Assessing Sparse Coding Methods for Contextual Shape Indexing of Maya Hieroglyphs

Edgar Roman-Rangel, Jean-Marc Odobez, Daniel Gatica-Perez
Idiap Research Institute, Martigny, Switzerland
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
Email: {eroman, odobez, gatica}@idiap.ch

Abstract—Bag-of-visual-words or bag-of-visual-terms (bov) is a common technique used to index multimedia information with the purposes of retrieval and classification. In this work we address the problem of constructing efficient bov representations of complex shapes as are the Maya syllabic hieroglyphs. Based on retrieval experiments, we assess and evaluate the performance of several variants of the recent sparse coding method KSVD, and compare it with the traditional \(k\)-means clustering algorithm. We investigate the effects of a thresholding procedure used to facilitate the sparse decomposition of signals that are potentially sparse, and we also assess the performance of different pooling techniques to construct bov representations. Although the bov’s computed via Sparse Coding do not outperform the retrieval precision of those computed by \(k\)-means, they achieve competitive results after an adequate enforcement of the sparsity, which leads to more discriminative bag representations with respect to using the original non-sparse descriptors. Also, we propose a simplified formulation of the HOOSC descriptor that improves the retrieval performance.

Index Terms—Indexing, clustering, sparse coding, shape descriptor, Maya culture, hieroglyph.

I. INTRODUCTION

The collection of digital imagery has been boosted in the last years by a whole new generation of devices that allow to gather thousands of high quality images, therefore generating the need for efficient tools to index large image data sets and to retrieve images that are similar to a given query in terms of visual content. This phenomenon is widely spread in different fields, such as photography, painting, the arts, and archaeology.

One instance of the above mentioned phenomenon is the AJIMAYA project (Hieroglyphic and Iconographic Maya Heritage) conducted by the National Institute of Anthropology and History of Mexico (INAH). Despite the success of the project towards gathering a collection of images of all existing monuments in some of the archaeological Maya sites within the Mexican territory, the manual cataloging of the hieroglyphs remains to be accomplished, mainly due to the large amount of information that has been generated, and the lack of automatic and semiautomatic tools to support the cataloging goal. For instance, Fig. 1 shows a Maya inscription with a large amount of hieroglyphs.

The Maya writing system is composed of two main types of hieroglyphs: logograms (words) and syllabograms (syllables), and the blocks found in inscriptions usually exhibit one or two logograms accompanied by one to four syllabograms complementing each other to build coherent sentences. Fig. 2(a) shows four blocks vertically arranged, each of them contains both syllabograms and logograms. A third type of Maya glyphs that correspond to Maya art is known as iconography, e.g., Fig. 2(b). In our work we focus on the description and retrieval of Maya syllabograms.

Currently, a rough estimate of 1000 different hieroglyphs have been discovered, from which only almost 80% of them have been deciphered. The other 20% remains unknown, and archaeologists continue finding new hieroglyphs that require to be identified and classified.

In this paper, we present recent advancements made towards the design of an efficient content-based retrieval engine for epigraphic versions of Maya hieroglyphs. We conducted a systematic study to assess the quality of recently proposed techniques to represent and retrieve images. More specifically, of bag-of-visual-terms representations constructed based on two indexing techniques: the KSVD algorithm, which is a recent method for sparse coding [1], and the traditional \(k\)-means clustering [2].

According to [3] sparse coding is a method to represent signals as sparse linear combinations of an overcomplete set of basis functions called dictionary. The method is inspired on research work by the neuroscience community, which suggests that the receptive field on...
mammalian primary visual cortex encodes natural images as sparse signals [4]. This approach has become common in a wide number of problems in multimedia research, for instance, images and video inpainting and denoising [5], image compression [6], image restoration [7], image classification [8], [9], and shape representation [10]. However, recent attempts to classify images based on sparse representations [11] suggest caution using this technique, as perhaps it is not completely suitable to deal with nonnatural images, or at least not when the level of noise is considerably high.

In our work, we have used the HOOSC descriptor [12] to represent the Maya glyphs, as it has recently proven to be effective in dealing with such complex shapes [13]. Since by construction the HOOSC descriptor is not a sparse signal in its original space, we conducted several studies to facilitate its sparse decomposition and to use it as a quantization technique comparing its performance in shape retrieval task. To the best of our knowledge, there are no previous works using sparse coding techniques as quantization method applied to local shape descriptors and retrieval of shape images.

More specifically, the contributions of this work are:

- The assessment and evaluation of the KSVD algorithm as method to compute over complete dictionaries and to construct bov representations of local shape descriptors based on their sparse decompositions.

- The assessment and evaluation of the performance of those bov’s in the task of content-based image retrieval. This evaluation includes the exploration of different pooling schemes used to construct the bov’s, different sizes of the dictionary, and different distance metrics. It also considers different evaluation criteria, i.e., retrieval precision, reconstruction error, intra-class and inter-class variability, and the potential to discover visual patterns that help differentiate classes of glyphs.

- The implementation of a thresholding procedure to the HOOSC descriptor that facilitates its sparse decomposition.

- The introduction of a new formulation of the HOOSC descriptor that accounts for a consistent 5% of improvement of the precision retrieving Maya hieroglyphs. This new version of the HOOSC also leads to a shorter descriptor.

