

# Balancing clusters to reduce response time variability in large scale image search

Romain Tavenard, Laurent Amsaleg, Hervé Jégou

► **To cite this version:**

Romain Tavenard, Laurent Amsaleg, Hervé Jégou. Balancing clusters to reduce response time variability in large scale image search. [Research Report] RR-7387, INRIA. 2010. <inria-00519490>

**HAL Id: inria-00519490**

**<https://hal.inria.fr/inria-00519490>**

Submitted on 21 Sep 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Balancing clusters to reduce response time  
variability in large scale image search*

Romain Tavenard — Laurent Amsaleg — Hervé Jégou

N° 7387

September 2010

Thème COG

*R*apport  
de recherche



## Balancing clusters to reduce response time variability in large scale image search

Romain Tavenard\* , Laurent Amsaleg , Hervé Jégou

Thème COG — Systèmes cognitifs  
Équipe-Projet Texmex

Rapport de recherche n° 7387 — September 2010 — 14 pages

**Abstract:** Many algorithms for approximate nearest neighbor search in high-dimensional spaces partition the data into clusters. At query time, in order to avoid exhaustive search, an index selects the few (or a single) clusters nearest to the query point. Clusters are often produced by the well-known  $k$ -means approach since it has several desirable properties. On the downside, it tends to produce clusters having quite different cardinalities. Imbalanced clusters negatively impact both the variance and the expectation of query response times. This paper proposes to modify  $k$ -means centroids to produce clusters with more comparable sizes without sacrificing the desirable properties. Experiments with a large scale collection of image descriptors show that our algorithm significantly reduces the variance of response times, at a slight cost with respect to the trade-off between efficiency and search quality.

**Key-words:** nearest neighbor search, large databases, quantization

\* Corresponding author.

## Réduction de la variabilité du temps de réponse pour la recherche d'image

**Résumé :** De nombreux algorithmes de recherche approchée de plus proches voisins en grande dimension partitionnent les données en clusters. Au moment de la requête, pour éviter une recherche exhaustive coûteuse, un index sélectionne un ou plusieurs clusters parmi les plus proches de la requête. Les clusters sont souvent obtenus par la méthode du  $k$ -means. Un des avantages de cette méthode est qu'elle tend à produire des clusters de tailles diverses. Ce déséquilibre entre les cardinalités des clusters a un effet négatif tant sur la variance que sur l'espérance du temps de réponse. Cet article propose de modifier les centroïdes obtenus par  $k$ -means dans le but de produire des clusters de tailles comparables. Les expériences effectuées sur une grande collection d'images décrites montrent que notre algorithme réduit significativement la variance du temps de réponse, en diminuant légèrement les performances en termes de compromis entre efficacité et qualité des résultats retournés.

**Mots-clés :** recherche de plus proches voisins, grandes bases de données, distance euclidienne, quantification

## 1 Introduction

Finding the nearest neighbors of high-dimensional query points still receives a lot of research attention as this fundamental process is central to many content-based applications. Most approaches rely on some different kinds of partitioning of the data collection into clusters of descriptors. At query time, an indexing structure selects the few (or a single) clusters nearest to the query point. Each candidate cluster is scanned, actual distances to its points are computed and the query result is built upon these distances.

There are various options for clustering points, the most popular being the  $k$ -means approach. Its popularity is caused by its nice properties: it is a simple algorithm, surprisingly effective and easy to implement. It nicely deals with the true distribution of data in space by minimizing the mean square error over the clustered data collection. On the downside, it tends to produce clusters having quite different cardinalities. This, in turn, impacts the performance of the retrieval algorithm: scanning heavily filled clusters is costly as the distances to many points must be computed. In contrast, under-filled clusters are cheap to process, but they are selected less often as the query descriptor is also less likely to be associated with these less populated clusters. Overall, having imbalanced clusters impact both the variance and the expectation of query response times. This is very detrimental to contexts in which performance is paramount, such as high-throughput settings where the true resource consumption can no more be accurately predicted by costs models.

This phenomenon has an even more detrimental impact at large scale. In this case, clusters must be stored on disks and the performance severely suffer when fetching large clusters due to the large I/Os. Furthermore,  $k$ -means is known to fail clustering at very large scale, and hierarchical or approximate  $k$ -means must be used, which, in turn, tend to increase the imbalance between clusters [4].

