On the burstiness of visual elements

Hervé Jégou, Matthijs Douze, Cordelia Schmid

On the burstiness of visual elements

Hervé Jégou Matthijs Douze Cordelia Schmid
INRIA Grenoble, LJK
firstname.lastname@inria.fr

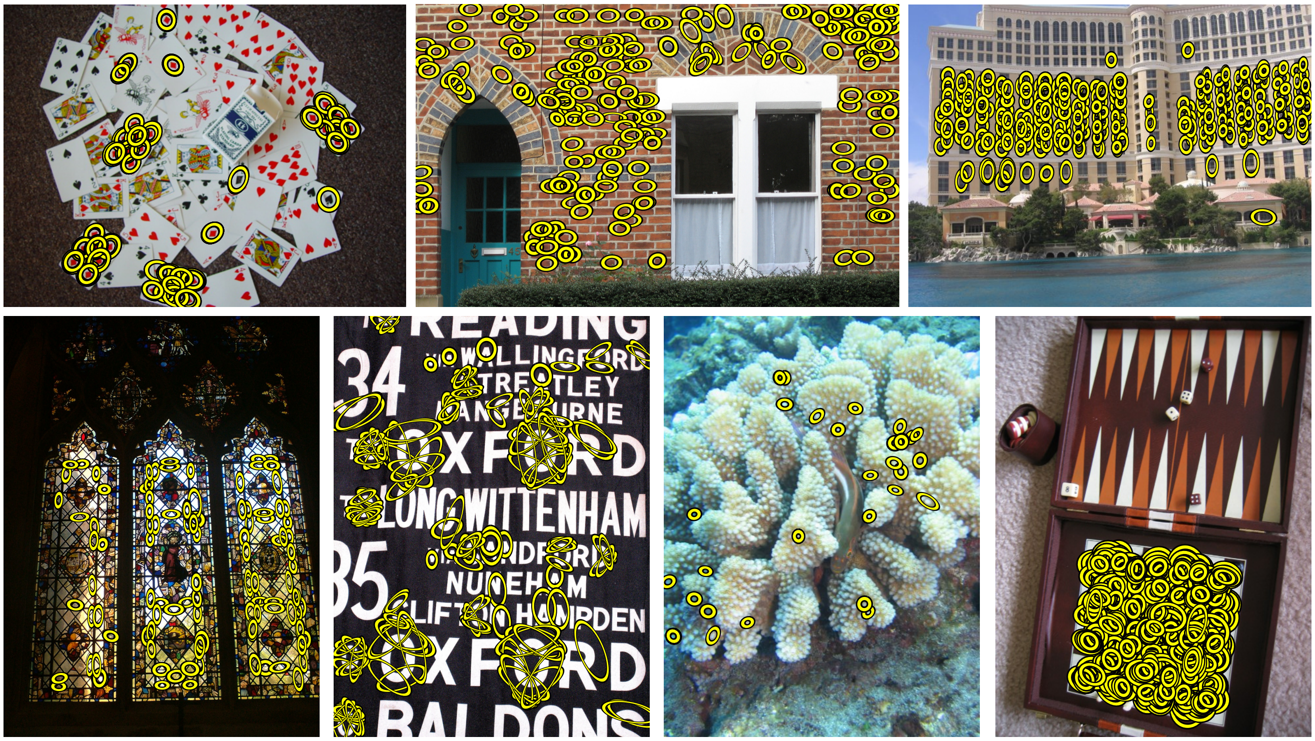

Figure 1. Illustration of burstiness. Features assigned to the most "bursty" visual word of each image are displayed.

Abstract

Burstiness, a phenomenon initially observed in text retrieval, is the property that a given visual element appears more times in an image than a statistically independent model would predict. In the context of image search, burstiness corrupts the visual similarity measure, i.e., the scores used to rank the images. In this paper, we propose a strategy to handle visual bursts for bag-of-features based image search systems. Experimental results on three reference datasets show that our method significantly and consistently outperforms the state of the art.

1. Introduction

Image search has received increasing interest in recent years. Most of the state-of-the-art approaches [1, 5, 14, 16] build upon the seminal paper by Sivic and Zisserman [20]. The idea is to describe an image by a bag-of-features (BOF) representation, in the spirit of the bag-of-words representation used in text retrieval.

This representation is obtained by first computing local descriptors, such as SIFT [9], for regions of interest extracted with an invariant detector [13]. A codebook is then constructed offline by unsupervised clustering, typically a k-means algorithm [20]. Several other construction methods, such as hierarchical k-means [14] or approximate k-means [15], have been used for efficiency. The resulting codebook is usually referred to as a *visual vocabulary*, and the centroids as *visual words*. The BOF representation is obtained by quantizing the local descriptors into the visual vocabulary, resulting in frequency vectors. This representation can be refined with a binary signature per visual word and partial geometrical information [4].