The rest of the paper is organized as follows. Section II discusses some of the relevant work in sparse coding, bag-of-words representations, and visual description of Maya hieroglyphs. Section III shows a schematic overview of our approach. Section IV describes a version of the HOOSC descriptor that has been used in this work. Section V explains the k-means clustering algorithm and the sparse coding approach, as well as the procedure to build efficient bag-of-words (bov) representations based on them. Section VI introduces the experimental setup to compare the performance of the clustering approaches. Section VII presents the analysis of results. Finally, we conclude in section VIII.

II. RELATED WORK

In our approach, we explore the use of sparse coding and vector quantization applied to bov representations for the retrieval of Maya hieroglyphs. Below, we present the related work in those directions.

**Shape representations.** They have been largely studied, mainly with Shape Context-like algorithms that have proven to be efficient methods to represent shapes with visual complexity ranging from low to high [12]–[15]. Two recent approaches have boosted the performance of retrieval systems by the incorporation of a constrained diffusion process [16], and the use of graph transduction [17], defining the state-of-the-art in retrieval of “generic” shapes.

**Sparse coding.** It was first introduced in [3], [4] as a method to find sparse linear combinations of basis functions to encode natural images. Given that the resulting sparse image codes have a high degree of statistical independence, the authors suggested that they are more suitable to be used for later stages in image processing. Even though the authors do not provide any quantitative evaluation of their method, they show that the sparse coding of natural images leads to a set of localized, oriented, bandpass fields that are similar to those found in the primary visual cortex of mammalians.

Since these seminal works, a large number of works used this approach in image and video processing, multimedia indexing, and image classification [18]. For instance, based on stochastic approximations, an online optimization algorithm for dictionary learning was proposed in [7] for in-painting and image restoration. The KSVD algorithm was introduced in [1] as a method to estimate sparse representations. This method was applied for restoring facial images and for image compression. It was extended in [5] to multi-scale sparse representations.
for the enhancement and restoration of color images and videos. In our work, we evaluate the applicability of the KSVD algorithm to deal with shape representations of Maya hieroglyphs. A previous work that investigated a similar problem is [6], where the authors presented a method to extract shift-invariant sparse features of shapes. This method was used to train a deep convolutional network for classification of shape images of numeric digits, and for compression of text document images achieving state-of-the-art results. However, digit shapes are far simpler compared with the high visual complexity of the Maya hieroglyphs.

In another direction, the problems of shape representation and recognition of multiple objects in images were approached with sparse decompositions of low-level features in [10]. However, these approaches were mainly evaluated on synthetic data, detection of simple shapes in aerial images, and reconstruction of brain magnetic resonance images in a qualitative manner. In general, there are a very few works that addressed shape encoding (rather than shape images) using sparse coding.

**Bag of words, and bag of words with sparse coding.** The bov representation is widely used in the image retrieval community [19]. One of the initial works for object matching in videos based on bov is [20], where objects are represented by quantized sets of viewpoint invariant region descriptors. For still images, the works in [21], [22] model visual scenes as bov that are designed based on vector quantization and probabilistic latent models. Such representations are used to perform image scene classification achieving state-of-the-art results.

Sparse coding has been investigated to improve bov representations. For instance, the work in [23] investigated the use of spatial pyramid matching as a method to generalize vector quantization to sparse coding. This work used SIFT sparse codes for image categorization obtaining state-of-the-art performance. This work was extended in [24] by the use of a Laplacian constraint, which overcomes the loss of spatial information of the bov construction process. However, none of them addressed the use of sparse coding for representation of complex shapes.

Several pooling schemes of sparse coding for vector quantization were evaluated in [8], [9], where a set of experiments for feature recognition and image classification showed that some pooling strategies perform better than others. Besides the success of sparse coding in the representation of natural images, a recent work in image recognition [11] has suggested that sparse coding might not be suitable if the input signal contain a reasonable level of noise.

**Applications in art and cultural heritage.** The problem of content-based retrieval of shape instances of Maya hieroglyphs was approached in [12] with a small data set and the proposal of a robust shape descriptor called HOOSC, which was improved in [13] to deal with an extended data set exhibiting higher visual complexity.

Using a heuristic approach in [25], a Mesoamerican symbol with high variability among its instances has been detected in drawings of steles. In other work [26], the tasks of artistic style recognition and authentication were successfully performed with sparse models to distinguish drawings by Pieter Bruegel the Elder from its imitations.

In our work we investigate the use of sparse coding as a quantization technique to build bag representations of Maya hieroglyphs based on HOOSC descriptors.

### III. Our approach

To start with, we present the general picture of the process we have followed to index Maya hieroglyphs.

The first column in Fig. 3 shows four examples of the query set, and how we preprocess them, extract their local descriptors, quantize them to estimate a dictionary, and build their respective bov representations. The second column shows a similar process applied to a query glyph, where its bov is computed based on the dictionary previously learned. The final row shows the 10 most similar candidates for the given query after being ranked by similarity. The details about these steps are explained in sections IV and V.

### IV. HOOSC descriptor

The Histogram of Orientations Shape-Context (HOOSC) [12] is a robust shape descriptor specifically designed to describe complex shapes in an efficient manner. It has been originally proposed to deal with Maya hieroglyphs, overcoming some of the issues that arise as consequence of their high visual complexity that traditional shape descriptors are not able to handle well [14], [15]. Fig. 2 shows several instances of Maya hieroglyphs with different degrees of visual complexity.