This paper proposes an extension of the traditional  $k$ -means algorithm to produce clusters of much more even size. This is beneficial to performances since it reduces the variance and the expectation of query response times. Balancing is obtained by slightly distorting the boundaries of clusters. This, in turn, impacts the quality of results since clusters do not correspond to the initial optimization criterion anymore. Section 2 defines the problem we are addressing and introduces the key metrics later used in the evaluation. Section 3 details the balancing strategy we propose. Section 4 evaluates the impact of balancing on the response time of queries when using large collections of descriptors computed over 1 million images from Flickr. It also shows result quality remains satisfactory with respect to the original  $k$ -means. Section 5 concludes the paper.

## 2 Problem statement

### 2.1 Base Clustering and Searching Methods

Without loss of generality, we partition a collection of high-dimensional feature vectors into clusters defining Voronoi cells. We typically use a  $k$ -means algorithm quantizing the data into  $k$  cells. Each cell stores a list of the vectors it clusters. This approach is widely adopted in the context of image searches,

where clustering is applied to local [14, 12] or global descriptors [2, 5]. A search strategy exploiting this partitioning is usually approximate: only one or a few cells are explored at query time. The quality of results is typically increased when multiple cells are probed during the search as in [8, 6, 4, 5]. The actual distances between the query point and the features stored in each such cell are subsequently computed [1, 11]. Therefore, the response time of a query is directly related to (i) the strategy used to identify the cells to explore and (ii) the total number of vectors used in distance computations. The cost for (i) is fixed and mainly corresponds to finding the  $m_p$  centroids that are the closest to the query point ( $L_2$ ). In contrast, the cost for (ii) heavily depends on the cardinality of each cell to process. It is of course linked to  $m_p$ . Note that (i) is often negligible compared to (ii).

## 2.2 Metrics: Selectivity and Recall

All approximate nearest-neighbor search methods try to find the best trade-off between result quality and retrieval time. The quality of the results can be seen as the probability to retrieve the correct neighbors at search time, given the total amount of data that is processed. This can be expressed in terms of selectivity and recall defined as:

- *Selectivity* is the total rate of vectors used in the distance calculations (with respect to the whole data collection). Obviously, the larger selectivity, the more costly is (ii).
- *Recall* is, for a query, the total rate of nearest neighbors correctly identified (with respect to the above selectivity). This measurement is called precision in [11], but *recall* is more accurate here. Observe that if the true nearest neighbor is found within any of the selected cells then it will be ranked first in the result list.

## 2.3 Imbalance Factor

As in [4], we measure the imbalance between the cardinalities of the clusters resulting from a  $k$ -means using an *imbalance factor*  $\gamma$  defined as:

$$\gamma = k \sum_{i=1}^k p_i^2 \quad (1)$$

where  $p_i$  is the probability that a given vector is stored in the list associated with the  $i^{\text{th}}$  cluster. For a fixed dataset of size  $N$ , this factor is empirically measured based on the number  $n_i \approx p_i N$  of descriptors associated with each list. As shown in [4], for  $m_p = 1$  and for a fixed  $k$ , the measure  $\gamma$  of the balancing is directly related to the search cost: a measure  $\gamma = 3$  means that the expectation of the search time is three times higher than the one associated with perfectly balanced clusters. Optimal balancing is obtained when  $n_i = n_{\text{opt}} = N/k$  for all  $i$ . In that case,  $\gamma = 1$  (lowest possible value) and the variance of query time is zero, as any cell contains exactly the same number of elements. This clearly appears in the analytical expression of the variance of the number of elements

in a given list:

$$\text{Var} = N^2 \sum_{i=1}^k p_i \left( p_i - \frac{1}{k} \right)^2. \quad (2)$$

### 3 Balancing Clusters

#### 3.1 The Balancing Process

Balancing clusters is an iterative post-processing step performed on the final output of a  $k$ -means type-of algorithm. The idea is to artificially enlarge the distances between the data points and the centroids of the heavily filled clusters. These penalties applied to distances depend on the population of clusters. Hence, the contents of cells and thus their population can be recomputed accordingly. This balancing process eventually converges to equally filled clusters.

The penalties are called *penalization terms* and are computed as follows:

$$\begin{cases} \forall i, b_i^0 = 1 \\ \forall l, b_i^{l+1} = b_i^l \left( \frac{n_i^l}{n_{\text{opt}}} \right)^\alpha \end{cases} \quad (3)$$

where  $\alpha$  controls the convergence speed. A small value for  $\alpha$  indeed ensures that balancing will be done in a smooth way, while it implies to iterate more in order to get even cell population. Note that, at each iteration  $l$ , the populations  $n_i^l$  are updated in order to take these penalization terms into account. More precisely, distances from any point  $\mathbf{x}$  to the  $i^{\text{th}}$  centroid are computed as

$$d_{\text{bal}}^l(\mathbf{x}, \mathbf{c}_i)^2 = d(\mathbf{x}, \mathbf{c}_i)^2 + b_i^l. \quad (4)$$

