IMPROVING VLAD: HIERARCHICAL CODING AND A REFINED LOCAL COORDINATE SYSTEM

Christian Eggert, Stefan Romberg, Rainer Lienhart

Multimedia Computing and Computer Vision Lab
University of Augsburg

ABSTRACT

The enormous growth of image databases calls for new techniques for fast and effective image search that scales with millions of images. Most importantly, the setting requires a compact but also descriptive image signature. Recently, the vector of aggregated local descriptors (VLAD) [1] has received much attention in large-scale image retrieval. In this paper we present two modifications for VLAD which improve the retrieval performance of the signature.

***Index Terms*—** VLAD, image retrieval, compact signatures, LCS

1. INTRODUCTION

In the field of image search, the bag-of-features (BoF) approach [2] has become very popular during the last decade. Hereby, an image is described by local features where the corresponding descriptors are quantized into discrete visual words. The image is then represented by a histogram of weighted visual word occurrences. The advantages of this representation are simplicity, robustness to occlusion, clutter and other image transformations, as well as the reduction of the high-dimensional image descriptors into a single histogram vector. However, despite its success, a simple *bag* of visual words still requires at least the memory to store the several thousand visual words it contains. Furthermore, for efficiency, the search is usually performed via an inverted index where each individual visual word is issued as query to the index. In contrast, the representation of an image by a vector of aggregated local descriptors (VLAD) [1] offers an effective way to encode the information from all the local feature descriptors into a single compact vector. In combination with PCA and product quantization, images can be represented by a signature with a size of a few bytes only. In contrast to other global features, VLAD still builds on the robustness and the descriptive power of local features while being a compact signature that can be used with image databases of several million images.

2. RELATED WORK

There is much work on global features that represent a whole image as scene within a single description. For instance, hashing-based methods such as Spectral Hashing [3] represent images by compact bit strings where the hamming distance between these resembles the underlying similarity between the corresponding images. This representation requires a few bits only, but the similarity search tends to work well only for images that are globally similar.

One technique inspired by random projection creates multiple mini-bag-of-words by randomly selecting components from the original bag-of-words histogram [4]. For retrieval, multiple queries are then issued with these mini-BoFs and the result sets are merged.

An alternative technique that bundles multiple visual words into a small number of sketches that describe the whole image is Geometric min-Hashing [5]. While a retrieval based on this particular method has high precision, recall is usually low, making steps like query expansion necessary.

Another way to compress bag-of-words histograms is to employ topic models such as pLSA and LDA. Instead of modelling the images as high-dimensional histograms of visual words, the image is represented by a mixture of (a few) topics [6], which require less memory than the full BoF histograms. However, the underlying generative representation tends to map many (conceptually) different visual words into the same topic yielding false positives during retrieval.

The Fisher vector [7] aims to overcome this problem. It combines a generative Gaussian mixture model with a discriminative coding scheme and yields a highly distinctive image signature which was successfully used for retrieval and image classification. Its downside are its computational complexity and increased memory requirements compared to VLAD.

3. VLAD

3.1. The original VLAD pipeline

The vector of locally aggregated descriptors (VLAD) describes an image by the difference of its local feature de-

We would like to thank the DFG for their financial support.

scriptors from a learned codebook. For that, VLAD utilizes a coarse visual codebook $q : X \rightarrow C$, $C = \{c_1, c_2, \ldots, c_k\}$, that has been learned offline and maps image descriptors to a set of centroids of size k. Here, X denotes the descriptor space and C the set of centroids. Typically, such a visual codebook is obtained by k-means clustering of descriptors of a training dataset. Common choices for the number of clusters range from $k = 32$ to $k = 256$.

Given an image represented by a set of m local descriptors $I = \{\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x_m}\}$, the original VLAD representation is obtained by encoding the descriptors in the following way:

$$\mathbf{v}_i = \sum_{\mathbf{x} \in I : q(\mathbf{x}) = \mathbf{c_i}} \mathbf{x} - \mathbf{c_i} \tag{1}$$

That is, each local descriptor is assigned to its nearest centroid and the residual with this centroid is computed. The residuals of all descriptors with centroid $\mathbf{c_i}$ are accumulated. Each centroid $\mathbf{c_i}$ in the codebook contributes a vector of aggregated residuals. The final VLAD signature $\mathbf{v}$ is obtained by concatenating the residual vectors $\mathbf{v_i}$ forming a $D = k \times d$ dimensional image signature where d is the dimensionality of the original descriptors. Finally, the concatenated vector $\mathbf{v}$ is L_2-normalized. In the following we refer to the resulting vectors as *uncompressed signatures*.

