# Visual word spatial arrangement for image retrieval and classification

Otávio A. B. Penatti[a,*], Fernanda B. Silva[a], Eduardo Valle[a,b], Valerie Gouet-Brunet[c], Ricardo da S. Torres[a]

[a]*RECOD Lab, Institute of Computing (IC), University of Campinas (Unicamp) – Av. Albert Einstein, 1251, Campinas, SP, 13083-852, Brazil*

[b]*Department of Computer Engineering and Industrial Automation (DCA), School of Electrical and Computer Engineering (FEEC), University of Campinas (Unicamp) – Av. Albert Einstein, 400, Campinas, SP, 13083-852, Brazil*

[c]*Paris-Est University, IGN/SR, MATIS Lab, 73 avenue de Paris, 94160 Saint-Mandé, France*

## Abstract

We present Word Spatial Arrangement (WSA), an approach to represent the spatial arrangement of visual words under the bag-of-visual-words model. It lies in a simple idea which encodes the relative position of visual words by splitting the image space into quadrants using each detected point as origin. WSA generates compact feature vector, making it useful for both image retrieval and classification. Experiments in the retrieval scenario show the superiority of WSA in relation to Spatial Pyramids. Experiments in the classification scenario show a reasonable compromise between those methods, with Spatial Pyramids generating larger feature vectors, while WSA provides adequate performance with much more compact features.

*Keywords:* visual words, spatial arrangement, image retrieval, image classification

## 1. Introduction

Content-based image retrieval is a key technique for improving image search engines. The most effective approaches currently used are based on a vocabulary of local patches, called visual dictionaries [1]. That model is inspired in text retrieval, where a simple but effective

[*] Corresponding author at +55 19 3521-5887 (phone)/+55 19 3521-5847 (fax), RECOD Lab, Institute of Computing (IC), University of Campinas (Unicamp) – Av. Albert Einstein, 1251, Campinas, SP, 13083-852, Brazil

*Email addresses:* `penatti@ic.unicamp.br` (Otávio A. B. Penatti), `fernanda@recod.ic.unicamp.br` (Fernanda B. Silva), `mail@eduardovalle.com` (Eduardo Valle), `valerie.gouet@ign.fr` (Valerie Gouet-Brunet), `rtorres@ic.unicamp.br` (Ricardo da S. Torres)

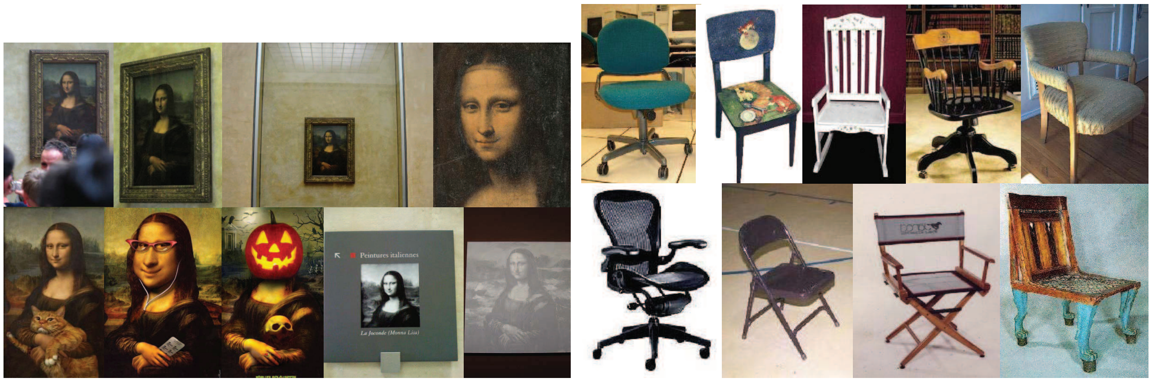
model takes documents simply as “bags” (multi sets) of words. In the same spirit, visual dictionary representations take images as bags of local appearances. That model has several important advantages, such as compactness (it encodes local properties into a single feature vector) and invariance to image/scene transformations.

Creating a visual dictionary takes several steps. First and foremost, local characteristics must be obtained from a set of training images, usually by extracting local patches and describing them. The patches may be taken around Points of Interest (PoI) [2] or by dense sampling [3], and image descriptors, like the popular SIFT [4], are used to extract feature vectors for each of them. Once the learning set of feature vectors is obtained, they are used to quantize the feature space (using, for example, k-means clustering) to choose a codebook of feature vectors representative of the training set. The clusters tend to contain visually similar patches and each cluster is a visual word of the dictionary. Once the dictionary is available, images are represented by statistical information about how they activate the visual words. The final image feature vector is commonly called bag of (visual) words (BoW).

When creating an image representation, one must be aware of its target application. Applications like copy detection or partial-duplicate image search, as shown in Figure 1(a)[1], require the creation of really discriminating representations. Very small differences between images or objects must be encoded, while still being robust to specific photometric/geometrical transformations related to the domain. Therefore, the representation must be very precise. The semantic-search application, as shown in Figure 1(b)[2], requires precise representations but, at the same time, general enough to comprise the intra-class variations. One may be interested in finding different types of the same object, like, for example, retrieving different types of chairs, instead of finding exactly the same chair [6]. Considering the topical problem of big data, more generic representations should be more interesting. Representations that are less specific to a certain application may be more suitable for addressing the big-data problem, where the big

[1] CreativeCommons images downloaded from Flickr (as of July 9th, 2012).

[2] Chairs from Caltech-101 dataset [5].



(a) (b)

Figure 1: Application examples: (a) retrieval of partial duplicates, where (parts of) the same object or scene are shared between the query and target images, possibly with transformations and noise; (b) semantic search, where query and target images share concepts (e.g., different instances coming from the same class of objects), but not necessarily objects or scenes.

volume of data makes it more complicated to extract several categories of features dedicated to specific scenarios (retrieval, classification, etc.), in terms of extraction time and of storage.

The research community has been very active in the area in the latest years and many new proposals over the visual dictionary model constantly appear. Special attention has been given to the lack of geometrical information encoded by the traditional bag-of-words representation [7, 8, 9, 10, 11, 12, 13]. The spatial arrangement of visual words in images is important to understand image semantics and is often crucial to distinguish different classes of scenes or objects. In that direction, approaches are proposed for image classification [10, 9] and retrieval [8, 12, 13, 11].

In the classification scenario, usually relied on Support Vector Machines (SVMs), the high dimensionality of vectors does not degrade effectiveness, because SVMs suffer less from the curse of the dimensionality. The popular Spatial Pyramids [9] are very successful for image classification and their vectors have high dimensionality. However, for retrieval experiments, which are generally based on computing distances between vectors, with the Euclidean distance, for example, vectors should be compact, or embedded in an index structure, to avoid the curse of the dimensionality [14, 15, 16, 8]. As dimensionality grows, the distribution of distances between features tends to become narrowly concentrated around an average value, reducing the contrast between similar and dissimilar features. Therefore, to create an image representation that works well in both classification and retrieval scenarios, one must be aware of the feature vector size. Many of the existing approaches for spatial pooling which are employed in the retrieval scenario leave the spatial verification as a post-processing step [12, 13]. They compute a simple representation and then, after finding the matching visual words between images, they compute the spatial representation and perform a spatial consistency verification, before reranking the images. Furthermore, some of the existing approaches used in the retrieval scenario are very specific and suitable for partial-duplicate image search [12, 13], thus their use for the semantic-search application is challenging.

In this paper, we present *Word Spatial Arrangement* (WSA), a spatial pooling approach for both image retrieval and classification. Our approach adds spatial information into the feature vector having the advantages of generating more compact vectors than the popular approaches for spatial pooling. It is also more precise than the traditional bag of words but keeps the generality desired for the semantic-search application, a good property also for the big-data problem. Our approach aims at addressing both retrieval and classification scenarios. In the retrieval environment, WSA encodes the spatial information of visual words into a single feature vector prior to any filtering step with matching visual words. Most of the approaches that encode spatial information of visual words in the retrieval scenario [8, 12] works solely with the assignment of a unique visual word to a point (hard assignment). WSA, however, also works with soft assignment, taking advantage of the good performance of soft assignment in classification experiments [17, 18, 19]. Soft assignment relies on considering a neighborhood around the point in analysis when assigning labels of visual words to it. In high-dimensional spaces, points tend to be in the frontier of many regions, therefore, assigning more than one visual word to them can be more robust.

WSA is simple and easy to understand. On top of that, WSA has the advantage of depending on almost no parameters, oppositely to many of the existing spatial pooling methods. The visual word spatial arrangement encoded by WSA is based on a sliding quadrant partitioning in the image space considering each point in the image as the origin of the quadrants and counting the visual word occurrences in each quadrant [7]. In this paper, we present an improvement to the original WSA method which we presented in [7]. Here, we are not concatenating the bag to the WSA information, therefore, in this paper, WSA refers to the spatial information only. Additionally, we have included several improvements over the original WSA propose. We explain how to use WSA with soft assignments introducing the threshold $t$ for very soft assignments, we evaluate the use of a weighted window, and we propose a distance function for image retrieval. Considering the experimental protocol, in this paper, we evaluate WSA for both image retrieval and classification using different datasets. We also provide an online interface based on Eva tool [20] to show the experiment results in the retrieval scenario[3] [4].

The rest of the paper is organized as follows. Section 2 presents other approaches for encoding spatial arrangement of visual words. Section 3 details the proposed method. Sections 4 and 5 show the experimental comparison in both retrieval and classification scenarios, respectively. Section 6 concludes the paper.

## 2. Related work

The related work section details the traditional representation schemes based on visual dictionaries in Section 2.1 and presents recent advances on encoding spatial information of visual words in Section 2.2.

### *2.1. Visual dictionaries*

The most popular and effective approach to represent visual content nowadays is based on visual dictionaries, which generate the so-called bag-of-words representation [1]. One of the benefits of using such a representation is its ability to encode local properties into a single feature vector per image. To generate a bag-of-words representation, one must first create the visual dictionary. Image local features, usually obtained by SIFT descriptor [4] computed on the detected interest points [2] or in a dense grid [3], are clustered in the feature space. Each cluster represents a visual word and tends to contain patches with similar appearance. Although k-means is still a popular method employed in the clustering step, due to the curse

[3] `http://www.recod.ic.unicamp.br/eva/view_images_base600.php` (as of June 19th, 2013).

[4] `http://www.recod.ic.unicamp.br/eva/view_images_paris.php` (as of June 19th, 2013).

of dimensionality, a simple random selection of points in the feature space creates dictionaries of similar quality [21, 22] and saves much time in dictionary generation.

To compute the image representation, image local patches are assigned to one or many visual words in the dictionary. The *hard* assignment assigns a local patch to the nearest visual word in the feature space. On the other hand, *soft* assignment reduces the effect of poor clustering results by assigning multiple visual words to a local patch [17, 19, 18]. One effective soft assignment computation method, presented in [17], is formally given by Equation 1:

$$\alpha_{i,j} = \frac{K_\sigma(D(v_i, w_j))}{\sum_{l=1}^{k} K_\sigma(D(v_i, w_l))}, \tag{1}$$

where $j$ varies from 1 to the dictionary size ($k$), $v_i$ is the feature vector of patch $i$, $w_i$ is the vector corresponding to visual word $j$, $K_\sigma(x) = \frac{1}{\sqrt{2\pi}\times\sigma} \times exp(-\frac{1}{2}\frac{x^2}{\sigma^2})$, and $D(a, b)$ is the distance between vectors $a$ and $b$. The $\sigma$ parameter indicates the smoothness of the Gaussian function: the higher the value, the larger the number of neighboring regions considered.