#### 3.2 Geometrical Interpretation

A geometrical interpretation of the balancing process described above is possible. Assume the balancing process first embeds the  $k$ -means clustered  $d$ -dimensional vectors into a  $(d + 1)$ -dimensional space. In this space, their  $d$  first components are the ones they had in their original space, while component  $d + 1$  is set to zero for all vectors. Centroids are also embedded in the same way, except for their last component. This last value for centroid  $i$  is set to  $\sqrt{b_i^0}$ . Then, while the balancing procedure iterates, it is set to the appropriate  $\sqrt{b_i^l}$  value. The intuition is that centroids are artificially elevated in an iterative manner from the hyperplane where vectors lie. The more vectors in one cluster, the more elevation its centroid gets. This is illustrated in Figure 1, where the  $z$ -axis corresponds to the added dimension. Along iterations, the updated vector assignments are computed with respect to the coordinates of the points lying in the augmented space. The artificial elevation of centroids tends to shrink the most populated clusters, dispatching some of its points in neighboring cells. Figure 2 exhibits the influence of the  $(d + 1)^{\text{th}}$  coordinate of the centroids on the position of the borders.

#### 3.3 Partial Balancing

The proposed balancing strategy empirically converges towards clusters having the same size. Several stopping criteria can be applied, the most simple being



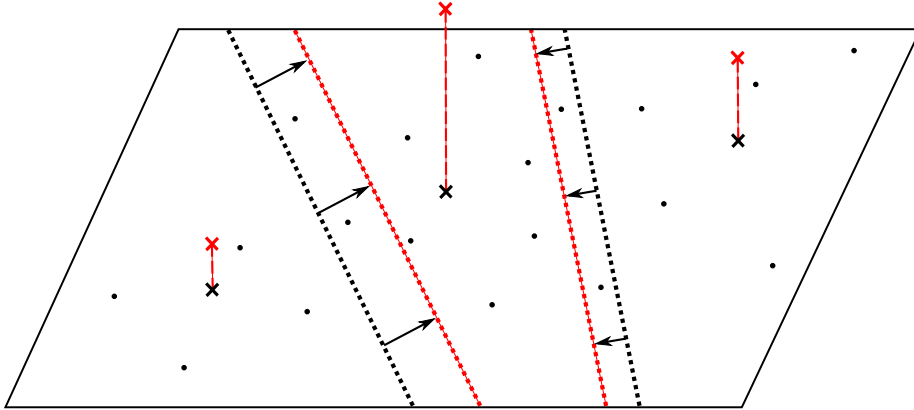


Figure 1: Data points and centroids embedded in a 3- $d$  example. Data points are plotted as dots while centroids are represented as crosses, with a non-null  $z$ -axis value after some iterations.

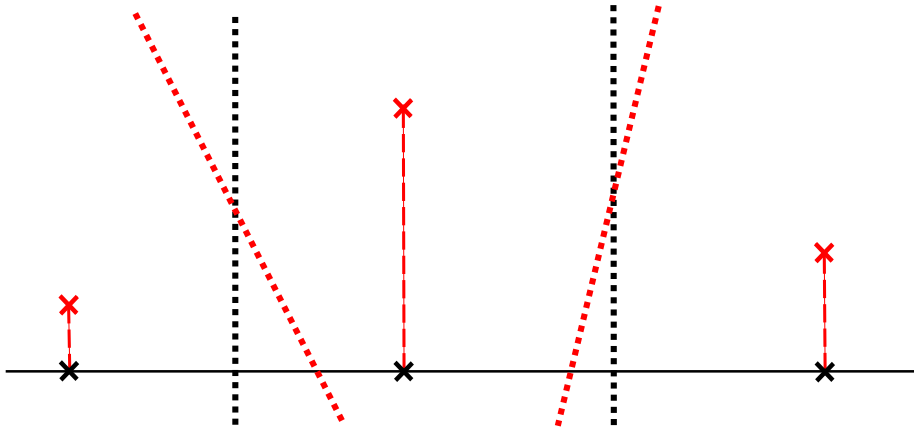


Figure 2: Voronoi cell boundaries shifted after some iterations. New boundaries are plotted as dashed red lines which shrink the central cluster because of its large population.

a fixed maximum number of iterations. It is also possible to target a particular value for  $\gamma$  which is recomputed at every step, either fixed or possibly in proportion of the original imbalance factor. Early stopping the balancing reduces the overall distortion of the Voronoi cells created by the original  $k$ -means.

