extents into a limited discrete set of clusters. We encode the extents as a space of two-dimensional vectors and perform vector quantization of this space. Quantization is performed with k-Means [Mac67] where we set $k = 10$. A lower count of clusters led to some strongly distorted subglyphs in our dataset. A higher count of clusters precludes the discovery of similar subglyphs since subglyphs are only compared with other candidates in the same cluster.

We then proceed with feature extraction for each rescaled subglyph in its respective shape cluster. We compute the Histogram-of-oriented-Gradients (HoG) descriptor with 8×8 histogram cells and 3×3 block normalization cells. Smaller cells led the descriptor to pick up on irrelevant noise while larger cells missed semantically meaningful detail like small point-like inscriptions. The resulting feature vectors are of the same size within shape clusters but of different size between shape clusters.

To extract clusters of visually similar subglyphs we perform another clustering of feature vectors of each shape cluster. Here, we require clusters to be highly visually similar, that is, all subglyphs to be below an visual dissimilarity threshold. Agglomerative clustering with a complete linkage criterion, as first shown by Defays in [Def77], achieves this goal of grouping subglyphs given a dissimilarity threshold.

The same visual dissimilarity threshold is different for each shape cluster as the distance computation is dependent on the feature vector length. We normalize these thresholds by normalizing the distance matrices of each shape cluster to $[0,1]$. Then, we compute the hierarchical clustering of the feature vectors for each shape cluster and uniformly cut the resulting clustering trees, which are due to the normalization all of height 1, at a height of 0.1. From the resulting clusters, we discard any clusters that contain only one subglyph. These do not provide any information for our subsequent network visualization of glyph connections.

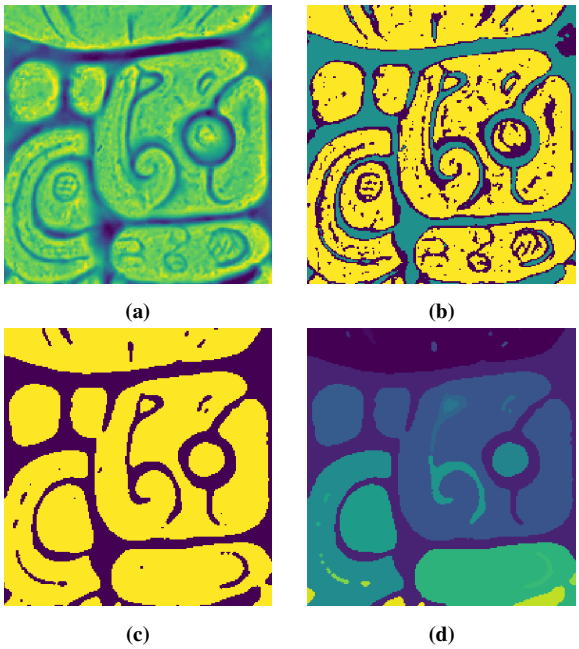


Figure 5: *Stages of subglyph segmentation. (a) Glyph segmented from a Maya panel using projection profiles. (b) Foreground (yellow) and background (teal) markers, set by thresholding MSII (surface curvature) values. (c) Independant regions (yellow) detected by random walker segmentation. (d) Disconnected components annotated by distinct labels.*

7. Glyph Network Visualization

From the clusters of visually similar subglyphs we construct a graph of glyphs that are related by sharing a subglyph. To explore the relations and semantic concepts expressed in such a graph, we employ a network visualization of the glyphs, where related glyphs are moved close together and unrelated glyphs farther apart.

We perform an iterative relaxation procedure, a force directed graph layout such as first shown by Eades in [Ead84], given three forces acting upon the nodes of the relatedness graph: (1) Nodes are pulled closer together along edges that connect them, (2) nodes are pulled closer together to all other nodes, (3) nodes are pushed apart if they are too close. Force (1) is used to keep related nodes close together, force (2) is much weaker than force (3) and ensures compactness of the layout, i.e. no unnecessary whitespace is present, and force (3) moves very close nodes apart to leave space for rendering glyph images.