After representing each image local patch according to the dictionary, the set of image patches are summarized as a single feature vector by pooling techniques [23]. The most popular pooling approaches are based on computing the *average* assignment value for each visual word in the image and on considering only the *maximum* (max) visual word activation. Results in literature show a better performance in classification experiments when using max pooling [23], which can be formally given by Equation 2 [23]:

$$h_j = \max_{i \in N} \alpha_{i,j} \tag{2}$$

where $\alpha$ is obtained in the assignment step (by Equation 1, for example), $N$ is the number of points in the image, and $j$ varies from 1 to the dictionary size ($k$).

### *2.2. Spatial information of visual words*

There are several recent works in the area which present interesting advances for creating better dictionaries [22, 24, 25] and better coding [17, 26] techniques. However, many of the re-

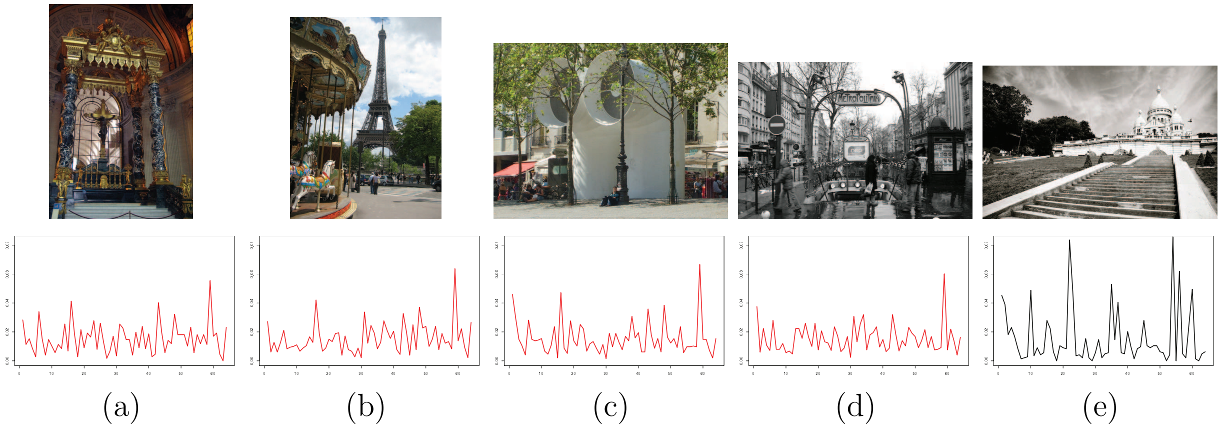

(a) (b) (c) (d) (e)

Figure 2: Examples of images (a–d) with different semantics but similar bags of visual words (BoW). The graph below each image shows its BoW, created using a dictionary of 64 words, hard assignment and average pooling. The horizontal axis is the label of the word (1–64) and the vertical axis is the frequency of occurrence of each word. Due to the loss of spatial information, unrelated images (a–d) may end-up sharing very similar BoWs. For sake of comparison, we show also an image with a dissimilar BoW (e).

cent advances built over the visual dictionary model relies on encoding geometrical information of visual words [7, 8, 9, 10, 11, 12, 13, 27, 28].

In the early days of the content-based image retrieval (CBIR) area [29], researchers faced the problem of having many different images with identical or very similar color histograms, motivating the creation of new methods for encoding the spatial arrangement of colors, like, for example by using color correlograms [30] or color-coherence vectors [31]. This issue is being revisited nowadays with the visual dictionary model. However, the element under analysis moved from single pixel values to local patches.

Spatial information of visual words, usually lost by the traditional pooling techniques like *average* and *max* pooling [23], may be very important for discriminating image content and for encoding image semantics. Consider the images shown in Figure 2. They have different semantics but their bag-of-words representations are very similar.

The development of methods to encode spatial information of visual words may take into account several aspects depending on the target application. Considering the semantic-search application, where we would like to be able to find different types of the same object or image, as shows the example in Figure 1(b), the representation needs to be specific enough to distinguish one class of objects from the others, but not too specific, otherwise only the same object instance will be considered similar. Therefore, capturing spatial information for semantic-search must be planed carefully for not loosing generality, which is one of the main strengths of the bag-of-words representation. On the other hand, in the partial-duplicate search application, where the changes among images exist but images still share some duplicate patches [12], the representation must be very precise. Several approaches include the geometrical verification as a post-processing step, keeping the representation simple and applying the geometrical constraints on a subset of matched visual words [12, 13].

Another important issue when developing a new method to encode the spatial information of visual words is related to the compactness of the representation. In the classification scenario, which is popularly based on SVMs, the curse of the dimensionality does not impact considerably the effectiveness of the methods, because SVM usually deals well with very large feature vectors [9, 26]. However, considering the retrieval scenario, the feature vector size considerably impacts the effectiveness of search approaches. The curse of the dimensionality is closely related to the action of computing distances between vectors, a frequent operation in retrieval systems. Therefore, some representations which work well for image classification may not work for image retrieval. Our approach aims at encoding the spatial arrangement of visual words being compact to be useful for both classification and retrieval scenarios.

The most popular approach to encode the spatial information of visual words is the *Spatial Pyramid* [9]. A spatial pyramid hierarchically splits the image into fixed-size tiles and generates one bag of words for each tile. For a pyramid level of 2, for example, 21 bags are generated. The first bag comes from the image without splitting. In the next level, the image is split into 4 tiles of the same size. The next level splits each of the 4 tiles into another set of 4 tiles. Therefore, there is 1 bag for level 0, 4 bags for level 1 and 16 bags for level 2. All the bags are weighted depending on their level [9] and then concatenated to create the image feature vector. The pyramids main advantage is their simplicity. Other advantage is that the hierarchical splitting tends to create a multi-scale image representation. Their main drawbacks are related to the large feature vector size, to the fact that no information regarding the image scale is taken into account, and that no spatial relationship among visual words is encoded.

Other recent approach for visual word spatial pooling, called spatial-bag-of-features [11], is based on creating linear and circular projections of the image. The linear projections consider the horizontal axis as reference. The image is split into $L$ vertical tiles and a bag of words is generated for each tile. The axis is then rotated by an angle of $\theta$ and each of the $L$ tiles generates another set of bags. This is performed by a predefined number of angles. The circular projections consider a set of points to be the center of the image splitting and then splits the image into $L$ sectors. A bag of words is computed for each sector. The final feature vector is a concatenation of all bags generated by linear and circular projections. The method also conducts reordering of bags in the feature vector to achieve rotation, translation, and scale invariance. Its main advantage lies in capturing more spatial configurations than the Spatial Pyramids, as these last ones could be considered particular cases of linear projections. Its main disadvantage is the feature vector size. Moreover, no spatial relationship information among visual words is explicitly encoded.

Another recent approach encodes the spatial relationship of visual words by using triangular relations among neighboring words [8]. All the triangular relationships between 3 points in the image are computed and, for each relationship, a set of signatures is created. There are signatures which depend on point labels, signatures considering the angles among points, and signatures considering point scales. Each relationship is indexed independently and is composed of a maximum of 7 signatures (7-D vector). The signatures maintain invariance to translation, rotation, scale, and flipping. To avoid a large number of triangular relationships, pruning strategies are employed. This method, called $\Delta$-TSR, explicitly encodes the spatial relationship among visual words, however, the description and its similarity measure were not designed for kernels, making it challenging to use in classification scenarios. On top of that, it was originally proposed to work with hard assignment only.

A recent spatial coding technique for partial-duplicate image search encodes the spatial relationship among every pair of points in the image by using binary spatial maps [12, 32]. The spatial verification is a post-processing step in the retrieval framework, applied only for matching visual words between query and database images. A horizontal spatial map is an $N \times N$ binary matrix, where each row $i$ says if the feature $i$ is at right (1) or at left (0) of each other feature. The vertical spatial map is analogous, having value 1, in row $i$, for points which $i$ is above and value 0, otherwise. The effect of the spatial maps calculations consists in splitting the image into 4 quadrants, using each point in the image as the origin. The method also considers rotation and scale issues, by rotating the image according to the origin point SIFT orientation [32] and by considering the distance between points (square maps). This method explicitly encodes the spatial relationship among visual words, but its representation is very specific making it not suitable for the semantic-search application. The spatial maps are computed only for matching words, therefore, changes in the representation are necessary to allow its use in classification scenarios. Our approach uses a similar idea of the image space splitting, however, our representation embeds the spatial information into the feature vector and works both for classification and retrieval scenarios. Furthermore, the applications considered are different and we intended to keep our representation more generic, that is, less specific to the application, with the aim of addressing big-data contents, where the volume makes more complicated the extraction of several kinds of features dedicated to specific scenarios.

Another recently proposed approach works specifically for image classification [10]. It is a geometric $l_p$-norm pooling method that learns the positions of visual words occurrences in an image dataset. For that, the method first puts all the images into the same resolution, discarding their aspect ratio, and uses a regular (dense) grid for image sampling. Therefore, all the images will have the same number $M$ of points. At the end, each visual word $k$ has a vector of dimension $M$, where each vector position $m$ corresponds to the activation of the visual word $k$ in the $m^{th}$ position of the dense grid. This approach can effectively learn the positions of visual words in the images, however it greatly depends on putting all the images into the same resolution and using the dense sampling. Additionally, the encoded properties represent the absolute visual word position in the image and objects translation inside the images will change considerably the final representation. Our proposed approach has some relation to the geometric $l_p$-norm pooling just presented [10]. The geometric $l_p$-norm pooling method encodes the *absolute* position of visual words in the images. Our method, on the other hand, by counting visual word positions in relation to all the other points in the image, discarding their visual word assignments, encodes the *relative* position of each visual word in the image. Our method is based on image sparse sampling (by interest point detectors) and geometric

$l_p$-norm pooling uses dense sampling. If the majority of points detected in the image are in the object of interest, our approach does not suffer from the translation problem mentioned for the geometric $l_p$-norm pooling. Other advantage of our method is that it also works in the retrieval scenario.

More recently, graph-based approaches for spatial pooling are emerging [28]. Graph methods are naturally suitable for encoding the spatial arrangement of visual words. The multi-layer local graph words (MLLGW) [28] creates several layers of graphs exploring different levels of adjacency of the points detected in the image. The layers are "nested" in the sense that each new layer is obtained by adding nodes to the local graphs of the previous layer. Based on the responses of the interest-point detector (in [28], SURF is used), the seeds for the first level of the multi-layer approach are selected. To build the next layer, three nodes must be added. At the end, four layers are obtained: the first having the seeds (raw SURF points), and the next layers having 3, 6, and 9 neighbors. For each layer, a visual dictionary is created. Each layer is then encoded (by hard assignment) according to its dictionary, and the final image representation is the concatenation of the bags of each layer. The MLLGW method is originally validated for image retrieval using the L1 distance function. The advantage of this method is that it explicitly encodes the spatial relationship of image points in a multi-layer approach. However, as a dictionary must be generated for each layer, it has higher computational cost, and no details regarding soft assignment are provided.

There are many other proposals for encoding the spatial information of visual words, like, for example, by using the co-occurrence of pairs visual words [33], by using correlograms [34], or by appending the point coordinates to their feature vectors before creating the dictionary [27]. Many of those methods face the problem of generating high-dimensional feature vectors, since including all the possible spatial configurations into the feature vector and keeping compactness is challenging. This is one reason that leads some approaches to leave the spatial verification as a post-processing step [12, 32, 13].

