# Elementary block extraction for mobile image search

José Mennesson, Pierre Tirilly, Jean Martinet

# ELEMENTARY BLOCK EXTRACTION FOR MOBILE IMAGE SEARCH

*José Mennesson, Pierre Tirilly, Jean Martinet*

LIFL UMR Lille1-CNRS 8022, IRCICA, 50 avenue Halley, 59658 Villeneuve d'Ascq, France

## ABSTRACT

In this paper, we propose an original content-based image retrieval method using bag-of-words dedicated to building matching on mobile devices. In the literature, the repetitiveness of visual words in natural scenes, and especially in building images, has been demonstrated. Assuming images are composed of a set of elementary blocks, we represent them using only a few well-chosen features. In the context of image search on mobile devices, this allows to considerably reduce the size of the data to be sent to the server. This method has been experimented using SIFT descriptors on three well-known databases. Experimental results show that this method can outperform the standard bag-of-words approach while reducing the number of features used to represent images. Moreover, this general framework can be used in conjunction with any kind of descriptors and indexing methods.

***Index Terms*—** image retrieval, bag-of-words, visual feature selection, building images, mobile applications

## 1. INTRODUCTION

This paper addresses the problem of content-based building image retrieval on mobile devices. In the literature, local methods [1, 2, 3] are among the most popular and effective to describe images of natural scenes. They consist in computing visual descriptors in the neighborhood of regions of interest. It produces sets of local features that allows to deal with cluttered images, occlusions, shape variations, etc. Two different methods can be used to represent and match images based on local features. The first one matches all features [4] to compare images, which is computationally expensive. The second one aims at reducing the matching complexity by compressing or aggregating the descriptors, and/or performing approximate descriptor matching. Such methods include bags-of-words (BoW) [5, 6], hashing [7], Hamming embedding [8], product quantization [9], descriptor compression [3], VLAD [10] or Fisher vectors [11].

With the rise of mobile devices, the need to adapt image retrieval methods to these devices' constraints is growing. Indeed, such devices are limited in memory, speed, energy and bandwidth. To deal with these issues, three scenarios were proposed in [12]. The first one consists in transferring a compressed version of the query image to the server, which is in charge of extracting descriptors, retrieving the most similar images and returning thumbnails of the results to the mobile device. However, highly compressed images tend to contain visual artifacts that make the detection of regions of interest difficult. The second scenario consists in performing the whole retrieval task on the mobile device. It requires the whole database index to be stored on the mobile phone. The memory available on the device being limited, it restricts the size of the database, even when using memory-efficient indexing methods. Moreover, the retrieval process can require more computational power than the device can provide. The last strategy proposes to extract the descriptors on the device and to transfer them to the server for the retrieval task, possibly after a descriptor compression step. This intermediate approach has been shown to provide a good trade-off between hardware constraints (low memory and bandwidth use) and retrieval effectiveness [3].

The approach proposed in this paper adopts the third strategy. We propose to further reduce the bandwidth use by reducing the amount of data required to describe images. By taking advantage of the repetitiveness (or burstiness [8]) of visual elements in images, subsets of local descriptors are represented as single representative descriptors, called elementary blocks, that are transmitted to the server. Compared to existing methods of descriptor elimination [13, 14], our approach allows to take into account all initial descriptors, and is able to reduce the quantity of data required to describe the images at little or no cost in terms of retrieval effectiveness.

This article is structured as follows : in Section 2, an overview of our approach is presented. Section 3 describes the computation of the elementary blocks, and Section 4 the method to use them in a BoW framework on the server side. Finally, experimental results on well-known datasets are provided and discussed in Section 5.