## 4 Experiments

### 4.1 Datasets and Imbalance Factors Analysis

Our analysis has been performed on descriptors extracted from a large set of real-world images. We downloaded from Flickr one million images to build the database and another set of one thousand images for the queries. Several description schemes were applied to these images, namely SIFT local descrip-

descriptor	dimensionality	$\gamma$	
		$k=256$	$k=1024$
SIFT	128	1.08	1.09
BOF	1000	1.65	1.93
GIST	960	1.72	3.75
VLAD	8192	5.41	6.23

Table 1: Imbalance factor for  $k$ -means clustered state-of-the-art descriptors, measured on a dataset of one million images for two values of  $k$ .

tors [7], Bag-of-features [14] (BOF), GIST [13] and VLAD descriptors [5]. SIFT were extracted from Hessian-Affine regions [10] using the software of [9]. The BOF vectors have been generated from these local descriptors, using a codebook obtained by regular  $k$ -means clustering with 1000 visual words. The VLAD descriptors were generated using a codebook of 64 visual words applied to the same SIFT descriptors, leading to vectors of dimension  $64 \times 128 = 8192$ . For GIST, we have used the most common setup, i.e., the three color channels and 3 scales, leading to 960-dimensional descriptors.

The global descriptors (BOF, GIST and VLAD) produce exactly one descriptor per image, leading to one million vectors for each type of descriptor. In order to keep the same number of vectors for the SIFT set, we have randomly subsampled the local descriptors to produce a million-sized set. In all cases, we assume a closed-world setup, i.e., the dataset to be indexed is fixed, which is valid for most applications.

Table 1 reports the imbalance factors obtained for each type of descriptors after performing a standard  $k$ -means clustering on our database. It can be observed that higher dimensional vectors tend to produce higher imbalance factors. BOF descriptors have an imbalance factor which is lower than GIST for a comparable dimension, which might be due to their higher sparsity. Note that the value of  $k$  has a significant impact on  $\gamma$ : larger values of  $k$  lead to significantly higher  $\gamma$  ( $k=256$  and  $k=1024$ ). The low values for  $k$  we have considered here probably explain why  $\gamma$  measured for the SIFT descriptors in Table 1 are lower than those of the literature: Jegou *et al.* [4] report 1.21 and 1.34 for codebooks of size  $k=20\,000$  and  $k=200\,000$ , respectively.

Our balancing strategy is especially interesting for global descriptors for which, in contrast to local descriptors, exactly one query vector is used. In this case, with perfectly balanced clusters, querying an image is performed in constant time. This is the rationale for focusing our analysis on the well known BOF vectors.

## 4.2 Evaluation of the proposed method

In this subsection we analyze the impact of our method on selectivity, recall and variability of the response time. We also analyze the convergence properties of our method. The parameter  $\alpha$  is set to  $\alpha=0.01$  in all our experiments.

**Selectivity/recall performance:** Figure 3 shows the performance in terms of this trade-off for different values of  $k$ . First note that the trade-off between selectivity and recall can be adjusted using the number  $k$  of clusters and the

