Large-Scale Document Image Retrieval and Classification with Runlength Histograms and Binary Embeddings

Albert Gordo[a,*], Florent Perronnin[b], Ernest Valveny[a]

[a]*Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain*
[b]*Xerox Research Centre Europe (XRCE), Grenoble, France*

Abstract

We present a new document image descriptor based on multi-scale runlength histograms. This descriptor does not rely on layout analysis and can be computed efficiently. We show how this descriptor can achieve state-of-the-art results on two very different public datasets in classification and retrieval tasks. Moreover, we show how we can compress and binarize these descriptors to make them suitable for large-scale applications. We can achieve state-of-the-art results in classification using binary descriptors of as few as 16 to 64 bits.

Keywords: visual document descriptor, compression, large-scale, retrieval, classification

*Corresponding author.
Email addresses: agordo@cvc.uab.es (Albert Gordo), florent.perronnin@xrce.xerox.com (Florent Perronnin), ernest.valveny@cvc.uab.es (Ernest Valveny)

1. Introduction

In the last few years there has been an increasing interest in dealing with large amounts of visual data, both in the research community and the business environment. Companies are dealing with millions of scanned or digital documents, and there is a real need to perform classification and retrieval tasks on these large corpora. For example, classifying an incoming document may be useful to automatically decide which workflow the document should be sent to depending on the category. We can also be interested in retrieving existing documents in a dataset similar to the incoming document. For example, if the documents in the dataset are annotated, retrieving the most similar documents may be useful to extract some metadata from them and transfer it to the incoming document.

This classification and retrieval problems have traditionally been approached using textual information. However, extracting that information may be complicated or unfeasible, since documents can be old, of low quality, in different languages, or may contain little textual information. In those cases it is necessary to rely on visual features. However, as opposed to the natural image domain, where there is a very active research on large-scale (see, *e.g.*, the recent [1, 2]), we are unaware of any work on large-scale document classification or retrieval based on visual features.

In fact, works dealing with document image classification that do not rely on OCR are not abundant. In [3], a combination of a variable length descriptor based on smearing and a sequence matching based on dynamic programming is used to retrieve document pages. In [4], an X-Y tree is constructed based on the layout of the page and tree edit distance is used to

classify the documents. Tree grammars are used to compensate for possible segmentation errors. In a similar way, [5] also constructs a tree-based representation of the form layout. Then the syntactic representations of the documents are used to infer a tree automaton for each one of the classes involved in the task. In [6], some classification methods based on layout distance such as the Minimum Weight Edge Cover or the Earth Mover Distance are tested on a publicly available dataset. In [7], The layout of the page is flattened into a sequence of blocks and compared with cyclic Dynamic Time Warping. The result is a rotation invariant distance measure, but the comparison between layouts is slow. In [8], a multi-scale density decomposition of the page is used to produce a fixed-length descriptor that can be compared by means of Euclidean distance. Feature vectors can be constructed efficiently using integral images and compared with fast operations as the dot product. In [9], a feature vector based on texture analysis is extracted from the image and the Euclidean distance is used to compare the feature vectors. In [10], a representation based on density changes after multiple morphology operations is used to represent the documents. These representations are later compared using the Euclidean distance. In [11], a feature vector is constructed based on image features such as percentages of text and non-text, column structures, density of content area, or connected components features. Then the feature vectors are classified using decision trees and self-organizing maps. In [12], a Latent Conditional Independence model is used along with variable-length Viola-Jones-based features to describe and classify forms of the NIST dataset.

When performing retrieval on large datasets, there are two key aspects

that must be taken into account. First, the *memory* cost: the memory footprint of the documents should be small enough so that all the database signatures fit in RAM. Otherwise, the response time of a query may collapse since the disk access is much slower than that of RAM access. Second, the *computational* cost: since we are dealing with large datasets, the computation of the distance between signatures should rely on efficient operations.

In general, none of the methods previously reviewed could fulfill the large-scale needs. Some of them rely on variable-length descriptors such as sequences, graphs, or trees [3, 4, 5, 6, 7, 12]. Although these are powerful representation, distances between such representations are not as efficient to compute as the distances between two vectors since they usually rely on costly methods. Moreover, variable-length descriptors are harder to compress than vectors of fixed length. Other methods rely on the layout analysis of the document [11, 3, 4, 5, 6, 7]. Although working directly with the layout could lead to more powerful representations, layout analysis is slow and error prone, and it is desirable to skip this step whenever possible. Finally, some methods are based on density decomposition [8, 10]. Unfortunately, this kind of representation is very sensitive to noisy documents. Since we are dealing with scanned documents of varying quality, this is a situation we have to deal with.

In this paper, we present a new document representation based on multi-scale runlength histograms. While runlength histograms have been used in the past in the document analysis community (*e.g.*, [13, 14, 15]), to the best of our knowledge, they have never been used to represent documents as a whole. This representation does not need any kind of layout analysis since

it is based solely on visual features. However, the use of multi-scale regions will provide some basic structural information. These runlength histograms have a fixed-length representation, thus allowing fast comparisons, and are less sensitive to noise than descriptors based on density decomposition. Runlength histograms are also fast to compute. As we show in the experimental section, these histograms achieve state-of-the-art classification results on two public datasets.