## 2. OVERVIEW OF OUR METHOD

Images are known to contain many repeated, visually similar, elements [8]. We propose to leverage this property to build a lightweight representation of images. In a given query image, subsets of visually similar local descriptors are identified, and each of them is described by a single representative vector $eB_i$ and the number of initial descriptors that it contains

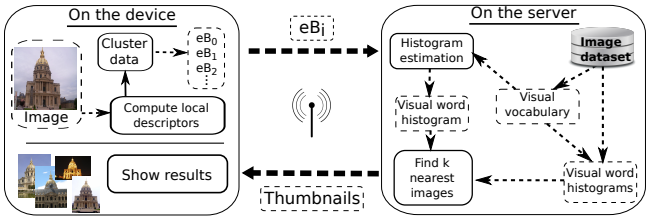



**Fig. 1**. Illustration of our method for mobile applications

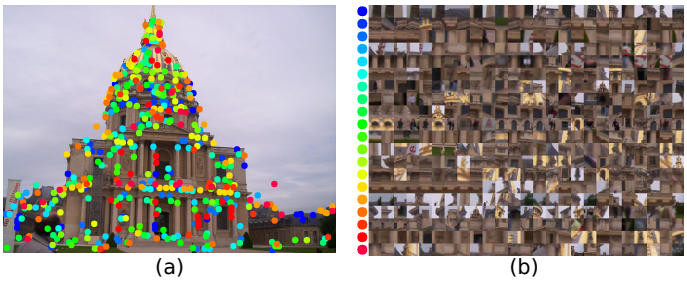



**Fig. 2**. Elementary blocks of *Les Invalides*. (a) SIFT keypoints are represented by circles; each color corresponds to a cluster. (b) Each row contains 25 sample patches from a given cluster.

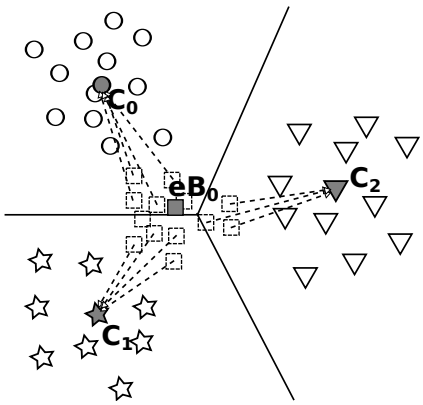



**Fig. 3**. Assignment of elementary block $eB_0$ to visual words $C_0$, $C_1$, and $C_2$.

$o(eB_i)$. In the remainder of this paper, a couple $(eB_i, o(eB_i))$ is called an elementary block. This is conceptually similar to BoW, only at the scale of a single image. Based on this notion and the standard BoW framework, our retrieval method (see Figure 1) is the following:

1. Keypoints are detected in a query image and described by local descriptors (e.g. SIFT descriptors [1]).
2. A number of elementary blocks are built from these descriptors.
3. Elementary blocks are transmitted to the server.
4. Elementary blocks are matched to the visual vocabulary to build the visual word histogram of the query image.
5. The query histogram is matched to the database histograms to retrieve similar images.
6. Search results are sent to the mobile device.

Although we use BoW as a basis in this paper, other quantization-based indexing methods (such as hashing [7] or VLAD [10]) could be used to perform step 5. Sections 3 and 4 describe steps 2 and 4 of the method, respectively.

## 3. ELEMENTARY BLOCK COMPUTATION

To build elementary blocks, local descriptors within an image have to be grouped into visually consistent sets. To do so, square-error partitioning methods like $k$-means, or hierarchical clustering techniques [15], which organize data into a nested sequence of groups, can be used. In this paper, the $k$-means algorithm is used, as it provides a good trade-off between precision and speed. Vectors $eB_i$ are defined as cluster centroids and $o(eB_i)$ is simply the number of features in the corresponding cluster. Figure 2 shows an example of elementary blocks built using $k$-means in an image of *Les Invalides* from the Paris dataset [16] (SIFT descriptors and 20 clusters, i.e. elementary blocks). Figure 2 shows the locations of the keypoints; each keypoint color corresponds to a given block. Figure 2(b) presents samples of visual elements belonging to each block. Some structures of the building emerge, such as pieces of roof, columns, balustrades, etc. The contribution to the block of "noisy" patches, like pedestrians in Figure 2(a), is limited because they are averaged with many actual building elements.