The next section details the proposed WSA representation.

## 3. Word Spatial Arrangement (WSA)

This section presents our approach to encode the spatial arrangement of visual words, which is called Word Spatial Arrangement (WSA). The main goal when designing WSA was to include the spatial information of visual words, aiming at increasing the precision of the traditional bag-of-words representation but keeping the generality desired for the semantic-search application. WSA was also designed to be able to work in both retrieval and classification scenarios.

As mentioned previously in Section 2, WSA presents some similarities with other methods from literature. Other important aspects of WSA are the following:

- the spatial information of visual words is embedded into the feature vector, therefore, in the retrieval scenario, no post-processing is required;
- WSA encodes the relative position of visual words in the image space;
- WSA representation is more compact than many of the spatial pooling approaches in the literature;
- WSA works with soft assignment as well as with hard assignment;
- WSA works with sparse sampling (interest-point detectors);

WSA is based on the idea of dividing the image space into quadrants [7] using each point as the origin of the quadrants and counting the number of words that appear in each quadrant. We count how many times a visual word $w_i$ appears in each quadrant in relation to all other points in an specific image. This counting will tell us the *spatial arrangement* of the visual word $w_i$. Intuitively, the counting will measure the positioning of a word in relation to the other points in the image. It reveals, for example, that a word $w_i$ tends to be below, at right, or surrounded by other points. By counting $w_i$ position in relation to the other points in the images, without considering the labels of other points (visual words assigned to them), we generate a not-too-precise representation, which is interesting for the semantic-search application.

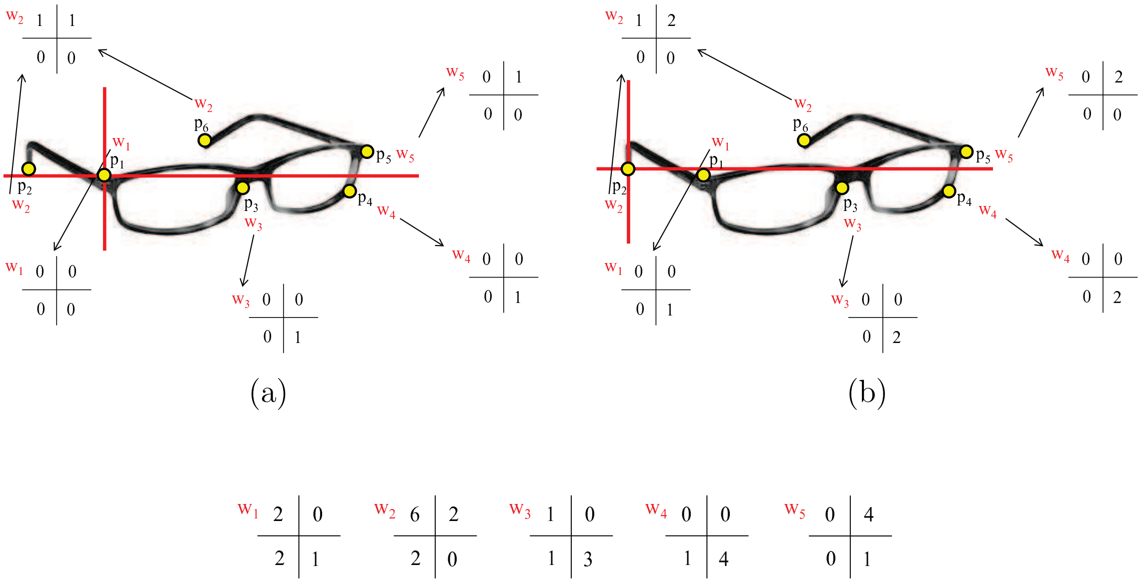
Figure 3 shows an example of partitioning the image space and counting. To generate the WSA vector, the image space is divided as follows: for each point $p_i$ detected in the image, we divide the space into 4 quadrants, putting the point $p_i$ in the quadrant's origin; then, for every other detected point $p_j$, we increment the counting of the visual word associated with $p_j$ in the position that corresponds to the position of $p_j$ in relation to $p_i$. For example, if $w_j$ is the visual word associated with $p_j$ and $p_j$ is at top-left from $p_i$, the counter for top-left position of $w_j$ is incremented. After all points are analyzed in relation to $p_i$, the quadrant's origin goes to the next point $p_{i+1}$, and the counting in relation to $p_{i+1}$ begins. When all points have already been the quadrant's origin, the counting finishes.

Each visual word is associated with 4 numbers, which tell the spatial arrangement of the visual word in the image. The same visual word can appear in several different locations in an image, however, there is only one set of 4 counters for each visual word. The complexity of this method to generate the feature vector is $O(n^2)$, while the traditional bag is $O(n)$, where $n$ is the number of points in the image.

When the counting is finished, each 4-tuple is normalized by its sum. Thus, each of the 4 counters of a given visual word will represent the percentage of times that the visual word appeared in the corresponding position in relation to the other points in the image. For instance, if the word $w_i$ has most of the counting values in its bottom-right counter, we can say that $w_i$ is a bottom-right word, as the word $w_4$ in Figure 3(c). If $w_i$ has top-left and top-right counters with high values, we can say that $w_i$ is a word that usually appears above other points. If all counters of $w_i$ are equally distributed, $w_i$ is surrounded by other points (middle-word) or it is a word that repeatedly surrounds other points (border-word).

WSA implementation is flexible to use either *hard* or *soft* assignment. In some methods of the literature, which are employed in the retrieval scenario using inverted files, only *hard* assignment is possible [8, 12]. In WSA, when using hard assignment, the increment in the visual word counters is always by 1. On the other hand, when using soft assignment, the increment is proportional to the activation of the point to every visual word. For example, considering that $p_i$ activated $w_1$ in 0.8 and $w_2$ in 0.2, we increment the corresponding counters of $w_1$ by 0.8 and the corresponding counters of $w_2$ by 0.2.

The final WSA feature vector is the concatenation of all 4 counters of each visual word, resulting in a feature vector of dimension 4×k, where $k$ is the dictionary size.



(a)

(b)

(c)

Figure 3: Example of partitioning and counting. The small circles are the detected points, tagged with their associated visual words ($w_i$'s). We start in (a), putting the quadrant's origin at $p_1$ and counting in the visual word associated with each other point, where the point is in relation to $p_1$. On the second step (b) the quadrant is at $p_2$; we add again the counters of the words associated with each other point in the position corresponding to their position in relation to $p_2$. We proceeded until the quadrant has visited every point in the image. Final counter values, before normalization, are shown in (c).

The initial results of WSA and soft assignment have shown that when the number of visual words activated by a point is large (very soft assignment), WSA performance is low. When the assignment is very soft, each point activates a large number of visual words in the dictionary and the activation levels can be very low for some visual words. Therefore, considering that we are encoding the spatial arrangement of visual words in the image space, taking into account a visual word that was few activated, might disturb the representation. A very low activation could mean that the visual word is almost not presented in a certain position. Thus, to avoid incrementing the WSA counters when the activation of a visual word is very low, we have used a threshold $t$ (activations smaller than $t$ are discarded).

We can point that another advantage of WSA is its low dependency of adjusting parameters, as the only parameter of WSA is $t$ and this parameter is necessary only when very soft assignments are used.

### *3.1. WSA-window-weighted*

As the WSA counting process considers all the points in the image, points that are far from the origin point and possibly belong to background or to other objects, will also be considered. Therefore, it would be better to consider in the counting process only points from the object where the origin point is located. We have implemented the use of windows around each origin point, aiming at capturing those points. The window size is determined by the scale of the origin point. Consequently, the approach keeps scale invariance.

In addition, the window has a Gaussian behavior over the counting process. Points near the origin have more weight in the counting than points far from it. The equation to compute the weight $w$ when $p_i$ is the origin is:

$$w = \frac{1}{\sqrt{2\pi} \times \sigma} \times exp(-\frac{1}{2} \times \frac{d^2}{\sigma^2}) \tag{3}$$

where $d = D_{L2}(p_i, p_j)$ is the Euclidean distance between $p_i$ and $p_j$ and $\sigma$ is the scale of $p_i$.

In the experiments, we call *WSA-ww* the version that use the Gaussian behavior of the window.

### *3.2. Distance function*

In the retrieval scenario a distance function is required to compare feature vectors. Therefore, we present here the distance function to be used with WSA.

The idea behind this function is somehow to assess if images contain the same visual words with the same spatial arrangement. Therefore, distances among points are computed only between corresponding visual words that present similar spatial arrangement. The effect is the same as first finding the matching visual words and then applying the spatial verification. However, as pointed in the beginning of Section 3, WSA does not require a post-processing step for spatial verification in retrieval scenarios. The reason is that, as the spatial information is already embedded into the feature vector, the spatial verification can be performed while going through the feature vector. By “post-processing”, we understand that, after finding the matching visual words, one is able to compute the spatial information and then perform the spatial verification, as it occurs with the methods presented in [12, 32], but not with WSA.

The retrieval schema is based on the following distance function:

$$D_{Q,I} = \frac{\sum_{j=1}^{N_{WC}} dist_j(WSA_j^{(Q)}, WSA_j^{(I)})}{(distMax \times N_{WC})} \tag{4}$$

where

$N_{WC}$ is the number of visual words in common between the query image $Q$ and the database image $I$,

$dist_j$ is a distance function for the WSAs of common words,

$WSA_j$ is the WSA (4-values set) of word $j$,

$distMax$ is the maximum distance for one pair of WSAs.

The number of words in common $N_{WC}$ depends on the images. The distance function $dist_j$ for each pair of WSAs can be any, like the popular Euclidean (L2) or Manhattan (L1) distances. The maximum distance $distMax$ between a pair of WSAs depends on the distance function used. For the Euclidean distance, for example, it is $\sqrt{2}$, while for Manhattan distance, it is 2.

To consider a pair of corresponding visual words as a match (words in common), the distance between their WSAs need to be lower than or equal to $\epsilon$. Otherwise, it is likely that the respective visual word is not present in both images. In the experiments, tests have been made with L1 and L2 distances, using $\epsilon$ equal to $\frac{1}{4}$, $\frac{1}{3}$, and $\frac{1}{2}$ of the maximum WSA distance ($distMax$).

## 4. Experiments for image retrieval

To evaluate the proposed approach considering the retrieval scenario, we have used two datasets. One dataset is composed of 600 synthetic images and the other collection is the popular Paris buildings dataset[5]. Both datasets can be classified in the partial-duplicate search

[5] `http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/` (as of June 19th, 2013).

| **Pooling method** | **Acronym** | **Feature vector size** |
|---|---:|---:|
| Average | avg | k |
| Max | max | k |
| Max pooling with Spatial Pyramids | max-SPM | 21k |
| Multi-layer local graph words | MLLGW | 4k |
| Word Spatial Arrangement | WSA | 4k |
| WSA using weighted windows | WSA-ww | 4k |

Table 1: Acronyms and feature vector sizes for the pooling methods being evaluated. $k$ is the dictionary size.

application because, for each category, the same object appears in different rotation and viewpoints.

The main questions to be answered by these experiments are:

- is the accuracy of WSA comparable to the best methods from literature?
- what is the impact of the soft assignment in WSA?