Once the coding is done, compression is performed in two rounds to minimize the required memory: First, a PCA is applied and only the components with the largest variance are retained. Typically, the number of retained dimensions D' is tuned for best performance, but $D' = 128$ has been reported [1] to be a good choice for most configurations. After the dimensionality reduction, the variance of the components is optionally rebalanced prior to product quantization. For this, a random but fixed orthogonal transformation is applied to the compressed vector. Finally, the resulting vector is then compressed by product quantization to a short code vector.

3.2. Further improvements

Power-Law Normalization: Bursty features can be caused by repeating structures in the original image and are known to be able to corrupt in the similarity metric [8] in BoF image retrieval as they might dominate other descriptors which are more useful to estimate similarity. The power-law normalization [9] given by

$$\widetilde{v_j} = sgn(v_j) \left| v_j^{\alpha} \right| \tag{2}$$

was proposed to downweight bursty components. In equation 2, $\alpha \in [0, 1]$ is a normalization parameter that needs to be tuned for best performance. It has been suggested that $\alpha = 0.2$ is a good choice [10]. The power-law normalization is applied prior to the L_2-normalization of the VLAD descriptor. Delhumeau suggested [10] that the effectiveness of the power-law normalization could be improved by transforming the descriptors to another coordinate system which is better suited for handling bursty components. This projection is learned through a PCA without dimensionality reduction. Further improvements can be achieved by learning one PCA projection per Voronoi cell [10], which allows to address a more diverse range of bursty patterns as the PCA can now adapt to the characteristics of each Voronoi cell. This scheme has been termed the *local coordinate system* (LCS).

Residual Normalization (RN): It has been shown that when the burstiness is addressed through a power-law normalization, the retrieval performance can be improved when all descriptors contribute equally to the aggregated residual vector [10]. Therefore it has been suggested [10] that the residuals should be L_2-normalized prior to accumulation. The coding scheme with standard VLAD coding, RootSIFT [11] descriptors – pre-processed by a PCA – and power-law normalized image signatures has been termed VLAD* [10].

Cluster Center Adaption: Arandjelovic et al. [12] have noted that the performance of VLAD depends significantly on the consistency of the coarse visual codebook. In other words, the average of all the local descriptors in the dataset assigned to a certain cluster should be the cluster center of this cluster. In order to keep a consistent codebook, it was proposed to update the centroids when images from a different dataset are processed.

Other Extensions: Further improvements were achieved by extracting multiple VLAD descriptors per image [12], by pooling descriptors by scale and orientation [13] and by using multiple vocabularies [14]. However all of those extensions tend to increase the dimensionality of the (uncompressed) image signature significantly.

4. HIERARCHICAL VLAD

The similarity between two VLAD-encoded images is typically measured through the cosine similarity between their signatures. However, the quality of this similarity measure strongly depends on the position of the codebook vectors within the corresponding Voronoi cell. Arandjelovic et al. [12] have shown that the consistency of the codebook has a great influence on the cosine similarity. Best results are achieved when the codebook vector is indeed the mean vector, such that the sum of all residuals over the whole dataset is zero for every Voronoi cell.

While the mean vector provides a good reference point to judge similarity over the whole dataset, this is not necessarily true for individual images. Descriptors which are located closer to the codebook vector can be discriminated very well. Small changes in the location of the descriptors produce a strong change in the direction of the residual. A descriptor located at the border of a Voronoi cell has to undergo a much larger shift in its position in order to produce the same directional change. Assuming that all points of the descriptor space are equally important for judging similarity, this means that the similarity measure is not equally sensitive to all descriptors. For greater sensitivity, the codebook vectors need to be located close to the descriptors. With a single codebook vector per Voronoi cell this is not always possible. Any vector different from the mean vector will improve the separability of a few data points at the expense of all the others. We therefore search for a way to encode descriptors that allows us to treat the individual descriptors more equally.