For a given set $M$ of 2-D points representing the contour of a shape, the HOOSC provides a set of shape descriptors, one per each point in a subset $N \subseteq M$; the points in this subset are called pivots. Each descriptor consists of a set of histograms localized in specific regions that are arranged in a log-polar grid whose center is the pivot to be described, these regions contain the subset of the closest remaining points $p_i \in M$. In turn, each histogram corresponds to the distribution of the local orientation of all the points inside the region it describes. Different concatenations of such histograms, and their adequate normalization lead to different versions of the HOOSC [13]. In general terms, the resulting HOOSC vector for a specific pivot can be thought as the description of the shape from a specific point of view: the point of view located at the $x$ and $y$ coordinates of that pivot.

The original HOOSC constructs the localized histograms using 8 bins to cover the interval $[0−\pi]$ inside each region, and uses a log-polar space divided in 12 orientation intervals that cover a complete circumference around the central pivot, and 5 distance intervals spanning up to twice the average pairwise distance of all the input points, therefore generating 60 regions placed around the center pivot. The resulting descriptor is a 480-D vector.

In [13] several improvements to the HOOSC were proposed to improve its retrieval performance. One of
Figure 3. Process for description and retrieval of Maya hieroglyphs with HOOSC. First column from top to bottom: four examples of the query set, their preprocess, computation of local descriptors, estimation of the dictionary, and bow representations. Second column: corresponding process for a query example. Last row: most similar glyphs retrieved for the given query.

Figure 4. (a) log-polar grid with contour points in blue whose orientations are used to describe the pivots points shown in red. (b) the polar space used in our work, with the local orientation of each point from (a). Note that with our proposed grid, all the points farther than once the average pairwise distance are not used for description.

By further investigating the splitting formulation of the spatial scope, we found another configuration that further improves the retrieval precision. Namely, we confirmed that the most external distance interval is not very informative. However, the most internal regions do contain important information regarding the local neighborhood for the pivot to be described. We have merged the three most internal distance intervals in a single one, such that the resulting polar grid has only two distance intervals: one for the interval $[0 - 0.5]$, and other for the interval $[0.5 - 1.0]$ times the average pairwise distance of the input set. By keeping the radial slitting fixed to 12 intervals, our new formulation consists of 24 regions $R_{p=1,...,24}$, and therefore, the descriptor becomes only $192 + 2$ dimensions (again the $x$ and $y$ coordinates are included). Fig. 4 shows both, the original log-polar grid of the HOOSC, and the polar grid we propose to use.

During the description process, the histogram of orientations in a given region $R_{r}$ is estimated with a density approximation procedure. More precisely, the density in the $b$-th bin of the histogram $H_{r}$ for the region $R_{r}$ is estimated as

$$H_{r}(b) = \sum_{p_{r} \in R_{r}} h_{r}(b),$$

where the summation is computed over all the points $p_{r}$ localized within the region $R_{r}$, and the density function $h_{r}(b)$ is computed as

$$h_{r}(b) = \sum_{\theta \in b} k_{r}(	heta),$$

where $\theta \in b$ denotes all the orientation values within the $b$-th bin of the kernel $k$ used to approximate the densities. In turn, the kernel $k$ implemented in the original HOOSC
is defined as
\[ k_i(\theta) = N(\theta; \mu, \sigma^2), \]
where \( N(\theta; \mu, \sigma^2) \) denotes the value of a Gaussian having mean \( \mu \) and variance \( \sigma^2 \). A value of \( \sigma = 10 \) is normally used [13] as it has proven to work well avoiding hard binning effects and dealing with imprecision in orientation estimation. Fig. 5 shows the Gaussian kernels \( k_{45}, k_{90}, \) and \( k_{180} \) for the respective angles of 45°, 90°, and 180°, and their corresponding density functions \( h_{45}, h_{90}, \) and \( h_{180} \).

We observed that the Gaussian density functions \( h_i \) have tails with densities very close to zero and that can be considered as “noise”. To get rid of that noise while keeping the advantage of such an efficient method against hard binning effects and imprecision in orientation estimation, we propose to use a truncated Gaussian assumption, i.e., we set to zero the 4 bins in each Gaussian density function corresponding to its smallest values. Therefore, the histogram of local orientations in each region is computed as the summation of only the 4 most probable orientation bins for each of the points within that region, i.e., the truncated Gaussians only contribute with their respective 4 most representative bins. The Gaussian densities shown in Fig. 5 have their 4 most representative bins in blue and their “noisy tails” in red.

V. INDEXING

The bag-of-words approach [19] consists in representing documents as simple unordered counts of prototype-terms (words in the case of text documents) defining a so-called dictionary.

This approach has been successfully generalized to different types of data such as images [20], where documents are represented by local image descriptors or patches instead of text words. Since local image descriptors contain continuous values, the generation of a finite dictionary \( D \) requires a quantization process, in which the prototypeterms or bases are first estimated, and then all the local descriptors are assigned to one or more of these bases. The final representation is commonly referred to as bag-of-words (bov) [21].

In the following, we review the k-means algorithm which is widely used to quantize continuous signals. Then we present the sparse coding approach, and more specifically, the KSVD algorithm [1] which has been derived as a generalization of k-means, and that has been used as a method to adapt dictionaries and to find sparse linear combinations for a given set of signals. We also explain some of the pooling strategies that can be used to estimate bov based on sparse representations.

