Inf Retrieval J (2015) 18:51–72
DOI 10.1007/s10791-014-9247-6

MDPV: metric distance permutation vocabulary

Vlastislav Dohnal · Tomas Homola · Pavel Zezula

Received: 1 May 2014 / Accepted: 3 October 2014 / Published online: 29 October 2014

Abstract Sub-image content-based similarity search forms an important operation in current image archives since it provides users with images that contain a query image as their part. Such a search can conveniently be implemented using the bag-of-features model. Its integral part is a construction of visual vocabulary. Most existing algorithms to create a visual vocabulary suffer from high computational (e.g. k-means) or supervisor-guidance (e.g. visual-bit classifier, or sparse coding) requirements. In this paper, we propose a novel approach to visual vocabulary construction called metric distance permutation vocabulary. It is based on permutations of metric distances to create compact visual words. Its major advantage over prior techniques is time and space efficiency of vocabulary construction and quantization process during querying, while achieving comparable or even better effectiveness (query result quality). Moreover, this basic concept is extended to combine more independent permutations. Both the proposals are experimented on well-known real-world data-sets and compared to other state-of-the-art techniques.

Keywords Feature quantization · Visual vocabulary · Bag-of-features model · k-Means clustering · Metric distance permutation vocabulary

V. Dohnal (✉) · T. Homola · P. Zezula
Faculty of Informatics, Masaryk University, Botanicka 68a, 602 00 Brno, Czech Republic
e-mail: dohnal@fi.muni.cz

T. Homola
e-mail: xhomola@fi.muni.cz

P. Zezula
e-mail: zezula@fi.muni.cz

1 Introduction

Image and especially sub-image content-based similarity search strongly depends on how images are described and characterized. During the last decade, various image characterization methods were introduced. Local visual features based on SIFT (Lowe 2004), its variant SURF (Bay et al. 2008) and others form the current state-of-the art. The rotation, scale and illumination-change invariance, and noise stability properties of SIFT descriptor are counterbalanced by its high complexity. Each SIFT descriptor is a 128-dimensional vector, whereas SURF descriptor is of 64 dimensions. Attempts to analyze, organize, search or index such high-dimensional spaces have to cope with the *dimensionality curse* (Beyer et al. 1999). Another weakness of SIFT and similar local descriptors is large, almost unpredictable, number of descriptors extracted from an image. There are techniques presented in literature that aim at decreasing the number of descriptors by using an improved local extrema detection algorithm (Chum et al. 2004) or by aggregating and compressing them (Turcot and Lowe 2009; Chum et al. 2008; Jégou et al. 2012; Bergamo et al. 2013). Nonetheless, it is still challenging to design an effective system that scales up to billions of images while being acceptably efficient (fast).

The bag-of-features (BOF) model has gained success and constitutes a state-of-the-art principle in image categorization and retrieval. Its idea is a transformation of image content-based retrieval problem to text retrieval problem that is efficiently implemented by current full-text search engines. This transformation is based on preprocessing database images by extracting local visual features and quantizing them to decrease their variability. Thus an image is described with a list of so-called *visual words*. The great advantage of such an approach is that an efficient implementation through inverted files, e.g. Apache Lucene search engine, is available. Of course, visual words can be weighted similarly to tf-idf scheme in text retrieval. A *visual vocabulary* is a set of all visual words. Obviously, the visual vocabulary is a critical component of the whole system. Each visual word should represent all feature descriptors that are semantically akin. In practice, we have to choose or define a proper similarity metric function that measures a pair of semantically-similar descriptors as close. Finally, a technique that clusters or partitions the feature space to a moderate number of "representatives" (visual words) must be employed.

There are two basic applications of BOF model: (1) object categorization and recognition, and (2) content-based image retrieval. The former typically requires training a classifier for each object category along with a special visual vocabulary that emphasizes some visual words for each of categories so a Bayes classifier, for example, can be applied. In Winn et al. (2005), clustering is used to construct a large vocabulary that is then reduced using pair-wise visual word merging for better categorization. In Yang et al. (2008b), a collection of discriminative vocabularies is learned by incorporating a classifier into the process of vocabulary creation. The latter application aims at retrieval of images that are similar to a query image or contain it as their part (Philbin et al. 2008; Mikulik et al. 2010), which is also the focus of this paper.

This paper is organized as follows. In the following subsections, we sketch out our proposal and relate it to relevant existing techniques. In Sect. 2, the formal definition of the proposed algorithm is given including complexity analysis. Section 3 presents two extensions to this proposal. Experimental trials on widely used data-sets are given in Sect. 4. The paper concludes in Sect. 5.

1.1 Paper contributions

We propose a novel algorithm metric distance permutation vocabulary (MDPV) for visual vocabulary creation for the need of generic image retrieval in large-scale image collections. Our approach is based on exploiting permutations of distances from a database/query descriptor to preselected descriptors. These permutations are then interpreted as codes of visual words. In other words, we select a set X of n descriptors and use them to construct a so-called l-prefixVoronoi diagram which differs from the order-l Voronoi diagram (Aurenhammer 1991) by exploiting not only l closest sites but also their ordering. In this respect, each l-prefixVoronoi cell is defined as the set of points of the feature space that have a particular sequence of l descriptors $(f_1,\ldots,f_l)$, where $f_i \in X$, as their l nearest neighbors. This sequence can be perceived as a permutation over X and it defines one visual word.

Next, we extend this rigid principle of l-prefixVoronoi diagram to an adaptive variant ($\langle 1..l\rangle$-prefix Voronoi diagram), where only the maximum prefix l is defined. This is convenient since Skala (2009) shows that not all Voronoi cells in a high-dimensional space may contain any point, so we can omit "empty" cells. To increase stability of quantization, several random but independent $\langle 1..l\rangle$-prefix tessellations can be defined to form one system, which is an idea similar to the random forests (Breiman 2001).

Advantages of our MDPV approach can be summarized as follows: (1) *fast preparation* of the visual vocabulary—majority of vocabulary construction costs are consumed by selecting proper feature descriptors that define the $\langle 1..l\rangle$-prefix Voronoi diagram over a carefully selected sites. However, experimental evaluations confirm that a random selection provides a good overall performance (Lu et al. 2012; 2) *low memory requirements*—a small number of descriptors lead to a large number of Voronoi cells; (3) *data type independence*—since our method is defined for a metric space, there is no restriction to multidimensional feature spaces only, so other application domains may benefit from using such a feature quantization; (4) *comparison* of the performance of MDPV with other state-of-the-art techniques on real-world data-sets.

1.2 Bag-of-features model

In computer vision the bag-of-features, or bag-of-words, model is an alias for a well-known vector-space model originating from text retrieval. It is extended to be applied to images as follows. Each database image I is processed by an extraction algorithm that reads the binary content of I, identifies patches and returns a set of visual local features $F = \{f_1,\ldots,f_m\}$, where each feature symbolizes a numerical description of a patch. Moreover, x, y coordinates and scale s defining the patch's location in I and its importance, can be associated with each feature. Since the feature descriptor f_i is typically a high-dimensional vector (128 for SIFT), its variability is extremal. So, a particular descriptor is not very likely to reoccur in I or any other image, thus a direct application of vector-space model would lead to poor quality search. A technique that reduces such variability must be applied first. This stage converts a descriptor to a *visual word*. The visual word is then a representative of several similar image patches. Finally, a set of all visual words is denoted as a *visual vocabulary*. For example, k-means clustering applied to all extracted local feature descriptors can be used to obtain k cluster centers that become the visual words. Thus, each image patch is mapped to a visual word by the clustering and the image is represented by a histogram of visual words.

1.3 Related work

Local visual feature descriptors are bulky, so existing techniques usually transform them to more compact representations. The motivation for doing so is twofold: to decrease memory requirements so a main-memory organization can be utilized, and to use a well-known and reliable data organization typically verified by previous successful deployment. We classify local feature compression schemes into three categories: *transform coding*, *hashing*, and *vector quantization*. Besides this, we also tackle indexing techniques that do not apply any transformation, but rather embed the data into a metric space (distance space) and use an indexing technique for such data.

Firstly, transform coding methods are mostly adopted in audio, image and video signal processing. They are application-specific and require deep knowledge about the data domain in advance. Such knowledge is then used to determine information to be discarded from the source versus information to be processed. A typical representative is the Karhunen Lòeve Transform (KLT) (Bigun 1992; Devaux et al. 2000) that gives good results for Gaussian data, causing the transformed coefficients to be statistically independent. Unfortunately, the statistics for SIFT nor SURF descriptors do not exhibit Gaussian data properties. In Chandrasekhar et al. (2009), the authors studied dimensionality reduction of SIFT and SURF descriptors using KLT but followed by an entropy coding. Discrete Cosine Transform (Chadha et al. 2011; Schwerin and Paliwal 2008; Makar et al. 2009) and Discrete Wavelet Transform (Grzegorzek et al. 2010; Lim et al. 2009) were also proposed as feature quantization methods, but they did not perform results comparable to other state-of-the-art methods.