In our experimental setup, the images were represented by different methods based on the bag-of-words approach. First, the Harris-Laplace detector [2] and the SIFT descriptor [4] were used to extract local feature vectors from images[6]. Dictionaries of 15 000 and 8 000 visual words were constructed by randomly selecting points in the feature space [21] (see Section 2.1) and they were used in Base-600 and Paris datasets, respectively. For those datasets, we have used the dictionary sizes recommended by [8]. We have varied the assignment method, using hard and soft assignment (according to Equation 1 in Section 2.1), the last with $\sigma$ varying in 30, 60, 90, and 150. The $\sigma$ values were based on the discussion found in [17], where, considering our experimental scenario, $\sigma$ values around 100 are good. So, we varied from harder to softer assignments than that. The following pooling methods were compared: average pooling, max pooling, max pooling with Spatial Pyramids, and multi-layer local graph words. For WSA, we have used the standard version (WSA) and the version that use the window around the origin point during the counting process: WSA-ww.

Table 1 summarizes the pooling methods compared and the size of their feature vectors. We have not used WSA with Spatial Pyramids because Spatial Pyramids enlarge the feature

[6] We have used the software of van de Sande et al. [3].

vectors and large vectors suffer from the curse of the dimensionality. This is noticed when using the max pooling with Spatial Pyramids in the following experiments. It is important to highlight that avg and max pooling generate more compact vectors than WSA, however, they are non-spatial pooling methods. Hence, when we argue that WSA is more compact, we refer to this property in relation to the spatial pooling methods only.

Max pooling with Spatial Pyramids (max-SPM) is our main baseline because it is a spatial pooling approach which can be directly used in retrieval and classification scenarios and works with hard and soft assignment, as well as WSA. Another approach used as baseline is MLLGW, which is a spatial pooling method based on graphs. Although MLLGW presents some limitations, like being originally proposed to work in hard assignment [28], and has higher computational cost, given that each layer has an independent dictionary, we have used it as baseline due the emerging importance of graph-based methods for spatial pooling.

Our implementation of the MLLGW baseline considers the use of SIFT features and the seeds for the first layer were randomly selected. Dictionaries of 15 000 and 8 000 words (Base-600 and Paris datasets, respectively) for each level were randomly created, like the other dictionaries used here.

Considering the other spatial pooling approaches presented in Section 2.2, the ones that are suitable for retrieval scenarios depend on performing the spatial verification as a post-processing step [12, 32] or they generate variable number of feature vectors per image [8]. Therefore, although they can work well in specific scenarios and conditions, they are not comparable with WSA in terms of flexibility.

The retrieval scenario requires distance computations between image representations. We have computed distances from the query images to all the other images in the datasets and ranked them according to the distances. For avg, max and max-SPM, the Euclidean distance (L2) was used to compare the vectors, while for MLLGW, L1 was used. For WSA, we have used both L2 and the distance function presented in Section 3.2, comparing their performances. As the proposed distance function has some parameters, we have tested the variations of them (see Equation 4) and the results presented in this section consider one of the best configurations: $\epsilon = \frac{1}{2} \times distMax$ and $dist_j = L1$.

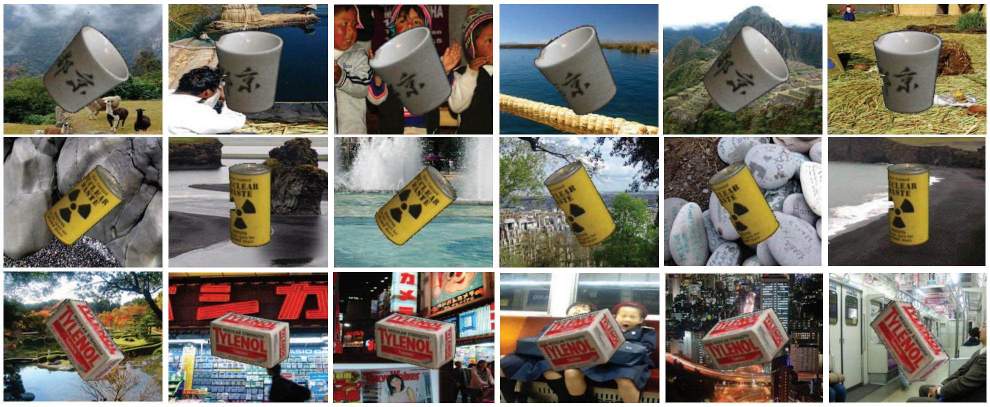


Figure 4: Sample images from the Base-600 dataset, highlighting 3 categories (one per row). There are 20 categories, each containing a particular object in different poses and orientations, and a random background.

The results are presented in terms of mean average precision (MAP) and precision for the top N retrieved images (P@N). It is important to highlight that, although MAP is a very popular measure to assess the effectiveness of CBIR methods, it does not reflect the ranking quality in the first positions. It only says how good a method is to retrieve all the relevant images. Considering an environment where the user analyze the retrieved images visually, like the Web, it is crucial to have a good set of 10 or 20 retrieved images even if the MAP value is not good. Therefore, in that case, we are more interested in good P@N values than good MAP values. Results are reported with confidence intervals for $\alpha$=0.05 and are based on the number of query images used.

*Base-600.* The first dataset used is composed of 600 synthetic images where there is a main object in the center over a heterogeneous background. This dataset, here called Base-600, simulates the partial-duplicate application and has 20 categories, each one containing 30 images. Each category refers to an object taken from Coil-100 dataset[7] and each view of it was inserted in a different background, while keeping it in the center of the images. Images have dimension of $352 \times 288$ pixels. The goal when using this dataset is to verify if the image representation is robust enough to encode the object properties without mixing background information. Good

[7] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php (as of June 19th, 2013).

(a) WSA: L2 distance × proposed distance function

| | L2 distance | | | Proposed distance | | |
|---|---|---|---|---|---|---|
| **Pooling** | **MAP (%)** | **P@10 (%)** | **Assignment** | **MAP (%)** | **P@10 (%)** | **Assignment** |
| WSA | 23.62 ± 0.64 | 50.13 ± 1.88 | Soft ($\sigma$=150) | 40.83 ± 1.23 | 92.12 ± 1.36 | Soft ($\sigma$=150) |
| WSA-ww | 27.73 ± 0.70 | 61.02 ± 2.00 | Soft ($\sigma$=150) | **41.57 ± 1.20** | **95.03 ± 1.20** | Soft ($\sigma$=150) |

(b) Baselines

| **Pooling** | **MAP (%)** | **P@10 (%)** | **Assignment** | - | - | - |
|---|---|---|---|---|---|---|
| Avg | 20.89 ± 0.78 | 44.32 ± 2.15 | Soft ($\sigma$=90) | - | - | - |
| Max | **33.41 ± 0.67** | **74.73 ± 1.99** | Soft ($\sigma$=150) | - | - | - |
| Max-SPM | 20.55 ± 0.53 | 39.47 ± 1.68 | Soft ($\sigma$=90) | - | - | - |

(c) Comparing methods in *hard* assignment

| **Pooling** | **MAP (%)** | **P@10 (%)** | **Distance function** |
|---|---|---|---|
| WSA | 21.10 ± 0.97 | 34.70 ± 2.00 | Proposed distance |
| WSA-ww | **27.62 ± 1.03** | **57.60 ± 2.50** | Proposed distance |
| Avg | 16.04 ± 0.68 | 31.88 ± 1.88 | L2 |
| Max | 13.14 ± 0.18 | 23.07 ± 0.59 | L2 |
| Max-SPM | 13.07 ± 0.18 | 22.68 ± 0.57 | L2 |
| MLLGW | **27.72 ± 0.92** | **59.53 ± 2.52** | L1 |

Table 2: Base-600: We can see that the proposed distance function is more adequate for WSA than L2. The best WSA configuration using the proposed distance function in (a) outperforms all the baselines in MAP and P@10 in (b). The best results in each table are shown in boldface. For each method, we show the best assignment (shown in the Assignment column). In (c), we show only the results of hard assignment.

precision values are obtained when images containing the same main object are retrieved, disregarding their background. Figure 4 shows some images from Base-600.

For Base-600, we used a dictionary of 15 000 visual words and all images were used as queries. The threshold $t$ of WSA for very soft assignments was experimentally determined as 0.03 (considers only visual word activations greater than 3%).

Table 2(a) shows how the proposed distance function improves the performance of WSA in relation to L2 distance. For all WSA variations the improvement is remarkably good. WSA presents MAP values around 24% for the L2 distance while for the proposed distance function, its MAP increases to more than 40%. Analyzing P@10, the values go from 50% to more than 90%. The use of the weighted window improved the results of WSA when using the Euclidean distance. However, when using the proposed distance function, only a very small improvement was observed in the P@10 measure.

The results for the baselines are presented in Table 2(b). We can see that max-SPM does not improve the performance over max pooling, giving a clear indication of the curse of dimensionality. Max-SPM presents one of the best results in the classification experiments (see Section 5), however, in the retrieval scenario its performance is degraded due to its large feature vector. Max pooling shows the best MAP and precision values in relation to the baselines when the assignment is very soft.

Table 2(c) compares the results of the methods in the condition of hard assignment. We present such comparison because the MLLGW method is based on hard assignment [28]. The results show that even in such conditions, WSA-ww provides comparable performance to MLLGW. However, considering that soft assignment is nowadays more adequate is many situations, WSA outperforms MLLGW, as we can see in Table 2(a).

Comparing the WSA configuration when using the proposed distance function with the best baseline (max pooling with soft assignment $\sigma$=150), we see a superiority of WSA. If we compare WSA with max-SPM (the spatial pooling baseline), we can see that WSA is very superior, both in terms of MAP (around 20 percentage points) and P@10 (around 50 percentage points).

We have created an interface based on Eva tool [20] to show the retrieved images of each pooling method and this interface is available online[3].

*Paris.* Paris dataset is composed of 6 392 images divided into 9 categories of different sizes. Each category represents a monument in the city of Paris, France. Although divided into 9 categories, the relevance between images is not necessarily based on the category division. A set of 55 query images was specifically released by dataset creators for standard evaluation purposes. Each query has its own pool of relevant images. We have computed the MAP and P@10 measures using the 55 query set and their respective pool (we have used the ground truth files with suffix "_ok"). Figure 5 shows examples of Paris dataset images. We have resized each image to have at most 512 pixels in the largest dimension (width or height), keeping the aspect ratio.

For the Paris dataset, we have used a dictionary of 8 000 visual words as it presented better performance in [8]. The threshold $t$ of WSA for very soft assignments was experimentally determined as 0.01 (considers only visual word activations greater than 1%).

Table 3(a) shows the large improvement in results of WSA when using the proposed distance function. MAP values increased from around 18% to more than 35% and P@10, from around 63% to more than 90%. Considering the use of the weighted window, for L2 distance no

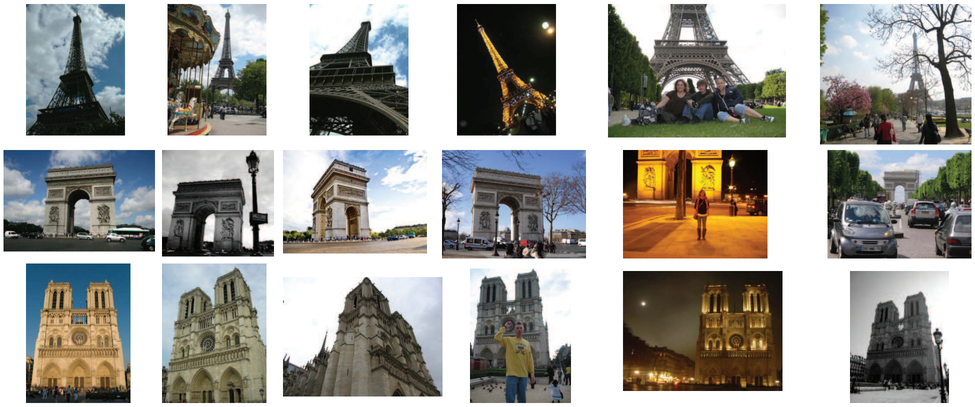


Figure 5: Sample images from the Paris dataset, highlighting 3 categories (one per row). There are 9 categories, each showcasing a landmark of the city of Paris, France.

improvement was observed. However, for the proposed distance, WSA was better without the window. Those results are different from the ones observed when using the weighted window in Base-600. Probably, the reason is that in Base-600 the main object, which is responsible for the dataset categorization, appears only in smaller size in relation to the image. Therefore, the use of windows was able to separate object and background information into the feature vector. In Paris dataset, the monument of interest has different sizes and appears in different positions into the images, therefore, a more general representation is necessary and was obtained by the WSA version without the windows. Another possible reason is that in Base-600 the background information is noisy, while in the Paris dataset, the background can provide information that helps the discrimination among objects of interest.

