Democratic match kernels for image search
— Supplemental material: Appendices —

Hervé Jégou
Inria
Rennes, France

Andrew Zisserman
Dept. of Engineering Science
University of Oxford

Abstract

The appendices provided in this supplemental material complement our paper in several aspects. We provide additional experiments, results and interpretations. We give pseudo-code of our democratization strategy and show how the democratic kernel relates to square-rooting normalization (powerlaw with $\alpha = 0.5$) in the case of bag-of-words vectors without inverse document frequency terms. Then, we report additional results, in particular on the UKB benchmark and Holidays merged with 1 million images. Finally, we provide complexity measurements.

Appendix A – ROC curves and discussion

Figure 1 gives the receiver operating curves associated with Fisher and T-embedding. We consider the same setup as in recent papers for evaluation of learned local descriptors [7], which is also used in Section 3.3. We learn on Liberty and test on NotreDame. The baseline is RootSIFT. Here, we use this framework to evaluate the impact of the encoding technique on the hypothesis test. More precisely we compare local descriptors *individually encoded* by the Fisher vector (mean components, no power-law component-wise normalization) and our encoding technique $\phi_\triangle$. We also evaluate our strategy *without* removing the low frequency terms.

The first observation is that the Fisher vector is not as good as RootSIFT with respect to the hypothesis test, *i.e.*, when vectors are compared individually with cosine similarity. Observe, also, that the benefit of T-embedding is especially significant for low false positive rates, which explains why T-embedding is more suitable than RootSIFT to detect visual bursts based on a high similarity threshold. Despite its relative poor performance on these curves, recall that the Fisher vector offers better performance than RootSIFT once aggregated with a sum.

This leads us to the following interpretation: the main benefit of these embedding functions is not the underlying quality of the comparison metric (although this also matters), but mainly the fact that they limit the interferences between descriptors once aggregated, thanks to the mapping to a higher dimensional space. This is also supported by the comparison of T-embedding before and after removing the first components. While the ROC curve of T-embedding is better when keeping the low frequencies (and better than RootSIFT), the recognition performance for image search is better when we remove them.

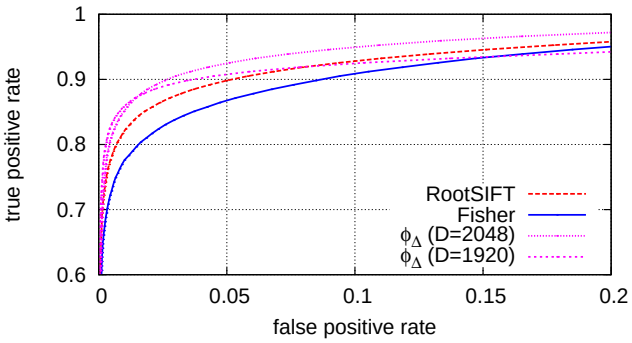

Figure 1. ROC curves: comparison of RootSIFT, Fisher (applied on RootSIFT, k=16), and T-embedding (k=16) before (D=2048) and after (D=1920) removing the first D components associated with the low frequencies.

Impact of removing the components associated with low frequencies. The previous interpretation is consistent with the fact that, before removing the low frequencies in our T-embedding, the variance of the cosine similarity for unrelated descriptors is comparable to that of RootSIFT after PCA. In this case, there is a lot of interferences between the aggregated descriptors.

After removing these components, the distribution of cosine similarities, for unrelated descriptors, is more comparable to that obtained with high-dimensional vectors distributed on the unit hypersphere: It is Gaussian-like with a small variance. These remarks are illustrated by Figure 2, which shows how the distributions of related and unrelated patches evolve when removing a varying number of components: 0 (=none removed), 64, 128, 192 and 256. Observe

that our removal strategy has a dramatic impact of the distributions of cosine similarities, in particular for the first d components, where d is the input (SIFT) dimensionality.