A. K-means as coding method

Given a set \( X \) of \( I \) input signals \( x_i \) (e.g., local image descriptors), k-means estimates the column elements (bases) of the dictionary matrix \( D = [d_1, d_2, \ldots, d_K] \) by looking iteratively for clusters \( c_j = \{x_i | g(x_i) = j\} \), where \( g(\cdot) \) denotes the cluster assignment function, such that the square of the euclidean distance of each descriptor \( x_i \) to the center of its respective cluster \( d_j \) (basis) is shorter than the distance to any other center \( d_k \) [2]:
\[ g(x_i) = j \iff \|x_i - D\omega_i\|^2 \leq \|x_i - D\omega_k\|^2, \forall k \neq j, \]
where \( \| \cdot \|^2 \) denotes the square of the l2 norm, and \( \omega_i \) is the unit weight row vector with its j-th entry being set to one and the rest to zero and it is associated to the signal \( x_i \). In other words, the problem consists in finding the solution to,
\[ \min_{D,\Omega} \{\|X - D\Omega\|^2_F\} \quad \text{s.t.} \forall i, \|\omega_i\|_0 = 1, \]
where \( \| \cdot \|^2_F \) denotes the Frobenius norm, \( \|\omega_i\|_0 \) is the l0 pseudo-norm defining the number of non-zero entries in \( \omega_i \), and \( \Omega \) is the matrix of weight row vectors \( \omega_i \). Allowing \( \omega_i \) to be a normalized vector with more than one non-zero entry corresponds to a weighted fuzzy assignment to more than one cluster [27].

A common option to chose the initial set of cluster centers, is to use a random subset of the input signals. Later in each iteration the cluster centers are recomputed as the component-wise mean of all the descriptors within each cluster.

Previous works on retrieval of shape images [12] have shown empirically that a variant of k-means that uses the ‘city block’ distance performs better than the ‘euclidean’ distance. That is, the distance between two vectors is computed by the l1 norm, and the centroid of each cluster is computed as the component-wise median of all the points within the cluster.

B. Sparse Coding via KSVD

The work in [28] presents sparse coding (SC) as a generalization of the quantization problem, representing the input signals \( X \) as (sparse) linear combinations of the bases in the dictionary \( D \). Due to the unfeasibility of computing the ideal solution [29], a good approximation is estimated by,
\[ \min_{D,\Omega} \{\|X - D\Omega\|^2_F\} \quad \text{s.t.} \forall i, \|\omega_i\|_0 \leq T, \]
where \( T \) is a parameter to control the number of basis functions allowed to be combined for the reconstruction of the input signals, i.e., Eq. (6) allows \( \omega_i \) to be a weight row vector with more than one non-zero entry.

Similar to k-means, the KSVD algorithm solves this minimization problem in two iterative steps. First, given a fixed dictionary \( D \), the coefficients \( \Omega \) are found by the use of any pursuit algorithm like Matching Pursuit [30] or Orthogonal Matching Pursuit [28]. After that, the dictionary is updated one basis at a time using singular value decomposition (SVD). This update of each basis function \( d_k \) is performed allowing changes in the components of the coefficients \( \omega_i \) associated to it, which results in an accelerated convergence [1]. Here as well, a common option to choose the initial dictionary is to use a randomly selected subset of the input signals.
D. Building bag models from sparse coefficients

When the quantization is made by k-means, each descriptor is associated to a single cluster, and computing the \( \text{bov} \) representation for a given image is as simple as applying a threshold filtering to the coefficient column vectors, which is also the dictionary elements. We have not modified this normalization but only the reconstruction error function (6) to be the \( l_1 \) norm. Thus for each HOOSC vector, we minimize:

\[
\|x_i - D\omega_i\|_1 \text{ s.t. } \forall i, \|\omega_i\|_0 \leq T. \tag{7}
\]

Note that KSVD relies in a singular value decomposition step which requires a \( l_2 \) normalization of the dictionary. We have not modified this normalization but only the reconstruction error function.

C. Sparse coding for HOOSC descriptors.

With a slight abuse of terminology, a signal is said to be sparse if it can be decomposed into a sparse linear combination of a set of basis functions. We noticed that building \( \text{bov} \) applying sparse coding decomposition directly to HOOSC descriptors performs very poor in retrieval experiments. We investigated the effects of enforcing sparsity by applying a threshold filtering to the components of the HOOSC descriptors.

More specifically, we have set to zero all the components in the HOOSC descriptors whose value was below a certain threshold \( \tau \). This thresholding step increases the sparsity of the input signals and facilitates their sparse decomposition. We observed that it helped improve the average precision of retrieval experiments that use \( \text{bov} \) representations computed based on the sparse coefficients.

Note that in order to achieve robust reconstruction of the input signals, the set of bases must be an over complete dictionary, i.e., the number \( K \) of basis functions must be (much) larger than the number \( N \) of input signals, \( K \gg N \) in Fig. 6.

Based on the observation that the ‘city-block’ distance improves the results obtained by \( k \)-means [12], we evaluated the effects of combining it with the KSVD algorithm. More specifically, we changed the norm in the reconstruction error function (6) to be the \( l_1 \) norm. Thus for each HOOSC vector, we minimize:

\[
\|x_i - D\omega_i\|_1 \text{ s.t. } \forall i, \|\omega_i\|_0 \leq T. \tag{7}
\]

Note that KSVD relies in a singular value decomposition step which requires a \( l_2 \) normalization of the dictionary elements. We have not modified this normalization but only the reconstruction error function.

C. Sparse coding for HOOSC descriptors.

With a slight abuse of terminology, a signal is said to be sparse if it can be decomposed into a sparse linear combination of a set of basis functions. We noticed that building \( \text{bov} \) applying sparse coding decomposition directly to HOOSC descriptors performs very poor in retrieval experiments. We investigated the effects of enforcing sparsity by applying a threshold filtering to the components of the HOOSC descriptors.