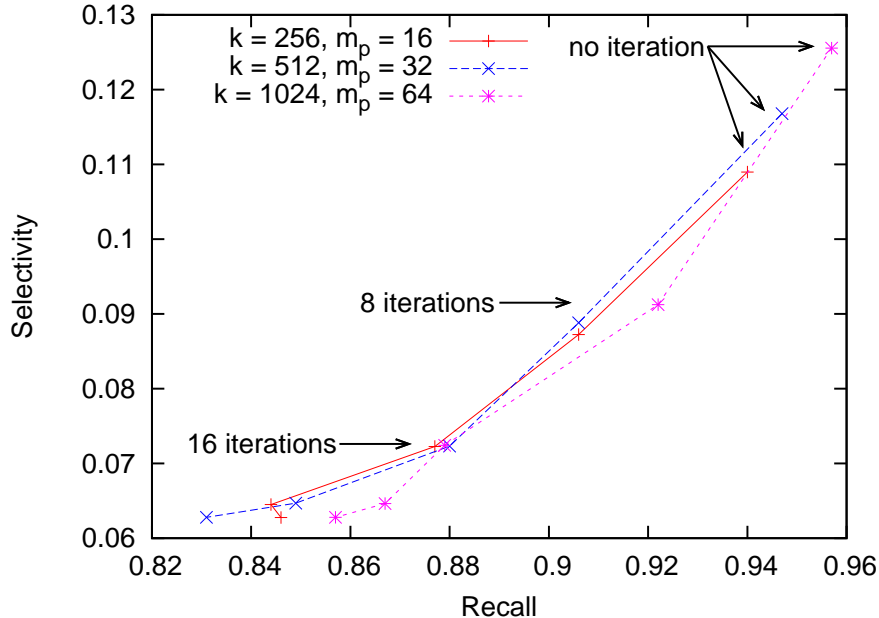


Figure 3: Selectivity/recall performance: impact of partial and full balancing on this trade-off. For each value of  $k$ , the top-right points correspond to the original k-means partition (no iteration). From top to bottom, the points of a given curve correspond to 8, 16, 32 and 64 iterations. Similar to choosing a high value of  $k$ , our method reduces the selectivity (i.e., provides better efficiency) at the cost of lower recall. The different trade-off selectivity/recall are obtained, for our method, with a significantly lower variability of the response time.

number  $m_p$  of probes. We keep the ratio  $m_p/k$  constant in order to better show the impact of our method, which exhibits comparable performance with that of the k-means clustering in terms of selectivity and recall. Figures 4 and 5 shows comparable results when  $m_p$  is constant. Note however that with our method a given selectivity/recall point is obtained with a much better (lower) variability of the response time, as shown later in this section.

**Impact of the number of iterations:** The number  $r$  of iterations performed by Equation 3 is an important parameter of our method, as it controls to which extent complete balancing is enforced or not. Figure 3 shows that selectivity is reduced in the first iterations with a reasonable decrease of the recall, i.e., comparable to what we would obtain by modifying the number of clusters. The next iterations are comparatively less interesting, as the gain in selectivity is obtained at the cost of a relatively higher decrease in recall. Modifying the stopping criterion allows our method to attain a target imbalance factor which is competitive with respect to the selectivity/recall trade-off.

**Convergence speed:** Figure 6 illustrates how the imbalance factor evolves along iterations. Only a few iterations are needed to attain reasonably balanced clusters. Our update procedure has a computational cost which is negligible compared with that of the clustering. Higher values of  $k$  do not require more

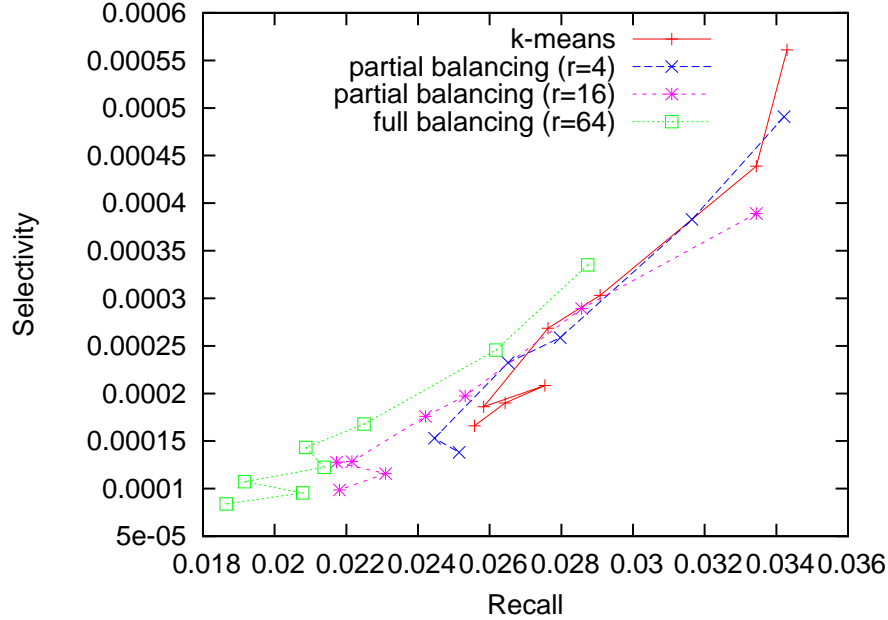


Figure 4: Selectivity/recall performance: impact of partial and full balancing on this trade-off for  $m_p = 1$ . For each balancing strategy, the top-right points correspond to small values of  $k$ . From top to bottom,  $k$  varies between 256 and 1024. Observe that if, for small values of  $k$ , balancing improves the performance in terms of this trade-off, for large  $k$ , balancing tends to deteriorate this trade-off.

iterations, which is somewhat surprising as more penalization terms have to be learned. Note that the convergence of our algorithm is not guaranteed, though in all the experiments presented in this paper it has been observed.