## 4. ELEMENTARY BLOCK ASSIGNMENT

Once the blocks $(eB_i, o(eB_i))$ are computed, they can be sent to the server. The following step consists in estimating the associated visual word histogram based on the blocks and the visual vocabulary. To do so, the $eB_i$ must be assigned to the visual words (or vocabulary centroids) $C_j$. Each $eB_i$ corresponds to a number of "virtual" descriptors distributed around it. As illustrated in Figure 3, there is no guarantee that all of them will fall into a single visual word: descriptors corresponding to $eB_0$ could be assigned to $C_0$, $C_1$ or $C_2$. To deal with this issue, we distribute all occurrences among several $C_j$ using a weighting function $w(eB_i, C_j)$. Euclidean distances between each $eB_i$ and all $C_j$ (denoted $D(eB_i, C_j) = ||eB_i - C_j||_2$) are computed, then normalized by rescaling them between 0 and 1 based on their minimum and maximum values. Weights (inspired from soft-assignment methods [16, 17]) are assigned to each couple

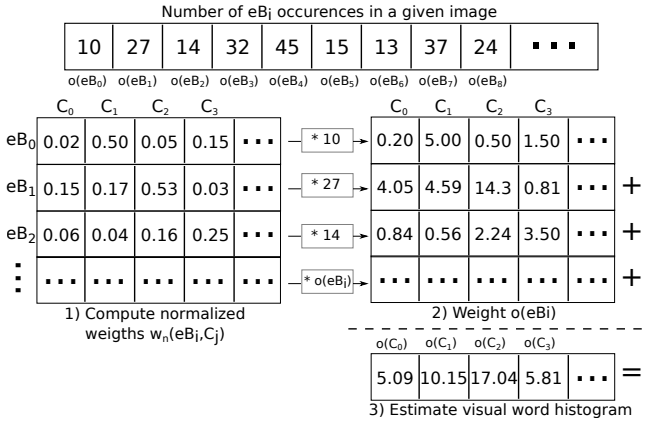



**Fig. 4**. Assignment of $eB_i$ to $C_j$ to estimate the associated visual word histogram.

$(eB_i, C_j)$ as follows:

$$w(eB_i, C_j) = exp\left(-\frac{D(eB_i, C_j)}{2 \times \sigma^2}\right) = exp\left(-\kappa \times D(eB_i, C_j)\right) \quad (1)$$

where $w$ is the weighting function, $\sigma \in \mathbb{R}^+$ is a parameter (set empirically) controlling the slope of the exponential function and $\kappa = \frac{1}{2\sigma^2}$. In order to estimate the distribution of all occurrences over the visual vocabulary, these weights are divided by the sum of these quantities over $j$:

$$w_n(eB_i, C_j) = \frac{w(eB_i, C_j)}{\sum_{j=0}^{N_C} w(eB_i, C_j)} \quad (2)$$

where $w_n$ is the normalized weighting function and $N_C$ is the size of the visual vocabulary. The final number of occurrences of visual word $C_j$ (denoted $o(C_j)$) is estimated as:

$$o(C_j) = \sum_{i=0}^{N_{eB}} w_n(eB_i, C_j) * o(eB_i) \quad (3)$$

where $N_{eB}$ is the total number of elementary blocks. This assignment process is illustrated in Figure 4.