Secondly, various hashing transformations like locality sensitive hashing (LSH) are applied. The main drawback of the original LSH method as described in Andoni and Indyk (2008) is that it requires a large number of hash tables or a long time to evaluate nearest neighbors. LSH was used to build a search engine without any intermediate visual-vocabulary-creation step in multi-probe LSH (Joly and Buisson 2008), kernelized-LSH (Kulis and Grauman 2009) and LDA-Hash (Strecha et al. 2011). Kang et al. (2004) use hierarchical and non-uniform bucket partitioning to handle non-homogeneous data-sets. In Yin et al. (2013), multi-probe LSH was enriched with principles of B^+ trees to optimize I/O costs, which significantly increased performance to retrieve nearest neighbors. An older, but interesting, approach, called Randomized Locality Sensitive Vocabulary (RLSV), applying LSH to create visual vocabulary has been proposed in Mu et al. (2010). It aggregates multiple visual vocabularies created from random projections without clustering and training efforts. RLSV's final quantization is based on the consensus of all such vocabularies. Unfortunately, the authors test this method on the problem of classification and near-duplicate detection rather than image retrieval. RLSV is highly relevant to our approach, so we provide a comparison with it in a near-duplicate video detection task in Sect. 4. Recently, sim-min-hash (Zhao et al. 2013) was proposed as an extension of min-hash (Broder 1997) and sketches (Chum et al. 2008). From the results, we can read that the speed of image retrieval was greatly improved however at the expense of lower query effectiveness.

Thirdly, vector quantization methods split feature space into a finite set of *clusters*. Each cluster consisting of many descriptors is represented by a single median computed from them or a single centroid selected from them. To the most known such methods, we count the k-means family of algorithms. K-means-based algorithms (Kanungo et al. 2002; Csurka et al. 2004; Nister and Stewenius 2006; Jegou et al. 2010b) are unsupervised methods that partition the feature space into a predefined number of clusters. A technique (Jégou et al. 2008) improves k-means quantization by additional partitioning in

hamming space. To this end, VLAD (Jegou et al. 2010b; Jégou et al. 2012) technique can be observed alike to our proposal, but it uses K preselected SIFT vectors as a codebook to form one aggregated vector for all SIFTs obtained from an image. Next, dimensionality reduction by slicing the aggregated vector is done to produce a short and compact code of few bytes for fast nearest neighbor search (Jegou et al. 2011). VLAD is a non-probabilistic version of fisher vector (FV) (Perronnin et al. 2010). This paper reports comparative results of FV with power normalization in the task of image retrieval. In Amato et al. (2013), VLAD descriptors are converted to a text representation and indexed by Lucene. Such a conversion is done by identifying k nearest-neighbors in a preselected set of pivots to a VLAD descriptor. So a list of "words" describing one image is obtained. Moreover, the number of occurrences of these words in the list depends on their distance from the VLAD descriptor—the closer the descriptor is, the more occurrences are in the list. This helps Lucene better estimate tf-idf weights. There are also methods that use prior knowledge/supervision to optimize clustering, e.g. random forests (Breiman 2001), informative vocabulary (Yang et al. 2008a), or sparse coding (Song et al. 2013). They are primarily used for object-recognition/classification task. Approaches to optimize visual vocabulary are soft assignment (Philbin et al. 2008) and fine vocabulary (Mikulik et al. 2010). The prior knowledge of domain gives a great advantage to all of the vector quantization approaches over other methods but, on the other hand, the vector quantization approaches usually suffer from the out-of-sample problem.

Lastly, generic metric-space-based indexing techniques (Zezula et al. 2006) can also be used for image retrieval, e.g. the system called MUFIN (Batko et al. 2010). Approaches relevant to our technique exhibit LSH properties and are based on distance permutations, e.g. M-index (Novak and Batko 2009), MI-file (Amato et al. 2014), ordering permutations (Chavez et al. 2008) and permutation prefixes (Esuli 2012). Issues of LSH in metric spaces were further studied in Kyselak et al. (2011). The ordering permutations technique builds on top of principles of Linear AESA (Micó et al. 1992). The major difference is that it stores a permutation of pivots in increasing order of their distances for each database object only. Thus, it is a very condensed representation because the distances themselves are not stored at all. The search is then done through the sequential scan over the database objects, however in decreasing order of permutation similarity measured to the permutation of a query, so the search can be terminated earlier to avoid full scan. Finally, this technique uses all pivots to index each database object. On the other hand, the permutation prefixes exploit only a subset of pivots that are the closest to a database object, so storage requirements are further reduced.

The primal idea of our method is to exploit distance permutations too, but its deployment requirements are principally different from ones known for indexing techniques. In particular, indexing structures define *a moderate number of high-capacity buckets*, but we aim at defining *a large number of low-capacity cells* to comply with the properties of a good visual vocabulary (Mikulik et al. 2010). Next, we do not need to support other than exact match queries, so visual vocabulary can be seen as a set of equivalence classes.

2 Metric distance permutation vocabulary

In this section, we describe our proposal for visual vocabulary creation and justify it with relevant existing techniques. We also analyze its space and time complexity to document its efficiency. Next, we extend this technique to be adaptive to data distribution in the feature space.

The idea of the *Metric Distance Permutation Vocabulary* (MDPV) is based on a Voronoi partitioning (Aurenhammer 1991) generalized to metric spaces. The major difference from k-means-based algorithms is in the definition of Voronoi tessellation. K-means approaches create K Voronoi cells by selecting K descriptors from the feature space. Each cell then corresponds to one cluster. To create a visual vocabulary good at image retrieval, the parameter K must usually be large, e.g., $K = 10^6$, so the computational time is extremely long. Hierarchical k-means pushes K down by creating a tree of cells. So a cell is recursively divided by a new set of K' descriptors selected from this cell. The parameter K' is much smaller than the original K, which leads to much faster computation and a reasonably shallow tree. Its depth is $log_{K'}K$. For example, we can define $K = 10^6$ cells by using $K' = 10$ and a 6-level tree. On the other hand, our approach (MDPV) uses a small number of preselected descriptors to recursively define a large number of Voronoi cells. We call this principle *l-prefixVoronoi partitioning*.

2.1 Construction of *l*-prefixVoronoi partitioning

Initially, a set of feature descriptors, called pivots (originally sites), is selected from descriptors extracted from database images, $P = \{p_1, \ldots, p_m\}$. The selection can be done at random. Next, a cell of l-prefixVoronoi partitioning is formed by descriptors that have the same sequence of pivots $(p_1, \ldots, p_l)$ as their l-nearest neighbors in P. Alternatively, an l-prefixVoronoi partitioning can be observed as a recursive application of the common Voronoi partitioning. All the pivots in P define initial Voronoi cells. Each of these cells is then divided on the next level with P again, but omitting the pivot that defined this cell. This is repeated until the maximum level l is reached. Figure 1b depicts an example of l-prefix2 Voronoi partitioning using four pivots. A formal definition of l-prefixVoronoi partitioning follows.

Preliminaries. Assume a metric space $\mathcal{M} = (\mathcal{D}, d)$, where $\mathcal{D}$ is a domain of objects (local image feature descriptors) and d is a total distance function $d : \mathcal{D} \times \mathcal{D} \mapsto \mathbb{R}$ satisfying metric postulates (identity, symmetry and triangle inequality), a fixed set P of n preselected descriptors (pivots) $p_1, p_2, \ldots, p_n$ from $\mathcal{D}$ and a descriptor (object) $f \in \mathcal{D}$, let

$$(\cdot)_f : \langle 1, n\rangle \mapsto \langle 1, n\rangle \tag{1}$$

be a permutation such that $\forall i, j \in \langle 1, n\rangle$:

$$\begin{aligned} (i)_f < (j)_f \Leftrightarrow\ & d(p_{(i)_f}, f) < d(p_{(j)_f}, f) \\ \vee\ [\ & d(p_{(i)_f}, f) = d(p_{(j)_f}, f) \wedge i < j\ ]. \end{aligned} \tag{2}$$

In other words, the pivots are ordered with respect to distances from f, so a sequence $p_{(1)_f}, p_{(2)_f}, \ldots, p_{(n)_f}$ is obtained.

Visual word creation. MDPV divides space $\mathcal{M}$ into $n \cdot (n-1) \cdot \cdots \cdot (n-l+1)$ cells by applying Voronoi partitioning recursively, where l is the maximum partitioning level ($1 \leq l \leq n$). For the first level, each feature descriptor f is assigned to its closest pivot p_i, so a cell C_i is formed. For the second level, each cell C_i from the previous level is re-partitioned with the pivots $P \setminus \{p_i\} = \{p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n\}$. It results in creating cells $C_{i,j}$, where j is the index of the second closest pivot. This partitioning process is repeated until the level l is reached. See Fig. 1 for illustration.