**Variance of the query response time:** The impact of our balancing strategy on the variability of the response time is illustrated by Figure 7, which gives the distribution of the number of elements returned by the indexing structure. The tight distribution obtained by our method shows that the objective of reducing the variability of the query time resulting from unbalanced clusters is fulfilled: the response time is almost constant with full balancing. The partial balancing also leads to significantly improve the shape of the distribution, which has a significantly reduced variance compared with the original one.

**Impact of the choice of descriptors on observed results:** In order to validate our approach on a different kind of descriptors, we tested it using Fisher kernels with 16 gaussians. The query set is the concatenation of the Holidays dataset [3] and the UKB one [12]. The results, as shown in figure 8 are strongly dependent on the value of  $k$ . This is due to the fact that  $k$ -means clustering for small values of  $k$  leads to well-balanced clusters ( $\gamma \leq 1.1$ ) while  $k = 1024$  reaches an imbalance factor of 2.2. In the latter case, balancing shows its efficiency in terms of selectivity, as expected.

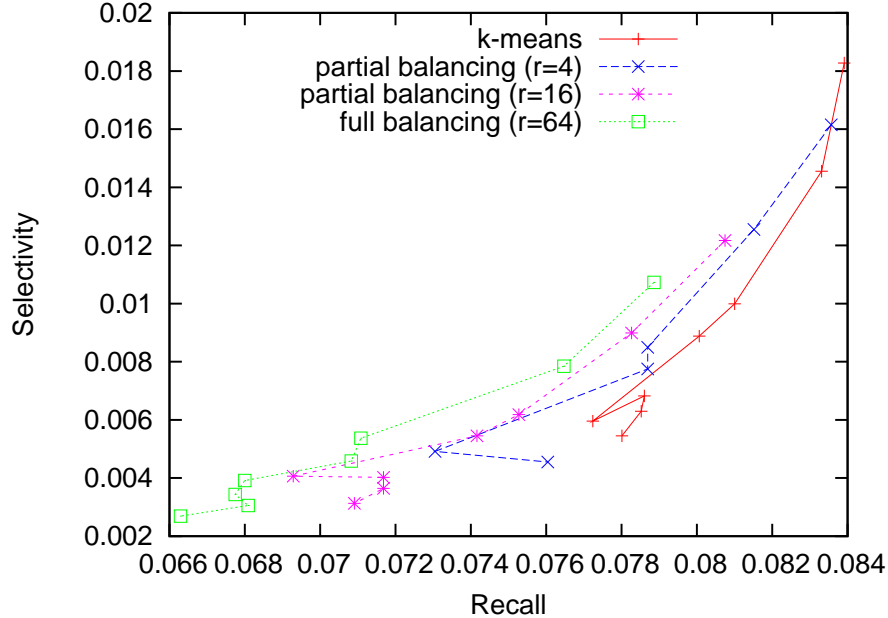


Figure 5: Selectivity/recall performance: impact of partial and full balancing on this trade-off for  $m_p = 32$ . For each balancing strategy, the top-right points correspond to small values of  $k$ . From top to bottom,  $k$  varies between 256 and 1024. In this case, balancing tend to deteriorate this trade-off.

### 4.3 Is closed-world setup mandatory ?

Previous section presented results obtained in a closed-world setup as it allows to achieve quasi-constant query time in all cases. However, figure 9 shows that, as soon as distribution of the learning set is reasonably close to the one of the database, comparable selectivity-versus-recall compromise can be achieved in the open-world case. In this example, the database is the same as the one used in the previous experiments. For both closed-world and semiclosed-world setups, another 1 million images from Flickr are used as a learning set to train  $k$ -means. The different between both setups is that in the semiclosed-world one, balancing is learnt on the database itself while in the open-world setup, it is optimized on the learning set, which could lead to unbalanced database clusters.

Note nevertheless that quality of the balancing in semiclosed and open-world setups strongly depends on the learning set having comparable distribution to the one of the database. Therefore, their usage should be restricted to cases where this assumption is likely to be verified, as for example in cases where the learning set is a subset of the entire database.