In addition, these histograms can be adapted to satisfy the aforementioned memory and computational needs. To make these histograms suitable for large-scale retrieval, we binarize them by means of Principal Component Analysis Embedding (PCAE) [16]. These binarized documents need significantly less memory and can be compared very efficiently using the Hamming distance. Finally, we will explore two ways to deal with shortcomings of PCAE: i) orthogonal random rotations, to better distribute the variances between the PCA dimensions, and ii) asymmetric distances between binarized and non-binarized documents [16].

The rest of the paper is organized as follows. In Section 2 we describe the runlength histogram representation for document images. In Section 3, we discuss the use of PCAE for compressing and binarizing runlength histograms, as well as the use of asymmetric distances. Finally, we present our experimental results in Section 4 and conclude with Section 5.

2. The Runlength Histogram Representation

The use of runlengths is not new in the document analysis community. In [14], runlength features are used to help classifying document zones as

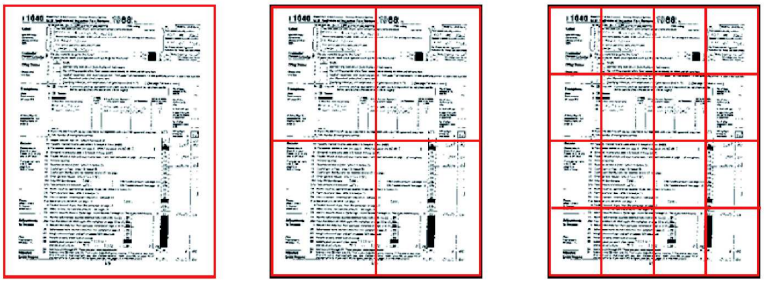

Figure 1: Original image and the 1×1, 2×2 and 4×4 partitions.

text or non text. In the recent [15], runlength histograms are used to detect the frames of double-paged document images. In [13], textures are described by means of runlengths. However, we are not aware of any use of runlength histograms for a whole page representation. We propose the use of multi-scale runlength histograms for such a task. The document encoding is based on the following steps:

Step 1. Normalization (optional): Several steps can be performed to normalize images, such as centering, cropping, re-scaling, skew correction, *etc*. Throughout our experiments, we have only performed a re-scaling of the images to 500,000 pixels while keeping the aspect ratio.

Step 2. Pixel quantization: The runlength encoding requires a small number of levels to be efficient. In our case, we deal with binary images, *i.e.*, with two levels.

Step 3. Region extraction: The image can be partitioned into different sub-regions at different scales using spatial pyramids [17]. This is a standard technique to add some basic structural information to the representation. These regions will later be described independently and finally concatenated. For example, we can see in Fig. 1 the splits corresponding to 1×1 (whole

image), 2×2, and 4×4 partitions, producing a total of 21 regions.

Note that computing the descriptors at different scales (*e.g.* 1×1 and 2×2) can yield important benefits over computing it only over the small regions (*e.g.*, 2×2): if we apply a non-linear transformation to the histograms – such as the square root that we apply, *cf.* step 6 – then the 1×1 region histogram can no longer be expressed as a linear combination of the 2×2 region histograms and therefore can bring some extra information.

Step 4. Runlength encoding of the regions: A *run* is a sequence of pixels of the same value. The length of the run is the number of pixels such a sequence contains, and the runlength histogram is a histogram of the lengths of the runs. Following [14] we propose to quantize the length of the runs in a logarithmic scale as follows:

$$[1], [2], [3-4], [5-8], [9-16], \ldots, [129-].$$

This results in 9 intervals for each of the quantized levels. When dealing with black and white images, this yields mini-histograms of length $2 \times 9 = 18$. We compute mini-histograms in horizontal, vertical, diagonal and anti-diagonal directions and concatenate them. Assuming again binary images, this yields a region descriptor of length $18 \times 4 = 72$. Note that this logarithmic-scale quantization of the runlengths makes the representation much less sensitive to noise, overcoming one of the main problems of density representations.

Step 5. Global image representation: To represent the document image, we simply concatenate the runlength histograms of all the regions. In our case, that would yield an histogram of $72 \times 21 = 1,512$ dimensions.

Step 6. Normalization: The histogram can later be normalized. Several approaches can be considered:

- Normalize each mini-histogram independently.
- Normalize each region independently.
- Normalize the final histogram as a whole.

Experimental results show little difference between these normalization approaches. In our case, we will perform an L1 normalization over the whole histograms and then square root each of its elements. As noted in [18], the dot-product on square-rooted vectors corresponds to an explicit embedding of the Bhattacharyya similarity, and so it is particularly suited for discrete probabilistic distributions such as our L1 normalized vectors. During preliminary experiments we confirmed this square root normalization to improve the accuracy of the system.

Step 7. Dimensionality reduction: Note that the quantization of the lengths of the runs we proposed is likely to produce lots of zeros in the final histograms, particularly in the sections corresponding to the smaller subregions. In fact, in our experiments, we noticed that approximately 35% of our histogram values are equal to zero. This, along with the fact that using multi-scale histograms may cause correlations between dimensions, suggests that the use of PCA could be beneficial and lead to better results, or, at least, be applied without significant loss.