For each feature descriptor $f \in \mathcal{D}$, MDPV is defined as the function $mdpv_l : \mathcal{D} \mapsto \mathbb{R}$ and assigns an integer to f using the formula:

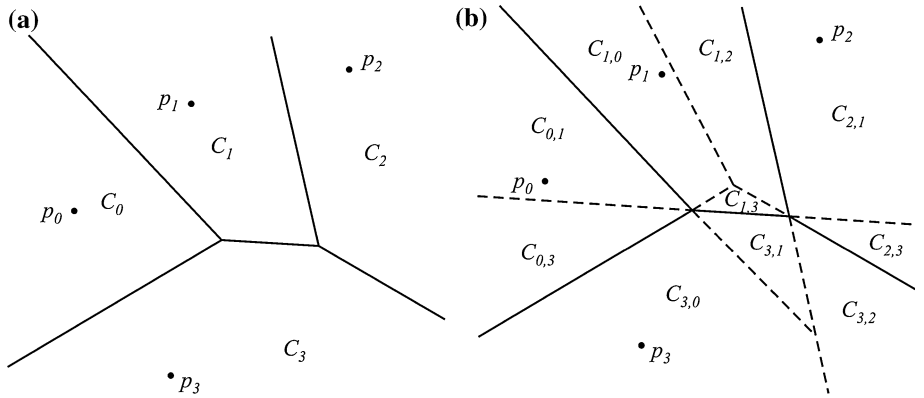

Fig. 1 An example of two l-prefixVoronoi partitionings in 2-D Euclidean space over the same four pivots. **a** l-prefix1 Voronoi partitioning is identical to the original Voronoi partitioning. **b** In l-prefix2 Voronoi partitioning each cell from l-prefix1 partitioning is split by the remaining three pivots (*dashed lines*). The resulting cells are denoted by C subscripted with the sequence of indexes of one and two closest pivots, respectively

$$mdpv_l(f) = \sum_{i=1}^{l} [(i)_f - 1] \cdot n^{(l-i)}. \tag{3}$$

The function $mdpv_l(f)$ identifies a cell at level l to which the feature descriptor f passed in the argument belongs. This number is equal to the number $i_1 \ldots i_l$ in the base n, so it falls into the decimal interval $[0, n^l]$.

2.2 Complexity analysis

Even though MDPV is generic and applicable to any metric space, we assume the feature space is a vector space with the Euclidean distance here. We use SIFT descriptors to compare MDPV with other state-of-the-art vocabulary-based methods, namely RLSV (Mu et al. 2010), ERCF (Moosmann et al. 2008), hierarchical k-means (hk-means) and standard k-means. The comparison is done in terms of theoretical space and time complexity to construct a vocabulary and to transform a feature to a visual word.

Table 1 presents the complexities, where D stands for feature-space dimensionality ($D = 128$ for SIFTs), K is the number of visual words created, N denotes the database size in the number of features, i is the number of iterations in k-means, and K' stands for the number of clusters created in each node of hierarchical k-means. To make the complexities comparable, we use K as the basis for stating all complexities. This implies that we have to estimate the number of non-empty/valid cells in an l-prefixVoronoi partitioning. Since it is still an open problem (Aurenhammer 1991), we estimate that $\sqrt[s]{K}$ pivots are needed in a l-prefix6 Voronoi partitioning to form K cells, where s is a constant in $(2, 3)$. This is based on our experimental observation, see Fig. 4a for details, and holds for moderate number of pivots (from 10 up to 150) and the partitioning level l up to 6. By analogy, RLSV uses a constant $c \in (1, 2)$ to handle trimming empty buckets.

MDPV is very space efficient since it stores only the pivots that define the tessellation. The construction time is $\mathcal{O}(\sqrt[s]{K})$, which covers that the pivots are selected at random without any

Table 1 Comparison of MDPV with other state-of-the-art methods in vocabulary occupation, vocabulary construction and transformation to visual word time, respectively

Alg.	Space	Construction Time	Transformation Time
MDPV	$\mathcal{O}(D\sqrt[s]{K})$	$\mathcal{O}(\sqrt[s]{K})$	$\mathcal{O}(D\sqrt[s]{K} + l\sqrt[s]{K})$
MDPV (dynamic prefix)	$\mathcal{O}(D\sqrt[s]{K} + K)$	$\mathcal{O}(DN\sqrt[s]{K})$	$\mathcal{O}(D\sqrt[s]{K} + l\sqrt[s]{K})$
RLSV (Mu et al. 2010)	$\mathcal{O}(D\log_c K)$	$\mathcal{O}(D\log_c K)$	$\mathcal{O}(D\log_c K)$
ERCF (Mu et al. 2010)	$\mathcal{O}(DK)$	$\mathcal{O}(\sqrt{D}N\log_2 K)$	$\mathcal{O}(D\log_2 K)$
hk-means	$\mathcal{O}(DK)$	$\mathcal{O}(iDNK'\log_{K'} K)$	$\mathcal{O}(DK'\log_{K'} K)$
k-means	$\mathcal{O}(DK)$	$\mathcal{O}(iDNK)$	$\mathcal{O}(DK)$

D, feature space dimensionality; K, number of visual words; $\sqrt[s]{K}$, number of pivots; N, number of features or training set cardinality; i, iterations of k-means to find local extremes; K', number of seeds used on each level in hierarchical k-means; l, maximum Voronoi partitioning level; c, a constant for eliminating empty buckets

additional checks. From a particular setting point of view, this complexity is constant because the number of visual words K is fixed. The cost to transform a feature descriptor to visual word is formed by locating l closest pivots to a feature descriptor to quantize. Currently, we apply a naïve solution to l-nearest neighbor search that compares the query descriptor with all pivots and keeps l closest (insertion sorting algorithm is used). However, any specialized technique such as compact codes (Jegou et al. 2011) or randomized k-d forests can be applied to speed this process up in case of vector feature spaces. As a result, MDPV is comparable to ERCF or even better than RLSV at transformation time. For example, if one-million-word visual vocabulary is used ($K = 10^6$), ERCF's transformation cost is $D \cdot 20$, whereas 80-pivot MDPV's transformation cost is $D \cdot 80 + 6 \cdot 80$, respectively. The extension of MDPV with dynamic prefixes is described in the following section.

3 Extending MDPV

MDPV, as has been described, creates a space and time efficient visual vocabulary, however it still exhibits two inconveniences. Firstly, the definition of Voronoi partitioning is, in a sense, rigidly defined by the set of pivots and the parameter l determining the prefix length. This often leads to improper clustering—some clusters are not identified as separate even though a clustering/quantization objective function would split them. To handle it, we propose a dynamic prefix partitioning. Secondly, it has been experimentally verified that organizing computer vision databases by quantization requires a large visual vocabulary (Mikulik et al. 2010). So, we propose combining more independent MDPVs in Sect. 3.2. It offers much larger vocabularies without the need to use a large number of pivots.

3.1 Dynamic prefix partitioning

In principle, dynamic l-prefixVoronoi partitioning allows using a varying number of nearest pivots to define each cell. So, a clustering objective function can be better modeled, which results to higher-quality visual vocabulary (Mikulik et al. 2010). In particular, to obtain good visual words, it is necessary to keep tracking the extent of cells during recursive re-partitioning to next levels. Compact cells with very close descriptors should not be split, so they use fewer pivots than l. To implement this, we maintain a *prefix tree*

that stores the maximum prefix length for each cell. This does not imply any major increase of memory requirements neither computational complexity, because it can be organized efficiently as a hashed associative array. The prefix tree has another important advantage since it allows dynamic expansion of the whole partitioning schema when some cells become overpopulated or clustering objective function has been refined, so it results in dynamic visual vocabulary. The prefix tree is initialized by monitoring cell extent during partitioning a training data-set T. The cost of this phase is linear to the training set size. So the construction time complexity is updated to $\mathcal{O}(D\,|T|\,\sqrt[s]{K})$ and the space complexity to $\mathcal{O}(D\sqrt[s]{K}+K)$. The transformation complexity stays obviously the same—this cost is increased by traversing the shallow prefix tree, thus it can be ignored. Such a tree traversal cost is ignored in complexity for the hierarchical k-means too. Please refer back to Table 1 for summary.

3.2 Combining MDPVs

We have identified the main weaknesses of MDPV as (1) the random selection of pivots to define Voronoi partitioning and (2) the limited number of cells that can contain any descriptor (Skala 2009). They both influence retrieval results negatively. Firstly, bad pivot selection may lead to some cells staying indivisible even by the full prefix length l. Secondly, the total number of cells in an l-prefix partitioning, theoretically stated as the number of permutations of l elements out of n ($l! \cdot C_l^n$), is impossible to reach (Skala 2009). In the following, we define a combination of independent MDPVs to diminish these negative effects. This principle is also used in ERCF (Moosmann et al. 2008) and RLSV (Mu et al. 2010).