## 5. EXPERIMENTS

### 5.1. Experimental settings

Images are described using SIFT descriptors with a difference of Gaussians (DoG) detector [1]. Several experiments using SURF descriptors (not reported in this paper) exhibited similar behavior (except that, as usually observed [18], SURF descriptors yields lower performance than SIFT). The $k$-means algorithm is used to build elementary blocks. Query BoW histograms are matched to database BoW histograms using the Euclidean distance. $N_{eB}$ is taken as a fixed ratio of the number of descriptors $N_D$: $N_{eB} = \alpha \times N_D$ ($\alpha \in ]0\ ;1]$). The visual vocabulary is computed using the $k$-means algorithm with sizes $N_C \in [5,000\ ;\ 50,000]$. Our method is compared to the classical BoW and a BoW using only random samples of SIFT descriptors from the queries on three well-known datasets:

**The Paris dataset** [16] is composed of $6,412$ images of Paris landmarks collected from Flickr. Retrieval performance is evaluated using $55$ queries provided along with their ground-truth. Query images contain $1,256$ keypoints on average.

**The Oxford dataset** [19] contains $5,062$ images of Oxford landmarks collected from Flickr. Retrieval performance is evaluated using $55$ queries provided along with their ground-truth. Query images contain $1,329$ keypoints on average.

**The Holidays dataset** [20] is composed of $1,491$ images of personal holidays photos. Query images are not only buildings images. Retrieval performance is evaluated using $500$ queries provided along with a their ground-truth. Query images contain $1,422$ keypoints on average.

### 5.2. Block building efficiency

As the number of SIFT keypoints per image is limited (typically $< 2,000$ on average), the running time of $k$-means for block computation is acceptable ($< 1$ second). Table 5.2 reports running times of $k$-means for various values of $k$ (i.e. $N_{eB}$) and various numbers of descriptors (denoted $N_D$) with a desktop computer (Intel core 2 CPU 2.66 GHz $\times$ 2, 2 GiB RAM).

| $N_D$ | 500 | 1,000 | 2,000 |
|---|---|---|---|
| $\alpha = 0.3$ | 0.05 s. | 0.20 s. | 0.58 s. |
| $\alpha = 0.5$ | 0.07 s. | 0.21 s. | 0.77 s. |
| $\alpha = 0.7$ | 0.08 s. | 0.26 s. | 0.87 s. |

**Table 1**. Running time (in seconds) of *k-means* with $N_{eB} = \alpha \times N_D$.

### 5.3. Choice of $\alpha$, $\kappa$ and vocabulary size

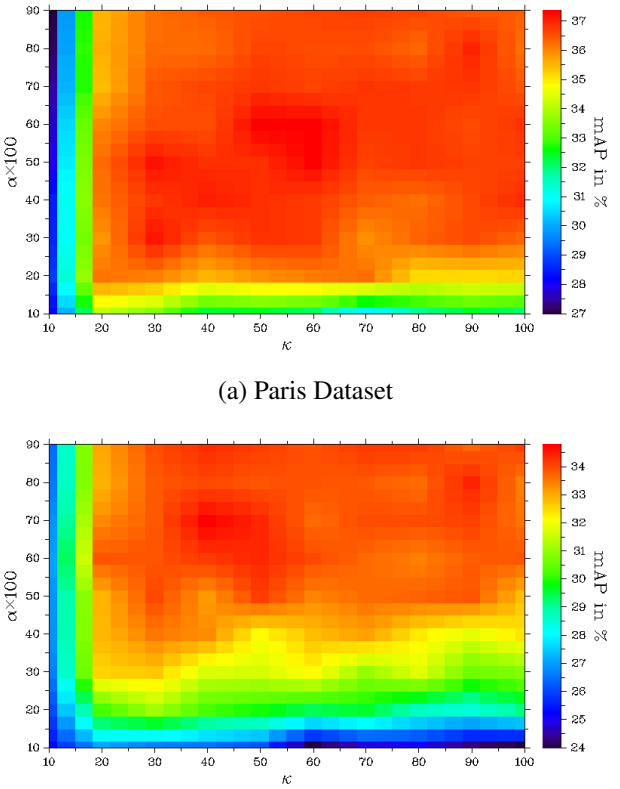
The effect of parameter $\kappa$ is shown in Figures 5(a) (Paris dataset) and 5(b) (Oxford dataset). The mAP (denoted by a color) is plotted as a function of $\kappa$ and $\alpha$. When $\kappa$ decreases (from 100 to 20), i.e. when the assignment gets "softer" (100 being the "hardest", i.e. similar to assigning to the closest visual word only), less $eB_i$ are needed to describe images. It confirms the need to assign the $eB_i$ to multiple $C_i$ as shown in Section 4. For $N_C = 5,000$, one can see that the best $\kappa$ is the same ($\kappa = 50$) as for $\alpha = 0.6$. A series of experiments (not reported here due to limited space) shows that the optimal value of $\kappa$ depends on the vocabulary size $N_C$. In the remainder of this paper, only results using the best settings of $\kappa$ are reported ($\kappa = 50$ for $N_C = 5,000$, $\kappa = 50$ for $N_C = 10,000$, $\kappa = 40$ for $N_C = 20,000$, and $\kappa = 20$ for $N_C = 50,000$).