More specifically, we have set to zero all the components in the HOOSC descriptors whose value was below a certain threshold \( \tau \). This thresholding step increases the sparsity of the input signals and facilitates their sparse decomposition. We observed that it helped improve the average precision of retrieval experiments that use \( \text{bov} \) representations computed based on the sparse coefficients.

D. Building bag models from sparse coefficients

When the quantization is made by \( k \)-means, each descriptor is associated to a single cluster, and computing the \( \text{bov} \) representation for a given image is as simple...
as counting how many descriptors of each cluster this image has. However, sparse coding approach associates each descriptor to several bases (visual words), where the (sparse) coefficients denote the strength of that association. Thus, we can explore different pooling criteria to find a function $f$ that maps the sparse coefficients into bov vectors. Fig. 6 shows the schematic representation of the process of mapping the matrix $\Omega$ of sparse coefficients of a given glyph to its bov representation. Note that the matrix $\Omega$ is indexed by $k = 1, \ldots, K$ (number of bases), and $i = 1, \ldots, N_I$ (number of local descriptors of the given glyph); whereas the bov is a vector indexed only by $k = 1, \ldots, K$.

Some of the pooling approaches available to compute bov representations are:

- **Average Pooling (AVP).** For a given glyph, it assigns as value to each visual word the average of its corresponding responses computed over the whole set of descriptors. In other words, the final bov is the average of the absolute values of each row in $\Omega$,

$$\bar{bov}_k = \frac{\sum_i \text{abs}(\Omega_{ki})}{N_I}, \quad (8)$$

where $\text{abs}(\cdot)$ denotes absolute value, and $N_I$ is the number of descriptors for the given glyph.

- **Max-N Weight Pooling (Max-NWP).** It consists in building the bov vector as the sum of the weights of the $N_{\text{max}}$ coefficients having the maximum responses. More precisely, let $f_{N_{\text{max}}} : \Omega \rightarrow \Omega^{N_{\text{max}}}$ be the function that generates a copy of $\Omega$ setting all its entries to zero, except for those corresponding to the $N_{\text{max}}$ components in each column of $\Omega$, i.e., for each vector of coefficients, $f_{N_{\text{max}}}$ keeps only the $N_{\text{max}}$ maximum responses. The bov is then computed as

$$\tilde{bov}_k = \frac{\sum_i \text{abs}(\Omega_{ki}^{N_{\text{max}}})}{N_{\text{max}}}, \quad (9)$$

- **Max-N Integer Pooling (Max-NIP).** This approach builds the bag representation as the integer count of the basis function associated with the coefficients having the maximum responses,

$$\tilde{bov}_k = \begin{cases} 1 & \text{if } \sum_i |\text{abs}(\Omega_{ki}^{N_{\text{max}}})| > 0, \quad (10) \\ 0 & \text{otherwise,} \end{cases}$$

where $|\cdot|$ denotes cardinality, i.e., the number of times its argument becomes true.

- **Max-N Binary Pooling (Max-NBP).** It builds the bov representation as the binary activation of the basis function associated with the coefficients having the maximum responses. The bov representation is computed as,

$$\tilde{bov}_k = \frac{\sum_i \text{abs}(\Omega_{ki}^{N_{\text{max}}})}{N_{\text{max}}}, \quad (11)$$

Note that strictly speaking, the bov representations are normalized vectors, i.e., all of the above mentioned vectors are normalized as,

$$bov_k = \frac{\tilde{bov}_k}{\sum_{j=1}^{K}\tilde{bov}_j}, \quad (12)$$

Among them, the **Max-1 Integer Pooling (Max-1IP)** method seems to have given the best performance in previous works for image classification task [9].

VI. EXPERIMENTAL PROTOCOL

In this section we present the details about the data set used in our experiments. We also explain the protocol we followed during the extensive evaluation performed under several criteria to assess the performance of the sparse coding and clustering approaches in the construction of bov representations of shape images.

A. Data

We used the data set presented in [13], which consists of 1270 syllabic Maya hieroglyphs gathered from different archaeological sources, and that are distributed over 24 visual classes (syllabic representations). All the 24 classes are subdivided in two subsets: candidates ($G_C$) and queries ($G_Q$). Around 80% of examples of each class are selected as candidates and used to build the representation model (clusters with $k$-means, or sparse dictionary with KSVD). The remaining 20% of the examples of each class are used as queries to evaluate the retrieval performance of the studied indexing techniques. Fig. 7 shows two tables of visual examples of this data set, the first table contains one candidate per class, and the second table has one query per class.

B. Evaluation metric for retrieval experiments

In all our retrieval experiments, we have used the average precision (AP) metric to evaluate the retrieval performance. This metric consists in ranking, in decreasing order, all the candidate elements according to their visual similarity to a given query, and then computing the average of the ranking precisions of all the candidates that are relevant to the query (i.e., belong to the same visual class). In the ideal case, all the relevant documents would be retrieved at the top of the ranking vector, thus their individual precision would be 1.0, as well as the AP.

To compare the performance of the different criteria used to construct bag representations, and their respective set of parameters, we used the mean of the average precision over the whole set of queries, denoted $mAP$.