Assume k different and independent sets of n pivots defining k l-prefixMDPVs, we define their combination as the function $k\text{-}\,mdpv_l : \mathcal{D} \mapsto \mathbb{R}^k$:

$$k\text{-}\,mdpv_l(f) = (mdpv_l^1(f), \ldots, mdpv_l^k(f)). \tag{4}$$

Thus, a vector of visual words is obtained. Figure 2 examples a combination of two l-prefix1 MDPVs of four pivots each, i.e. 2- $mdpv_1(\cdot)$ function.

From the figure, we can see that some cells in the combination cannot contain any descriptor. So even here in 2-D, four cells out of 16 possible combinations cannot contain any data: $C^{0,2}$, $C^{1,3}$, $C^{2,1}$ and $C^{3,1}$. This geometrical property of Voronoi partitioning is also discussed in Skala (2009), but combinations of independent partitionings create many more non-empty cells compared to a single l-prefixpartitioning with many pivots, which is also our practical experience. To ensure independence of pivots among MDPVs, a more sophisticated method for choosing pivots than the random selection can be adopted (Figueroa and Paredes 2014). Experiments we have undertaken show that even the random selection provides a good trade-off.

The vector obtained by k-MDPV can be interpreted in various ways: it can become a new visual word by simple concatenation of vector components; or a quorum can be defined on the number of MDPVs returning the same result for a pair of two descriptors. Such a voting scheme would provide similar effects as the soft assignment technique (Philbin et al. 2008). In the following, we take the whole vector as a new visual word. The rationale behind it is that an image database can be directly organized in an inverted file after quantization through these visual words.

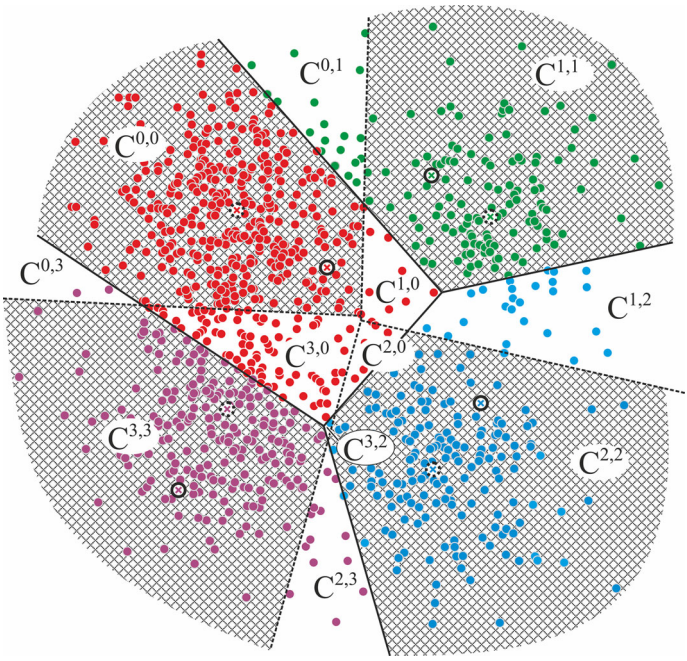

Fig. 2 Example of 2- $mdpv_1$—a combination of two l-prefix1 MDPVs with 4 pivots each. *Dashed lines* separate cells of the first MDPV, while *solid lines* the cells of the second MDPV. Cells resulting from the combination are denoted as $C^{x,y}$, where x and y are the evaluations of the first and the second MDPV, respectively

4 Experimental evaluation

The proposed MDPV approach was validated in two retrieval tasks on four well-known data-sets. We primarily focus on the performance in terms of retrieval accuracy, but we also include wall-clock time to construct a vocabulary and to transform a feature to a visual word.

Firstly, we analyze properties of MDPV-based vocabularies. Secondly, MDPV in image retrieval task is evaluated and compared with many state-of-the-art approaches that may not necessarily be based on visual vocabularies exclusively. Lastly, we present results in a near-duplicate video retrieval.

4.1 Evaluation protocol

We used the following image data-sets for the purpose of evaluation and comparison. They are typically shipped with SIFT descriptors extracted. If it is not the case, we used VLFeat library[1] to extract them.

Kentucky object recognition benchmark data-set[2] (denoted as UKBench) (Nister and Stewenius 2006) consists of 2,550 quadruples, each depicting the same object from four

[1] http://www.vlfeat.org/.

[2] http://www.vis.uky.edu/~stewe/ukbench/.

different angles. Thus, the ground truth is trivially defined. Each image's resolution is 640 by 480 pixels and we extracted about 13 million SIFT descriptors out of all 10,200 images.

Oxford Buildings data-set[3] (denoted as Oxford) (Philbin et al. 2007) is formed by 5,062 images downloaded from Flickr and expose several landmarks of Oxford. The authors provide five queries for each of 11 selected landmarks and list correct answers. There are 15,877,710 SIFT descriptors available.

Belga Logos data-set[4] (denoted as Belga) (Joly and Buisson 2009) is composed of 10,000 images covering all aspects of life and current affairs. Maximum width of each image is 800 pixels. This data-set has 26 queries containing company logos. Ground-truth in the form of a list of images containing a particular logo is available. We extracted 19.5 million SIFT descriptors in total.

INRIA Holidays data-set[5] (denoted as Holidays) (Jégou et al. 2008) contains 1,491 personal holiday photos, where 500 of them are queries and the remaining 991 are corresponding relevant images. Ground-truth is again available. The data-sets is shipped with 4,455,091 SIFT descriptors.

CCWebVideo data-set[6] was used in near-duplicate retrieval task. It contains 12,790 short video clips (up to 10 min) collected from YouTube, Google Video, and Yahoo! Video by issuing 24 textual queries (Wu et al. 2007, 2009). Ground-truth files defining near-duplicate videos are available. The authors also provide key-frames corresponding to shot boundaries. We used HSV color histograms as descriptors here. In particular, a 24-dimensional HSV color histogram was obtained each key-frame.

To compare our approach with other approaches, we use the *mean average precision* (mAP) and the results presented were typically obtained as averages over five independent runs of MDPV. A prototype of MDPV was implemented in Java 7 and experiments were done on a common PC with a dual-core 3 GHz CPU and 4 GB RAM.

4.2 Basic analysis

In this part, we study different parameters and properties of MDPV in a number of experimental trials. We focus namely on adjusting the prefix length parameter, which inherently influences vocabulary size, its memory requirements, time to build the vocabulary, and time to quantize a feature.

Maximum prefix length. Firstly, we would like to state the maximum value of l to be used to define an l-prefixVoronoi partitioning. A sub-set of five million SIFT descriptors was taken from the Oxford data-set and used to initialize a prefix tree for various number of pivots. We observed the number of non-empty cells on each level. The objective function used to trigger splitting a cell was set to the cell occupation in the number of data objects—1,024 SIFT descriptors in each cell at maximum. Figure 3 shows the relative number of non-empty cells for each level of prefix tree and for varying number of pivots that define the partitioning. The presented values are averages over five runs. The outcome of this experiment is that the maximum prefix length $l = 6$ is sufficient for constructing visual vocabularies, since the majority of cells (> 80 %) are created on the third and fourth level for 40 and more pivots and the number of cells to be split on the sixth level is marginal.

[3] http://www.robots.ox.ac.uk/∼vgg/data/oxbuildings/.

[4] http://www-sop.inria.fr/members/Alexis.Joly/BelgaLogos/BelgaLogos.html.

[5] http://lear.inrialpes.fr/∼jegou/data.php.

[6] http://vireo.cs.cityu.edu.hk/webvideo/.

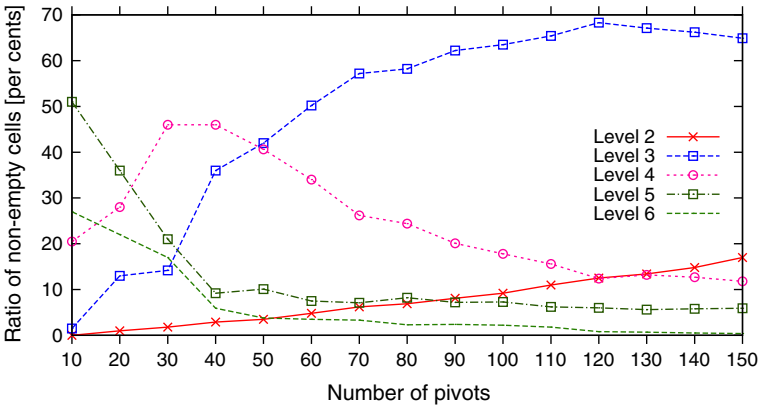

Fig. 3 Ratio of non-empty cells on individual levels of MDPV's prefix tree to the total number of non-empty cells

Vocabulary size. We compare MDPV to the standard solution of k-means-based visual vocabulary. We tested various hierarchical k-means algorithms described in Kanungo et al. (2002) to build Vocabulary Tree. The EZ-hybrid method exhibited best results, so we adopted this technique and denote it simply as "k-means" in results. Figure 4 reports on the number of non-empty cells (i.e., valid visual words) in MDPV and k-means algorithms for the UKBench and Oxford data-sets separately. A sample of five million descriptors was taken to set up k-means and prefix tree of MDPV.