The strategy has also an effect on the true matches: the corresponding similarities decrease when removing components. Even though a significant proportion of the matches remain associated with strong similarities, the hypothesis test on individual descriptors is weakened. Our strategy therefore aims at optimizing a trade-off between

- reducing the interferences in the match kernel
- and keeping the hypothesis test as good as possible.

By removing more than d=128 components, the benefit of variance reduction is less significant for unrelated patches: compare the Figures associated with 128 and 256 removed components, which have comparable shapes for unrelated matches. In contrast, the similarity of true matches continue to suffer. In particular, more patches have a similarity close to 0. These observations are consistent with our preliminary experiments, where the optimal performance, with respect to search accuracy, is approximately obtained when removing d components.

As a final remark, recall that the embeddings we consider are learned in an unsupervised manner: The ROC curves would be certainly improved by using metric learning techniques such as those proposed for descriptor learning [6]. However, as discussed in this section, the ROC performance is not directly related to the image search performance because of the interferences. This shows the limit of the ROC evaluation setup in the context of match kernels.

Appendix B – Modified Sinkhorn: Pseudo-code

The algorithm below gives pseudo-code for our democratization strategy, solved by a symmetric variant of the Sinkhorn scaling algorithm. We set $\gamma = 0.3$, typically. Note however that using a value lower than $\gamma = 0.5$ (like in the regular Knight variant) is critical only if we re-compute the kernel matrix from the weighted descriptors at each iteration of the Sinkhorn algorithm.

Input: Gram matrix K % of size $n \times n$
parameters γ and $\mathrm{n_{iter}}$
Output: Weight vector $\boldsymbol{\lambda}$
Initialization: $\boldsymbol{\lambda} = \mathbf{1}_n$

For i=1 to $\mathrm{n_{iter}}$
$\boldsymbol{\sigma} = \mathrm{diag}(\boldsymbol{\lambda}) \times \mathsf{K} \times \mathrm{diag}(\boldsymbol{\lambda}) \times \mathbf{1}_n$ % Sums of rows
$\forall i,\ \lambda_i := \lambda_i / \sigma_i^\gamma$ % Update

Note, we provide a package associated with paper, which includes a Matlab implementation of this algorithm: `http://tinyurl.com/democratic-kernel`.

