EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2017)
R. Sablatnig and B. Štular (Editors)

Short Paper

Histogram of Oriented Gradients for Maya Glyph Retrieval

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

[1]Heidelberg University, Interdisciplinary Center for Scientific Computing (IWR),
Forensic Computational Geometry Laboratory (FCGL), Germany
{felix.feldmann, bartosz.bogacz, 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 effort that has already started in the early 19th century. Inexpertly-created drawings of Maya writing systems resulted in a large number of misinterpretations concerning the contents of these glyphs. As a consequence, the decryption 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 Maya glyph retrieval based on a machine learning pipeline. A Support Vector Machine (SVM) classifier is trained based on the Histogram of Oriented Gradients (HOG) feature descriptors of the query glyph and random background image patches. Then a sliding window classifies regions into viable candidates on the scale pyramid of the document image to achieve scale invariance. The algorithm is demonstrated on two different data sets. First, photographs from a hand written codex and second 3D scans from stone engraved monuments. A large amount of future extensions lies ahead, comprising the extension to 3D, but also more sophisticated classification algorithms.

CCS Concepts

•Computing methodologies → Shape representations; Object identification; •Applied computing → Graphics recognition and interpretation; Optical character recognition;

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 [MKJB10] 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 a printed catalogue, which contains glyphs that have already been decoded e.g. [TS62].

This work develops a machine learning pipeline for object recognition for automatic Maya glyph retrieval. For this purpose, the *Histogram of Oriented Gradients (HOG)* detector by Dalal and Triggs [DT05, Dal06] is used. Originally, the authors introduced the HOG descriptor to detect humans in pictures. 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 a *Support Vector Machine (SVM)* is trained and a sliding window is used to search for similar objects in other 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 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].

2. Data Sets

Two different data sets with different properties have been used, 2D scans of original ink drawings and renderings of 3D measurement data. The first one is the *Dresden Maya Codex* [GB12], which is one of the best preserved Maya writings consisting of 78 pages written in a fanfold style which appears to be an old-style calendar by the Maya priests. The calendar was used to obtain current astronomical

DOI: 10.2312/gch.20171301

forecasts and therefore to plan related rituals accordingly [GB12, p. 32]. As far as the researchers know, the codices were written by six different writers, hence differences in the shape of the same representative of a glyph are unavoidable.

Photos of the codex exist at the online archive of the Dresden museum, SLUB (Saxon State and University Library Dresden). The maximum resolution of the images is 1000×2000 pixels, which makes the work with extractions from smaller glyphs more challenging. The commonly extracted size of a glyph is smaller than 95×58 pixels for the codex data set.

The second data set consists of three 3D scans of cultural monuments, taken by Christian Prager during the exhibition "Maya. Das Rätsel der Königsstätte" in Speyer, Germany [SG16]. 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.

After 3D acquisition, features have been computed using the *Multi-Scale Integral Invariant* method [MKJB10] using the *GigaMesh* framework. This method extracts multi-scale curvature-based features of a 3D mesh. The generated data is exported to different resolutions so that more features can be extracted later on by object recognition algorithms. Nevertheless, for this work the lack of glyphs in the 3D data generates problems for the evaluation process, because with only a few glyphs a reliable test and trainset could not be extracted (Table 1).

To examine the machine learning algorithms used in this article it is necessary to select symbols which occur more frequently than others. Those symbols were chosen as they were representing dates or recurring events in the Maya culture, such as the enthronement of a new leader.

All selected classes of the hieroglyphs have been labeled with their class name using a bounding box surrounding the glyph. This data is used to generate the ground-truth data to train the classifier with the extracted features and for evaluating the applied method.

Figure 1: *Page 59-61 of the Dresden Codex. SLUB Dresden: http://digital.slub-dresden.de/werkansicht/dlf/2967/1/1/*

Glyph	Name & No.	Occurence	Avg. size
	*chi*, T671	36	95×40px
	*ka*, T25	33	86×26px
	*ETZ'NAB*, T527	31	81×66px
	*nik*?, T533	55	83×58px
	*li*, T24	46	85×37px
	*WI'IL*, T158	21	89×87px
	*KAN*, T506, (3D)	14	403×443px

Table 1: *Glyphs for retrieval with their occurrence and average size.*

3. Maya Feature Extraction

The basic idea of the *Histogram of Oriented Gradients* descriptor is that the local appearance and shape of an object in an image can be described by the distribution of intensity gradients.

Image division - Cells & Blocks

The first step consists of dividing the image into cells. The cell size is very important for the classification, it describes the number of pixels that will be used for computing the HOG and therefore retrieves the necessary information describing the glyph for classification. A very large cell size, more than 14×14 pixels yields a coarse collection of features, because not enough relevant information can be gathered. Similarly, a very small cell size yields a large feature vector. Typically, the cell size is selected in accordance to the amount of detail that needs to be described. It is worth mentioning that a large cell size does not necessarily lead to better results for the classification, for our retrieval a cell size of 6×6 pixels retrieved most glyphs.

It is obvious, that each cell differs in brightness. To handle these differences in illumination, blocks are used to normalize the contrast over a larger range and blocks do overlap with 50%. The block size is twice the size of the cell size so information of the illumination over an area which is four times greater then a single cell is gathered.