| $N_C$ | 5,000 | | | 10,000 | | | 20,000 | | | 50,000 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Database | Paris | Oxford | Holidays | Paris | Oxford | Holidays | Paris | Oxford | Holidays | Paris | Oxford | Holidays |
| BoW (Baseline) | 36.83 | 33.48 | **48.41** | 36.18 | 33.33 | **44.85** | 40.53 | 35.85 | 44.39 | 50.40 | 42.89 | **60.10** |
| Random 30% | 33.74 | 29.71 | 42.78 | 34.98 | 32.43 | 43.64 | 39.26 | 33.33 | 43.11 | 44.17 | 36.39 | 53.34 |
| Random 50% | 35.60 | 31.78 | 44.26 | **36.31** | 32.79 | 44.25 | 39.89 | 35.11 | 43.71 | 47.91 | 40.14 | 57.28 |
| Random 70% | 36.59 | 32.21 | 46.93 | 35.93 | 33.12 | 44.50 | **40.53** | 35.79 | **44.84** | 49.16 | 40.42 | 59.40 |
| Ours ($\alpha = 0.3$) | **37.20** | 31.25 | 43.16 | **37.88** | **34.49** | 43.93 | **41.70** | **36.27** | 41.76 | 46.86 | 36.75 | 49.99 |
| Ours ($\alpha = 0.5$) | **37.25** | **33.69** | 46.12 | **37.52** | **33.98** | 44.76 | **41.60** | **36.65** | 43.31 | 49.82 | 40.52 | 55.83 |
| Ours ($\alpha = 0.7$) | **37.16** | **33.97** | 46.95 | **36.79** | **33.78** | 44.30 | **41.43** | 35.77 | 43.74 | **50.66** | **42.90** | 58.99 |

**Table 2**. mAP (in %) on Paris, Oxford and Holidays datasets for BoW, BoW with randomly sampled descriptors, and our method using different values of $N_{eB}$, $N_C$ (results above the BoW baseline are shown in **bold**).

### 5.4. Results

Table 2 reports the mAP obtained by our method and the two baselines (regular BoW and BoW with random sampling) using increasing values of $\alpha$ and $N_C$. It shows that setting $\kappa \leq 50$ and $N_C \leq 20,000$ allows our method to achieve results comparable to BoW with $\alpha = 0.3$, i.e. using only one third of the initial descriptors, on two of the datasets. This allows to significantly reduce transmission times without decreasing retrieval effectiveness. Our method performs better than BoW with a random sampling of descriptors, which yields almost always worse results than regular BoW. On the Holidays dataset, our method's performance does not match the performance of regular BoW. Further analysis of the results showed that, contrary to queries from the Paris and Oxford datasets, query images from Holidays require various, query-specific, numbers of $eB_i$ to compete with BoW. Therefore, setting $\alpha$ automatically according to the query would more more suited than using a single value for all queries as done here. Finally, for large-sized dictionaries, more $eB_i$ ($\alpha = 0.7$) are required to improve the BoW results for Paris and Oxford datasets.