Table 3(b) shows the results for the baselines using L2 distance. We can see that max pooling has the best effectiveness. As observed for Base-600, the use of Spatial Pyramids (max-SPM) does not improve the results of max pooling, giving an indication of the curse of dimensionality. Comparing the results of max-SPM and WSA, we can see that WSA is very superior both in terms of MAP and P@10. Therefore, considering the use a spatial pooling method in retrieval experiments, WSA shows to be a promising choice, being more recommended than Spatial Pyramids because of its compact feature vector.

Table 3(c) compares the methods in hard assignment, a condition to the MLLGW method. We can see that, in such conditions, MLLGW achieves the highest rates and WSA is the second

(a) WSA: L2 distance × proposed distance function

| | | **L2 distance** | | | **Proposed distance** | |
|---|---|---|---|---|---|---|
| **Pooling** | **MAP (%)** | **P@10 (%)** | **Assignment** | **MAP (%)** | **P@10 (%)** | **Assignment** |
| WSA | 17.78 ± 2.63 | 62.91 ± 8.05 | Soft ($\sigma$=90) | **35.77 ± 4.22** | **90.73 ± 4.37** | Soft ($\sigma$=90) |
| WSA-ww | 19.68 ± 3.70 | 58.00 ± 8.84 | Soft ($\sigma$=60) | 21.00 ± 4.60 | 55.82 ± 9.08 | Hard |

(b) Baselines

| **Pooling** | **MAP (%)** | **P@10 (%)** | **Assignment** | - | - | - |
|---|---|---|---|---|---|---|
| Avg | 15.03 ± 3.64 | 58.18 ± 9.30 | Soft ($\sigma$=90) | - | - | - |
| Max | **28.68 ± 5.03** | **79.64 ± 7.08** | Soft ($\sigma$=150) | - | - | - |
| Max-SPM | 12.87 ± 2.71 | 50.00 ± 9.41 | Soft ($\sigma$=150) | - | - | - |

(c) Comparing methods in *hard* assignment

| **Pooling** | **MAP (%)** | **P@10 (%)** | **Distance function** |
|---|---|---|---|
| WSA | 25.26 ± 4.01 | 60.55 ± 7.02 | Proposed distance |
| WSA-ww | 21.00 ± 4.60 | 55.82 ± 9.08 | Proposed distance |
| Avg | 12.53 ± 3.38 | 46.00 ± 9.86 | L2 |
| Max | 4.53 ± 0.55 | 10.36 ± 0.50 | L2 |
| Max-SPM | 4.20 ± 0.54 | 10.18 ± 0.36 | L2 |
| MLLGW | **32.42 ± 5.21** | **80.55 ± 7.52** | L1 |

Table 3: Paris: The proposed distance function boosts WSA effectiveness in relation to L2. Comparing the best WSA with the proposed distance in (a) to the best baseline in (b), we can see that WSA has better average values, but the confidence intervals intersect. The best results in each table are shown in boldface. For each method, we show the best assignment scheme (shown in the Assignment column). In (c), we show only the results of hard assignment.

best method. However, considering that soft assignment is more adequate in many situations nowadays, WSA would be more recommended than MLLGW, as we can see in the results presented in Table 3(a).

The results presented in Table 3 show the superiority of WSA in relation to the baselines, however a simple statistical analysis using the confidence intervals shows that the gain in favor of WSA is not statistically meaningful (confidence intervals intersect). Therefore, we have performed a per-query analysis to better understand the difference between the methods. This kind of analysis puts into the statistical model the query variability, oppositely to the analysis shown in previous tables. The previous analysis excludes the query variability considering that their differences are noise in the statistical model. This is one of the reasons for the large confidence intervals presented previously. However, the previous analysis is useful to have a general idea of the performance of the methods evaluated. The per-query analysis solves this problem and gives a deeper understanding of how methods differ from each other. It is important to mention that for Base-600, as the main object for each category is always the same in the middle of the image and has only few variations, a per-query analysis is not necessary, and the intra-class differences can be considered noise.

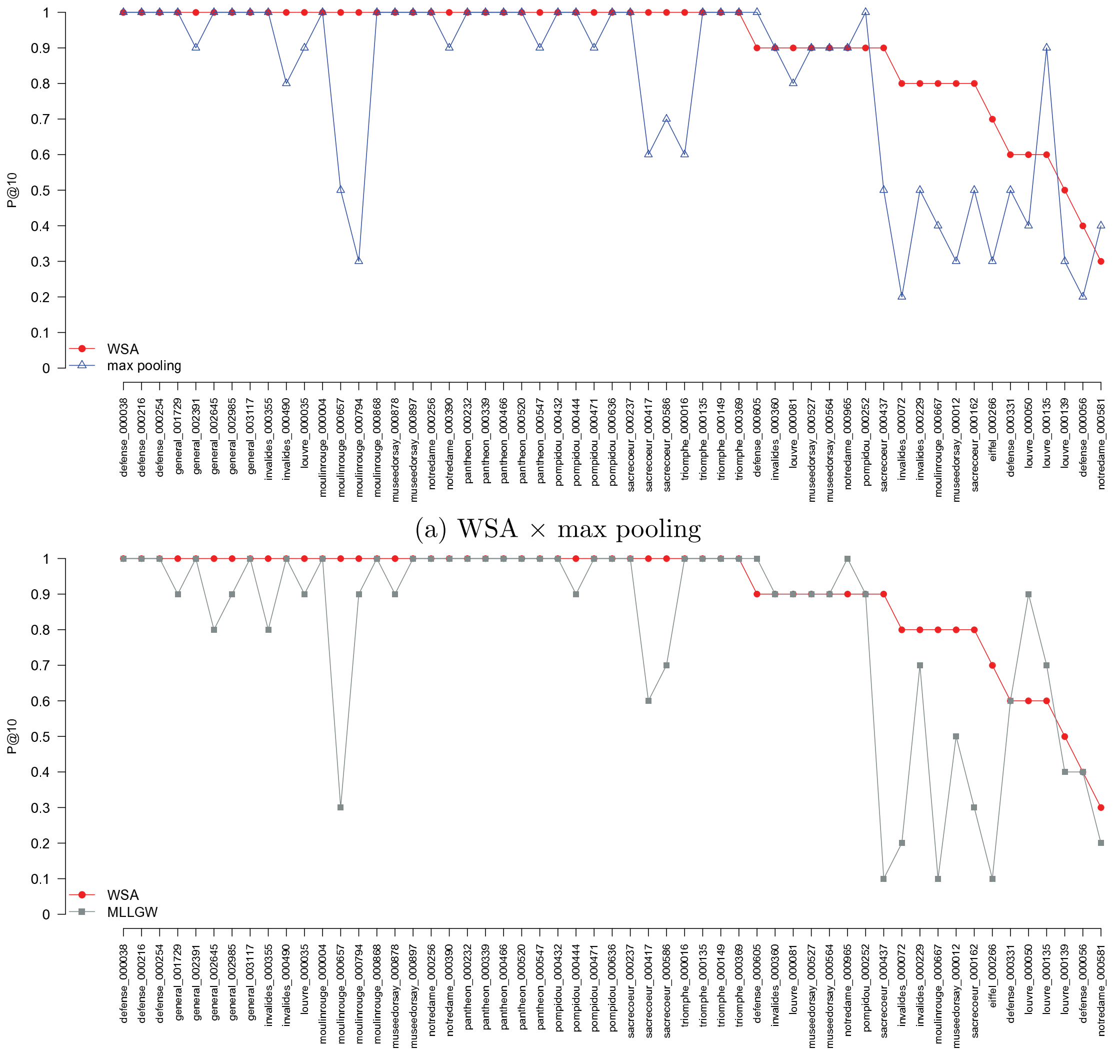
We have selected the best WSA configuration to compare with the best baseline configurations. The best WSA performance considering P@10 values was obtained when using soft assignment ($\sigma$=90) and the proposed distance function. The best performance of a non-spatial baseline was obtained by max pooling with soft assignment ($\sigma$=150) and the L2 distance. And the best spatial baseline was MLLGW with hard assignment.

Our analysis uses S-curves and a paired-test. S-curves put in comparison the effectiveness measures obtained for each of the 55 query images. To plot a S-curve, we have selected WSA as the reference method, sorted the precision values of each query in decreasing order, and plotted them into the graph. Using the same query order obtained, we plot the precision values for max pooling and MLLGW methods. Figure 6 shows the results.

Analyzing the S-curves, we can see that WSA is better than max pooling and also than MLLGW for many of the queries. In Figure 6(a), we can see that max pooling wins for only 4 queries, while WSA wins for 23. In Figure 6(b), MLLGW also wins for only 4 queries, while WSA wins for 20. The largest precision difference in favor of WSA in relation to max pooling is around 70% and around 80% in relation to MLLGW. On the other hand, when max pooling or MLLGW are better than WSA, the differences in precision are smaller, being at most 30%.

Figure 7 shows the results for the paired-test. In a paired test, we compute the differences of MAP values (or P@N values) for two methods for all corresponding pair of queries. Then, we compute the average and the confidence intervals of those differences. If the confidence interval includes the zero, the two methods are equivalent at that confidence level. Otherwise, the sign of the difference indicates the best method. In Figure 7, WSA is the first method and max pooling or MLLGW is the second, therefore, a positive value would indicate that WSA is better and, a negative value, that max pooling or MLLGW is better. Thus, for a confidence of 95%, WSA is better than max pooling for P@5, P@10, P@20, P@30, and MAP (as the confidence intervals do not intersect the zero and all values are above zero). Comparing with MLLGW, WSA is better in all measures except MAP, the only case where the interval touches the zero.

An online interface based on Eva tool [20] is available to show the retrieved images of some pooling methods[4].