## 5 Conclusion

Many high-dimensional indexing schemes rely on a partitioning of the feature space into clusters obtained from a  $k$ -means type-of approach. These schemes are efficient because they process a very small number of cluster for answering

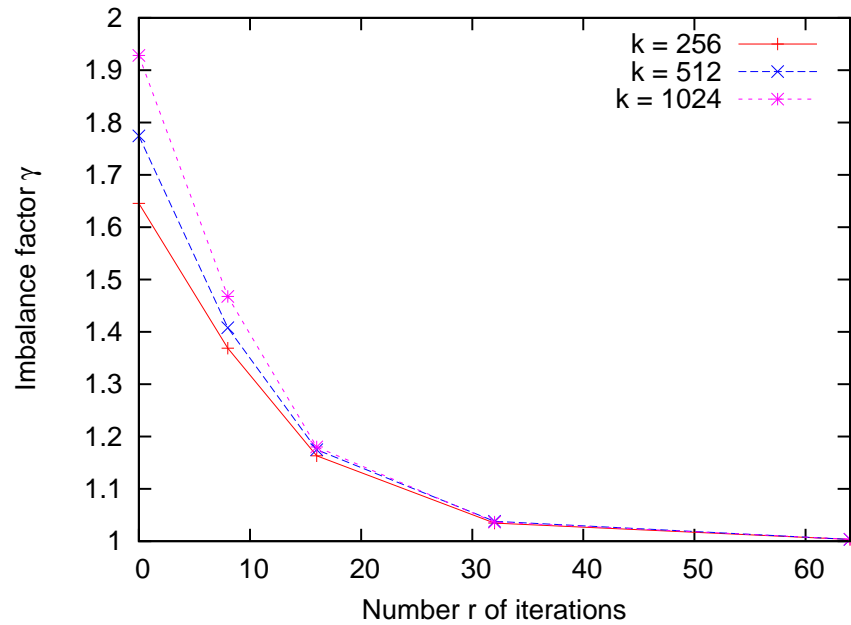


Figure 6: Convergence speed

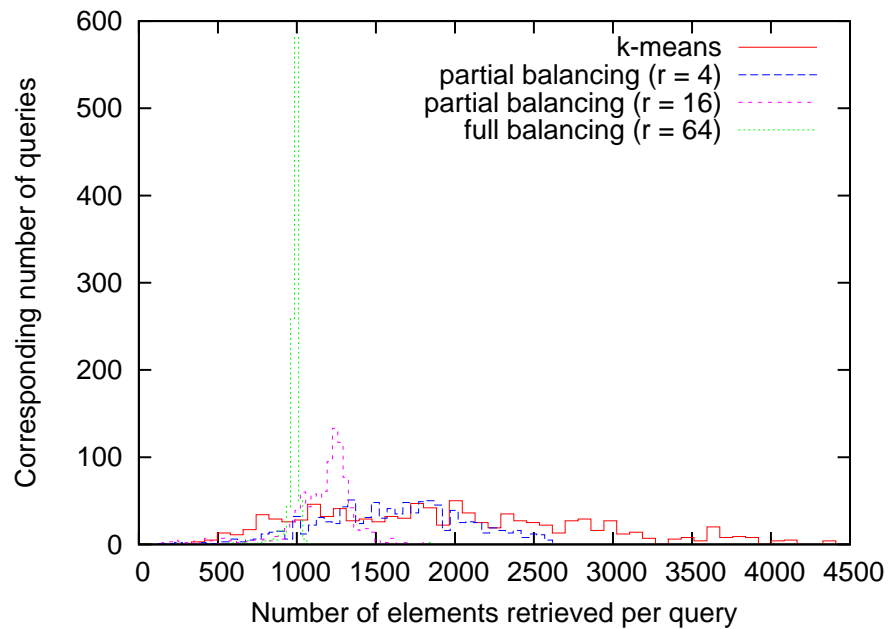


Figure 7: Histograms of the number of elements returned, computed over our 1000 queries, for the original k-means and our algorithm with three number of iterations. Observe the tightness of the distribution in the case of our method, which reflects a very low variability in response time.