3. Binarizing Runlength Histograms

One of the key issues in the large-scale classification and retrieval domain is keeping a low memory footprint. For instance, if we consider descriptors of 1,512 dimensions such as our runlength histograms, each document would produce a signature of 6KB when using 4 byte floating-point arithmetic, and a million documents would need approximately 5.5GB of memory. All these documents need to be kept alive in RAM in order to provide a reasonable response time for retrieval applications.

Several algorithms exist for compressing and encoding descriptors into binary codes oriented to nearest neighbor search, *e.g.* Locality Sensitive Hashing (LSH) [19], Spectral Hashing (SH) [20], Product Quantization (PQ) [21], or PCA embedding (PCAE) [16]. Then the binary codes can be compared very efficiently by means of the Hamming distance, *i.e.*, counting the number of bits that differ between the representations. Of those methods, PCAE has been shown to produce very competitive results while being conceptually very simple.

Let $x \in \mathbb{R}^D$ be a document signature. We assume that we have access to a training set of unlabeled document signatures. Let μ be the data mean of that train set, and let w_k be the eigenvector associated to the k-th largest eigenvalue of the data covariance matrix. We define the following 1-D embedding:

$$h_k(x) = q(g_k(x)), \qquad \text{with} \tag{1}$$

$$q(u) = 1 \quad \text{if } u \geq 0 \text{ , 0 otherwise,} \qquad \text{and} \tag{2}$$

$$g_k(x) = w_k'(x - \mu). \tag{3}$$

To produce a b bit embedding function h, we concatenate the first b bits, *i.e.*, $h(x) = [h_1(x)\ h_2(x) \dots h_b(x)]$.

As presented, this approach exhibits two shortcomings. First, in PCAE not all dimensions contain a similar amount of information. However, by binarizing, we give the same weight to each dimension. Second, binarizing the query is not necessary, and produces a significant loss of accuracy. We now describe the shortcomings in more detail as well as explore possible solutions. Note that these solutions are orthogonal and can be applied together.

Balancing the variances: One problem with the PCAE approach is that, after binarization, all dimensions have the same weight when compared with the Hamming distance. However, PCA projections contain much more information in the first dimensions than in the subsequent ones. The last dimensions contain very little information and can be considered noise. Since, after binarization, these noisy dimensions have the same weight as the first dimensions, they may negatively impact the accuracy.

One way to mitigate this problem is to rotate the data with a random orthogonal matrix after applying the PCA projection and before binarizing the data, as done, for example, in [1]. To construct the random orthogonal matrix, we follow [22] and perform a QR decomposition of a random matrix drawn from a $\mathcal{N}(0, 1)$ distribution.

After rotating the data, the information is more scattered across dimensions, as illustrated in Figure 2. We plot the accumulated energy (*i.e.* variance) of each dimension with and without an orthogonal random rotation (RR) for one of the datasets we will use in the experimental section. We can observe how the energy is much more scattered after the rotation. This

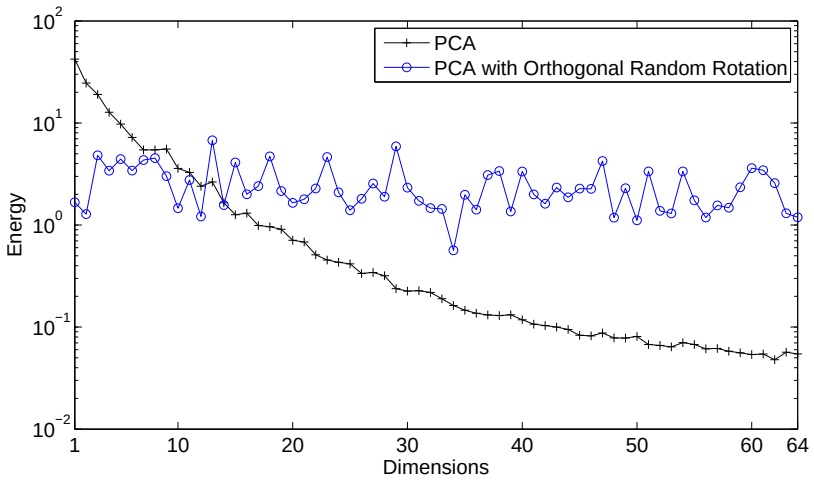

Figure 2: Energy (*i.e.* variance) of the first 64 dimensions of projected NIST signatures. In the PCA case, the energy is concentrated in the first few dimensions. When an orthogonal random rotation is applied after the PCA, the energy is scattered across a larger number of dimensions. Note that the y-axis is in logarithmic scale.

makes the last dimensions carry much more information and should mitigate the aforementioned problem. However, it should be noted that this rotation does not necessarily lead to overall better results. For example, when aiming at a really low number of bits, random rotations will *decrease* the energy of the first dimensions, thus likely producing worse codes. Random rotations can also decrease the accuracy when the dimensions of the descriptors are correlated. Note that, in our case, after projecting with PCA, all dimensions are uncorrelated and therefore this should not be an issue. As we will see through the experiments of Section 4, random rotations generally improve the results.

Asymmetric distances: The binarized descriptors can be efficiently compared by means of the Hamming distance. However, it has been noted that compressing the query is not mandatory [23]. Indeed, the additional cost of storing in memory a single non-binarized signature can be neglected. In this case, we compute distances between a non-binarized query and a binarized dataset. As the distance is computed in two different spaces, these algorithms are referred to as asymmetric. A major benefit of asymmetric algorithms is that they can achieve higher accuracy for a fixed compression rate because they take advantage of the more precise location of the non-binarized query in the descriptor space.