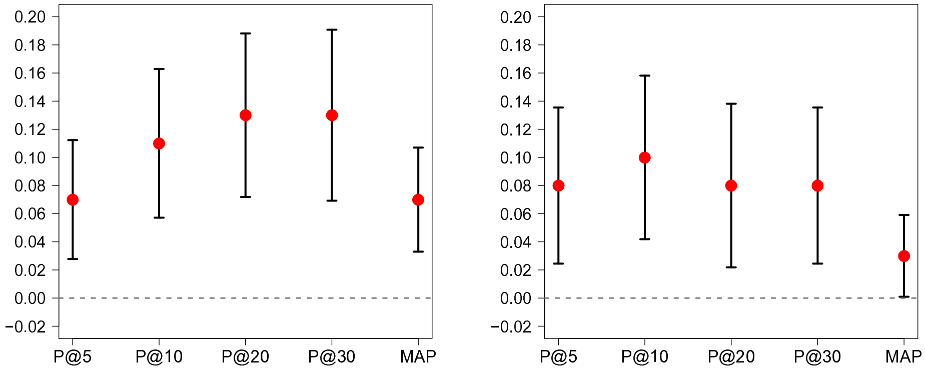


(a) WSA × max pooling

(b) WSA × MLLGW

Figure 6: S-curves for the best WSA configuration (red line with circle) and the best baselines (max pooling and MLLGW) in Paris dataset. The S-curves highlight the performance (P@10 measure in the vertical axis) of the two methods for each of the queries (horizontal axis), allowing to appreciate visually how often one outperforms the other. We can see that WSA has better effectiveness than (a) max pooling and (b) MLLGW in a large number of queries.

In this work, we focus on evaluating the retrieval effectiveness, therefore, we are not providing experiments measuring the efficiency of methods. We know that scalability is an important aspect for retrieval systems and some works target that by compacting the image representation [8, 35], highlighting the importance of compact feature vectors in that scenario. We can point that as WSA has a larger feature vector than avg and max pooling, it will be less efficient.



(a) WSA × max pooling

(b) WSA × MLLGW

Figure 7: Paris dataset: paired-test comparing WSA with (a) max pooling and (b) MLLGW. As the average of the differences (vertical axis) including the confidence intervals are always positive (do not include the zero, except for MAP in b), the test indicates a superiority of WSA for all measures used (horizontal axis).

However, considering most of the spatial pooling approaches and specially the Spatial Pyramids, WSA has a more compact feature vector, which makes it more efficient. Additionally, as WSA computes a vector of 4 dimensions for each visual word, indexing them to accelerate retrieval does not represent a challenge. Tools like inverted files or customized trees such as in [8] should be considered as relevant solutions.

*Conclusions.* Considering the questions presented in the beginning of this section, we can point that WSA has better effectiveness than the most popular approach for spatial pooling, the Spatial Pyramids. WSA presented the best results in both datasets used. The use of the proposed distance function has largely improved the effectiveness of WSA when compared to the L2 distance.

Soft assignment was also important for improving the quality of the WSA representation.

The results presented also indicate the importance of compact feature vectors in the retrieval scenario. We could note that the use of Spatial Pyramids did not improve the performance of max pooling, having, in fact, reduced its discriminating power.

We could observe that the use of the weighted window in the counting process was good only for Base-600, where the main object is small and centrally located in all the images. However, the gain obtained was small. For the Paris dataset, WSA without windows had better effectiveness. Therefore, in most of the situations, we argue that WSA in its raw format (without the windows) is more recommended. In a specific scenario similar to the one presented in Base-600, the weighted window should be considered.

As summary, we conclude that the spatial information encoded by WSA can improve the effectiveness of retrieval systems without suffering from large feature vectors, usually generated by many spatial pooling methods. We also would like to emphasize the importance of the spatial information for improving image representations. As WSA encodes only the spatial arrangement of visual words and not explicitly their activations in the feature space, we can conclude that knowing *where* visual words tend to appear in an image can be more discriminant than knowing *how frequent* visual words appear in an image.

## 5. Experiments for image classification

The experiments in the classification scenario are based on traditional image datasets which comprise the semantic-search application. We focus our experiments on evaluating scene categorization using 15-Scenes dataset [9] and object categorization using Caltech-101 dataset [5].

The main questions to be answered by these experiments are:

- is the accuracy of WSA comparable to the best methods from literature?
- what is the impact of soft assignment in WSA?
- can WSA performance be improved by combining it with spatial pyramids?

The images were represented using the same configurations presented in the retrieval experiments in Section 4: random dictionaries based on the Harris-Laplace detector and the SIFT descriptor, combined with some coding and pooling strategies. However, dictionaries of 1 000 words were used because in 15-Scenes and Caltech-101 datasets small dictionaries are commonly used [23, 10]. The following pooling strategies were employed: average, max, max with Spatial Pyramids (max-SPM), WSA, and WSA with Spatial Pyramids (WSA-SPM1). For WSA, we have used Spatial Pyramids of level 1 (5 WSA vectors concatenated). We have not used Spatial Pyramids of level 2 for WSA, because this would make the feature vector larger

| **Pooling method** | **Acronym** | **Feature vector size** |
|---|---:|---:|
| Average | avg | k |
| Max | max | k |
| Max pooling with Spatial Pyramids | max-SPM | 21k |
| Word Spatial Arrangement | WSA | 4k |
| Word Spatial Arrangement with Spatial Pyramids | WSA-SPM1 | 20k |

Table 4: Acronyms and feature vector sizes for the pooling methods being evaluated. $k$ is the dictionary size.

than max-SPM. WSA-SPM1 (5×4×k) is still more compact than max-SPM (21×k). Table 4 summarizes the pooling methods compared and their feature vector sizes.

Spatial Pyramids (SPM) are used as our main baseline for spatial pooling of visual words, because, although there are many new approaches with better and comparable results to SPM, SPM still are the most widely used. Another advantage is that SPM can be used together with many new methods, as well as with WSA. The other spatial pooling methods adequate for the classification scenario presented in Section 2.2 were not used because they present limitations. The spatial-bag-of-features [11] generates extremely large feature vectors and the geometric $l_p$-norm pooling [10] depends on resizing all the images to the same size. Using dimension reduction techniques or special treatments for individual methods were not in the scope of our experiments, because they can create advantages for a specific method and make the comparison unfair.

We are also not showing the results of WSA-ww, because it presented inferior accuracy than the WSA version that does not use windows. This also happened in the retrieval experiments on the Paris dataset (see Section 4). WSA-ww was good only in the retrieval experiments on Base-600, where the main object was in the middle of the image and in small size in relation to the whole image. These characteristics are not present in the 15-Scenes dataset neither in the Paris dataset, therefore we would expect that WSA without windows would perform better than WSA-ww. In relation to Caltech-101, many categories contain the object in the middle of the image as in Base-600, however, in Base-600 the object is exactly the same for a given class while this is not true for Caltech-101. On top of that, in Base-600 the background information is noisy while in Caltech-101 the background is mostly white.

For the classification setup, we have employed SVMs with linear kernel ($c$=1.0) and a balanced validation. A number of samples per class (nTrain) was taken for training and the rest were used for testing. We have varied nTrain from 5 to 100 in the 15-Scenes dataset and from 5 to 30 in the Caltech-101 dataset. Results are reported with confidence intervals for $\alpha$=0.05 for the 5 runs of each balanced validation. We have initially performed grid search to determine the best $c$ parameter of SVM, however, as the improvements were proportional to all methods, we have fixed $c$=1.0. Our objective was to create a suitable environment to perform a fair comparison between the methods, even knowing that were are not showing the state-of-the-art results for those datasets. We know that by using other classification strategies, like fusion or by employing specialized image representations for the application we could achieve higher rates. Nonetheless, our classification protocol is suitable to evaluate the methods.

It is important to highlight that, although we have proposed a distance function for WSA to be used in the retrieval experiments, we have not employed that function in the classification experiments in the SVM kernel, for instance. Thus, the classification results are based on the linear kernel only. Some tests were also made using the RBF kernel, but WSA has shown to work better with the linear kernel.

*15-Scenes.* The 15-Scenes dataset [9] is composed of 4 485 images divided into 15 categories [9]. Each category comprises a different scene (both indoor and outdoor) and has a variable number of images. The scene categories with their sizes in parenthesis are: bedroom (216), CALsuburb (241), industrial (311), kitchen (210), livingroom (289), MITcoast (360), MITforest (328), MIThighway (260), MITinsidecity (308), MITmountain (374), MITopencountry (410), MIT-street (292), MITtallbuilding (356), PARoffice (215), and store (315). Images have dimension of at most 552 pixels in width or 411 pixels in height. The threshold $t$ of WSA for very soft assignments was experimentally determined to consider only visual word activations greater than 2% ($t$=0.02).

Figure 8 shows how the WSA descriptors react to different assignment softness. We can see that WSA-SPM1 is less affected by the changes in the assignment softness. We can also note that WSA-SPM1 has a larger increase in accuracy as the training set grows. This means that WSA alone is more robust in conditions of smaller training sets.

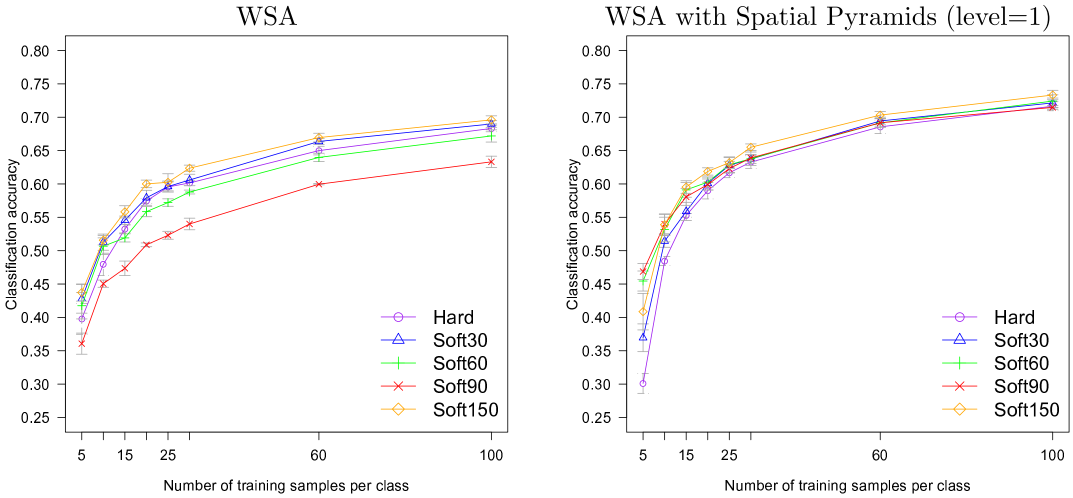


Figure 8: Evaluating the effect of soft assignment for WSA and WSA-SPM1 in the 15-Scenes dataset for variable training set sizes. WSA-SPM1 is less affected than WSA when the assignment softness changes, however, their best results are using the most soft assignment.

Figure 9 aggregates all the results for nTrain=100. The graph shows how each method performs when changing the assignment softness. While max pooling has an increase in accuracy as the assignment becomes softer, average pooling saturates at $\sigma$=150. WSA is also benefited by softer assignments, but the improvement is small.

By analyzing Figure 9, we can also compare the methods in each assignment schema. For harder assignments (hard and soft $\sigma$=30), WSA outperforms avg and max pooling and is close to max-SPM. In relation to max pooling, the differences in accuracy in favor of WSA, considering the confidence intervals, are around 4% and 3% for the above mentioned assignments, respectively. Although WSA is outperformed by max-SPM in those assignments, the differences in favor of max-SPM, considering the confidence intervals, are around 2.5% and 3.5%, respectively.

Considering the use of Spatial Pyramids with WSA (WSA-SPM1), we can see an improvement in accuracy in relation to WSA alone. The gain goes from around 2 percentage points in harder assignments (hard and soft $\sigma$=30) to around 7 percentage points for soft $\sigma$=90. Also, WSA-SPM1 has equivalent accuracy to max-SPM for harder assignments.