Vocabulary tree (hierarchical k-means) of six levels and ten cells in each node was used to build a visual vocabulary with one million words maximally. In the figure, the reader may observe that it is not always possible to obtain full vocabulary tree on a data sample since some nodes (clusters) may become empty earlier than on the sixth level.

MDPV was compared in both *basic* and *dynamic* variants. In the former, the full prefix of $l = 6$ was used and no initialization with sample data was done, so the number of non-empty cells corresponds to the number of unique visual words created by quantizing the whole sample set. The later variant results to comparatively much fewer non-empty cells. However, each cell corresponds to many more descriptors (up to 1,024) here, so their visual similarity is better handled. Figure 4b details the number of visual words created by the dynamic variant of MDPV on Oxford and UKBench data-sets. The pivots were taken at random in all cases.

This group of experiments gave an overview of visual vocabulary size and comparison among various settings of MDPV and k-means. Even though the dynamic prefix partitioning produces a small number of cells, we will show its good results in retrieval.

Memory requirements. Table 2 summarizes MDPV's memory occupation in bytes. The basic variant of MDPV requires to store pivots only, whereas the dynamic one adds an associative array of valid visual words implemented as `HashSet` organizing `Long` integers. Our implementation in 64-bit Java 7 requires 290 bytes per SIFT descriptor (pivot) and approximately 56 bytes per visual word. These results are in compliance with MDPV's space complexity stated in Table 1. The largest MDPV configuration requires 3 MiB only, which is in a sharp contrast of 241 MiB required by hierarchical k-means to define its 879,569 visual words. For up-to-date server configurations, memory occupation of several tens of megabytes may not be an issue. However, if we accept an image search

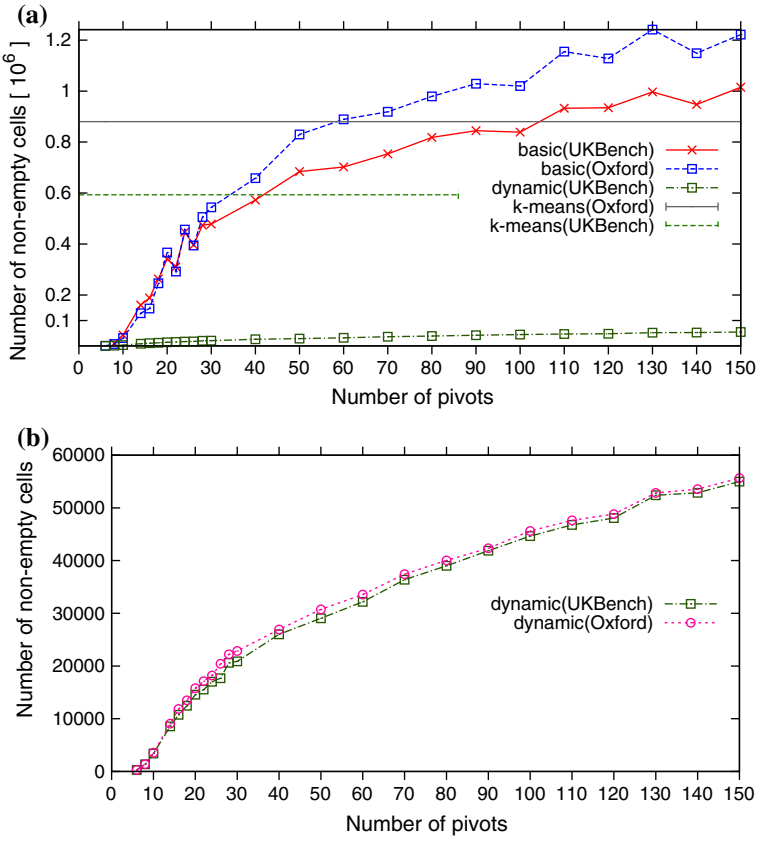

Fig. 4 Number of visual words generated by $mdpv_6$ for varying the number of pivots selected from UKBench and Oxford data-sets. The number of visual words refers to the number of unique cell IDs obtained by transforming the whole data-set (Oxford and UKBench). **a** Comparison of basic MDPV, dynamic MDPV and hierarchical k-means on UKBench and Oxford. **b** Details on dynamic MDPV variants on UKBench and Oxford

engine can also become part of an application in smart-phones in the future, the compactness of individual techniques will become important.

Building requirements. We used a C++ optimized implementation of hierarchical k-means to prepare a vocabulary on 5 million sample of Oxford data-set and it took 178 h to get the visual words. Whereas, the dynamic variant of MDPV (in Java) over 150 pivots and the same sample was initialized in 2 h, so MDPV was 89 times faster. Since the limit on the assignment/update loop in k-means was set to 100 (the variable i), such results correspond to the complexities stated in Table 1.

Transformation time. Since the transformation of a feature descriptor to a visual word is rather a trivial operation in MDPV that uses few pivots, we have implemented the search for l nearest neighbors as full scan and we do not include any experimental results here. On the other hand, this is certainly an issue for large vocabularies created by k-means, where the lookup for one nearest neighbor in a collection of one million centroids cannot be done efficiently by full scan. Approximate nearest-neighbor search with k-d forest as is

Table 2 Comparison of MDPV memory requirements with hierarchical k-means on Oxford data-set

Alg.	Number of pivots				
	10	40	60	100	150
MDPV (basic)	2,904 B	11,544 B	17,304 B	28,824 B	43,224 B
MDPV (dynamic prefix)	193,808 B	1,467,544 B	1,818,992 B	2,528,888 B	3,122,104 B
k-means	253,315,896 B (241 MiB)				

implemented in the FLANN library (Muja and Lowe 2009), is then a suitable solution. If the hierarchical k-means is used to prepare a large vocabulary, a shallow tree with few centroids in each node is a typical representation. For example, 6 levels and 10 pivots in each node creates 1 million visual words, but the transformation to a visual word requires 60 distance evaluation only.

4.3 Variants of MDPV

This section is devoted to comparison of basic and dynamic variants of MDPV in image retrieval effectiveness. In particular, we aim at showing the positive influence of extensions to the basic MDPV design proposed in Sect. 3. Please recall, the extensions are the dynamic prefix Voronoi partitioning and the combination of more independent partitionings.

Retrieval effectiveness is justified by evaluating queries on UKBench data-set and measuring mAP. Figure 5a exposes the combinations of 1, 3, and 5 dynamic prefix MDPVs (k- $mdpv_6$ for $k = 1, 3, 5$). We can directly observe a positive effect on 3- $mdpv_6$ over 1- $mdpv_6$. The visual vocabulary of 3- $mdpv_6$ is fine-grained enough to filter out false positives. On the other hand, 5- $mdpv_6$ vocabulary is too fine for this kind of image-retrieval. It is competitive with small number of pivots only (from 15 to 20). Combinations of more MDPVs have also a positive effect on the stability of results, where the values of mAP are quite stable for k-MDPV with at least 40 pivots.

We also studied the influence of pivot selection. In particular, Fig. 5b depicts the influence of source of pivots used to prepare prefix trees in 3-MDPV. We compared results for pivots taken from the data-set itself (UKBench), pivots selected from Oxford, and synthetically generated pivots (random sample from uniformly distributed vectors generated in $[0, 255]^{128}$ Euclidean space). Synthetic pivots lead to unsatisfactory results which are even worse than the results achieved by the basic design of MDPV (full-length prefix). MDPV initialized with pivots picked from Oxford data-set exhibit only slightly worse results in retrieval of UKBench images than the configuration with pivots from UKBench data-set itself.

The outcome of this group of experiments is that 3-MDPV with 40–80 pivots provides a good solution to create visual vocabulary. Behavior similar to the results on UKBench was experienced on the INRIA Holidays and Belga Logos data-sets. So, MDPV can be perceived as a general purpose technique for image retrieval, since these three data-sets represent instances of different image retrieval tasks. Please recall UKBench verifies performance of *object retrieval* with respect to rotation invariance. INRIA Holidays data-set stands for search in *general holiday photographs*, while Belga Logos tests quality of *sub-image retrieval*. It has also been shown that MDPV initialized using a set of real-life pivots from a different data-set can be successfully used, so MDPV does not suffer from the so-called out-of-sample problem.