In [16], two asymmetric distances for binary embeddings were introduced. Both were observed to produced similar improvements over the Hamming distance. Here, we focus on the Lower-Bound (d_{LB}) asymmetric distance, since it is simpler and does not requiere any training.

Let d be the squared euclidean distance [1]. Let x be a query document and let y be a database document. Then, $d(x, y) = \sum_k d(x_k, y_k)$. Also, since PCA projections preserve (at least approximately) the Euclidean distance, we have:

$$d(x, y) \simeq \sum_k d(g_k(x), g_k(y)). \tag{4}$$

In the asymmetric case, $g_k(x)$ (and therefore $h_k(x)$) is available, but $g_k(y)$ is not accessible, since the dataset is binarized and we only have access to

[1] We use the following abuse of notation for simplicity: d denotes both the distance between the vectors $g(x)$ and $g(y)$, *i.e.* $d : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$, and the distance between the individual dimensions, *i.e.* $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.

$h_k(y)$. In a nutshell, the idea behind d_{LB} is to lower-bound the sum (4) between the non-binary query and the binary element of the dataset, without actually having to use $g_k(y)$.

Let $\bar{\delta}_{i,j}$ be the negation of the Kronecker delta, *i.e.*, $\bar{\delta}_{i,j} = 0$ if $i = j$ and 1 otherwise. Then, $d_{LB} = \sum_k \bar{\delta}_{h_k(x),h_k(y)} d(g_k(x), 0)$ is a lower bound of $d(x, y)$.

Proof. For each dimension k, we have two different situations: either $h_k(x) \neq h_k(y)$, or $h_k(x) = h_k(y)$.

- $h_k(x) \neq h_k(y)$, *i.e.*, the projected query has a different sign than the dataset element. Then $g_k(x)$ and $g_k(y)$ are on different sides of the plane and a lower bound is $d(g_k(x), g_k(y)) \geq d(g_k(x), 0)$. Moreover, $\bar{\delta}_{h_k(x),h_k(y)} = 1$ and so $d^k_{LB} = d(g_k(x), 0)$. Therefore, $d(g_k(x), g_k(y)) \geq d^k_{LB}$.

- $h_k(x) = h_k(y)$, *i.e.*, the projected query has the same sign than the dataset element. Then we have the following obvious lower bound: $d(g_k(x), g_k(y)) \geq 0$. Moreover, $\bar{\delta}_{h_k(x),h_k(y)} = 0$ and so $d^k_{LB} = 0$. Therefore, $d(g_k(x), g_k(y)) \geq d^k_{LB}$.

Since $d(g_k(x), g_k(y)) \geq d^k_{LB}$ is always true for every k, then $d(x, y) \geq d_{LB}$ and d_{LB} is a lower bound of $d(x, y)$. □

4. Experimental Results

4.1. Datasets and experimental set-up

For our experiments, we report results on two publicly available datasets, the NIST Structured Forms dataset [24], and the Medical Article Records

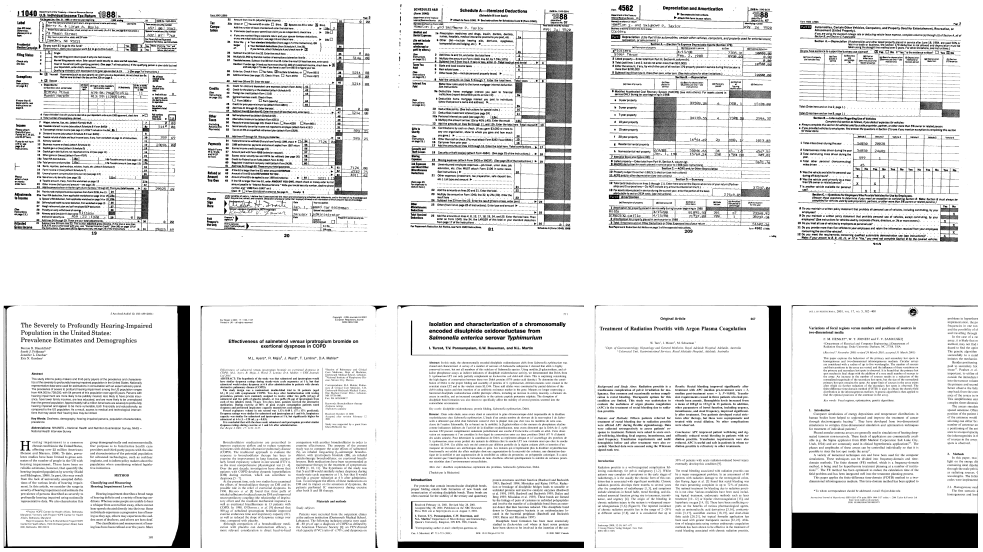

Figure 3: Samples of different classes drawn from NIST and MARG datasets. Top: NIST. Bottom: MARG.

Groundtruth (MARG) dataset [25]. The NIST dataset consists of 5,590 binary documents from 20 different classes of tax forms. The MARG dataset consists of 1,553 documents, first pages of medical journals, and is divided in 9 different layout types.