Regarding the best scenario for each method, max-SPM has an average accuracy of 75.62% (soft $\sigma$=150) while WSA has 69.61% (soft $\sigma$=150). Considering the confidence intervals, the

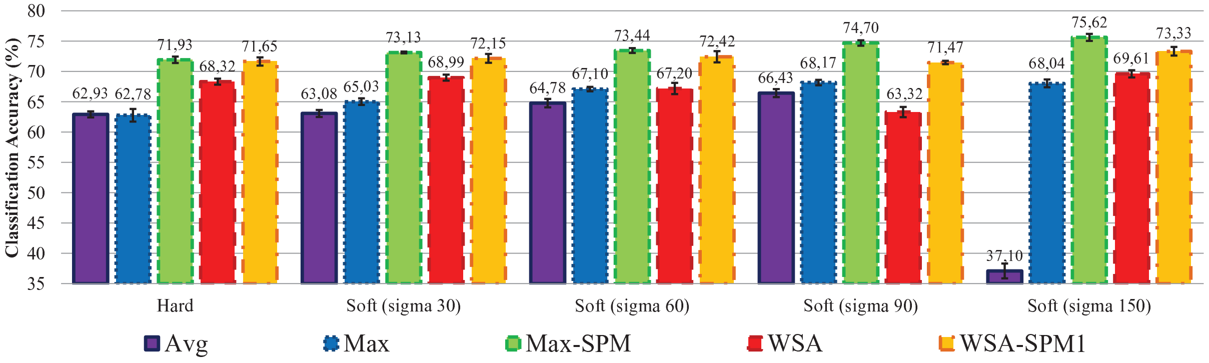


Figure 9: 15-Scenes: average classification accuracies with confidence intervals for nTrain=100.

difference in favor of max-SPM is only around 4%, even WSA presenting a vector more than 5 times smaller than max-SPM.

Table 5 shows a comparison for individual classes of 15-Scenes dataset considering the best non-spatial baseline configuration (max pooling with soft assignment $\sigma$=90) and the best WSA configuration (WSA with soft assignment $\sigma$=150). Methods are equivalent in most of the classes, but in some of them there is statistical difference. A paired-test comparing the results per class shows that, for nTrain=100, the methods are equivalent.

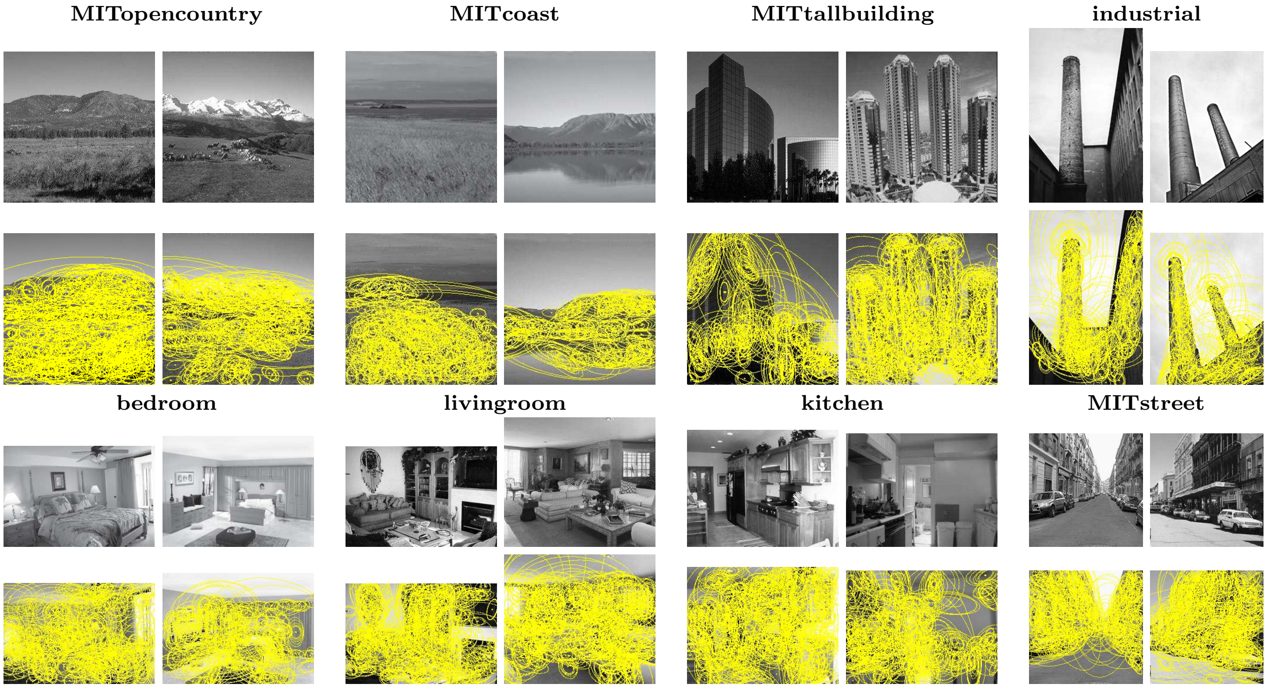
Table 5 also shows images from the classes where there is a meaningful difference between WSA and max pooling. We are also showing images from classes which are confusing for the methods. They were obtained from an analysis in the confusion matrices of each method.

WSA is worse than max pooling in classes *MITforest*, *MITopencountry*, and *MITtallbuilding*. For *MITopencountry*, WSA mostly confuses it with *MITcoast*. We could note that many images from both classes have clear sky, as the ones shown in Table 5, which means that no points are detected in the top part of the images, but many points appear the lower part (see the images just below each original image in Table 5). Therefore, the spatial relationship between the lower parts of those images were probably not enough to distinguish between the two classes.

WSA is confusing the class *MITtallbuilding* with the class *industrial*. We can suggest that the spatial relationship between the tall structures are generating similar WSA representations.

WSA is better than max pooling in classes *bedroom*, *industrial*, *livingroom*, *MITinsidecity*, and *MITstreet*. Max pooling makes confusion between *kitchen*, *bedroom*, and *livingroom*. There

| Class | Max(soft-$\sigma$=90) | WSA(soft-$\sigma$=150) | Winner |
|---:|---:|---:|:---:|
| bedroom | 49.66 ± 2.70 | 55.86 ± 1.80 | WSA |
| industrial | 37.16 ± 3.16 | 44.36 ± 1.68 | WSA |
| livingroom | 45.50 ± 1.80 | 54.07 ± 4.11 | WSA |
| MITforest | 92.72 ± 1.38 | 86.49 ± 2.11 | Max |
| MITinsidecity | 65.77 ± 1.09 | 68.27 ± 1.33 | WSA |
| MITopencountry | 67.48 ± 1.41 | 63.29 ± 1.63 | Max |
| MITstreet | 67.92 ± 2.39 | 73.02 ± 2.25 | WSA |
| MITtallbuilding | 78.52 ± 1.39 | 75.16 ± 1.76 | Max |



**MITopencountry** **MITcoast** **MITtallbuilding** **industrial**

**bedroom** **livingroom** **kitchen** **MITstreet**

Table 5: Contrasting the performance of WSA and max pooling in the classes of 15-Scenes dataset for nTrain=100. WSA significantly outperforms max pooling in classes *bedroom*, *industrial*, *livingroom*, *MITinsidecity*, and *MITstreet*. The opposite happens in classes *MITforest*, *MITopencountry*, and *MITtallbuilding*. We show examples of images from classes where there is meaningful difference between WSA and max pooling. There are also images from the classes which are confused by the methods, which were obtained by analyzing the confusion matrices of the results. Below each image, we show the points detected by Harris-Laplace detector.

must be a large intersection between the visual words of those classes. Therefore, knowing only the visual word activations was not enough to distinguish among them. That is, their spatial relationship were more important.

For class *MITstreet*, the spatial arrangement of visual words present in the tall structures (buildings for *MITstreet* and chimneys for *industrial*) and the other structures could improve significantly the discrimination between those classes, and this information was not captured by max pooling.

To summarize the results in the 15-Scenes dataset, WSA outperforms the non-spatial base-

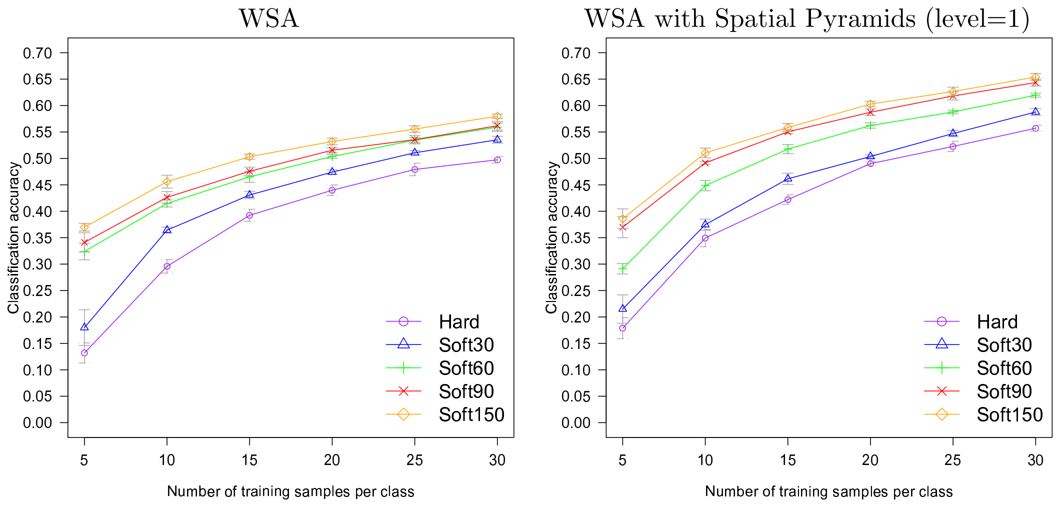


Figure 10: Evaluating the effect of soft assignment for WSA and WSA-SPM1 in the Caltech-101 dataset for variable training set sizes. In both graphs, the softer the assignment, the better for WSA.

lines (avg and max pooling) and, in harder assignments, is close to max-SPM. Although WSA does not have a decrease in accuracy in very soft assignments, max-SPM presents a larger improvement than WSA. However, as WSA has a vector more than 5 times smaller than max-SPM, it would be more efficient in terms of time and space. Therefore, WSA is a good option to encode the spatial arrangement of visual words for scene categorization while saving storage space and classification time.

*Caltech-101.* The Caltech-101 dataset [5] is a very popular dataset used to evaluate object recognition approaches. Although there are known problems with this dataset [36], it is still largely used in the literature. Caltech-101 has 9 144 images divided into 101 object categories and 1 category of distractors (BACKGROUND_Google). The number of images per class varies from 31 (inline_skate) to 800 (airplanes). We have resized the images to half of their original size (keeping the aspect ratio). The threshold $t$ of WSA for very soft assignments was experimentally determined to consider only visual word activations greater than 1% ($t$=0.01).

Figure 10 shows how WSA methods perform in different assignment schemes when varying the training set size. We can see that both methods improve the accuracy as the assignment becomes softer.

