Residual Enhanced Visual Vectors for On-Device Image Matching

David Chen[1], Sam Tsai[1], Vijay Chandrasekhar[1], Gabriel Takacs[1], Huizhong Chen[1],
Ramakrishna Vedantham[2], Radek Grzeszczuk[2], Bernd Girod[1]
[1]Department of Electrical Engineering, Stanford University [2]Nokia Research Center, Palo Alto

***Abstract*—Most mobile visual search (MVS) systems query a large database stored on a server. This paper presents a new architecture for searching a large database directly on a mobile device, which has numerous benefits for network-independent, low-latency, and privacy-protected image retrieval. A key challenge for on-device MVS is storing a memory-intensive database in the limited RAM of the mobile device. We design and implement a new compact global image signature called the Residual Enhanced Visual Vector (REVV) that is optimized for the local features typically used in MVS. REVV outperforms existing compact database representations in the MVS setting and attains similar retrieval accuracy in large-scale retrieval tests as a Vocabulary Tree that uses $26\times$ more memory. The compactness of REVV consequently enables many database images to be queried on a mobile device.**

I. Introduction

Many mobile visual search (MVS) applications have been successfully developed for recognition of outdoor landmarks [1], product covers [2], and printed documents [3], [4], amongst other categories. In each case, the user snaps a photo with a mobile device to retrieve information about an object of interest. Robust image-based recognition is achieved using local scale-and-rotation-invariant features like SIFT [5], SURF [6], CHoG [7], and RIFF [8].

Equally important for large-scale visual search is the method used to index the billions of local features extracted for a database containing millions of images. Sivic and Zisserman developed the popular Bag-of-Features (BoF) framework [9]. Nístér and Stewenius subsequently extended the BoF framework to use a large codebook of up to 1 million visual words [10]. Their Vocabulary Tree (VT) and subsequent variants [11], [12], [13] are widely used today.

Fig. 1(a) shows a possible architecture for MVS that relies on a database stored in the cloud. On the mobile device, features are extracted and encoded, and the features are transmitted to a remote server. Then, on the server, the database images are quickly scored using a data structure like the VT to generate a ranked list of candidates, and geometric verification is performed on the shortlist of the top-ranked candidates. As mobile computing power and hardware resources improve, operations typically performed on a server, like the database search, can be performed directly on the mobile device, giving rise to the new architecture shown in Fig. 1(b). Searching a database directly on the mobile device has several advantages: (1) In regions with unreliable or no cell phone service, a visual query can still be performed with the mobile device. (2) We can reduce traffic on a remote server that is already handling many incoming queries. (3) Since no data are sent to a remote server, the privacy of photos taken with the mobile device is protected. (4) Querying the locally stored database can be faster than querying a database on a remote server, because data transmission delays are avoided. New mobile applications can reap these benefits, provided there is a way to efficiently store and search a large visual database on the mobile device.

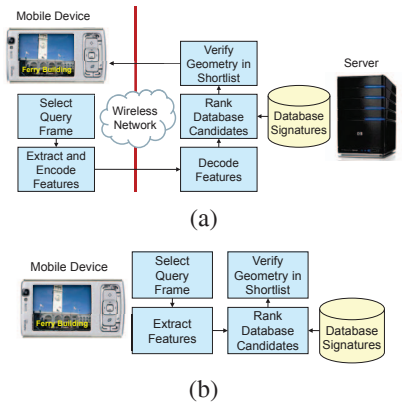


Fig. 1. Two different system architectures for mobile visual search. (a) Feature extraction occurs on the mobile device, while database search occurs on a remote server. (b) All operations occur on the mobile device.

A key challenge to performing on-device MVS is fitting the entire database in the mobile device's limited random access memory (RAM). Mobile devices typically have two orders of magnitude less RAM than a standard server. BoF-based schemes can use up to 9 KB per image [14]. Additionally, a VT with 1 million leaf nodes requires around 70 MB [15]. Thus, if we employ the VT for on-device MVS, the number of database images that we can query is severely constrained by the limited RAM.

Prior works have also recognized the need for a memory-efficient database, although not for the on-device MVS scenario. An image decomposition model for BoF-based retrieval is presented in [16]. A compressed inverted index was developed for the VT in [14]. In an interesting new direction, the Vector of Locally Aggregated Descriptors (VLAD) in [17] and the Compressed Fisher Vector (CFV) in [18] both create a compact global image signature by aggregating vector residuals of descriptors quantized to a small set of visual words.