Figure 3 shows a few sample documents from different classes of both datasets. It is interesting to note how, albeit their very different nature, in both datasets the distinction between classes is based on the structural content of the document.

We report results on classification and retrieval tasks. For learning the PCA transformation of NIST (resp. MARG), we will use a subset of 1,000 documents randomly drawn from NIST (resp. MARG).

4.2. Classification

We first describe the evaluation protocols we used on both document datasets:

NIST: To the best of our knowledge, the best reported results on the NIST dataset where those of [12], based on Viola-Jones-like features: a 99.82% classification accuracy with a 1-NN classifier. The protocol of [12] is as follows: the training set consists of 10 randomly drawn documents from each class (200 documents in total). The testing set comprises all 5,590 documents of the dataset, *including* the 200 documents in the training set. Note that since they use a nearest neighbor classifier, the 200 samples that appear in the training set will always be correctly classified, which makes the reported result an slight upper bound of the real accuracy.

We follow a very similar approach, but repeat the experiment 10 times with different training partitions and average the results, while the experiments of [12] were performed only with one fold. Although we report results following the optimistic evaluation of [12] for fair comparison purposes, we would like to remark that we also performed experiments without including the training samples in the test set and observed no significant decrease in the accuracy results.

MARG: A layout-based classification benchmark over the MARG dataset has been published in [6, 26]. They report results using different layout distance methods such as the Minimum Weight Edge Cover (EC), Assignment (ASS), or the Earth Mover Distance (EMD). They also explore the influence of the block distance used (overlap, Manhattan / Euclidean distance, etc). We follow their procedure and use a 1-NN classifier with a leave-one-out

evaluation protocol.

Table 1 reports our results on the NIST and MARG datasets. We provide two baselines, i) the uncompressed baseline, which computes the Euclidean distance between the original runlength vectors, and ii) the PCA baseline which computes the Euclidean distance between the vectors after PCA but with no binarization (referred to as "PCA no bin" in the tables). Then we report results with PCAE binarization using both the symmetric Hamming distance (Ha) and the asymmetric lower-bound distance (d_{LB}), both with and without random rotations. Based on these results, we consider relevant to address the following points:

PCA compression: As previously hinted, we can apply PCA and greatly reduce the dimensionality of the signatures without significant loss of accuracy. In NIST, we can reduce down to 16 dimensions while still retaining a 100% accuracy. In MARG, we can reduce it down to 64 dimensions with minimal loss: from 94.78% (uncompressed) to 94.46% (64 dimensions).

PCAE binarization: In both datasets, the results with PCAE binarization and Hamming distance show the behavior we predicted: increasing the number of bits can worsen the quality of the descriptor. We can observe how both techniques, RR and d_{LB}, significantly mitigate this problem.

Random rotations: PCAE after random rotations no longer exhibits the problem. However, as we predicted, the results with 8 bits are worse with random rotations than without them, since the random rotations are decreasing the energy in those first dimensions. In fact, the results in MARG, in general, are worse with random rotations, even if the problem is no longer present.

Asymmetric vs symmetric: The asymmetric d_{LB} distance significantly improves the classification accuracy, particularly when the number of bits is small. In the MARG dataset, d_{LB} obtains an improvement of 5.38% absolute and 8% relative for 16 bit signatures, and 4.55% absolute and 5.3% relative for 32 bit signatures. In the NIST dataset the improvements are smaller since the Hamming results are already very high. For 16 and 32 bit signatures, we obtain a 0.77% absolute and 0.1% relative improvement, but achieving a 99.99% accuracy. Also as expected, asymmetric distances also reduce the PCAE problem when the number of dimensions increase

Random rotations + asymmetric distances: Both techniques can be combined to yield even better results. Note however that if random rotation did not produce good results as in the case of the MARG dataset, asymmetric distances should be applied *without* random rotations.

Finally, Table 2 compares our results with the state-of-the-art results published in [12] and [6], as well as the results on the NIST dataset published in [11] and [5][2]. We can observe how, in both datasets, the uncompressed baseline outperforms the state-of-the-art methods (99.82% vs 100% in NIST and 92.6 vs 94.78% in MARG). Moreover, we can significantly compress the signatures while retaining state-of-the-art-results, down to 16 bits in the case of NIST and 64 in the case of MARG.

Furthermore, [12] reports times of "[...] only a few seconds [per page] with an unoptimized Java implementation [...]". With our non-optimized C++ code, it takes approximately 130ms to compute the descriptor of a

[2]Note that [11] and [5] use different evaluation protocols that make use of more training data, and so the comparison of the results should be exercised with caution.

NIST form. After compressing the NIST signatures to 64 bit descriptors, it takes less than 20ms to compare the 5,590 documents against the 200 training samples, using a single CPU of a 3.16GHz Intel Xeon X5460 with 32GB of RAM. In the case of MARG, [26] reports times of 54s for the EC distance calculation and 62s for the ASS distance calculation using an Opteron CPU of 2.4GHz. Using 64 bit descriptors and asymmetric distances, we can compare the 1,553 documents in a leave-one-out strategy in less than 25ms.