The graph in Figure 11 shows the overall results for all methods in the different assignment

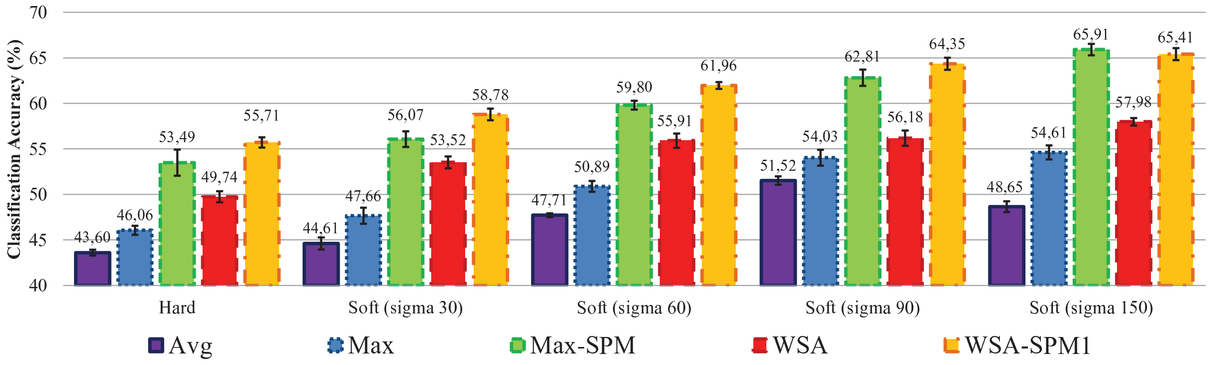


Figure 11: Caltech-101: average classification accuracies with confidence intervals for nTrain=30.

schemes tested for Caltech-101 dataset, using nTrain=30. We can see that WSA outperforms both avg and max pooling. In relation to max pooling, the differences in accuracy in favor of WSA, considering the confidence intervals, go from around 1% (soft $\sigma$=90) to around 4% (soft $\sigma$=30).

WSA is outperformed by max-SPM, but the differences, considering confidence intervals, are around only 1%, for hard assignment and soft assignment ($\sigma$=30). For assignments softer than those, max-SPM increases its accuracy more than WSA.

Considering WSA combined with Spatial Pyramids (WSA-SPM1), we can note again a great improvement in the performance in relation to WSA alone. The accuracy difference increases from around 4% for harder assignments to around 6% for softer assignments. Besides that, WSA-SPM1 outperforms max-SPM for harder assignments ($\sigma$<90).

Comparing the methods in their best scenarios, we can see that max-SPM outperforms WSA. It achieves accuracy of 65.91% (soft $\sigma$=150) while WSA achieves 57.98% (soft $\sigma$=150), a difference of almost 7% considering the confidence intervals.

A per-class analysis was also performed for Caltech-101 considering the best configurations of the non-spatial baseline (max pooling with soft assignment $\sigma$=150) and the best WSA configuration (soft assignment $\sigma$=150). The results point that WSA is statistically superior to max pooling, as we have already observed by analyzing the average accuracies in Figure 11.

As summary, the experiments in Caltech-101 show that WSA does not win in classification accuracy in relation to max-SPM. However, WSA improves accuracy over max pooling and is very close to max-SPM in harder assignments. This means that WSA can improve object categorization by including spatial information of visual words in a compact feature vector, being an interesting alternative to save storage space and classification time in relation to Spatial Pyramids. If accuracy is more important than efficiency, WSA can be used together with Spatial Pyramids (WSA-SPM1) to achieve a better accuracy than max-SPM, yet saving some space (in harder assignments). Therefore, if storage is a constraint in the classification system and classification time is important, WSA can include spatial information of visual words keeping compact feature vectors and still increasing accuracy rates over non-spatial methods.

*Conclusions.* Considering the questions presented in the beginning of this section, we can draw our conclusions about the performance of WSA in the classification scenario:

- WSA has better performance than the non-spatial baselines.
- WSA is close to Spatial Pyramids, specifically for harder assignments ($\sigma \leq 60$).
- Soft assignment improves WSA performance.
- Combining WSA and Spatial Pyramids improves the classification accuracy.

Our classification experiments show that WSA is a good option for classification systems requiring better accuracies than traditional non-spatial pooling methods. WSA is recommended in place of max-SPM if storage is an important constraint for the system, because it saves space in a compromise of loosing a few accuracy in some cases in relation to Spatial Pyramids, but having comparable performance in others, specially for harder assignments. Smaller feature vectors also lead to faster classification, which is another advantage of WSA.

Although we are not showing surprisingly high accuracies in the datasets used, we show the power of the spatial information encoded by WSA. WSA does not encodes how many times visual words appear in an image. The whole dictionary model is based on knowing the frequency of occurrence of visual words in the image space. However, even without considering this information, we could obtain an image representation that is equally or more discriminating than those representations. This shows the importance of considering the spatial information of visual words in the image space.

## 6. Conclusions

This paper presented WSA (Word Spatial Arrangement), a spatial pooling approach to encode the spatial arrangement of visual words. WSA has the advantage of working both in retrieval and classification scenarios. WSA encodes the relative position of visual words in the image by splitting the image space using each point as the origin of a four-quadrant structure and counting the number of points in each quadrant.

To work in the retrieval scenario, we have also proposed a distance function to be used with WSA. Experimental results show that the proposed distance function remarkably improves the effectiveness of WSA over the Euclidean distance. Experiments in the retrieval scenario also show that WSA outperforms the most popular approach for spatial pooling, the Spatial Pyramids. The latter degraded the performance of max pooling, giving a clear indication of the curse of the dimensionality in scenarios where distance computations are required. A per-query analysis by S-curves and paired-tests have shown that WSA is also superior to max pooling and to a recent graph-based approach, the best baselines in our retrieval experiments. We also provide an online interface based on Eva tool [20] to navigate through the results.

Experiments in the classification scenario have shown that WSA has close accuracy to max pooling with Spatial Pyramids (max-SPM) in harder assignments. For configurations of very soft assignments, max-SPM is superior. However, WSA computes vectors more than 5 times smaller than max-SPM, which is a clear advantage considering efficiency, both in terms of time and space. Anyhow, if accuracy is priority, WSA can also be combined with Spatial Pyramids, boosting its performance.

An important conclusion of our experiments is related to the importance of the spatial information of visual words in the image space to improve image representations. WSA encodes only the spatial information of visual words and does not take into account their frequency of occurrence in the image space. The occurrence of visual words in images is one of the main aspects of the visual dictionary model. We could show that even without using the information about the frequency of occurrence of visual words in the image space, we could create an image representation that is equally or more discriminating than representations based on them. This highlights the importance of encoding the spatial arrangement of visual words in the image space. WSA has shown to be an effective solution for that purpose.

For future work, we plan to evaluate WSA in other datasets, considering again both the retrieval and classification scenarios. We also plan to investigate how to improve WSA in very soft assignments.

## 7. Acknowledgements

We would like thank Nguyen Vu Hoàng for the discussion on the description of spatial relationships. We also would like to thank Fapesp (grant numbers 2009/10554-8 and 2009/05951-8), CNPq, Capes, and Microsoft Research for infrastructure and financial support.

## References

[1] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: International Conference on Computer Vision, Vol. 2, 2003, pp. 1470–1477.

[2] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Gool, A comparison of affine region detectors, International Journal of Computer Vision 65 (2005) 43–72.

[3] K. E. A. van de Sande, T. Gevers, C. G. M. Snoek, Evaluating color descriptors for object and scene recognition, Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1582–1596.

[4] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[5] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, Conference on Computer Vision and Pattern Recognition Workshop 12 (2004) 178.

[6] A. Andreopoulos, J. K. Tsotsos, 50 years of object recognition: Directions forward, Computer Vision and Image Understanding 117 (8) (2013) 827–891.

[7] O. A. B. Penatti, E. Valle, R. d. S. Torres, Encoding spatial arrangement of visual words, in: Iberoamerican Congress on Pattern Recognition, Vol. 7042, 2011, pp. 240–247.

[8] N. V. Hoàng, V. Gouet-Brunet, M. Rukoz, M. Manouvrier, Embedding spatial information into image content description for scene retrieval, Pattern Recognition 43 (2010) 3013–3024.

[9] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2169–2178.

[10] J. Feng, B. Ni, Q. Tian, S. Yan, Geometric lp-norm feature pooling for image classification, in: Conference on Computer Vision and Pattern Recognition, 2011, pp. 2609–2704.

[11] Y. Cao, C. Wang, Z. Li, L. Zhang, L. Zhang, Spatial–bag–of–features, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 3352–3359.

[12] W. Zhou, Y. Lu, H. Li, Y. Song, Q. Tian, Spatial coding for large scale partial-duplicate web image search, in: International Conference on Multimedia, 2010, pp. 511–520.

[13] H. Jégou, M. Douze, C. Schmid, Improving bag-of-features for large scale image search, International Journal of Computer Vision 87 (2010) 316–336.

[14] R. Weber, H. Schek, S. Blott, A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, in: International Conference on Very Large Data Bases, 1998, pp. 194–205.

[15] J. C. Traina, A. Traina, C. Faloutsos, B. Seeger, Fast indexing and visualization of metric data sets using slim-trees, Transactions on Knowledge and Data Engineering 14 (2) (2002) 244–260.

[16] H. Kang, M. Hebert, T. Kanade, Image matching with distinctive visual vocabulary, in: IEEE Workshop on Applications of Computer Vision, 2011, pp. 402–409.

[17] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, J.-M. Geusebroek, Visual word ambiguity, Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1271–1283.

[18] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases, in: Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[19] L. Liu, L. Wang, X. Liu, In defense of soft-assignment coding, in: International Conference on Computer Vision, 2011, pp. 1–8.

[20] O. A. B. Penatti, R. d. S. Torres, Eva - an evaluation tool for comparing descriptors in content-based image retrieval tasks, in: International Conference on Multimedia Information Retrieval, 2010, pp. 413–416.

[21] V. Viitaniemi, J. Laaksonen, Experiments on selection of codebooks for local image feature histograms, in: International Conference on Visual Information Systems: Web-Based Visual Information Search and Management, 2008, pp. 126–137.

[22] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: International Conference on Computer Vision, Vol. 1, 2005, pp. 604–610.

[23] Y.-L. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, Conference on Computer Vision and Pattern Recognition (2010) 2559–2566.

[24] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: European Conference on Computer Vision: Part I, Vol. 5302, 2008, pp. 304–317.

[25] S. Avila, N. Thome, M. Cord, E. Valle, A. de A. Araújo, Bossa: Extended bow formalism

for image classification, in: International Conference on Image Processing, 2011, pp. 2966–2969.

[26] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European Conference on Computer Vision, Vol. 6314, 2010, pp. 143–156.

[27] E. Mbanya, S. Gerke, P. Ndjiki-Nya, Spatial codebooks for image categorization, in: International Conference on Multimedia Retrieval, 2011, pp. 50:1–50:7.

[28] S. Karaman, J. Benois-Pineau, R. Mégret, A. Bugeau, Multi-layer local graph words for object recognition, in: Advances in Multimedia Modeling, Vol. 7131, 2012, pp. 29–39.

[29] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1349–1380.

[30] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, R. Zabih, Image indexing using color correlograms, in: Conference on Computer Vision and Pattern Recognition, 1997, p. 762.

[31] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: ACM Multimedia, 1996, pp. 65–73.

[32] W. Zhou, H. Li, Y. Lu, Q. Tian, Large scale image search with geometric coding, in: ACM Multimedia, 2011, pp. 1349–1352.

[33] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, W. T. Freeman, Discovering objects and their location in images, in: International Conference on Computer Vision, Vol. 1, 2005, pp. 370–377.

[34] S. Savarese, J. Winn, A. Criminisi, Discriminative object class models of appearance and shape by correlatons, in: Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2033–2040.

[35] H. Jegou, M. Douze, C. Schmid, P. Perez, Aggregating local descriptors into a compact image representation, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 3304–3311.

[36] A. Torralba, A. A. Efros, Unbiased look at dataset bias, in: Conference on Computer Vision and Pattern Recognition, 2011, pp. 1521–1528.