Given that some visual words are more frequent than others, most of the existing approaches use an *inverse document frequency* (*idf*) word weighting scheme, similar to text retrieval [17]. It consists in computing an entropic weight per vocabulary word which depends on its probability across images [20]. Although *idf* weighting reduces the impact of frequent visual words, it has two limitations. First, it has been designed for a finite alphabet, not for a continuous feature space. Consequently, it cannot measure the quality of the matches based on the distances between descriptors. Second and most importantly, *idf* weighting does not take into account the *burstiness* of the visual elements: a (visual) word is more likely to appear in an image if it already appeared once in that image.

This paper is organized as follows. The burstiness phenomenon in images is presented in Section 2. In Section 3 we introduce our image search framework and in Section 4 we propose three strategies that take into account burstiness in the matching scores. The first one removes multiple matches that are counted in a BOF framework. Due to the burstiness of visual elements, such multiple matches often occur when matching two images based on their visual words, see Fig. 2. The second and third strategies are more sophisticated reducing the scores of intra- and inter-images bursts, respectively. Finally, in the experimental section 5 we report our results for three reference datasets. They significantly and consistently outperform state-of-the-art methods.

2. The burstiness phenomenon

In the bag-of-words framework [7] terms are assumed to be conditionally independent. The overall frequency of a term is the main information. The central problem with this assumption is that words tend to appear in bursts [2, 6], as opposed to being emitted independently, i.e., if a word appears once, it is more likely to appear again. For instance, this article features a burst of 36 instances of the rare term "burstiness" ! Church and Gale [2] model burstiness by representing a term's distribution pattern with a Poisson distribution. Katz [6] models the within-document burstiness using K-mixtures. Burstiness has recently been shown to improve performance in the context of text classification [10] and text clustering [3].

Here, we show that burstiness translates to images, see Fig. 1. For each of the example images, we display features assigned the most frequent visual word. One can observe

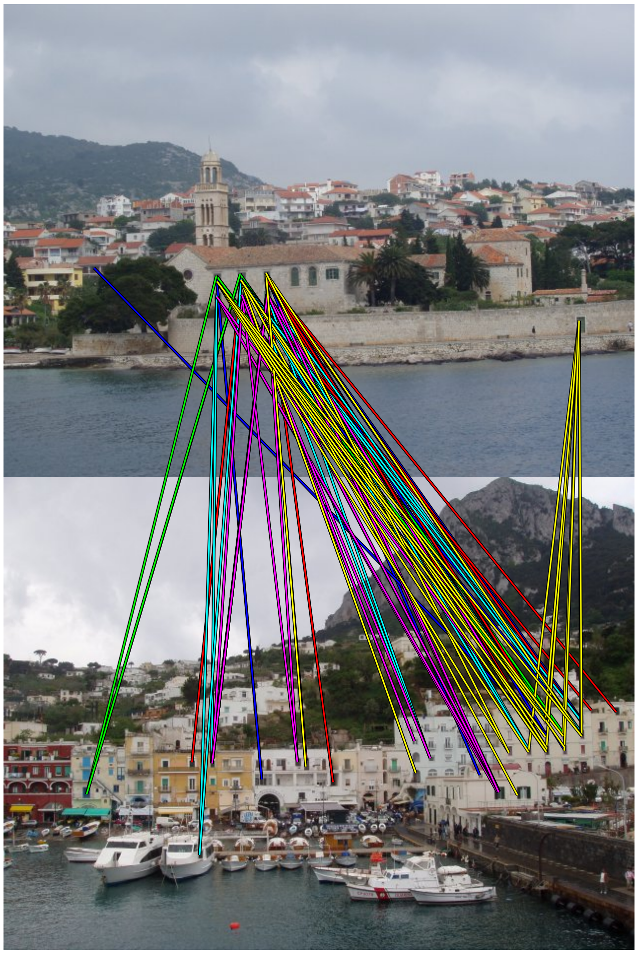

Figure 2. Example of non-corresponding images with multiple point matches. Lines of the same color represent matches based on the same visual word.

that many regions (detected by the Hessian-affine detector) are assigned to the same visual word, the maximum being 445 regions assigned to a single visual word on the "cards" image. The examples include man-made objects such as buildings, church windows and playing cards as well as textures such as a brick wall and corals. In both cases the repetitiveness stems from the scene property, for example the windows of the buildings are very similar and the bricks are repeated. More surprising is the burstiness of text. Here, the main burst appears on a finer scale than that of the entire letter, for example at the extremities of the O, R, D and S letters, which share similar parts.