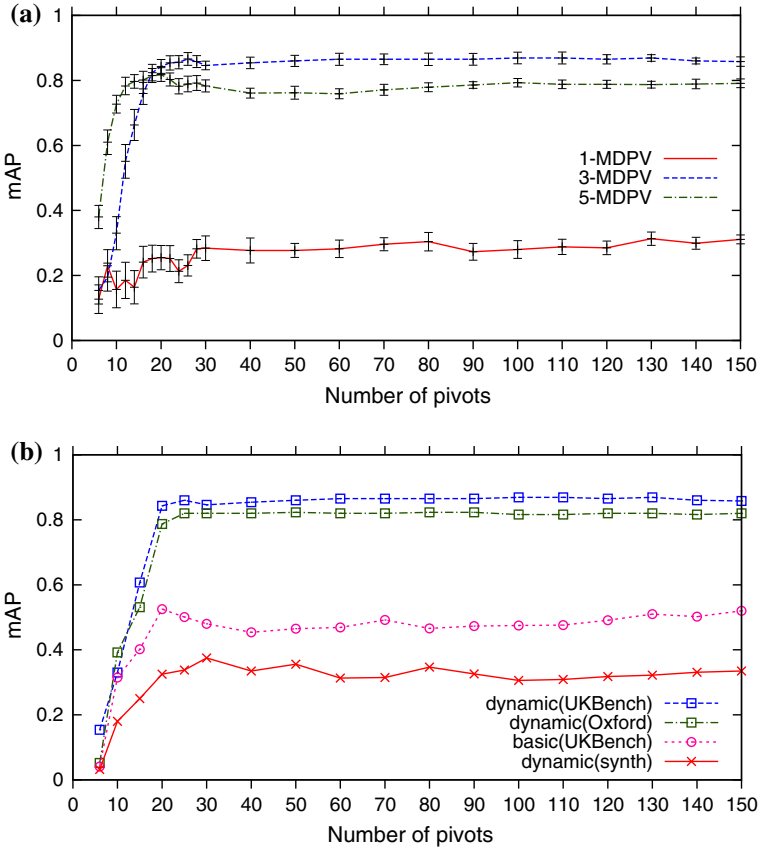

Fig. 5 Retrieval performance on UKBench data-set. **a** Combinations of MDPVs (1-, 3- and 5-$mdpv_6$) initialized with pivots from UKBench. **b** Comparison of 3-$mdpv_6$ initialized with pivots from different data-sets (UKBench, Oxford and synthetic) with the original proposal (full-length prefix)

4.4 Comparison with others in image retrieval

In this section, we present performance comparison of MDPV with several state-of-the-art methods in image retrieval task on UKBench, Belga and Holidays data-sets. Namely, we compare our results with Vocabulary Tree (Nister and Stewenius 2006), RLSV (Mu et al. 2010), min-hash (Chum et al. 2008), sim-min-hash (Zhao et al. 2013), weighted Hamming Embeddings with multiple assignment (Jégou et al. 2010a), contextual weighting of local features (Wang et al. 2011), and VLAD (Jégou et al. 2012). As for the vocabulary tree, we took the recommended configuration of tree—ten elements in each node and six levels. This leads to a tree representing up to one million visual words. In Wang et al. (2011), even larger vocabulary tree (10^7 words) is used and contextual weighting of local descriptors is added. We use 3-$mdpv_6$ with dynamic prefix tree prepared on pivots selected at random from the same data-set as the descriptors to be quantized.

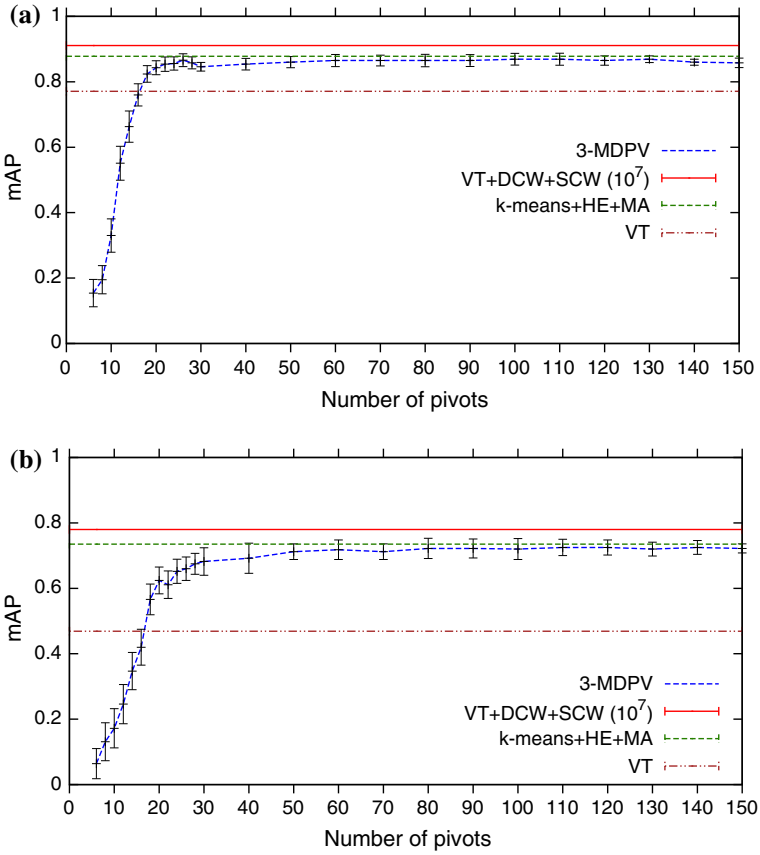

Fig. 6 Results for 3-$mdpv_6$ and k-means based state-of-the-art methods on UKBench and Holidays data-sets. Mean and standard deviation at MDPV is over five runs. **a** UKBench data-set. **b** INRIA Holidays data-set

Figure 6 shows results on UKBench and Holidays data-sets for varying the number of pivots used to define MDPV. Moreover, it contains results for the following state-of-the-art k-means-based methods: original vocabulary tree (VT) (Nister and Stewenius 2006), vocabulary tree with descriptor and spatial contextual weighting (VT+DCW+SCW) (Wang et al. 2011), and hierarchical k-means with weighted Hamming embedding and multiple assignment (k-means+HE+MA) (Jégou et al. 2010a). From the results, the combination of three MDPVs is by far better than simple vocabulary tree and it reaches the performance of VT with several improvements. Only VT with contextual weighting extension (VT+DCW+SCW) is better by 10 % on both the data-sets.

In Table 3, we present the results for all comparable methods on UKBench, Holidays and Belga data-sets. On UKBench, MDPV is outperformed by methods extending vocabulary tree with descriptor and spatial contextual weighting (DCW+SCW) (Wang et al. 2011) and with Hamming embedding and multiple assignment (Jégou et al. 2010a). On Holidays data-set, only the DCW+SCW method with 10^7 visual words provides higher accuracy in comparison to MDPV. On Belga data-set, MDPV clearly outperformed all

Table 3 Comparison of MDPV with state-of-the-art methods

Algorithm	mAP	Top 4
Data-set UKBench		
VT + DCW + SCW ($k = 10^7$) (Wang et al. 2011)	0.911	3.56
VT + DCW + SCW ($k = 10^6$) (Wang et al. 2011)	0.897	3.46
k-means + weighted HE + MA (Jégou et al. 2010a)	0.878	3.42
Fisher vector ($K = 64, D' = 4096$) (Jégou et al. 2012)	–	3.35
VLAD ($K = 64, D' = 4096$) (Jégou et al. 2012)	–	3.28
min-hash (Chum et al. 2008)	–	3.17
3-MDPV (3- $mdpv_6(\cdot)$, 19 pivots)	0.834	3.14
VT (Jegou et al. 2009)	0.771	2.95
sim-min-hash (Zhao et al. 2013)	–	2.70
VT, normal	0.420	2.54
Data-set holidays		
VT+DCW+SCW ($k = 10^7$) (Wang et al. 2011)	0.780	
3-MDPV (3- $mdpv_6(\cdot)$, 60 pivots)	0.761	
VT+DCW+SCW ($k = 10^6$) (Wang et al. 2011)	0.744	
k-means + weighted HE+MA (Jégou et al. 2010a)	0.735	
VLAD with LCS, RN ($K = 64$) (Delhumeau et al. 2013)	0.658	
Fisher vector ($K = 64, D' = 4096$) (Jégou et al. 2012)	0.595	
VT (Jegou et al. 2009)	0.469	

Data-set Belga	mAP Full data-set	mAP Subset
3-MDPV (3- $mdpv_6(\cdot)$, 19 pivots)	0.342	0.661
Joly (Joly and Buisson 2009)	0.257	0.563
VT ($k = 10^5$)	0.263	0.309
RVP (Jiang et al. 2012)	–	0.295
VT ($k = 10^6$)	0.176	0.192

Results are sorted descendingly by mAP values

VT vocabulary tree, *k* is the number of visual words, *HE* Hamming embedding, *MA* multiple assignment, *DCW* descriptor contextual weighting, *SCW* spatial contextual weighting, *VLAD* vector of aggregated local descriptors, *RVP* random visual phrases

methods. Here, RVP approach (Jiang et al. 2012) was tested on a subset of 6 queries only, so we present results for all queries and the subset.