(a) Paris Dataset

(b) Oxford Dataset

**Fig. 5**. mAP on (a) Paris and (b) Oxford datasets using SIFT, $N_C = 5,000$ and several values of $\alpha$ and $\kappa$ (best seen in color).

## 6. CONCLUSION

This paper proposes an original method to reduce the amount of data sent to the server to perform building image retrieval from mobile devices. Experiments on Paris and Oxford datasets show that using elementary blocks allows to reduce significantly the data size with no or little loss of effectiveness. This general framework can be used in conjunction with other descriptors such as *cHoG* [3], and other indexing methods such as VLAD [10], hashing [7], etc.

Further work include developing automatic methods to determine the optimal number of elementary blocks to be used for each query. This could be done using techniques such as the *Silhouette* [21]; however, such methods usually have a high computational cost that could be detrimental in our context. The estimation of the visual word histogram could also be improved using different weighting functions like the ones proposed in [22]. An automatic way to set the parameter $\kappa$ can also be explored, for example by considering the distribution of the descriptors used to create the blocks.

## 7. ACKNOWLEDGEMENT

This work is supported by the TWIRL (ITEA2 10029 - Twinning Virtual World On-line Information with Real-World Data Sources) project.

## 8. REFERENCES


[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999, pp. 1150–1157.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *CVIU*, vol. 110, no. 3, pp. 346–359, 2008.

[3] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: Compressed histogram of gradients a low bit-rate feature descriptor," in *CVPR*, 2009, pp. 2504–2511.

[4] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE PAMI*, vol. 19, pp. 530–535, 1997.

[5] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, 2003, vol. 2, pp. 1470–1477.

[6] T. Chen and K.-H. Yap, "A discriminative bow framework for mobile landmark recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, 2013.

[7] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *CVPR*, 2008.

[8] H. Jegou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *CVPR*, 2009, pp. 1169–1176.

[9] H. Jegou, M. Douze, and C. Schmid, "Product Quantization for Nearest Neighbor Search," *IEEE PAMI*, vol. 33, no. 1, pp. 117–128, Jan. 2011, QUAERO.

[10] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010, pp. 3304–3311.

[11] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *ECCV*, Berlin, Heidelberg, 2010, pp. 143–156, Springer-Verlag.

[12] B. Girod, V. Chandrasekhar, R. Grzeszczuk, and Y.A. Reznik, "Mobile visual search: Architectures, technologies, and the emerging MPEG standard," *IEEE MultiMedia*, vol. 18, no. 3, pp. 86–94, 2011.

[13] P. Tirilly, V. Claveau, and P. Gros, "Language modeling for bag-of-visual words image categorization," in *Proceedings of the International Conference on Content-based Image and Video Retrieval*, New York, NY, USA, 2008, CIVR '08, pp. 249–258, ACM.

[14] D. Awad, V. Courboulay, and A. Revel, "Saliency filtering of SIFT detectors: Application to CBIR," in *Advanced Concepts for Intelligent Vision Systems*, vol. 7517 of *LNCS*, pp. 290–300. Springer Berlin Heidelberg, 2012.

[15] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, 2nd edition edition, Nov. 2001.

[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008, pp. 1 – 8.

[17] B. Girod, V. Chandrasekhar, D. M. Chen, N.-M. Cheung, R. Grzeszczuk, Y. A. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham, "Mobile visual search.," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 61–76, 2011.

[18] J. Chao, A. Al-Nuaimi, G. Schroth, and E. Steinbach, "Performance comparison of various feature detector-descriptor combinations for content-based image retrieval with jpeg-encoded query images," in *IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, 2013, pp. 029–034.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.

[20] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, Berlin, Heidelberg, 2008, pp. 304–317, Springer-Verlag.

[21] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 0, pp. 53 – 65, 1987.

[22] J. C. Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders, "Kernel codebooks for scene categorization," in *ECCV*, Berlin, Heidelberg, 2008, pp. 696–709, Springer-Verlag.