If residual normalization is used, another problem arises: In this case only the directional contribution of a feature descriptor is encoded. If the centroid and two descriptors form a collinear set in the descriptor space the descriptors cannot be distinguished, even if a large euclidean distance between them might indicate a strong dissimilarity. In order to alleviate this problem, we introduce multiple reference points for every Voronoi cell. To adapt the position of those reference points according to the density of the distribution of data points this can be done by clustering the data points within each Voronoi cell. The result is a hierarchical codebook similar to [15] with the difference that the vocabulary tree is limited to two levels and the branch factor is not necessarily fixed. We call the Voronoi cells of the original codebook *coarse Voronoi cells* and the Voronoi cells on the second level of the tree *fine Voronoi cells*.

Formally, let $\mathbf{c} : X \to C$ be a quantizer which maps the feature space X to the set of coarse centroids $C = \{\mathbf{c_1}, \mathbf{c_2}, \ldots \mathbf{c_K}\}$. Furthermore let $\mathbf{f} : X \to F$ be a quantizer which maps the feature space X to the set of fine centroids $F = \{\mathbf{f_1}, \mathbf{f_2}, \ldots \mathbf{f_L}\}$. An image – represented by a set of local descriptors $I = \{\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x_m}\}$ – is encoded by HVLAD in the following way:

$$\mathbf{v_i} = \sum_{\mathbf{x} \in I : \mathbf{c}(\mathbf{x}) = \mathbf{c_i}} \frac{\mathbf{x} - \mathbf{f}(\mathbf{x})}{\|\mathbf{x} - \mathbf{f}(\mathbf{x})\|} \tag{3}$$

Note that the dimensionality of the original descriptors is not increased. The coarse centroid determines where the descriptors should be accumulated and the fine centroid acts as a reference point for the encoding. This is illustrated in figure 1.

5. IMPROVING THE LOCAL COORDINATE SYSTEM

Repeating structures in images can yield accumulations of similar features. These feature are called bursty features [8]. In the context of VLAD, this manifests itself in the following way: Since the descriptors of these features are similar to each other they will likely be assigned to the same centroid where they will be accumulated. When lots of similar descriptors are aggregated some descriptor dimensions will dominate the other dimensions, corrupting the similarity metric. This problem is usually addressed by a component-wise power-law normalization which downweights large components.

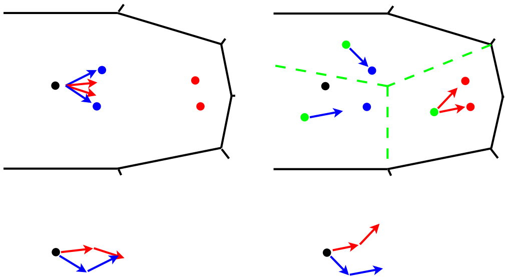

Fig. 1. Illustration of the encoding mechanism of VLAD (left) and HVLAD (right). The blue and red points represent features from two different images (other descriptors used to learn the Voronoi cell are not shown to avoid clutter). At the bottom the resulting aggregated residual vectors are shown. Within every fine Voronoi cell the separability is improved by moving the reference point closer to the datapoints.

Delhumenau et al. [10] introduced another method to improve the effectiveness of the power-law normalization. A PCA without dimensionality reduction is learned for the descriptors. This effectively rotates the features into a coordinate system whose axis are aligned with the directions of the greatest variance – the principal components. In this new coordinate system the power-law normalization is more effective. Compared to a global PCA which rotates the whole feature space, an even greater variety of bursty patterns can be captured when a rotation matrix $\mathbf{X_i}$ is learned separately for every Voronoi cell. This improvement has been called the local coordinate system (LCS) [10]. When using residual normalization the aggregation for VLAD* is done in the following way:

$$\mathbf{v_i} = \sum_{\mathbf{x} \in I : \mathbf{q}(\mathbf{x}) = \mathbf{c_i}} \mathbf{X_i} \frac{\mathbf{x} - \mathbf{c_i}}{\|\mathbf{x} - \mathbf{c_i}\|} \tag{4}$$