To sum up, a combination of three MDPVs, each defined on about 50 pivots, is a very competitive configuration on any data-set, especially if we judge MDPV's definition is based purely on permutation of close pivots without any additional statistical enhancements. The results of VT+DCW+SCW method are counterbalanced by the following negatives. Firstly, the time spent in both contextual weightings (DCW+SCW) during the vocabulary preparation and descriptor quantization time (transformation time) is not marginal. Secondly, memory requirements are increased because visual words stored in vocabulary tree are enriched with this weighting information. Finally, due to the statistical principles of these weightings, the out-of-sample problem arises when previously unseen

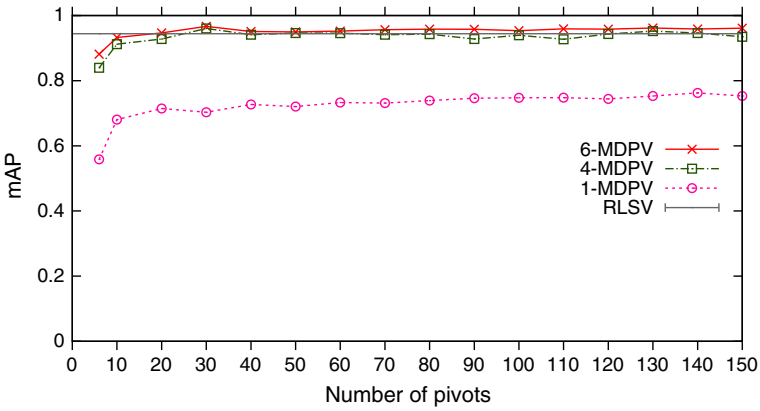

Fig. 7 Performance of several combinations of MDPV and RLSV on CCWebVideo data-set

Table 4 Comparison of 6- $mdvp_6$ over 30 pivots with RLSV and k-means on CCWebVideo

Algorithm	mAP	Construction time (s)	Conversion to visual word (s)
6-MDPV	0.9669	$<1 \times 10^{-6}$	5.0×10^{-6}
RLSV	0.9411	2.29×10^{-4}	3.4×10^{-3}
k-means	0.9280	7.76	1.85×10^{-2}

images are to be stored in the database. Nonetheless, we plan to employ contextual weighting enhancements to boost MDPV's performance further.

4.5 Near-duplicate video retrieval

Lastly, we demonstrate MDPV in the task of near-duplicate video retrieval on the CCWebVideo data-set. Each video is described by a sequence of descriptors extracted from video key-frames. Descriptors are 24-dimensional HSV color histograms extracted by the procedure described in Mu et al. (2010). To measure similarity between a query video sequence and a database sequence, we applied a global sequence alignment algorithm (namely, Needleman-Wunsch), which is a different approach from the trials in previous sections. This similarity was used to rank all database sequence according to a query and mAP was computed against the data-set's ground-truth. MDPV was used to quantize the feature descriptors to visual words to speed up evaluations of the similarity function.

Combinations of up to six MDPVs were tested and the results for selected combinations are exposed in Fig. 7. To compare with RLSV, we included the best average mAP value obtained by a hierarchical method of RLSV (Mu et al. 2010). In this type of retrieval, a combination of more independent MDPVs leads to a fine-grained visual vocabulary that can effectively filter out replicas. Best results were obtained for 6- $mdpv_6$ with at least 30 pivots. Table 4 summarizes also the performance indicators captured during trials' evaluation. The results confirm the MDPV's outstanding speed and effectiveness over the other state-of-the-art methods.

5 Conclusion

We have introduced a novel approach to quantization of high-dimensional vectors in image feature spaces called Metric Distance Permutation Vocabulary (MDPV). It is based on an *l*-prefix Voronoi tessellation defined in metric spaces. The advantages over existing techniques for visual vocabulary construction are very low memory requirements, no expensive training/learning phase, and very efficient transformation of a feature descriptor to a visual word. Our method has been validated in two computer vision tasks: image retrieval and near-duplicate video retrieval. MDPV shows very good accuracy and efficiency in comparison to state-of-the-art methods such as Randomized Locality Sensitive Vocabulary, Vocabulary Tree, k-means with hamming embedding and min-hash. We have also proposed an extension that tunes granularity of Voronoi partitioning to a particular data-set to further improve performance. A recommended configuration is a combination of three MDPVs each built over about 50 pivots for the task of image retrieval and six independent MDPVs over the same number of pivots for near-replica detection task. Pivots can be picked at random from a real-life data-set that may not be the same as the data-set to organize but similar in its nature. This is another advantage—higher resistance to out-of-sample problem that methods creating visual vocabularies usually suffer from.

Our hypothesis that the visual words created by MDPV are good is based not only on the result we have presented but also on the results of other researchers. As it was shown and analyzed in Chavez et al. (2008); Esuli (2012), the sequence of identifications of the closest pivots that describes a database object, is precise enough to substitute the original distance function directly. In this paper, we exploited this outcome by taking only a prefix of the whole permutation to define a feature quantization technique for the bag-of-features model.

For the future work, we would like to focus on the issue of comparing vectors resulting from a combination of MDPVs. A perspective solution would be a voting schema, where only a certain quorum of complying MDPV components has to be reached. Secondly, we will analyze possibilities to employ various contextual information (Wang et al. 2011) in MDPV design. Finally, we would like to focus on other issues of sub-image retrieval task, such as pattern re-occurrence.

Acknowledgments This work was supported by the Project No. P103/12/G084 by the Czech Science Foundation. To execute experimental trials, an access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme "Projects of Large Infrastructure for Research, Development, and Innovations" No. LM2010005, is greatly appreciated.

References

Amato, G., Bolettieri, P., Falchi, F., & Gennaro, C. (2013). Evaluating inverted files for visual compact codes on a large scale. In *Proceedings of 10th international workshop on large-scale and distributed systems for information retrieval (LSDS-IR), co-located with ACM WSDM* (pp. 44–49). Roma, Italy: ACM Press.

Amato, G., Gennaro, C., & Savino, P. (2014). MI-file: using inverted files for scalable approximate similarity search. *Multimedia Tools and Applications*, *71*(3), 1333–1362.

Andoni, A., & Indyk, P. (2008). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *ACM Communications*, *51*(1), 117–122.

Aurenhammer, F. (1991). Voronoi diagrams—A survey of a fundamental geometric data structure. *ACM Computing Surveys*, *23*(3), 345–405.

Batko, M., Falchi, F., Lucchese, C., Novák, D., Perego, R., Rabitti, F., et al. (2010). Building a web-scale image similarity search system. *Multimedia Tools and Applications, 47*. http://www.springerlink.com/content/u6112378t8k63382/.

Bay, H., Ess, A., Tuytelasrs, T., & Vangool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346–359.

Bergamo, A., Sinha, S. N., & Torresani, L. (2013). Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 763–770). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6618948.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "Nearest Neighbor" meaningful? In *Proceedings of the international conference on database theory (ICDT)* (pp. 217–235).

Bigun, J. (1992). Unsupervised feature reduction in image segmentation by local Karhunen–Loeve transform. In *Proceedings of the 11th IAPR international conference on pattern recognition, 1992. Vol. II. conference B: Pattern recognition methodology and systems* (pp. 79–83).

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Broder, A. (1997). On the resemblance and containment of documents. In *Proceedings of the compression and complexity of sequences 1997* (pp. 21–29). doi:10.1109/SEQUEN.1997.666900.

Chadha, A., Vaidya, P. P., & Roja, M. M. (2011). Face recognition using discrete cosine transform for global and local features. *Computer Vision and Pattern Recognition*. abs/1111.1423.

Chandrasekhar, V., Takacs, G., Chen, D., Tsai, S. S., Singh, J., & Girod, B. (2009). Transform coding of image feature descriptors. In *Proceedings of SPIE, visual communications and image processing 2009* (vol. 7257, pp. 725710–725719).

Chavez, E., Figueroa, K., & Navarro, G. (2008). Effective proximity retrieval by ordering permutations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(9), 1647–1658. doi:10.1109/TPAMI.2007.70815.