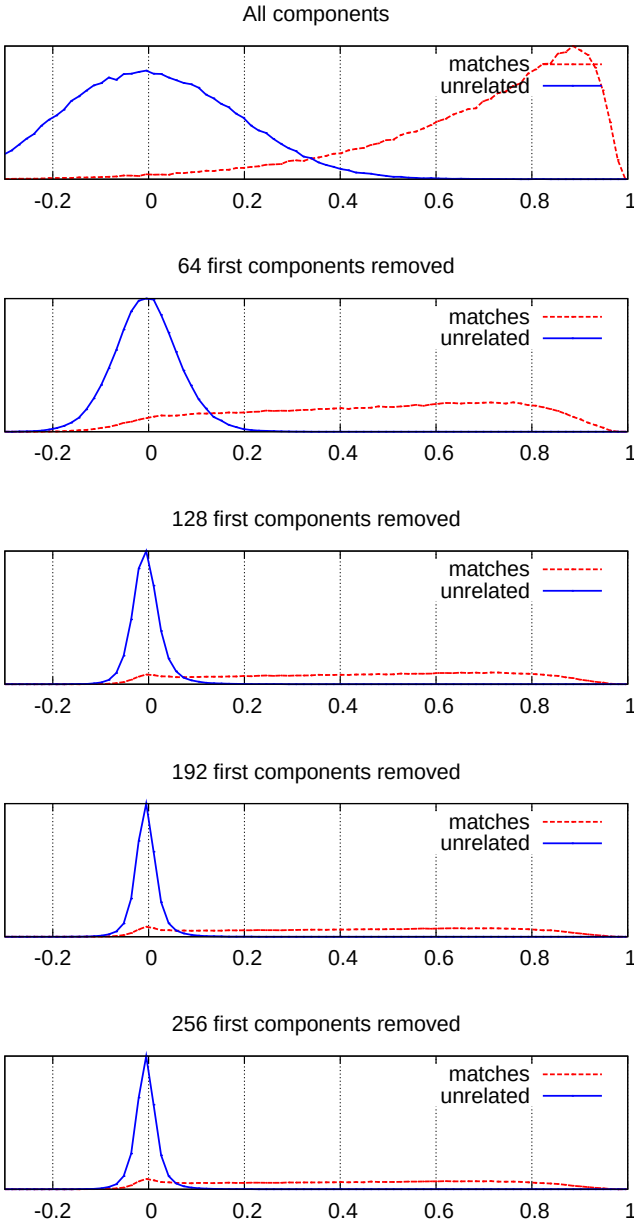

Figure 2. Impact of on the cosine similarity distributions of removing highly-energetic components in our T-embedding (k=16).

Appendix C – Bag-of-visual-words: Link between democratization with square-root normalization

In this section, we discuss the particular case of our method when applied to the bag-of-word representation with cosine similarity. We do not consider inverse-document-frequency terms. In this case, ϕ_{BOW} is defined by

$$\phi_{\mathrm{BOW}}(x) = [0, \ldots, 0, 1, 0, \ldots, 0]^\top. \qquad (1)$$

These mapped vectors are summed up to produce the

bag-of-visual-words vector

$$\text{BOW}(\mathcal{X}) = \alpha(\mathcal{X}) \times [m_1(\mathcal{X}), \ldots, m_j(\mathcal{X}), \ldots, m_n(\mathcal{X})]^\top, \tag{2}$$

where $m_j(\mathcal{X})$ is the number of descriptors assigned to visual word j in $\mathcal{X}$.

The match kernel matrix $\mathsf{K}_{\text{BOW}}(\mathcal{X}, \mathcal{X})$ is, up to a permutation (to assume that the vectors are ordered by increasing visual word indices), block diagonal with only 1 in the blocks:

$$\mathsf{K}_{\text{BOW}}(\mathcal{X}, \mathcal{Y}) = \begin{array}{cccc} \boxed{\mathbf{1}} & \boxed{\mathbf{0}} & \ldots & \boxed{\mathbf{0}} \\ \boxed{\mathbf{0}} & \boxed{\mathbf{1}} & \ldots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \boxed{\mathbf{0}} & \ldots & \ldots & \boxed{\mathbf{1}} \end{array} \begin{array}{c} \updownarrow m_1 \\ \updownarrow m_2 \\ \vdots \\ \updownarrow m_n \end{array} \tag{3}$$

This matrix is positive, which means that the strategy to enforce only positive values has no effect. Similarly, the mapped vectors are already normalized to unit norm. A trivial solution to

$$\Lambda \mathsf{K} \Lambda \mathbf{1}_n = C \mathbf{1}_n, \tag{4}$$

is such that

$$\lambda(x) = 1/\sqrt{m_j} \tag{5}$$

for a vector assigned to visual word j such that $m_j \neq 0$. The resulting vector

$$\psi_{\text{d}}(\mathcal{X}) \propto [\sqrt{m_1}, \ldots, \sqrt{m_i}, \ldots, \sqrt{m_n}]^\top \tag{6}$$

is a democratic kernel that plainly satisfies the "democratic" condition of Equation 4. Interestingly, it is the same vector as the one obtained by applying square-root component-wise normalization [3, 5], which significantly improves bag-of-words performance.

Consider now the symmetric version of the Sinkhorn algorithm (Algorithm 1), where we set $\gamma = 0.5$. The first iteration computes the sum of each row. If the i^{th} vector is assigned to the visual word j, then the sum is m_j and $\lambda_i = 1/\sqrt{m_j}$. In other terms, the algorithm reaches the fixed-point in a single iteration. For other values of $\gamma < 0.5$, the algorithm also converges to this fixed point.