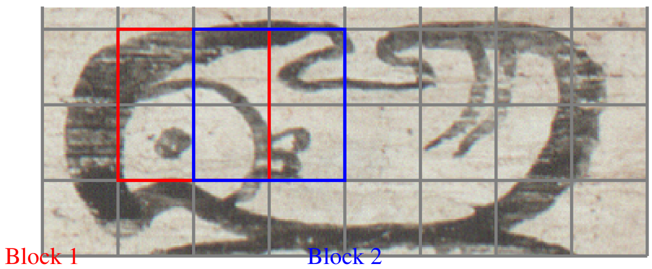
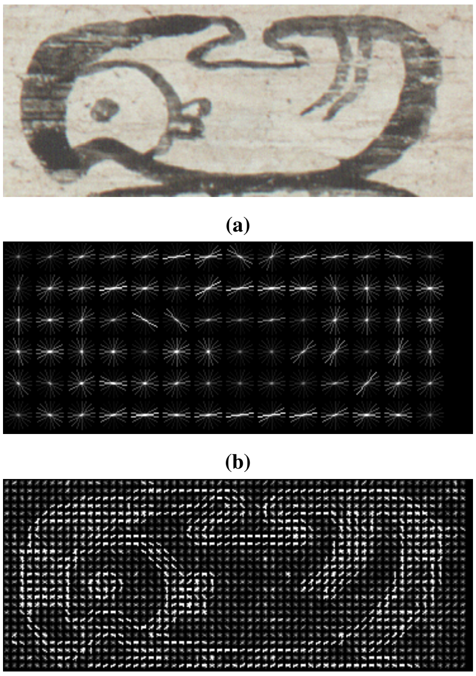


Figure 2: *Example of overlapping blocks (blue, red rectangle) and cells (grey lines).*



(a)

(b)

(c)

Figure 3: *Example of HOG applied on the Query Image* chi *(a) with different cell sizes (b) with coarse cell size of* 14×14 *(c) with optimal cell size of* 6×6

Gradient Computation

The essential part of the algorithm is to compute gradients using the cells generated by the previous step. In order to compute the gradients a 1D filter without any smoothing has been applied to the image, because smoothing reduces the information that can be captured, it especially reduces the edge contrast, which is important for the descriptor. One advantage of the simple filter without any smoothing is that it can be computed quicker with less computational effort.

Orientation Cells

The orientation of a gradient is in the range from 0-360 degrees, named *signed gradient* or from 0-180 degrees, thus *unsigned gradient*, where the opposite direction of the vector will be mapped to the corresponding value for the unsigned gradients. As studies by Dalal and Triggs show, the signed gradients decrease the performance, thus only signed gradients have been tested for the Maya glyph retrieval.

For computing the *Histogram of Oriented Gradients*, first the orientation of 180 degrees is divided by a certain number, which is called bins. Each of the bins represents a range of degrees, to which each gradient with its magnitude as a vote is mapped to. If a gradient is exactly between one of the bins, the gradient is voted for both bins, where the votes are accumulated.

A very low number of bins, e.g. 1 to 4, or a large number of bins leads into a very coarse description of the distribution of gradients. For this work, a value of 9 bins has been used; this corresponds to the author's original proposal [Dal06, p. 38].

Normalization over Blocks

Considering, that gradient strengths change over a wider range due to changes in illumination, a normalization of each cell is necessary for better invariance to brightness and contrast. Therefore, the previously created blocks are used, which are twice the size of a cell. Thereby, for each area of a block the cells are normalized in its changes of illumination. Each block overlaps with 50% so that each cell contributes with different normalization to the final feature vector, several times.

We chose a modified L_1 norm as a normalization function as it yielded the best classification performance.

Feature Descriptor and Classifier Training

Previously the gradients have been calculated for each cell and have been voted into bins. After normalizing the cells over a larger area, each block with its normalized histograms are concatenated into a 1D matrix. The size of a typical feature vector is blocksize $\times$ cellsize $\times$ number of bins.

For every query image from the training set the previously described HOG descriptor is computed and passed to the *SVM* as positively labeled data. Negative examples are taken from the background, where no glyphs appear.

4. Detecting Glyphs

In order to retrieve glyphs in the test set a sliding window is used to retrieve the information on the clutter image. Therefore the sliding window uses the average size of the glyphs that were originally used to train the classifier. In each step the HOG of the current window is computed and the resulting feature descriptor is being classified with the previously trained model inside the SVM.

As this is a basic approach, the sliding window that is supposed to find representatives of the glyphs, is limited to the size of the window. It may occur that some representatives of a glyph class appear bigger than the window size used for training. To retrieve those glyphs and to add invariance with respect to scale to the process, an *image pyramid* is used.

The image pyramid consists of six different levels. Starting from the original size of the detection window where each image in each level is scaled. Using more than one scale in the sliding window did not lead to better results, which has the effect that glyphs appearing in the Codex have a similar size.

To avoid counting glyphs that have already been retrieved several times, due to a not perfectly coordinated sliding over the glyph, a *Non-Maximum Suppression* is applied, to suppress multiple detections in the same area.