Chum, O., Philbin, J., & Zisserman, A. (2008). Near duplicate image detection: Min-hash and tf-idf weighting. *Proceedings of the British Machine Vision Conference*, *3*, 4.

Chum, O., Urban, M., Pajdla, T., & Matas, J. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, *22*(10), 761–767.

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. *Earth*, *1*(May), 22.

Delhumeau, J., Gosselin, P. H., Jégou, H., & Pérez, P. (2013). Revisiting the vlad image representation. In *Proceedings of the 21st ACM international conference on multimedia, MM '13* (pp. 653–656). ACM, New York, NY, USA. doi:10.1145/2502081.2502171.

Devaux, J. C., Gouton, P., & Truchete, F. (2000) Aerial colour image segmentation by Karhunen–Loeve transform. In *Proceedings of the 15th international conference on pattern recognition, 2000* (vol. 1, pp. 309–312).

Esuli, A. (2012). Use of permutation prefixes for efficient and scalable approximate similarity search. *Information Processing and Management*, *48*(5):889–902; large-Scale and Distributed Systems for Information Retrieval.

Figueroa, K., & Paredes, R. (2014). An effective permutant selection heuristic for proximity searching in metric spaces. In *Proceedings of Mexican conference on pattern recognition (MCPR)*, Lecture Notes in Computer Science (pp. 1–10). Berlin: Springer.

Grzegorzek, M., Sav, S., O'Connor, N. E., & Izquierdo, E. (2010). Local wavelet features for statistical object classification and localization. *IEEE MultiMedia*, *17*(1), 118–128.

Jégou, H., Douze, M., & Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. In D. Forsyth, A. Z. Philip Torr (Eds.), *European conference on computer vision, LNCS* (vol. I, pp. 304–317). Berlin: Springer.

Jegou, H., Douze, M., & Schmid, C. (2009). On the burstiness of visual elements. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1169–1176).

Jégou, H., Douze, M., & Schmid, C. (2010a). Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, *87*(3), 316–336.

Jegou, H., Douze, M., Schmid, C., Perez, P. (2010b). Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition* (pp. 3304–3311), IEEE.

Jegou, H., Douze, M., & Schmid, C. (2011). Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(1), 117–128.

Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., & Schmid, C. (2012). Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(9), 1704–1716. doi:10.1109/TPAMI.2011.235.

Jiang, Y., Meng, J., & Yuan, J. (2012). Randomized visual phrases for object search. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3100–3107).

Joly, A., & Buisson, O. (2008). A posteriori multi-probe locality sensitive hashing. In *Proceeding of the 16th ACM international conference on multimedia—MM '08* (p. 209). New York, New York, USA: ACM Press.

Joly, A., & Buisson, O. (2009). Logo retrieval with a contrario visual query expansion. In *MM '09: Proceedings of the seventeen ACM international conference on multimedia* (pp. 581–584).

Kang, Z., Ooi, W. T., & Sun, Q. (2004). Hierarchical, non-uniform locality sensitive hashing and its application to video identification. In *2004 IEEE international conference on multimedia and expo 2004 (ICME '04)* (vol. 1, pp. 743–746).

Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., & Wu, A. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 881–892.

Kulis, B., & Grauman, K. (2009). Kernelized locality-sensitive hashing for scalable image search. In *2009 IEEE 12th international conference on computer vision* (pp. 2130–2137), IEEE.

Kyselak, M., Novak, D., & Zezula, P. (2011). Stabilizing the recall in similarity search. In *Proceedings of the fourth international conference on similarity search and applications (SISAP)* (p. 43). New York, New York, USA: ACM Press.

Lim, J., Kim, Y., & Paik, J. (2009). Comparative analysis of wavelet-based scale-invariant feature extraction using different wavelet bases. In *Signal processing, image processing and pattern recognition, communications in computer and information science* (vol. 61, pp. 297–303). Berlin, Heidelberg: Springer.

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

Lu, W., Shen, Y., Chen, S., & Ooi, B. C. (2012). Efficient processing of k nearest neighbor joins using mapreduce. *PVLDB*, *5*(10), 1016–1027.

Makar, M., Chang, C. L., Chen, D., Tsai, S. S., & Girod, B. (2009). Compression of image patches for local feature extraction. In *IEEE international conference on acoustics, speech and signal processing 2009 (ICASSP 2009)* (pp. 821–824).

Micó, M. L., Oncina, J., & Vidal, E. (1992). An algorithm for finding nearest neighbors in constant average time with a linear space complexity. In *Proceedings of the 11th international conference on pattern recognition (ICPR 1992)* (vol. II, pp. 557–560), The Hague, The Netherlands.

Mikulik, A., Perdoch, M., Chum, O., & Matas, J. (2010). Learning a fine vocabulary. In *Computer vision, ECCV 2010* (pp. 1–14).

Moosmann, F., Nowak, E., & Jurie, F. (2008). Randomized clustering forests for image classification. *IEEE Transcations on Pattern Analysis and Machine Intelligence*, *30*(9), 1632–1646.

Mu, Y., Sun, J., Han, T. X., Cheong, L. F., & Yan, S. (2010). Randomized locality sensitive vocabularies for bag-of-features model. In *European conference on computer vision (ECCV), lecture notes in computer science* (vol. 6313, pp. 748–761). Berlin: Springer.

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *International conference on computer vision theory and application VISSAPP'09)* (pp 331–340), INSTICC Press.

Nister, D., & Stewenius, H. (2006). *Scalable recognition with a vocabulary tree*, IEEE.

Novak, D., & Batko, M. (2009). Metric index: An efficient and scalable solution for similarity search. In *2009 Second international workshop on similarity search and applications* (pp. 65–73), IEEE.

Perronnin, F., Liu, Y., Sanchez, J., & Poirier, H. (2010). Large-scale image retrieval with compressed fisher vectors. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3384–3391). doi:10.1109/CVPR.2010.5540009.

Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE conference on computer vision and pattern recognition* (pp. 1–8), IEEE.

Schwerin, B., & Paliwal, K. (2008). Local-DCT features for facial recognition. In *2nd International conference on signal processing and communication systems, 2008 (ICSPCS 2008)* (pp. 1–6).

Skala, M. (2009). Counting distance permutations. *Journal of Discrete Algorithms*, *7*(1), 49–61.

Song, X., Jiao, L., Yang, S., Zhang, X., & Shang, F. (2013). Sparse coding and classifier ensemble based multi-instance learning for image categorization. *Signal Processing*, *93*(1), 1–11.

Strecha, C., Bronstein, A., Bronstein, M., & Fua, P. (2011). LDAHash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(1), 66–78.

Turcot, P., & Lowe, D. G. (2009). Better matching with fewer features: The selection of useful features in large database recognition problems. In *2009 IEEE 12th international conference on computer vision workshops* (pp. 2109–2116). ICCV Workshops, IEEE.

Wang, X., Yang, M., Cour, T., Zhu, S., Yu, K., & Han, T. (2011). Contextual weighting for vocabulary tree based image retrieval. In *IEEE International conference on computer vision (ICCV)* (pp. 209–216).

Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. *Proceedings of IEEE Internation Conference on Computer Vision*, *2*, 1800–1807.

Wu, X., Hauptmann, A. G., & Ngo, C. W. (2007). Practical elimination of near-duplicates from web video search. In *Proceedings of the 15th international conference on multimedia (Multimedia 2007)* (pp. 218–227). New York, NY, USA: ACM. doi:10.1145/1291233.1291280.

Wu, X., Ngo, C. W., Hauptmann, A. G., & Tan, H. K. (2009). Real-time near-duplicate elimination for web video search with content and context. *IEEE Transactions on Multimedia*, *11*(2), 196–207.

Yang, L., Jin, R., Sukthankar, R., & Jurie, F. (2008a). Unifying discriminative visual codebook generation with classifier training for object category recognition. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)* (pp. 1–8), IEEE Computer Society.

Yang, L., Jin, R., Sukthankar, R., & Jurie, F. (2008b). Unifying discriminative visual codebook generation with classifier training for object category reorganization. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition (CVPR)* (pp. 1–8), IEEE Computer Society.

Yin, S., Badr, M., & Vodislav, D. (2013). Dynamic multi-probe LSH: An I/O efficient index structure for approximate nearest neighbor search. In H. Decker, L. Lhotska, S. Link, J. Basl, & A. Tjoa (Eds.), *Database and expert systems applications, lecture notes in computer science* (vol. 8055, pp. 48–62). Berlin, Heidelberg: Springer.

Zezula, P., Amato, G., Dohnal, V., & Batko, M. (2006). *Similarity search: The metric space approach, advances in database systems* (vol. 32). Berlin: Springer.

Zhao, W. L., Jégou, H., & Gravier, G. (2013). Sim-min-hash: An efficient matching technique for linking large image collections. In *ACM multimedia conference (MM)* (pp. 577–580). Barcelona, Spain: ACM.