4.3. Retrieval

For the retrieval tasks, we will follow a leave-one-out strategy. We will query each dataset item in turn, and we will rank all the remaining documents. As in the classification experiments, we will use the Euclidean distance for the uncompressed and PCA non binarized descriptors, Hamming distance (with and without random rotations) for the PCAE binary descriptors, and d_{LB} for the binarized descriptors in the asymmetric case (also with and without random rotations). The reported result will be the mean Average Precision (mAP) of all the queries. The Average Precision can be interpreted as the area beneath the Precision-Recall curve, and it is a standard measure in retrieval systems. We will follow the same procedure both for the NIST and the MARG datasets.

The retrieval results can be seen in Table 3. Note how we can apply the same conclusions that we drew after the classification experiments:

1. PCA compression can be used to significantly reduce the dimensionality of the vectors with no noticeable loss of precision.
2. PCAE binarization and Hamming distance still produce worse quality results as the number of bits increase. This is even more noticeable

Table 1: Classification accuracy (in %) as a function of the number of dimensions / bits on the NIST and MARG datasets with and without an orthogonal random rotation (RR). Ha: Hamming distance. d_{LB}: Asymmetric lower-bound distance. The uncompressed NIST baseline is 100%. The uncompressed MARG baseline is 94.78%.

NIST					
dimensions	8	16	32	64	128
PCA no bin	99.99	100	100	100	100
Ha	92.73	99.25	97.40	92.51	84.48
Ha (RR)	89.6	99.91	99.92	**100**	**100**
d_{LB}	**93.89**	99.92	99.95	99.94	99.95
d_{LB} (RR)	90.01	**99.99**	**99.99**	**100**	**100**

MARG						
dimensions	8	16	32	64	128	256
PCA no bin	74.31	89.76	92.79	94.46	94.66	94.91
Ha	33.47	68.22	84.85	90.21	92.48	91.72
Ha (RR)	31.37	62.70	80.71	88.94	92.20	93.57
d_{LB}	**33.70**	**73.60**	**89.40**	**93.86**	**94.51**	**95.05**
d_{LB} (RR)	30.71	68.71	86.17	92.65	93.73	94.51

here than in the classification experiments. Rotating the data helps mitigating this problem. In the retrieval experiments, rotating the data also yields better results in both datasets.

3. Asymmetric distances still produce significantly better results than the

Table 2: Comparison of classification accuracy (in %) results on the NIST and MARG datasets.

NIST	
Method	Acc (%)
[12] Viola-Jones-based features	99.82
[11] Decision tree	99.70
[11] SOM	96.85
[5] Decision tree	98.82
Ours Uncompressed	**100**
Ours Asymm RR 16 bits	**99.99**

MARG	
Method	Acc (%)
[6] EC (Overlap + Manhattan)	92.6
[6] EC (Overlap)	91.8
[6] ASS (Overlap)	77.1
[6] EMD (Overlap)	79.8
Ours Uncompressed	**94.78**
Ours Asymm 64 bits	93.86

Hamming distance for the binarized vectors.

One may also note the low retrieval results obtained in the MARG dataset, compared to the high results obtained in classification. A plausible expla-

nation to this effect lies in the way MARG is groundtruthed. The criteria used to define the nine categories are the position and shape of some information as the title, authors, affiliation, abstract, etc (see Fig 4). However, the number of columns is *not* taken into account. Indeed, documents of any given layout category exist in one, two, or, sometimes, even three columns format. When retrieving documents of, *e.g.*, one column, it is reasonable to expect the two-column documents of that category to be badly ranked, thus significantly dropping the retrieval results as we have experienced.

4.4. Large-scale Retrieval

Unfortunately, we are not aware of any publicly-available large-scale documents dataset in which to perform the large-scale experiments. Instead, we will combine the documents of NIST with those of an in-house, real-world dataset. This in-house dataset contains approximately 50,000 unlabeled documents. This dataset is more heterogeneous than NIST, and the documents range from IDs and forms (similar to those of NIST) to coupons or handwritten letters. We refer to this combined dataset as NIST+50k.

In this experiment, we will query each of the original 5,590 NIST documents and rank all the elements in the NIST+50k set, which include the relevant items. As in the previous experiment, we will report the mean AP of all the queries.

Results can be seen in Table 4. We can see how, as in the previous experiments, performing a random rotation is very important when we are interested in the pure Hamming distance results (compare the 23.99% *vs.* the 99.98% at 128 bits), and also how, for a given output size of bits, using asymmetric distances can yield very significant improvements, particularly

Table 3: MAP (in %) as a function of the number of dimensions / bits on the NIST and MARG datasets. The NIST uncompressed baseline is 100%. The MARG uncompressed baseline is 31.97%.

NIST					
dimensions	8	16	32	64	128
PCA no bin	99.99	100	100	100	100
Ha	88.55	95.81	77.23	57.14	42.22
Ha (RR)	90.06	98.94	99.64	100	100
d_{LB}	92.94	**100**	99.78	99.74	99.73
d_{LB} (RR)	**94.29**	**99.92**	**100**	**100**	**100**

MARG						
dimensions	8	16	32	64	128	256
PCA no bin	27.52	30.09	31.55	31.95	32.01	32.00
Ha	21.82	25.64	27.40	27.00	25.45	23.64
Ha (RR)	22.98	26.20	28.34	29.35	30.75	31.39
d_{LB}	23.67	27.33	29.06	29.76	29.91	29.96
d_{LB} (RR)	**24.33**	**27.40**	**29.66**	**30.57**	**31.39**	**31.80**

when aiming at a low number of bits (71.27% *vs.* 80.66% at 8 bits). We can also observe how the results are similar to those obtained in the NIST dataset (Table 3), suggesting that the multi-scale runlengths preserve their highly discriminative power in large-scale scenarios, even when reduced to as few as 32 bits.