Formally, let i, j denote the indices of the glyphs segmented from our dataset of Maya panels and let k, l denote the indices of subglyphs subsequently sliced from the aforementioned glyphs. Then, let $\downarrow i = \{k, l, \dots\}$ denote the operation by which we get the subglyph indices $\{k, l, \dots\}$ of a glyph i.

We denote the graph of relatedness between glyphs by an adjacency matrix $a_{ij} \in A$ where each a_{ij} indicates the presence $a_{ij} = 1$ or absence $a_{ij} = 0$ of a connection between glyphs i and j. The adjacency is computed by connecting glyphs sharing the same subglyphs as computed by the hierarchical clustering. Let c_k denote the cluster index of a subglyph k, then the elements of A are computed as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } c_{\downarrow i} \cap c_{\downarrow j} \neq \emptyset \\ 0, & otherwise \end{cases} \tag{1}$$

For laying out the graph let $v_i^t \in \mathbb{R}^2$ denote a position of a node associated with glyph i at time step t. We define the three aforementioned forces acting upon nodes associated with glyph i, force (1) $\vec{f^a}_i$ for keeping related nodes close, force (2) $\vec{f^b}_i$ for keeping the layout compact, and force (3) $\vec{f^c}_i$ for avoiding overlapping nodes. The vector $\vec{n_{ij}}$ denotes the normal pointing from node v_j^t to v_i^t.

$$\vec{n_{ij}} = \frac{(v_i^t - v_j^t)}{\|(v_i^t - v_j^t)\|} \tag{2}$$

$$\vec{f^a}_i = \sum_j 0.1 * \vec{n_{ij}} * \|(v_i^t - v_j^t)\| * a_{ij} \tag{3}$$

$$\vec{f^b}_i = \sum_j 0.01 * \vec{n_{ij}} * \|(v_i^t - v_j^t)\| \tag{4}$$

$$\vec{f^c}_i = \sum_j 0.01 * (-\vec{n_{ij}}) * \frac{1}{\|(v_i^t - v_j^t)\|} \tag{5}$$

Then, the layout is computed by first initializing the node positions v_i^0 at the beginning randomly and iterating the following equation.

$$\vec{v_i^{t+1}} = \vec{v_i^t} + \vec{f^a}_i + \vec{f^b}_i + \vec{f^c}_i \tag{6}$$

The final positions of the nodes v_i^t denote the layout of the graph. We found that 1000 iterations provide enough time for the layout to settle.

8. Results

The resulting visualization, shown in Figure 6, of the glyph network provides an intuitive view into the groups of glyphs and their relations formed by sharing similar subglyphs. This method of analysis allows us to explore and discover interesting commonalities between glyphs without the need of a-priori query statements, such as those necessary in typical database systems. For example, since we analyze more than one panel at once, we can easily discover glyphs which are unique to a panel or common in the processed dataset. While specific glyphs may be unique for a panel, subglyphs of unique glyphs can be common in a set of panels.

The main limitations of our work are: (i) the glyph segmentation and thus line detection not being flexible enough. One of the panels in our dataset is warped leading to non-parallel line separators. Employing Hough transform [DH72] or line detection with seam carving [AS14] alleviates such issues. (ii) Feature extraction is not powerful enough. Either employ an siamese neural network to compare image patches [ZK15] or use texture classification methods such as local binary features [LFG*17] and SIFT-like methods [CMK*14]. (iii) The visualization does not scale with significantly larger datasets. In this case showing a higher-order view of panels as nodes, instead of glyphs, connected by edges of similar

glyphs and sub-glyphs increases the capacity of the visualization to display large datasets.

9. Summary & Future Work

In this work we presented an approach for Maya glyph and subglyph segmentation and clustering to enable the extraction and network visualization of connections of glyphs by means of their subglyphs.

First, applying a vertical and horizontal projection profile, we extract and segment glyph images from panels. Then, a random walker segmentation extracts subglyphs from the glyph images based on the surface convexity of original 3D scan. We cluster the space of image extents and normalize subglyphs to common sizes. Then, subglyphs in their respective size cluster are clustered hierarchically based on their HoG features. From these the adjacency of glyphs is computed and laid out in two dimensions using a force directed layout.