In the original design [10] $\mathbf{X_i}$ is learned on the descriptors assigned to this particular centroid $\mathbf{c}_i$. However, the power-law normalization does not operate on the individual descriptors but on the aggregated residuals. Therefore it might be beneficial to use the aggregated residuals to obtain the PCA projection matrix instead of the descriptors as they better represent the data in the input space. Because of the distributivity of the matrix multiplication we can first aggregate the residuals and then rotate the aggregated residual in order to get the same effect as in equation 4. Therefore our learning procedure works as follows: We compute the VLAD*/HVLAD image representation for the training set without applying the power-law normalization. From the image signatures we extract the components belonging to the different (coarse) centroids and use these vectors to learn the rotation matrices. This improvement which we call LCS$^+$ can be used in the context of both VLAD and HVLAD. For the indexing step,

	8	16	32	64	128
VLAD	0.487	0.531	0.544	0.548	0.555
HVLAD	**0.540**	**0.555**	**0.575**	**0.575**	**0.585**
VLAD*	0.558	0.597	0.590	0.614	0.622
HVLAD *	**0.594**	**0.611**	**0.610**	**0.618**	**0.638**
VLAD*+LCS	0.582	0.622	0.614	0.628	0.648
VLAD*+LCS$^+$	0.620	0.642	**0.651**	**0.691**	**0.699**
HVLAD *+LCS$^+$	**0.622**	**0.649**	0.640	0.670	0.691

Table 1. Comparison of the mAP performance of several (uncompressed) VLAD signatures evaluated on the Holidays dataset for different (coarse) vocabulary sizes.

	8	16	32	64	128
VLAD	0.236	0.276	0.306	0.318	0.338
HVLAD	**0.278**	**0.311**	**0.334**	**0.363**	**0.392**
VLAD*	0.271	0.326	0.351	0.384	0.417
HVLAD *	**0.300**	**0.330**	**0.361**	**0.408**	**0.434**
VLAD*+LCS	0.315	0.348	0.380	0.412	0.447
VLAD*+LCS$^+$	**0.350**	**0.378**	**0.426**	**0.447**	**0.475**
HVLAD *+LCS$^+$	0.338	0.375	0.405	0.446	0.472

Table 2. Comparison of the mAP performance of several (uncompressed) VLAD signatures evaluated on the Oxford dataset for different (coarse) vocabulary sizes.

equation 4 can still be used. For HVLAD (see equation 3), the aggregation has to be modified accordingly. When used in conjunction with VLAD we found that $\alpha = 0.2$ still remains a good choice as a normalization parameter. For HVLAD, the normalization parameter needs to be adjusted slightly for best results. We obtained the best results with $\alpha = 0.4$.

6. EVALUATION

We evaluate the different VLAD variants on two well known datasets: The Holidays dataset [16] and the Oxford5k dataset [17]. For all experiments we use the Flickr60k dataset [16] to train all our parameters which includes the vocabulary, PCA, LCS and the codebook for the product quantizer. As features we always use RootSIFT [11] descriptors detected with a DoG [18] interest point detector. Images were scaled down to a maximum width of 1024 pixels prior to feature extraction.

Table 1 shows the mAP performance on the Holidays dataset using the uncompressed VLAD signatures. Residual normalization is used in all experiments and for the HVLAD variants the size of the fine codebook is 64. Unsurprisingly HVLAD generally performs best when the size of the coarse codebook is small. Compared to the standard VLAD variant the relative performance gain ranges between 4.5% and 10.9%, depending on the size of the coarse codebook. For the VLAD* variant the performance gain is noticeably smaller but HVLAD * consistently outperforms VLAD* by an average margin of 2.7%. When a local coordinate system is used, the results are mixed: HVLAD + LCS$^+$ still consistently outperforms the standard LCS by an average margin of 5.7% but it is clear that most of the performance gain is due to the improved LCS. Since the standard LCS cannot be applied to HVLAD, comparable measurements could not be obtained. The evaluation on the Oxford5k dataset (Table 2) shows similar results.

	16 ($D' = 64$)	64 ($D' = 80$)
VLAD	0.501	0.528
HVLAD	**0.526**	**0.553**
VLAD*	0.555	0.598
HVLAD *	**0.572**	**0.603**
VLAD*+LCS	0.584	0.617
VLAD*+LCS$^+$	0.585	0.619
HVLAD *+LCS$^+$	**0.595**	**0.626**

Table 3. Performance on the Holidays dataset with compressed signatures (16 Bytes) for different vocabulary sizes. Results are averaged over 10 runs.