Table 4: MAP (in %) as a function of the number of bits on the NIST+50k dataset.

NIST+50k					
dimensions	8	16	32	64	128
Ha	71.27	79.03	55.40	35.06	23.99
Ha (RR)	73.59	90.77	96.24	99.49	**99.98**
d_{LB}	**80.66**	**96.22**	94.57	93.73	93.44
d_{LB} (RR)	79.58	95.39	**99.38**	**99.88**	**99.97**

5. Conclusions

In this paper we introduced a new visual descriptor for document images based on multi-scale runlength histograms. This descriptor does not require any kind of layout analysis and can be efficiently computed. The descriptor can be compressed with PCA to a low number of dimensions while still retaining (or even improving) its discriminative qualities. We tested this descriptor on two public datasets and obtained state-of-the-art results in classification tasks.

Furthermore, we have shown how these descriptors can be binarized for large-scale tasks by means of PCAE and compared either with Hamming or asymmetric distances. We have also shown how applying an orthogonal random rotation after PCAE can lead to significantly better results, particularly when using the Hamming distance. These compressed descriptors still provide state-of-the-art results in the NIST dataset when compressed to as few as 16 bits and in the MARG dataset when compressed to 64 bits.

6. Acknowledgments

Albert Gordo and Ernest Valveny are partially supported by the Spanish projects TIN2008-04998, TIN2009-14633-C03-03, and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

References

[1] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[2] F. Perronnin, Y. Liu, J. Sánchez, H. Poirier, Large-scale image retrieval with compressed fisher vectors, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[3] J. Li, Z.-G. Fan, Y. Wu, N. Le, Document image retrieval with local feature sequences, in: *Proceedings of the International Conference on Document Analysis and Recognition*, 2009.

[4] S. Marinai, E. Marino, G. Soda, Layout based document image retrieval by means of xy tree reduction, in: *Proceedings of the International Conference on Document Analysis and Recognition*, 2005.

[5] I. Perea, D. López, Syntactic modeling and recognition of document images, in: *Proceedings of the International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, 2004.

[6] J. van Beusekom, D. Keysers, F. Shafait, T. M. Breuel, Distance measures for layout-based document image retrieval, in: *Proceedings of the*

International Conference on Document Image Analysis for Libraries, 2006.

[7] A. Gordo, E. Valveny, A rotation invariant page layout descriptor for document classification and retrieval, in: *Proceedings of the International Conference on Document Analysis and Recognition*, 2009.

[8] P. Heroux, S. Diana, A. Ribert, E. Trupin, Classification method study for automatic form class identification, in: *Proceedings of the International Conference on Pattern Recognition*, 1998.

[9] J. F. Cullen, J. J. Hull, P. E. Hart, Document image database retrieval and browsing using texture analysis, in: *Proceedings of the International Conference on Document Analysis and Recognition*, 1997.

[10] A. Bagdanov, M. Worring, Multiscale document description using rectangular granulometries, *International Journal on Document Analysis and Recognition* (2003).

[11] C. Shin, D. Doermann, A. Rosenfeld, Classification of document pages using structure-based features, *International Journal on Document Analysis and Recognition* (2001).

[12] P. Sarkar, Image classification: Classifying distributions of visual features, in: *Proceedings of the International Conference on Pattern Recognition*, 2006.

[13] X. Tang, Texture information in run-length matrices, IEEE *Transactions on Image Processing* (1998).

[14] D. Keysers, F. Shafait, T. M. Breuel, Document image zone classification - a simple high-performance approach, in: *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2007.

[15] N. Stamatopoulos, B. Gatos, T. Georgiou, Page frame detection for double page document images, in: *Proceedings of the International Workshop on Document Analysis Systems*, 2010.

[16] A. Gordo, F. Perronnin, Asymmetric distances for binary embeddings, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[17] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[18] F. Perronnin, J. Sánchez, Y. Liu, Large-scale image categorization with explicit data embedding, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[19] P. Indyk, R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in: *Proceedings of the ACM Symposium on Theory of computing*, 1998.

[20] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: *Advances in Neural Information Processing Systems*, 2008.

[21] H. Jegou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE *Transactions on Pattern Analysis and Machine Intelligence* (2010).

[22] H. Jégou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: *Proceedings of the European Conference on Computer Vision*, 2008.

[23] W. Dong, M. Charikar, K. Li, Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces, in: *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2008.

[24] The NIST Structured Forms Database (NIST Special Database 2), http://www.nist.gov/ts/msd/srd/nistsd2.cfm.

[25] The Medical Article Records Groundtruth Dataset, http://marg.nlm.nih.gov/roverintro.asp.

[26] J. van Beusekom, Document Layout Analysis, diploma thesis, Technische Universität Kaiserslautern, 2006.