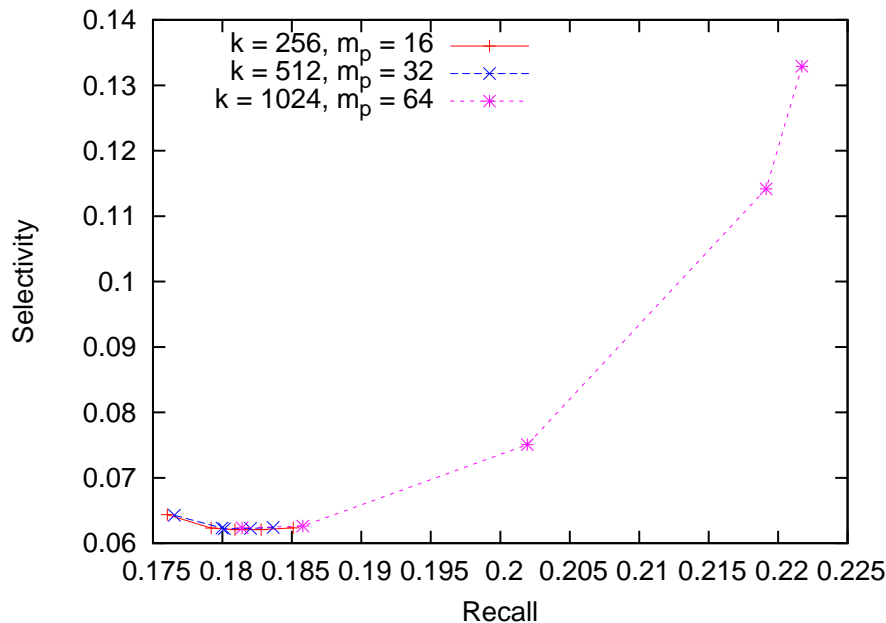


Figure 8: Selectivity/recall performance: impact of partial and full balancing on this trade-off for Fisher kernel descriptors. From top to bottom, the points of a given curve correspond to 8, 16, 32 and 64 iterations.

each query. Their performance suffer, however, from having to process clusters with very different cardinalities since this causes great variations in the response time to queries. This paper presents an algorithm that iteratively balances clusters such that they become more equal in size. Reducing the variance and the expectation of response times is a key issue when targeting high-performance settings, especially when data has to be read from disk. Our experiments demonstrated that clusters are better balanced without significantly impacting the search quality. We are planning to index much data collections where the imbalance factor will be higher, as for the promising VLAD descriptors [5], increasing the need for a more uniform cluster distribution.

## References

- [1] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. of the Symp. on Computational Geometry*, 2004.
- [2] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *Proc. CIVR*, 2009.
- [3] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In A. Z. David Forsyth, Philip Torr, editor, *European Conference on Computer Vision*, volume I of *LNCS*, pages 304–317. Springer, oct 2008.

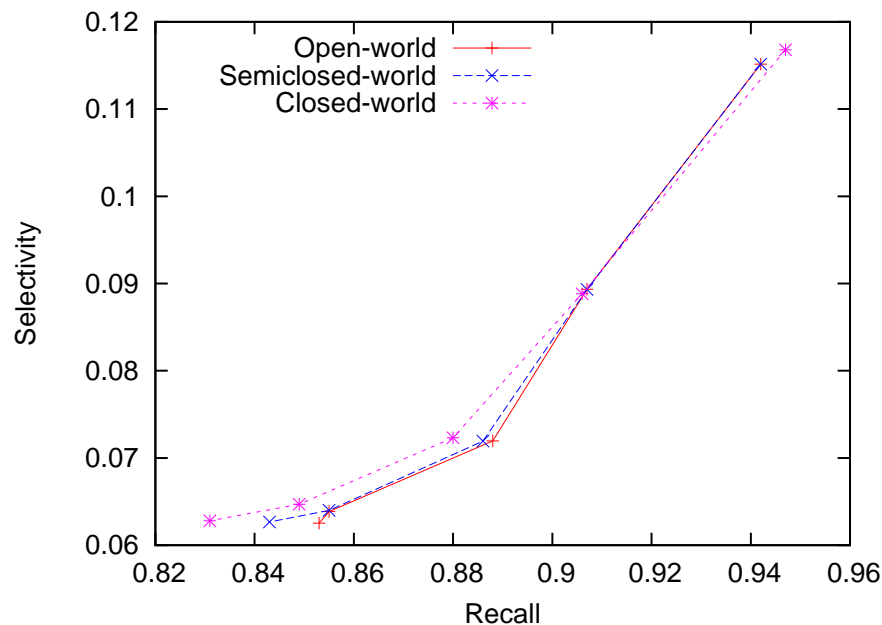


Figure 9: Compared selectivity/performance for 3 different setups. Here, we used  $k = 512$  and  $m_p = 32$ .

- [4] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *Int. Journal of Computer Vision*, 87(3), 2010.
- [5] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, June 2010.
- [6] A. Joly and O. Buisson. A posteriori multi-probe locality sensitive hashing. In *Proc. ACM MM*, 2008.
- [7] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2), 2004.
- [8] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Multi-probe LSH: Efficient indexing for high-dimensional similarity search. In *Proc. VLDB*, 2007.
- [9] K. Mikolajczyk. Binaries for affine covariant region descriptors, 2007.
- [10] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Int. Journal of Computer Vision*, 60(1), 2004.
- [11] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.
- [12] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, 2006.



- [13] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. Journal of Computer Vision*, 42(3), 2001.
- [14] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.



---

Centre de recherche INRIA Rennes – Bretagne Atlantique  
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex  
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier  
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq  
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex  
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex  
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex  
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399