The performance under compression is shown in table 3. All measurements were obtained on the Holidays dataset. Again, for the HVLAD variants the size of the fine vocabulary is 64 and residual normalization is used in all instances. Because of the random initialization of k-means when training the product quantizer, the results shown there are given as the mean over ten runs using different random seeds. Similar to the performance on the uncompressed descriptors, the HVLAD variants outperform the traditional VLAD variants by a margin of up to 1.9%. However, compared to the evaluation of the uncompressed descriptors the performance gain is not as distinguished, especially when LCS is used. This behavior is consistent with the behavior observed by [10] who noted that the compression and quantization steps tends to diminish the performance gain obtained on the uncompressed descriptors. Due to the two-step quantization during the encoding, HVLAD is slightly more computationally demanding than the traditional VLAD variants. However, because of the small vocabularies the added complexity is negligible. The computational complexity on the retrieval side remains unaffected.

7. CONCLUSION

We have presented two improvements to the VLAD image signature and evaluated their impact on two well known datasets. Both measures do not increase the size of the image signature. We have shown that HVLAD works particularly well with small vocabularies and does improve the retrieval results when no local coordinate system is being used. When a local coordinate system is used, our modified training procedure improves its effectiveness and outperforms the state of the art in VLAD-like signatures.

8. REFERENCES

[1] Herve Jegou, Matthijs Douze, Cordelia Schmid, and Patrick Perez, "Aggregating local descriptors into a compact image representation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3304–3311, 2010.

[2] Josef Sivic and Andrew Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *IEEE International Conference on Computer Vision*, vol. 2, pp. 1470–1477 vol.2, 2003.

[3] Yair Weiss, Antonio Torralba, and Rob Fergus, "Spectral Hashing," *Advances in Neural Information Processing Systems*, , no. 1, pp. 1753–1760, 2008.

[4] Hervé Jégou, "Packing bag-of-features," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2357–2364, 2009.

[5] Ondrej Chum, Michal Perdoch, and Jiri Matas, "Geometric min-Hashing: Finding a (thick) needle in a haystack," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 17–24.

[6] Rainer Lienhart and Malcolm Slaney, "pLSA on Large Scale Image Databases," *IEEE Conference on Acoustics, Speech and Signal Processing*, pp. 1217–1220, 2007.

[7] Gabriela Csurka and Florent Perronnin, "Fisher Vectors : Beyond Bag-of-Visual-Words Image Representations," *Computer Vision, Imaging and Computer Graphics*, pp. 28–42, 2011.

[8] Hervé Jégou, Matthijs Douze, and Cordelia Schmid, "On the burstiness of visual elements," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1169–1176, 2009.

[9] Florent Perronnin, J Sánchez, and Thomas Mensink, "Improving the fisher kernel for large-scale image classification," in *European conference on Computer Vision*. 2010, pp. 143–156, Springer.

[10] Jonathan Delhumeau, Philippe-Henri Gosselin, Hervé Jégou, and Patrick Pérez, "Revisiting the VLAD image representation," *ACM Multimedia*, 2013.

[11] Relja Arandjelovic and Andrew Zisserman, "Three things everyone should know to improve object retrieval," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 32, pp. 2911–2918, 2012.

[12] Relja Arandjelović and Andrew Zisserman, "All about VLAD," *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[13] Wan-Lei Zhao, Hervé Jégou, and Guillaume Gravier, "Oriented pooling for dense and non-dense rotation-invariant features," *British Machine Vision Conference*, pp. 99.1–99.11, 2013.

[14] Hervé Jégou and Ondej Chum, "Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening," *European Conference on Computer Vision*, pp. 774–787, 2012.

[15] David Nister and Henrik Stewenius, "Scalable Recognition with a Vocabulary Tree," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2161–2168, 2006.

[16] Herve Jegou, Matthijs Douze, and Cordelia Schmid, "Hamming embedding and weak geometric consistency for large scale image search," *European Conference on Computer Vision*, pp. 304–317, 2008.

[17] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman, "Object retrieval with large vocabularies and fast spatial matching," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 3613, pp. 1–8, 2007.

[18] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.