Fig. 3 presents a quantitative measure of burstiness. It shows the probability that a given word occurs in a document exactly x times in a real image database. It has been measured on one million images. This empirical distribution is compared with a simulated distribution. To produce this synthetic curve, we use the same number of descriptors and visual word probabilities as in our one-million-image

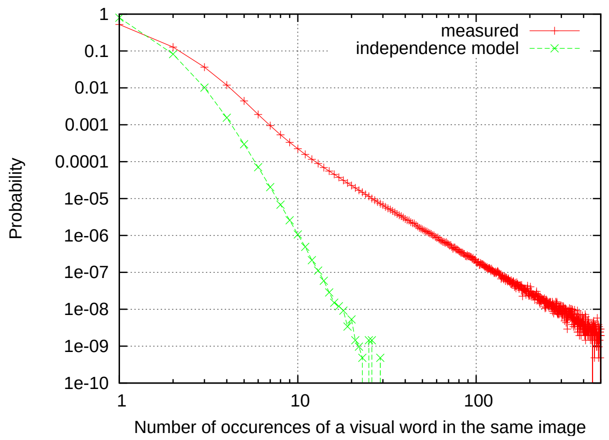

Figure 3. Probability distribution of the number of occurrences of a visual word in an image.

database, assuming they are drawn independently. The difference between the synthetic and the observed distributions clearly shows the extent of the burstiness phenomenon.

In what follows, we distinguish between 1) the *intra-image burstiness*, which usually appears due to repetitive patterns, i.e., when the same visual element appears several times in the same image, and 2) the *inter-image burstiness*, which corresponds to visual elements that appear in many images. Intra-image burstiness is related to the self-similarity property used in [19] and obviously appears on near-regular textures [8]. Feature self-similarity was used in [18] to discard elements occuring more than 5 times in the same image using intra-image indexing, which is a simple way of handling bursts. Usually unbalanced visual word frequencies are addressed by applying *idf* weights. However, the *idf* weights do not take into account the burstiness phenomenon. Moreover, they do not reflect the strength of the matches, that can for example be obtained from the distance measures between SIFT descriptors. Therefore, the burstiness of visual elements cannot be handled by simply translating the models introduced in text, where the underlying alphabet is discrete. In Section 4, the inter- and intra-burstiness phenomena will be addressed independently.

3. Image search framework

Our baseline system builds upon the BOF image querying method [20] and recent extensions [4, 15]. In the following we briefly describe the steps used in this paper.

Local descriptors and assignment. We extract image regions with the Hessian-affine detector [13] and compute SIFT descriptors [9] for these regions. To obtain a bag-of-features representation for an image, we learn a 20k visual vocabulary and assign the descriptors to the closest visual word (Euclidean distance). The visual vocabulary is obtained by k-means clustering performed on an independent dataset of Flickr images. Such a nearest-neighbor quantizer, which assigns an index $q(x)$ to a descriptor x, implicitly divides the feature space into *cells*, i.e., the regions of a a Voronoï diagram corresponding to the space partitioning.

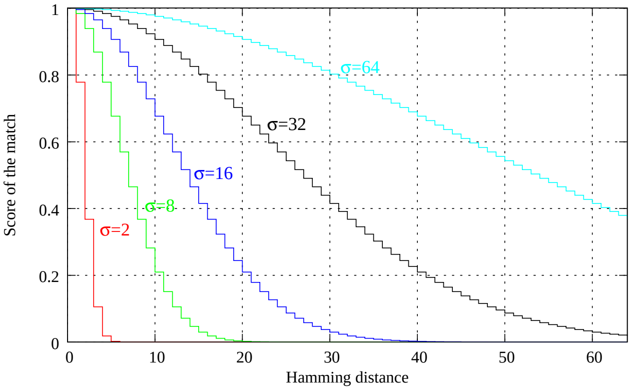

Figure 4. The match score as a function of the Hamming distance.

Hamming Embedding (HE). HE provides a more precise representation of the descriptors than the quantized index [4], i.e., it adds a compact binary representation. This representation subdivides each cell associated with a given visual word into regions. Associating a binary signature $s(x)$ with a descriptor x refines the descriptor matching, as two descriptors x and y match if they are assigned to the same visual word, i.e., if $q(x) = q(y)$, *and* if the Hamming distance $h(s(x), s(y))$ between their binary signatures is lower or equal than a threshold h_t. We set the signature length to 64 and $h_t = 24$, as done in [4].