C. Evaluation procedure

We started performing the dictionary learning process via $k$-means or KSVD. From the candidates subset ($G_C$), we chose randomly 1500 descriptors from each of the 24 classes and used them to estimate dictionaries of different sizes. Then, using the dictionary model of ($G_C$), we computed the bov representation of each glyph. For the KSVD cases, the bov construction step was repeated several times according to different pooling techniques.
Previous works on retrieval of shape images [12] have shown empirically that there is a variant of k-means that uses the ‘city block’ distance and performs better than the ‘euclidean’ distance. That is, the distance between two vectors is computed by the $l_1$ norm, and the centroids are considered to be the component-wise median of all the points within each cluster, we implemented this approach and refer to it as k-means-$l_1$. In order to compare the reconstruction error between k-means and KSVD, we used both distances with the sparse coding approaches. We refer to them as KSVD-$l_1$ and KSVD-$l_2$.

During the retrieval experiments, we build the bov representation of each query glyph using the tested pooling technique, and compare it against the bov of the glyphs in $G_C$ using $l_1$ distance, we then rank the resulting distances. Finally, we estimate the AP of each query from the ranking of the candidate glyphs belonging to the query class, and compute the mAP of the current representation to evaluate its performance.

Namely, the experiments we performed are:

1) We evaluated the retrieval performance of the four pooling techniques explained in section V-D to build the bov representations. To this end, we considered different numbers $N_{max}$ of the bases having the maximum responses.
2) We also evaluated the impact of the thresholding procedure used to enforce sparsity for different values of the parameter $\tau$.
3) We performed retrieval experiments using dictionaries of different sizes, estimated with k-means-$l_1$, KSVD-$l_1$, and KSVD-$l_2$. We did not evaluate the performance of k-means-$l_2$ as it has been shown that k-means-$l_1$ gives better retrieval precision.
4) To acquire a clearer idea of in the behavior of the mAP of the different approaches, we compared the reconstruction errors achieved by k-means and KSVD, both with $l_1$ and $l_2$ distances.
5) To investigate the combination of methodologies that better discriminate visual classes of glyphs, we have computed the inter-class distance between two hieroglyph classes $A$ and $B$ as the average pair-wise distance between each instance of class $A$ with respect to each instance of class $B$ (also obtaining the intra-class distance when $A = B$). We performed this inter-class similarity study comparing the k-means and KSVD approaches that achieved the best retrieval precision, and using two different distance metrics: Euclidean and Jensen-Shannon Divergence [31].

6) A study was conducted to evaluate the potential of our methods to automatically discover visual patterns in shape descriptors of Maya hieroglyphs. A tool with such a capacity is of great interest for archaeologist, as it could suggest visual similarities of symbols based on local visual patterns. To this end, we localized the most frequent visual words in each class and its associated closest pivots, then we looked at its neighbor points that are used to construct its HOOSC descriptor.

**VII. RESULTS**

In this section we present the results of our extensive evaluation. To facilitate their reading, we decided to show these results divided by subsets, that is, each subsection discusses the results that are the most relevant to it.

A. Pooling schemes evaluation

First, we evaluated the performance of the four different pooling schemes to construct bov representations based on KSVD: Average Pooling (AVP), Max-N Binary Pooling (Max-NBP), Max-N Integer Pooling (Max-NIP), and Max-N Weighted Pooling (Max-NWP). Fig 8(a) shows the mAP retrieval results obtained when comparing bov vectors that are computed using the AVP (flat-continuous line), it also shows the performance curves for the Max-N Binary Pooling for various values of $N_{max}$ (see section V-D). In general, using hard assignments to only the basis with the highest response gives better results than any of the other options.
In Fig. 8(b), the retrieval results of Max-NIP are shown for distinct values of $N_{\text{max}}$. Note that the curves in the binary and integer cases have similar behavior, and that the less coefficients used to estimate the $\text{bov}$ representation, the better the retrieval results. Since the coefficients represent weights for linear combinations, this might sound counter intuitive as it could be expected that a weighted assignment to visual words could help reconstruct better the original signal. This was not the case in practice.

The curves shown in Fig. 8(c) present results when the $\text{bov}$ are computed combining the actual weights of the coefficients corresponding to the highest responses (Max-NWP). In this case, varying the maximum number of coefficients has little impact in the performance. Also, all of the results in Fig. 8(c) perform below the Max-1 Integer pooling shown in blue-diamond in Fig. 8(b). Furthermore, the Max-1 Integer pooling strategy outperforms any of the other approaches.

### B. Facilitating the sparse decomposition

We compared the performance of the KSVD method to build $\text{bov}$ representations after performing a thresholding step that sets to zero all the HOOSC components below the threshold $\tau$. By doing so, the sparse decomposition of the HOOSC descriptors results in $\text{bov}$ representations that allow for better retrieval precision. Fig. 9 shows these results. We can see that $\tau = 0.01$ provides slightly better result than $\tau = 0.005$ and $\tau = 0.03$, and that higher threshold values generate very poor results.

### C. Combining $L_1$ with KSVD

The combination of the KSVD algorithm with the $L_1$ (city-block) distance (KSVD-$L_1$) resulted in a slight decrease of the retrieval precision with respect to the original KSVD that uses the $L_2$ distance. Fig. 10 presents the subset of the most relevant curves resulting from this assessment, note that non of them achieves as much precision as the original KSVD.

We noticed that these curves behave similar when we vary the value of the threshold $\tau$, and that there is not significant difference when the pooling strategy is changed. Regarding the HOOSC formulation to be used, there is an important improvement in the retrieval precision achieved by the formulation proposed in this paper (sHOOSC) with respect to the version presented in [13] (HOOSC). However, the improvement after performing the thresholding procedure (stHOOSC) remains modest.