For future work a significantly larger dataset would allow for more salient discoveries of glyph connections and common subglyphs across a corpus of Maya panels. Additionally, a large corpus allows for identifying unique and isolated glyphs, that is, glyphs sharing no subglyphs with others, with significantly higher confidence.

Acknowledgments

We thank the *Fundación La Ruta Maya*, Guatemala and the *Historisches Museum der Pfalz, Ausstellung "Maya. Das Rätsel der Königsstätte"*, Speyer for their kind support. This his work is partially funded by the *Massnahme 5.4 – Wissenschaftliches Rechnen – Zukunftskonzept* (institutional strategy) of the 2nd *German University Excellency Initiative*. Additional support is provided by the *Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences* (HGS MathComp).

References

[AS14] ARVANITPOULOS N., SÜSSTRUNK S.: Seam Carving for Text Line Extraction on Color and Grayscale Historical Manuscripts. *International Conference on Frontiers in Handwriting Recognition* (2014). 5

[BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 4 (2002), 509–522. 2

[CMK*14] CIMPOI M., MAJI S., KOKKINOS I., MOHAMED S., VEDALDI A.: Describing textures in the Wild. *Computer Vision and Pattern Recognition* (2014). 5

[DCRC09] DOS SANTOS R. P., CLEMENTE G. S., REN T. I., CALVALCANTI G. D. C.: Text Line Segmentation Based on Morphology and Histogram Projection. *International Conference on Document Analysis and Recognition* (2009). 3

[Def77] DEFAYS D.: An efficient algorithm for a complete linkage method. *The Computer Journal* (1977). 4

[DH72] DUDA R. O., HART P. E.: Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM* (1972). 5

[DT05] DALAL N., TRIGGS B.: Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), vol. 1, pp. 886–893. 2

[Ead84] EADES P.: A Heuristic for Graph Drawing. *Congressus Numerantium* (1984). 5

[Gra06] GRADY L.: Random Walks for Image Segmentation. *Transactions On Pattern Analysis and Machine Intelligence* (2006). 4

[LFG*17] LIU L., FIEGUTH P., GUO Y., WANG X., PIETIKÄINEN M.: Local binary features for texture classification: Taxonomy and experimental study. *Pattern Recognition* (2017). 5

[Mac67] MACQUEEN J. B.: Some Methods for Classification and Analysis of Multivariate Observations. *Berkeley Symposium on Mathematical Statistics and Probability* (1967). 4

[MK13] MARA H., KRÖMKER S.: Vectorization of 3D-Characters by Integral Invariant Filtering of High-Resolution Triangular Meshes. *International Conference on Document Analysis and Recognition* (2013). 2

[MKJB10] MARA H., KRÖMKER S., JAKOB S., BREUCKMANN B.: 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)* (2010), Eurographics Association, pp. 131–138. 3

[RRMM17] ROMAN-RANGEL E., MARCHAND-MAILLET S.: Assessing Deep Learning Architectures for Visualizing Maya Hieroglyphs. *Mexican Conference on Pattern Recognition* (2017). 2

[RRPOGP09] ROMAN-RANGEL E., PALLAN C., ODOBEZ J.-M., GATICA-PEREZ D.: Retrieving ancient maya glyphs with shape context. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (2009), pp. 988–995. 2

[RRPOGP11] ROMAN-RANGEL E., PALLAN C., ODOBEZ J.-M., GATICA-PEREZ D.: Analyzing ancient maya glyph collections with contextual shape descriptors. *International Journal of Computer Vision 94*, 1 (2011), 101–117. 2

[SG16] SCHUBERT A., GRUBE N. (Eds.): *Maya. Das Rätsel der Königsstädte*. Hirmer, München, 2016. 2

[SZ15] SIMONYAN K., ZISSERMAN A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations* (2015). 2

[TS62] THOMPSON J. E. S., STUART G. E.: *A catalog of Maya hieroglyphs*. University of Oklahoma Press, Norman, Oklahoma, 1962. 2

[ZK15] ZAGORUYKO S., KOMODAKIS N.: Learning to compare image patches via convolutional neural networks. *Computer Vision and Pattern Recognition* (2015). 5