Weighting based on HE. In [4], the Hamming distance results in a binary decision, i.e., two descriptors match or not. However, the distance reflects the closeness of descriptors and should be taken into account. Since we have higher confidence in smaller distances, we weight them with a higher score. The fundamental difference between this weighting scheme and the soft assignment of [16] is that their weighting depends on the distance between the query descriptor and the visual word centroid, whereas our method uses a distance between the binary signatures, which reflects the distance between the descriptors.

The weight $w(h_d)$ associated with a Hamming distance $h_d = h(s(x), s(y))$ between binary signatures $s(x)$ and $s(y)$ is obtained with a Gaussian function:

$$w(h_d) = \exp\left(\frac{-h_d^2}{\sigma^2}\right). \quad (1)$$

Figure 4 shows the weighting functions obtained for different values of σ. In the following, we set $\sigma = 16$. Note that 16 is not the optimal value for a particular dataset, but is a good choice given that distances above $h_t = 24$ are not significant. For efficiency, we keep a threshold h_t above which the matching score is set to 0. The matching score is finally multiplied by the square[1] of the *idf* factor $\mathrm{idf}(x)$ associated with the visual word $q(x)$. This reduces the impact of the visual words that are more frequent over the entire database. In summary, the matching score between two descriptors x and y is given by

$$\mathrm{score}(x,y) = \begin{cases} w(h(s(x),s(y))) & \text{if } q(x)=q(y) \\ \quad \times \mathrm{idf}(q(x))^2 & \quad \text{and } h(s(x),s(y)) \leq h_t \\ & \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Weak Geometric Constraints (WGC). WGC uses partial geometric information for all images, even on a very large scale [4]. It is a simple verification that checks for consistency of the rotation and scale hypotheses obtained from matching point pairs. Furthermore, we use priors to favor "natural" geometrical transformations between matching points.

Weighting based on WGC. We have observed that points detected at larger scales tend to produce more reliable matches. Therefore, the characteristic scales of the points are used to weight the scores obtained with our HE weighting scheme. This gives a modest improvement in precision at almost no computational cost: 0.005 on the mAP for the Holidays dataset described in Section 5.

Score normalization: We use L2 normalization, as in [4]. The overall score of a database image with respect to the query image is obtained as the sum of the individual matching scores of (2) divided by the L2 norm of the histogram of visual occurrences.

Variant: multiple assignment (MA). As a variant, a given query descriptor is assigned to several visual words instead of only to one, i.e., to the k nearest neighbors. This gives an improvement when a noisy version of the descriptor is not assigned to the same visual word as the original descriptor. This is in the spirit of the multiple assignment of [16], but does not use soft weighting. Instead, we use the weights obtained from the Hamming distance as in the case of a single assignment (SA). In contrast to the method of [5], MA is performed on the query side only. The memory usage is, therefore, unchanged. The complexity is higher than for the standard SA method but lower than for symmetric multiple/soft assignment strategies.

We also require that the distance d of a candidate centroid satisfies $d < \alpha d_0$, where d_0 is the distance to the nearest neighbor. This avoids assignments to irrelevant centroids when there is one clear nearest visual word. For our experiments, we set $\alpha = 1.2$. On average, a descriptor is assigned to 4.3 visual words for a vocabulary size of 20000 and $k = 10$.

Spatial verification (SP). Given a set of matching descriptors, we use a robust estimation procedure to find a subset of matches consistent with a 2D affine transformation [4, 9, 16]. Since this procedure is costly, we only apply it to re-rank the 200 best results returned by our system. Because the estimation may fail to match some relevant images, we append the remaining images to the list of SP-verified ones.

4. Burstiness management strategy

In this section, we propose three strategies that penalize the scores associated with bursts. The first one removes multiple matches that are intrinsically counted in a BOF framework. The second and third approaches are more sophisticated strategies that reduce the impact of intra- and inter-images bursts.

4.1. Removing multiple matches

The cosine measure between two bag-of-features is equivalent to a voting score [4]. Given this interpretation we can see that multiple matches are not disambiguated in a BOF comparison: a single descriptor can "vote" several times for one image of the database. This phenomenon clearly appears in Fig. 2, where the descriptors from the top image are matched with many of the descriptors from the bottom image, dramatically altering the quality of the comparison.

The multiple match removal (MMR) strategy proposed in this subsection addresses this problem by removing multiple matches. It is similar to [12], where each point votes only once for an image in the database, i.e., for each database image only the best match associated with a query descriptor is kept: a descriptor cannot vote several times for the same database image. MMR is performed on-the-fly when querying the inverted file, which makes it tractable for large databases.