After having applied the non-maximum suppression, the retrieved area is verified against the previously labeled ground-truth data. A correct result is registered if the retrieved area overlaps the ground-truth by more than 30%. The data set consists of images containing one or more glyphs. For each glyph class 70% of the labeled glyphs are used for the training, and the rest of the images, containing the 30% test data, are used for testing purposes. The search has been done per glyph class and only on document images where at least one glyph of the class is present.

5. Results

For evaluating the retrieval setup a line search has been applied to find the best parameters. Hereby, it is important to use a cell size of 6×6, which is quite close to the proposed value for human detection of 8×8. A too coarse collection of information in to large cell sizes does not lead into a good retrieval by looking at the average glyph size. In average at a *Recall* of 50% a *Precision* of 40% could be achieved.

As the graphs show, there are quite huge differences in the performance of glyph retrieval for each class. The two simplest glyphs *ka* and *li* can be retrieved quite rarely, even if their occurrence in the data set is higher than e.g. the *wi'il* glyph. It appears that complexity of the glyph with respect to its representation and its shape results in increasing retrieval. The more distinct the shape of the glyph, the better it can be retrieved, especially with the two *ka*, *li* glyphs where the HOG is quite similar to e.g. the Maya numbers. Even researchers who have been working in that field for years sometimes cannot clearly distinguish these simple glyphs. For the 3D data set only one frequent glyph could be used for the retrieval, which appeared 14 times. Therefore, it is quite challenging to evaluate the retrieval for this data set.

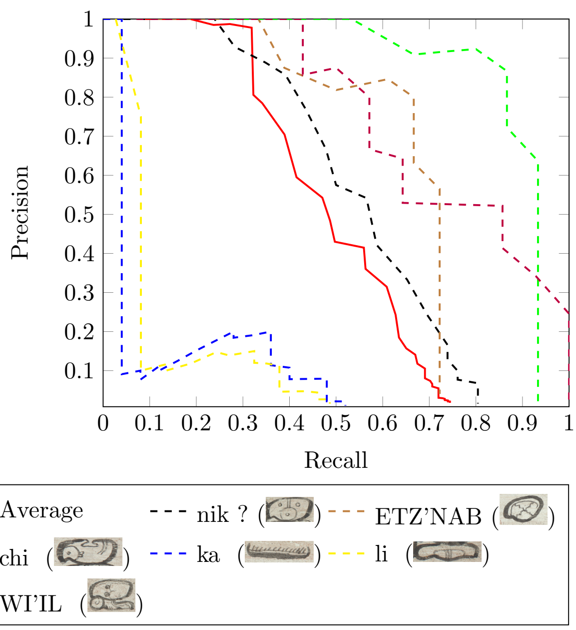


Figure 4: *Comparison of PR for the 6 different glyph classes with 70% training data*

6. Summary & Future Work

The main focus on this work lies on the retrieval and the spotting of Maya glyphs within raster images using a machine learning approach combined with Histogram of Oriented Gradients for feature extraction. By performing a parameter search for the algorithm, the overall precision and recall led to classifiers that can be used for glyph retrieval in real data. The utility of this approach highly depends on the given glyph. This approach thus also illustrated the limits of using classifiers for Maya glyphs, as a sufficient amount of data is required to sufficiently gain many significant results. As the evaluation shows, the retrieval rate for geometrically simple glyphs is rather low.

To understand why the retrieval setup misclassified some glyphs, an evaluation proposed in [VKMT13] can be used, where the detection will be visualized and it is possible that in some images the created Histogram of Orientations has the same appearance as the retrieved glyph.

It is possible to apply different feature extraction methods, in particular for 3D data sets, because this type of data does not, compared to the drawn codex, suffer from too much noise and smoothed-like appearance. For this, the *Shape Context* algorithm [BMP02] for comparing shapes of binary images can be used, as well as the word spotting algorithm by Howe [How13] addresses the same spotting problem.

Finally the retrieval of Maya glyphs can be improved if a segmentation of all glyph blocks as well as an extraction of the glyphs found in the glyph blocks is performed before classification.

References

[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. 1, 4

[Dal06] DALAL N.: *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble (INPG), 2006. 1, 3

[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. 1

[GB12] GRUBE N., BÜRGER T.: *Der Dresdner Maya-Kalender*. Herder Verlag GmbH, Freiburg im Breisgau, Germany, 2012. 1, 2

[How13] HOWE N. R.: Part-structured inkball models for one-shot handwritten word spotting. In *IEEE International Conference on Document Analysis and Recognition (ICDAR)* (2013), pp. 582–586. 4

[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. 1, 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. 1

[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. 1

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

[TS62] THOMPSON J. E. S., STUART G. E.: *A catalog of Maya hieroglyphs*. University of Oklahoma Press, Norman, Oklahoma, 1962. 1

[VKMT13] VONDRICK C., KHOSLA A., MALISIEWICZ T., TORRALBA A.: Hoggles: Visualizing object detection features. In *IEEE International Conference on Computer Vision (ICCV)* (2013), pp. 1–8. 4