Appendix D – Parameter and method evaluation

We complement the parameter study of Section 5.3 by providing the same experiments performed on the Oxford5k dataset. We also include the variant method $\phi_\triangle + \psi_s + RN$ in the plot. The results are shown in Figure 3 for the vocabulary size $|\mathcal{C}|$ and in Figure 4 for the exponent α involved in power-law normalization.

The conclusions drawn in our main paper are identical for both datasets:

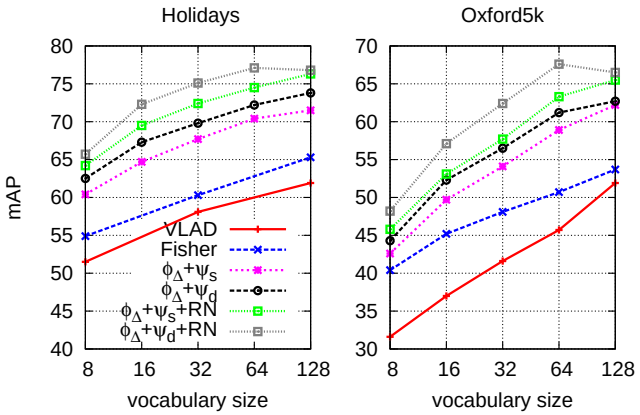

Figure 3. Impact of the vocabulary size $|\mathcal{C}|$ on the performance on Holidays and Oxford5k for several strategies. VLAD, Fisher, our T-embedding $\phi_\triangle$ with sum aggregation ψ_{s} and democratic pooling ψ_{d}. All methods use the same input descriptors with RootSIFT post-processing [1]. Parameter $\alpha = 0.5$.

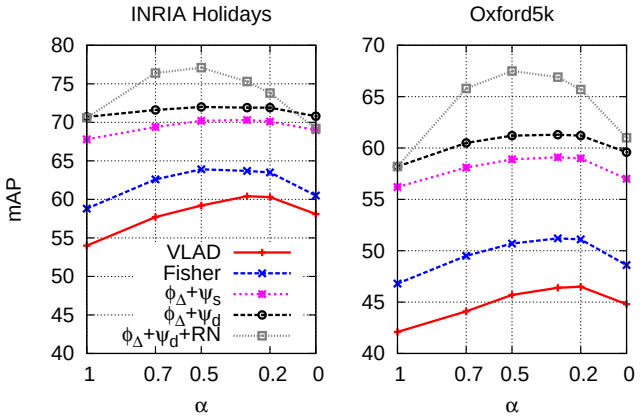

Figure 4. Impact of power-law normalization (parameter α) on the performance. We fix $|\mathcal{C}| = 64$ for all methods. Note that $\alpha = 0$ amounts to binarizing the vector.

- The performance grows as a function of the vocabulary size for all methods, apart from $|\mathcal{C}| = 128$ with the best variant. For this last point, a possible explanation is that the number of learning images (10k for Holidays, the 6k images from Paris for Oxford105k) is no large enough to accurately learn the rotation matrix.
- Our democratic aggregation strategy is complementary with power-law normalization, and improves upon our embedding;
- RN is worth applying and combining with power-law normalization in most cases.
- When using RN, the best performance is noticeably attained for $\alpha = 0.5$.

Table 1 gives a enlarged set of results for accuracy as a function of the vocabulary size on Oxford5k. The conclusion is consistent for all vocabulary sizes, except for ψ_{d}+RN which is better for $|\mathcal{C}| = 64$ than for $|\mathcal{C}| = 128$. How-