The best match for a given database image is the one maximizing (2), i.e., the one corresponding to the smallest Hamming distance between binary signatures. All the other matches associated with this particular query descriptor are discarded. In case of a tie, we arbitrarily choose the first descriptor. We measured that, on average, about 13% of the descriptor matches are discarded with this approach. Note that this selection only marginally increases the complexity of the voting scheme.

4.2. Intra-image burstiness

As we will show in the experimental section, the previous strategy improves the results. However, the penalization applied to sets of matches is too strong compared with unique matches. Hereafter, we improve this strategy.

[1] Squaring the idf factor is not an arbitrary choice: it is consistent with the computation of the L2 distance between BOF. See [4] for details.

Let x_i be the i^{th} descriptor of the query image and $y_{b,j}$ be the j^{th} descriptor of the database image b. In case of multiple assignment we assume that we have distinct descriptors. The matching score between x_i and $y_{b,j}$ is denoted by $m(i, b, j)$. This score is 0 if x_i and $y_{b,j}$ are not assigned to the same visual word or if the Hamming distance between the corresponding binary signatures is above the Hamming threshold h_t. The score of a query descriptor i for image b is obtained as a sum of scores over matching descriptors:

$$t_{\mathrm{q}}(i, b) = \sum_{j / q(y_{b,j}) = q(x_i)} m(i, b, j). \tag{3}$$

The score of a match is then updated as

$$m(i, b, j) := m(i, b, j) \sqrt{\frac{m(i, b, j)}{t_{\mathrm{q}}(i, b)}}. \tag{4}$$

If a query descriptor is matched to a single descriptor in the database image, the strength of the match is unchanged. Otherwise, its score is reduced. Inversely, if there are several query descriptors assigned to the same visual word (a burst), their scores are penalized. The choice of (4) is motivated in the experimental section, where we show results for different update functions. This strategy can be used even if the scores $m(i, b, j)$ are binary, that is even if no HE weighting scheme is applied.

4.3. Inter-image burstiness

The two previous methods address the bursts within a image. However, some visual elements are also frequent *across* images. This problem is usually addressed by using *idf* weights. Although this strategy is useful, it is a pure text retrieval approach that only takes into account the number of descriptors associated with a given visual word in the database, without exploiting the quality of the matches, i.e., the closeness of the descriptors in feature space. Therefore, it cannot exploit the scores provided by HE distances or any similarity measure between descriptors.

The strategy proposed hereafter can be seen as an extension of *idf* weighting that takes into account these measures. We first define the total $t_{\mathrm{b}}(i)$ of the matching scores of a given query descriptor for all the database images as

$$t_{\mathrm{b}}(i) = \sum_b \sum_j m(i, b, j). \tag{5}$$

The matching scores are updated using the same weighting function as in (4):

$$m(i, b, j) := m(i, b, j) \sqrt{\frac{m(i, b, j)}{t_{\mathrm{b}}(i)}}. \tag{6}$$

This update penalizes the descriptors that vote for many images in the database. By contrast to pure *idf* weighting, the penalization is computed on-the-fly to take into account the particular amount of votes received by a given query descriptor. This is more precise than assigning a weight at the visual word level only.

Dataset	# images	# queries	# descriptors
Kentucky	10,200	10,200	19.4 M
Oxford	5,063	55	15.9 M
Holidays	1,491	500	4.4 M
Distractors	1,000,000	N/A	2.1 G

Table 1. Characteristics of the datasets used in our experiments.

5. Experiments

5.1. Datasets and evaluation

We present results for three reference datasets used in state-of-the-art papers to evaluate image search systems. All of them are available online. The characteristics of these datasets are summarized in table 1.

The Kentucky object recognition benchmark[2] depicts 2550 objects. Each object is represented by 4 images taken under 4 different viewpoints. The viewpoint changes are so significant that matching the images geometrically requires wide baseline stereo techniques. However, there are neither significant occlusions nor changes in scale. Each image of the database is used as a query. The correct results are the image itself and the three other images of the same object. Since many distinct objects are taken in front of the same background, the algorithm should be robust to clutter.

The Oxford building dataset[3] contains photos from Flickr that were tagged with keywords related to Oxford. Each query is a rectangular region delimiting the building in the image. The correct results for a query are the other images of this building. The dataset is challenging because of cropped and cluttered images, changes in imaging conditions (different seasons, cameras, viewpoints, etc), and image quality. The database contains only 55 query images, which may result in noisy performance measures. It has a bias towards building images and repeating scenes (some buildings are represented more than 200 times).