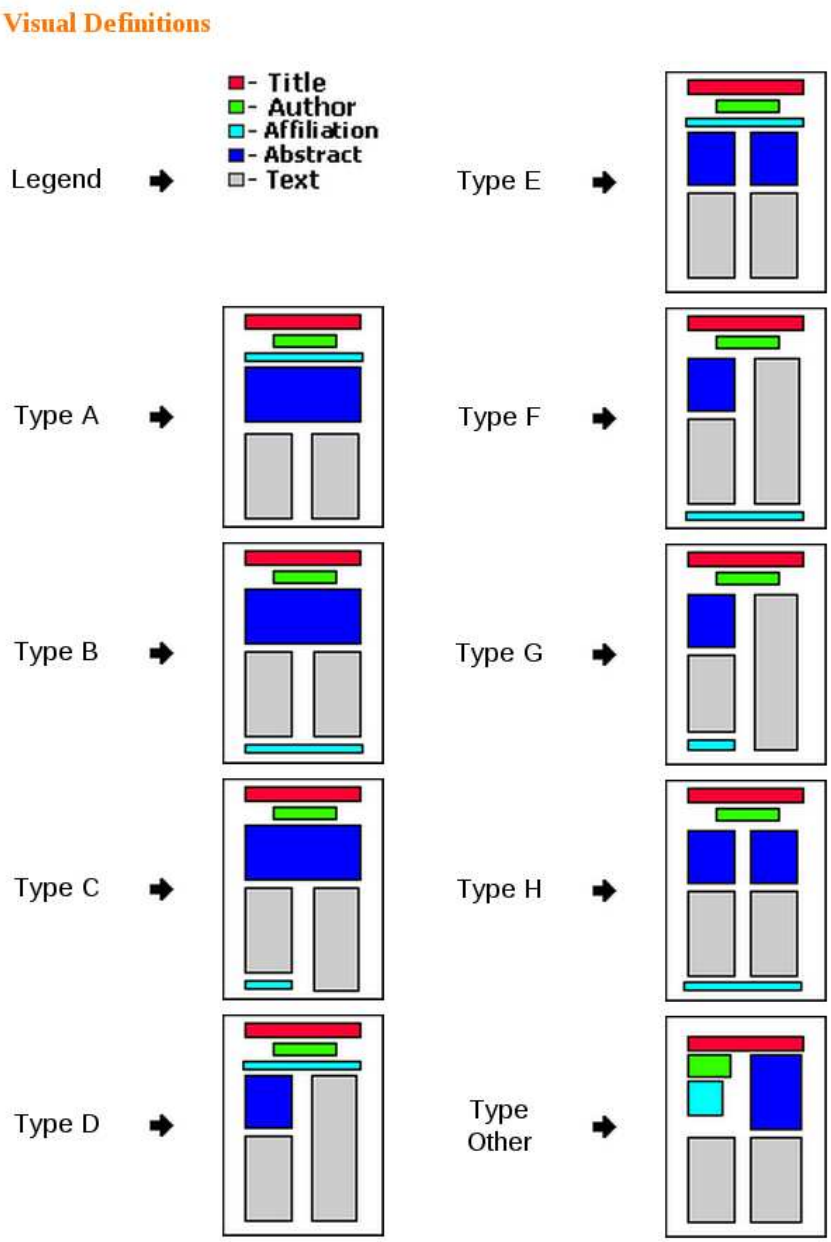

Figure 4: Layout categories in MARG, obtained from http://marg.nlm.nih.gov/gtdefinition.asp. The number of columns is *not* relevant to decide the category of a document.

Albert Gordo received his BSc degree in Computer Engineering and his MSc degrees in Intelligent Systems from the University Jaume I in Castellón, Spain, in 2007 and 2009, respectively. He also holds a degree in Computational Math obtained in 2007. He is currently pursuing his PhD in the Computer Vision Center of Barcelona, Spain in collaboration with the Xerox Research Centre Europe in Grenoble, France, under the supervision of Dr. Ernest Valveny and Dr. Florent Perronnin.
His main research interests include document image analysis, document retrieval and large-scale problems.

Florent Perronnin received his Engineering degree in 2000 from the Ecole Nationale Supérieure des Télécommunications (Paris, France) and his Ph.D. degree in 2004 from the Ecole Polytechnique Fédérale de Lausanne (Lausanne, Switzerland). From 2000 to 2001 he was a Research Engineer with the Panasonic Speech Technology Laboratory (Santa Barbara, California) working on speech and speaker recognition. In 2005, he joined the Xerox Research Centre Europe (Grenoble, France). His main interests are in the practical application of machine learning to computer vision tasks such as image classification, retrieval or segmentation.

Ernest Valveny is an Associate Professor at the Computer Science Department of the Universitat Autònoma de Barcelona (UAB), where he obtained his PhD degree the in 1999. He is also member of the Computer Vision Center (CVC) at UAB. His research work has mainly focused on symbol recognition in graphic documents. Other areas of interest are in the field of computer vision and pattern recognition, more specifically in the domain of document analysis, including shape representation, character recognition, document indexing and layout analysis. He is currently a member of the IAPR TC-10, the Technical Comitee on Graphics Recognition, and IAPR-TC-5 on Benchmarking and Software. He has been co-chair of the three editions of the International Contest on Symbol Recognition, supported by IAPR-TC10. He has worked in several industrial projects developed in the CVC and published several papers in national and international conferences and journals.