	Oxford5k			
$\|\mathcal{C}\|$	ψ_s	ψ_d	ψ_s+RN	ψ_d+RN
8	42.6 ±1.9	44.3 ±2.0	45.8 ±2.8	48.2 ±2.2
16	49.7 ±0.5	52.3 ±0.3	53.1 ±0.4	57.1 ±1.0
32	54.1 ±0.4	56.5 ±0.4	57.7 ±0.1	62.4 ±0.7
64	58.9 ±0.3	61.2 ±0.4	63.3 ±0.9	67.5 ±0.2
128	62.2 ±0.5	62.7 ±0.9	65.5 ±1.4	66.5 ±1.9
	Holidays			
$\|\mathcal{C}\|$	ψ_s	ψ_d	ψ_s+RN	ψ_d+RN
8	60.4 ±0.0	62.5 ±0.2	64.2 ±0.2	65.7 ±0.1
16	64.7 ±0.6	67.3 ±0.4	69.5 ±0.8	72.3 ±0.6
32	67.7 ±0.3	69.8 ±0.4	72.4 ±0.5	75.1 ±0.5
64	70.4 ±0.4	72.2 ±0.2	74.5 ±0.4	77.1 ±0.7
128	71.5 ±1.2	73.8 ±0.8	76.3 ±0.6	76.8 ±1.3

Table 1. Performance of different combinations. For all the methods, we report the average over 3 distinct vocabularies, which are the same for all the methods. We set $\alpha = 0.5$.

$\|\mathcal{C}\|$	UKB (score/4)	Holidays+Flickr1M (mAP)
D=1920	3.53	51.9
$\rightarrow$ 1024	3.51	49.4
$\rightarrow$ 512	3.49	46.9
$\rightarrow$ 256	3.45	43.7
$\rightarrow$ 128	3.40	38.7

Table 2. $\phi_\triangle + \psi_d$: Performance on UKB and Holidays+Flickr1M before (D=1920) and after dimensionality reduction to 1024, 512, 256 and 128 components. $|\mathcal{C}| = 16$. Note that, depending on the desired output dimensionality, the choice of $|\mathcal{C}| = 16$ is not optimal, as it depends on the target final dimensionality.

ever note the larger variance: On Oxford5k, the best result mAP=68.6% is actually obtained with a vocabulary of size $|\mathcal{C}| = 128$ (Best score with $|\mathcal{C}| = 64$: mAP=67.7%).

Appendix E – Performance on UKB and Holidays+Flickr1M

The performance of our best method, namely our T-embedding associated with Sinkhorn democratization, is presented in Table 2 for two additional datasets, namely

- **The University of Kentucky Benchmark** (UKB) [4]. This dataset contains 10,200 images, organized by group of 4 images. All images are submitted in turn. The performance measure is the average number of images returned in the first 4 positions.

- **Holidays+Flickr1M**: Following common practice [2], we merge the Inria Holidays dataset with another set of 1 million images retrieved from Flickr. The performance measure is mean average precision.

$\|\mathcal{C}\|$	$\phi_\triangle + \psi_s$	$\phi_\triangle + \psi_d$
8	0.086	1.754
16	0.096	2.603
32	0.107	4.720
64	0.147	18.740
128	0.237	56.753

Table 3. CPU timings (in seconds) for encoding. We do not include the cost of extracting the SIFT descriptors.

Appendix F – Complexity analysis

Table 3 reports the timings measured to compute our representations for different vocabulary sizes $\mathsf{k_c}$. The measures are obtained on the query images of Oxford5k. The measurements are carried out on a Xeon E5-2650/2.00GHz (32 cores). We report the CPU times (larger than elapsed ones because CPU time cumulates all active threads). On a quad-core laptop with multi-threading, the timing is typically 20ms per image for $\phi_\triangle + \psi_\mathrm{s}$.

The computation of $\phi_\triangle$ is fast. The bottleneck is democratic aggregation, when adopted. This is partially due to the low degree of optimization: Democratic aggregation is done in plain Matlab, while we have optimized the computation of $\phi_\triangle$ with a mex file. This also suggests that further optimization strategies should be considered for larger vocabularies. A simple effective one is to threshold the gram matrix by setting to 0 all values below a threshold (typically, 0.1), to get it sparse at a small accuracy cost.

References

[1] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

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

[3] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.

[4] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[5] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010.

[6] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *PAMI*, 2014.

[7] S. Winder and M. Brown. Learning local image descriptors. In *CVPR*, 2007.