The INRIA Holidays dataset[4] is divided in small groups of photos of the same object or scene. The first image of the group is the query, the others are the images which are relevant for this query. There are viewpoint changes, occlusions, photometric changes, blur, in-plane rotations, etc. There is a bias on the image properties, as most groups have been shot with the same camera and on the same day.

[2] http://vis.uky.edu/%7Estewe/ukbench/

[3] http://www.robots.ox.ac.uk/˜vgg/data.html

[4] http://lear.inrialpes.fr/˜jegou/data.php

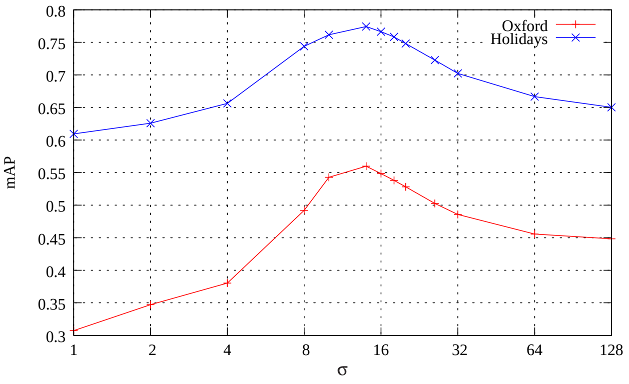

Figure 5. Impact of the weighting parameter σ on the accuracy.

For large-scale experiments, we use the approach of [15] and [4], i.e., the Oxford and Holidays datasets are merged with a distractor set of up to one million random images downloaded from Flickr. We assume that the distractor set contains only irrelevant results for all queries (which occasionally proves wrong because both the Oxford and Holidays datasets contain photos of landmarks that may appear in random photo collections).

Evaluation measures. For all experiments we measure the mean average precision (mAP) of the search, as defined in [15]. For the Kentucky dataset, we also report the average number N_s of TPs retrieved in the 4 first results, as this is the performance measure usually reported for this dataset.

5.2. Implementation details

We handle information at the interest point level to evaluate our matches. This information is available while scanning the inverted file, since we store one entry per database descriptor, including the binary signature and the geometrical quantities used by WGC. To handle the inter-image burstiness, we store all the relevant descriptor matches obtained while scanning the inverted file. Each descriptor match generates an 11-byte structure containing the index of the query descriptor, the index of the database image, WGC-related information and a matching score.

During the inverted file scan, the structures are stored in an array lexicographically ordered by (i, b, j) (indexes are defined in Section 4). Due to the filtering of matches by the HE check, this array (1100 MB on average for one million images and MA) is an order of magnitude smaller than the inverted file itself (24 GB). The inter-image burst weighting stages are applied per database image, so the array must be "transposed" to be in (b, i, j)-order. This transposition is performed in a cache-efficient way using a blocked algorithm. The average query time (for description, quantization, search and normalization, but without SP) in a one-million-image set is 6.2 s. This is a bit slower than the timings reported in [4] for the same vocabulary size.

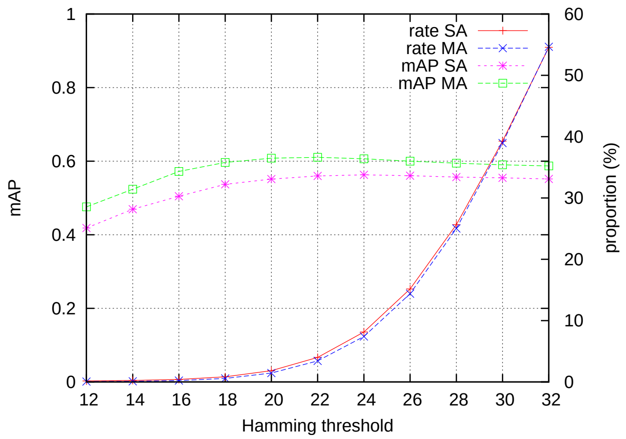

Figure 6. Proportion of descriptors kept and corresponding mAP as a function of the Hamming threshold test when searching the Oxford dataset. Results on the Holidays dataset are similar.

5.3. Impact of the parameters

For all our experiments, we have mostly followed our previous experimental setup [4]. The software from [11] was used with default parameters to extract the Hessian-Affine regions and compute the SIFT descriptors. For the Kentucky dataset, we adjusted the detector threshold in order to obtain the same number of descriptors as in [5]. The visual vocabulary was learned on an independent dataset downloaded from Flickr. We used k-mean for clustering and 20k visual words in all our experiments.

HE Weighting. The impact of the parameter σ introduced in (1) is shown in Fig. 5. One can see that the mAP scores are almost equivalent for a large range of values (from $\sigma =$ 10 to $\sigma = 20$). This behavior is consistent over all datasets. We choose to set $\sigma = 16$ in the following.