### D. Comparing HOOSC formulations

All the results presented in sections VII-A, VII-B, and VII-C where actually computed for all possible combinations "pooling strategy - threshold value - HOOSC formulation", thus resulting in a large amount of tables to be
analyzed. In this paper, we only present the most relevant results to facilitate their reading, i.e., the results shown about pooling schemes correspond to those computed with the threshold fixed to $\tau = 0.01$, whereas the curves regarding the facilitation of the sparse decompositions correspond to using Max-1 Integer pooling. In all cases we present the results obtained with the HOOSC formulation proposed in this paper.

In Fig. 11 we compare the performance for the different formulation of the HOOSC descriptor when using the KSVD approach with the best combination of parameters, i.e., with the $l_1$ distance (KSVD-$l_1$), using Max-1 Integer pooling, and a threshold fixed to $tau = 0.01$. The formulation of the HOOSC descriptor correspond to the HOOSC presented in [13], and the version introduced in this paper (section IV), both before and after the thresholding procedure.

Different from the result obtained with KSVD-$l_1$, these results show that the simplified HOOSC (sHOOSC) proposed in this paper performs slightly lower than the original HOOSC with a drastic decrease after 4500 bases when the KSVD method is applied. However, after the implementation of the thresholding procedure (stHOOSC), its performance is notably increased, reaching its maximum when 5000 bases are used as dictionary elements.

Fig. 11 also shows the results obtained by the $k$-means quantization method (for these experiments we only show the results of the ‘city-block’ distance). We refer to the corresponding results as $k$-means-$l_1$. The simplified HOOSC (sHOOSC) has a consistently better performance, around 5% more than the HOOSC in [13]. Note that the performance of the descriptor that uses the threshold step (stHOOSC) has no considerable difference with respect to the simplified HOOSC. In general, the performance of the three versions of the descriptor tends to degrade after 4000 clusters.

Overall, we noticed that KSVD does not seem to achieve as good retrieval results as the traditional $k$-means method. The best result obtained with KSVD (0.554) using 5000 bases and the euclidean distance, is lower than the corresponding result of 5000 clusters of $k$-means (0.582), and lower than the best result of $k$-means, obtained with only 3500 clusters (0.585).

### E. Comparing the numerical error

To better understand the behavior of $k$-means and KSVD, we compare the reconstruction error achieved by both methods. However, given that by nature these two metrics can have different order of magnitude, a direct comparison of them might not be correct. This is due to that the $l_1$ distance accumulates the absolute sum of the dimension-wise differences between two vectors, while the $l_2$ correspond the their euclidean distance, i.e., in the case of density functions, $l_1$ will result in higher values.

In Table I, we show the average reconstruction error achieved by $k$-means and KSVD during the dictionary learning process using the $l_1$ distance. In each case, we present the result corresponding to the best HOOSC formulation, that is: sHOOC for $k$-means and stHOOSC for KSVD, with 3500 and 5000 visual words respectively.

<table>
<thead>
<tr>
<th>visual-words</th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5000</th>
<th>6000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-means-$l_1$</td>
<td>1.233</td>
<td>1.127</td>
<td>1.029</td>
<td>0.976</td>
<td>0.914</td>
<td>0.861</td>
</tr>
<tr>
<td>KSVD-$l_1$</td>
<td>1.452</td>
<td>1.413</td>
<td>1.363</td>
<td>1.374</td>
<td>1.440</td>
<td>1.519</td>
</tr>
</tbody>
</table>

In Table II, we show the average reconstruction error achieved by $k$-means and KSVD during the dictionary learning process using the $l_2$ distance. In each case, we present the result corresponding to the best HOOSC formulation, that is: sHOOC for $k$-means and stHOOSC for KSVD, with 3500 and 5000 visual words respectively.

<table>
<thead>
<tr>
<th>visual-words</th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5000</th>
<th>6000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-means-$l_2$</td>
<td>0.032</td>
<td>0.030</td>
<td>0.026</td>
<td>0.024</td>
<td>0.018</td>
<td>0.012</td>
</tr>
<tr>
<td>KSVD-$l_2$</td>
<td>0.270</td>
<td>0.267</td>
<td>0.201</td>
<td>0.145</td>
<td>0.132</td>
<td>0.127</td>
</tr>
</tbody>
</table>

We can see that $k$-means has a consistent lower reconstruction error than KSVD when the $l_1$ distance is used. Also note that, for $k$-means, the reconstruction error tends to decrease as the number of bases increases, whereas it exhibits a local minimum at 3000 bases in the case of KSVD. In Table II we show similar results computed with the $l_2$ distance. In this case the reconstruction error continues decreasing as the number of bases decreases for both algorithms, and $k$-means remains consistently as the method with lower reconstruction error.

To graphically illustrate our results, in each row of Fig. 12 we present one retrieval example per class. The first column shows the corresponding best query in each class, i.e., the query with the highest AP value. The remaining columns correspond the the most similar candidate ($G_C$) hieroglyphs retrieved in the top 10 positions of the ranking vector. The candidates enclosed in a blue rectangle correspond to relevant documents (hieroglyphs of the same visual class). These results have been generated using the
best retrieval method, i.e., $k$-means-$l_1$ with sHOOSC (Fig. 11). Note that some classes are easy to match as most of the top 10 retrieved elements proved to be relevant. However, there are still few classes whose elements are confused due to a high inter-class visual similarity, e.g., some classes share visual patterns such as lattices and horizontally elongated shapes.