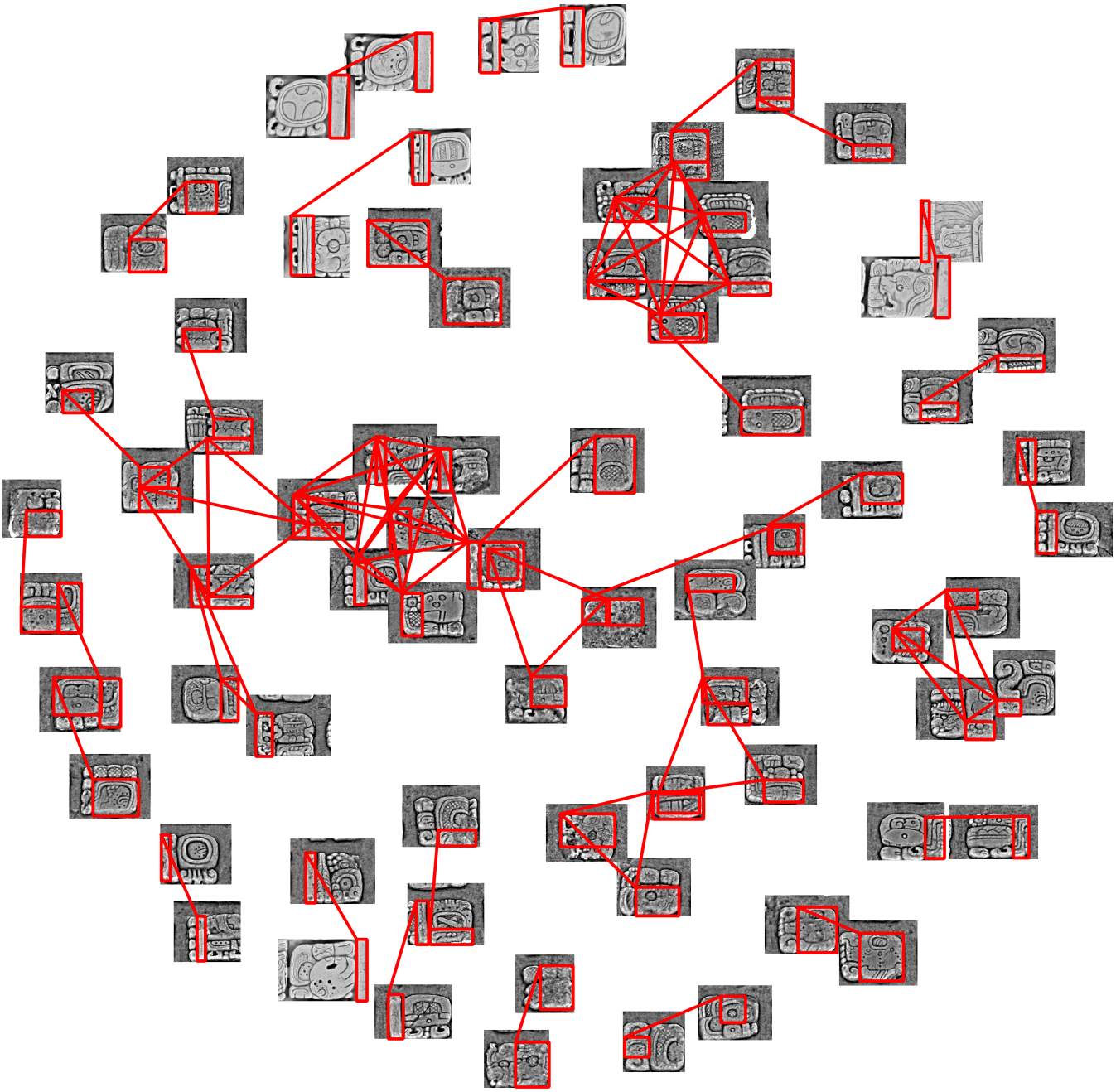


Figure 6: *Glyph connection visualization. Based on the adjacency matrix computed from shared visually similar subglpyhs, a force directed layout is computed. The respective shared subglyphs are highlighted by red bounding boxes and red lines between glyphs. Tight clusters and many red connections indicate a high degree of interrelatedness, i.e. many shared subglyphs between glyphs, and a high degree of composability, i.e. glyphs are composed of common subglpyhs instead of containing original subglyphs.*