Fig. 6 shows that for very low Hamming thresholds, i.e., if keeping only 0.1% of the points, we still get excellent results. There is an optimal threshold of 24 for the standard single assignment method (SA) and 22 for multiple assignment (MA). The mAP decreases slightly for higher values, because too many noisy descriptors are introduced (especially with MA). We choose a threshold of 24 for all experiments, which removes 93% of the matches.

Burst weighting functions. Table 2 compares several functions inspired by text processing techniques [17], here applied to handle intra-image burstiness. These functions correspond to the right term in (4). Normalizing directly by the number of occurrences of the visual word (Function #2) improves the score, but this normalization is too hard. It is advantageously weakened by a square root (#3). It is also beneficial to take into account the matching scores at this stage, as done in (4), where the normalizer is the square

		Oxford		Holidays	
	function	SA	MA	SA	MA
#1	None	0.563	0.606	0.768	0.815
#2	$\frac{1}{N(i,b)}$	0.579	0.624	0.788	0.816
#3	$\frac{1}{\sqrt{N(i,b)}}$	0.582	0.626	0.793	0.824
#4	$\sqrt{\frac{m(i,b,j)}{t_\mathrm{q}(i,b)}}$	0.581	0.625	0.790	0.826
#5	$\log(1 + \frac{m(i,b,j)}{t_\mathrm{q}(i,b)})$	0.582	0.627	0.792	0.820

Table 2. Comparison of intra-image burst normalization functions in terms of mAP on two datasets. In addition to the notations of Section 4, $N(i,b)$ denotes the number of occurrences of the visual word $q(x_i)$ in the image b .

root of the score divided by the sum of scores obtained by this visual word (#4). Replacing the square root by a log (#5) leads to similar results. Overall, the three normalization function #3, #4 and #5 give equivalent results.

5.4. Comparison with the state-of-the-art

Baseline: Table 3 shows the improvements due to our contributions. In similar setups, our baseline compares favorably with the algorithms of [4, 5, 14, 16]. Results for the baseline BOF representation are reported for reference.

HE weighting scheme and MA: The combination of the Hamming distance weighting scheme with multiple assignment provides a significant improvement of 0.06 in mAP on the Oxford and Holidays datasets. The mAP of 0.606 that we obtain using this combination is significantly better than the score of 0.493 reported in [16] for a similar setup, i.e. without spatial verification or query expansion.

Burstiness: For each database image, the descriptor matching scores are updated according to Section 4. The MMR approach, that removes multiple matches, is shown to be of interest, but performs poorly when combined with the other methods. Table 3, both the intra- and inter-burstiness methods are shown to significantly improve the results.

In the following we provide a comparison on each reference dataset with the best results reported in the literature in a similar setup, i.e., without spatial verification or query expansion.

- *Kentucky*: we obtain $N_s = 3.54$. This result is slightly below the score of 3.60 obtained in [5] by using the *contextual dissimilarity measure*, but only the non-iterative version of this approach can be used on a large scale. Our method compares favorably with this scalable approximation, for which $N_s = 3.40$.

- *Oxford:* we obtain mAP=0.647, which is significantly better than the score of 0.493 reported in [16].

- *Holidays:* our mAP of 0.839 significantly outperforms the mAP of 0.751 reported in [4].

The spatial verification takes a shortlist of the 200 best results, here obtained with our burstiness management strategy, and refines the results by robustly estimating an affine 2D model. The verification strongly improves the results for the Oxford dataset, which contains a lot of planar and geometrical elements that are suitable for an affine 2D model. Improvements on the two other databases are more modest.

Combination with distractors. The curves in Fig. 7 show the results for the distractor dataset Flickr1M combined with Oxford and Holidays. All our proposed methods improve the performance. On Holidays, the improvement is higher for large databases, and the performance of our best method decreases very slowly when the database grows. Before SP, the accuracy obtained on the one-million-image database is better than our previous result [4] on the Holidays dataset alone. On Oxford combined with 100,000 images, we obtain a better mAP value (0.628) than the query expansion[5] method of Chum et al. [1, 16].

6. Conclusion

In this paper, we have shown the burstiness phenomenon of visual elements and proposed a strategy to address this problem in the context of image search. The resulting image search system significantly improves over the state-of-the-art on the three different reference datasets.

Acknowledgements

We would like to thank the ANR project RAFFUT and the QUAERO project for their financial support.

References

[1] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007.

[2] K. W. Church and W. A. Gale. Poisson mixtures. *Natural Language Engineering*, 1:163–190, 1995.