### F. Distance estimation between visual classes

As mentioned in section VI-C, having a method to estimate class distances of Maya hieroglyphs would allow to find the most probable visual classes of new discovered symbols. Fig. 13 shows the inter-class distances for each pair of classes, computed as Euclidean distance and as the Jensen-Shannon divergence ($D_{JS}$) for the best results of $k$-means-$l_1$ and KSVD-$l_2$, i.e., 3500 clusters and 5000 bases, respectively. Note that the main diagonal of the matrices showed correspond to the intra-class similarity.

In general, using $D_{JS}$ the intra-class distance is always smaller than the inter-class distances, whereas this is not true for the Euclidean distance. This suggests that using $D_{JS}$, it is possible to discriminate new symbols with higher accuracy. Table III shows the percentage of the average number of times the intra-class distance is smaller than the inter-class distance.

### Table III. Percentage of times the intra-class distance is minimal compared to the inter-class distance.

<table>
<thead>
<tr>
<th></th>
<th>$k$-means</th>
<th>KSVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>$D_{JS}$</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

### G. Visual patterns recovery

We noticed that some visual patterns (visual words) are more descriptive than other for certain classes, i.e., some visual patterns contribute more than other in the \textit{bov} representations of glyphs within a given class. From the 3500 visual words estimated with $k$-means-$l_1$ and the sHOOSC approach, in Fig. 14 we show graphical examples of the two most common visual words for some of the Maya syllabic classes. Each graphical example corresponds to the closest point to the two most popular clusters used in the \textit{bov} representations within each class. Note that the sHOOSC method only uses the distance scope up to 1 (see section IV), therefore only the red points are part of the descriptors; we show the whole image with the purpose of providing visual context to the reader.

Overall, our approach produces \textit{bov} representations that are visually consistent with one another, as they have similar weight in their corresponding components. That is to say, they use the same visual patterns in similar proportions. Furthermore, the sHOOSC often assigns such visual patterns to the same visual words. We believe that in the future, this observation might allow to describe Maya hieroglyphs based on localized visual patterns automatically discovered.

### VIII. Conclusions

In this work, we have presented recent advancements towards the visual description and automatic retrieval of Maya hieroglyphs.

The main contribution of this work is the evaluation of the performance of two quantization approaches in the construction of bag representations of local shape descriptors, and more specifically, the HOOSC descriptor. We have assessed the retrieval performance of the sparse
coding algorithm KSVD in the task of content-based shape-images retrieval. Sparse coding is a recent trend that has gained popularity for description and classification of natural images, and our work is a first exploration of the use of sparse coding decompositions as quantization method of contextual shape descriptors. We have evaluated the retrieval performance of this method with different pooling strategies. We believe that this assessment allows to confirm the conclusions from previous classification works that (i) depending on the given application, sparse techniques might or not perform better, and (ii) in the context of sparse techniques with pooling, often max-pooling provides the best results.

We proposed a version of the HOOSC descriptor that produces better retrieval results, and that also resulted in vectors with smaller size. We implemented an efficient method to facilitate the sparse decomposition of the HOOSC descriptor, this method consists in setting to zero all the signal components that are below a certain threshold. The evaluation of this method shows that with it the bag of visual words representations can achieve better retrieval performance. We also compared the performance of KSVD with the traditional k-means with respect to k-means clustering for

max-pooling provides the best results.

We compared the performance of KSVD with the traditional k-means clustering for

max-pooling provides the best results.

In our study, we also proposed a method to measure distances between pairs of visual classes of Maya hieroglyphs. This measure can be used to find the most probable classes of glyphs recently discovered, and to analyze visual relationships among visual classes. Finally, we manually analyzed the visual patterns that our methods are able to encode and recover. We observed that the visual patterns encoded by HOOSC descriptors via k-means clustering are consistent across shapes that share similar visual aspects. We believe that this method could be potentially used to automatically discover visual patterns that represent hieroglyphs, and shapes in general, in a robust manner. We plan to investigate deeper this particular idea in future work.

ACKNOWLEDGMENT

We thank the support of the Swiss National Science Foundation through the CODICES project. We thank INAH for the data of the AJIMAYA project, and specially Carlos Pallan Gayol (University of Bonn) for his effort in the selection of the instances used in this work.

REFERENCES

Figure 14. For some of the visual classes, the two most common visual patterns recovered by sHOOSC under $k$-means-$l_1$ clustering. A whole glyph is plotted to show visual context, though only the points in red are actually used for description.


Edgar Roman-Rangel is a Ph.D candidate at the École Polytechnique Fédérale de Lausanne (EPFL), and research assistant at the Idiap Research Institute, Switzerland. He received his MS degree in Computer Sciences from the Monterey Institute of Technology and Higher Education (ITESM) in 2006. His research interests include multimedia analysis and retrieval, computer vision, and machine learning.

Jean-Marc Odobez is a Senior Researcher at Idiap since 2011 and Maitre d’Enseignement et de Recherche (MER) at EPFL. Between 1996 and 2001, he was Associate Professor at the University du Maine, France. He has worked for several years on the development of statistical models for image representation and segmentation, object recognition, tracking, human activity recognition and multimedia content analysis. He has published over 20 journal papers and 80 conference papers and holds several patents on motion analysis. He is the co-founder of Klewel, a company dedicated to recording and analysis of multimedia presentations.

Daniel Gatica-Perez is Senior Researcher at Idiap Research Institute and Maitre d’Enseignement et de Recherche at the Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland, where he directs the Social Computing group. His research has been funded by the Swiss National Science Foundation, the US government, the European Union, and industry. He has served as Associate Editor of the IEEE Transactions on Multimedia, Image and Vision Computing, Machine Vision and Applications, and the Journal of Ambient Intelligence and Smart Environments.