[3] Q. He, K. Chang, and E.-P. Lim. Using burstiness to improve clustering of topics in news streams. In *Proceedings of the IEEE International Conference on Data Mining*, 2007.

[4] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, Oct 2008.

[5] H. Jégou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *CVPR*, 2007.

[5] Query expansion is suitable for databases containing many images of the queried object. We implemented a version of the AQE query expansion [1] and obtained a mAP score of 0.747 (respectively, 0.692) on Oxford (resp., Oxford+100K), an improvement over the results reported in [16].

						Kentucky				Oxford		Holidays	
						mAP		N_s		mAP		mAP	
Method	HE weights	MMR	intra	inter	SP	SA	MA	SA	MA	SA	MA	SA	MA
BOF						0.780	0.701	2.99	2.68	0.338	0.260	0.469	0.313
BOF			x			0.831	0.663	3.23	2.54	0.349	0.193	0.500	0.294
HE+WGC						0.867	0.888	3.36	3.45	0.542	0.585	0.751	0.790
HE+WGC	x					0.874	0.894	3.39	3.47	0.563	0.606	0.768	0.815
HE+WGC	x	x				0.884	0.900	3.43	3.50	0.580	0.630	0.780	0.817
HE+WGC	x		x			0.889	0.904	3.46	3.52	0.581	0.625	0.790	0.826
HE+WGC	x			x		0.885	0.902	3.44	3.51	0.586	0.635	0.786	0.828
HE+WGC	x		x	x		0.892	0.907	3.47	3.54	0.596	0.647	0.807	0.839
HE+WGC	x		x	x	x	0.926	0.930	3.62	3.64	0.654	0.685	0.845	0.848

Table 3. Search results with various methods on the three datasets. N_s=Kentucky specific score, mAP=mean average precision, SA=single assignment, MA=multiple assignment, SP=spatial verification, MMR=multiple match removal, intra and inter: see Section 4.

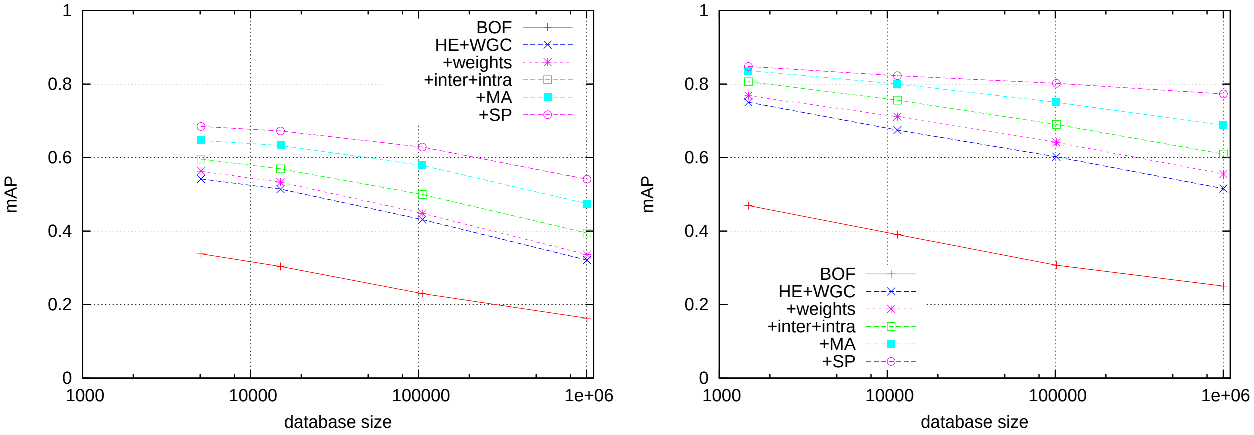

Figure 7. mAP for Oxford (left) and Holidays (right) combined with a varying number of distractor images.

[6] S. M. Katz. Distribution of content words and phrases in text and language modeling. *Natural Language Engineering*, 2:15–59, 1996.

[7] D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *ECML*, pages 4–15, 1998.

[8] W.-C. Lin and Y. Liu. A lattice-based MRF model for dynamic near-regular texture tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):777–792, 2007.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[10] R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the dirichlet distribution. In *ICML*, 2005.

[11] K. Mikolajczyk. Binaries for affine covariant region descriptors, 2007.

[12] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision*, volume 1, pages 525–531, 2001.

[13] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[14] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

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

[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.

[17] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[18] F. Schaffalitzky and A. Zisserman. Automated location matching in movies. *Computer Vision and Image Understanding*, 92:236–264, 2003.

[19] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, June 2007.

[20] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.