In [17], [18], the retrieval systems that use VLAD and CFV extract on average 3,000 Hessian-Affine SIFT features per image. For MVS applications, however, low-latency retrieval is very important, and extracting 3,000 Hessian-Affine SIFT features per query image on a mobile device would be unacceptably slow. In our past work, we have achieved feature extraction latencies near 1 second per query on a mobile device, using fast interest point and descriptor computations and targeting around 500 features per image [2].

In this paper, we design a new compact global image signature called the Residual Enhanced Visual Vector (REVV). Like VLAD and CFV, REVV starts by generating visual word residuals. However, REVV is optimized for fast on-device MVS and has several new enhancements: (1) Improved residual aggregation, using mean aggregation instead of sum aggregation. (2) A new outlier rejection mechanism for discarding unstable features during vector quantization. (3) Classification-aware dimensionality reduction, using linear discriminant analysis in place of principal component analysis. (4) Discriminative weighting based on correlation between image signatures in the compressed domain. With these enhancements, REVV attains similar retrieval performance as a VT, while using $26\times$ and $6\times$ less memory than a VT with uncompressed and compressed inverted indices, respectively.

The rest of the paper is organized as follows. First, Sec. II reviews existing methods for large-scale image retrieval. Then, Sec. III presents the design of our new compact signature for on-device MVS. Experimental results in Sec. IV demonstrate that REVV attains the same retrieval accuracy as a VT on two large-scale data sets, while using substantially less memory. The large memory savings directly translate into the ability to search the signatures of many database images in the RAM of a mobile device.

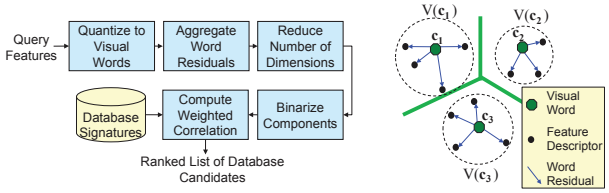
II. Large-Scale Retrieval Techniques

A VT is trained by hierarchical k-means clustering of many database feature descriptors [10]. To have a discriminative vocabulary, a large number of visual words, e.g., 1 million, must exist at the leaf level. Misclassification through the VT can be alleviated using greedy-N best paths [11], where the N most promising paths are explored at each level, and soft binning [12], where each descriptor is assigned with fractional counts to the M nearest visual words. Each image is represented as a histogram of visit counts over the visual words. Throughout the rest of the paper, we use a VT with $k = 1$ million visual words, $N = 10$ for greedy-N best paths, and $M = 3$ for soft binning, as we find these parameters yield very good retrieval performance.

VLAD [17] differs from the VT in that it can attain good retrieval performance using a much smaller set of visual words, typically just $k = 64$ to $k = 256$ words trained by flat k-means clustering. VLAD computes (1) the vector difference between each feature descriptor and the nearest visual word, which is called a word residual (WR) and (2) the sum of WRs surrounding each visual word. The aggregated WRs for all k visual words are concatenated together to form an image signature. For a memory-efficient representation, principal component analysis (PCA) and product quantization (PQ) are subsequently applied to the WR vector. In the next section, we will show how to enhance the discriminative capability of WR-based schemes to build a compact database representation for large-scale MVS.

III. Design of Residual Enhanced Visual Vector

Fig. 2(a) shows an overview for REVV, depicting how a compact image signature is formed for a query image and then compared against database REVV signatures to produce a ranked list of database candidates. We will explain each block in detail in the following sections.



(a) (b)

Fig. 2. (a) Overview of how feature descriptors for a query image are converted into a REVV signature and compared against database REVV signatures. (b) Visual words, feature descriptors, and sets of word residuals.

A. Image-Level Receiver Operating Characteristic

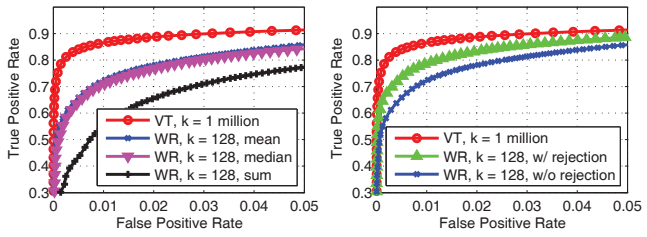
To systematically optimize the performance of REVV, we first employ an image-level receiver operating characteristic (ROC). Later in Sec. IV, we will validate our signature's performance on large-scale retrieval tests. For training, 16,000 matching and 16,000 non-matching image pairs are collected from the Oxford Buildings Data Set [19] and the University of Kentucky Benchmark Data Set [20]. For testing, 8,000 matching and 8,000 non-matching image pairs are collected from the Stanford Media Cover Data Set [21] and the Zurich Buildings Data Set [22]. Since we target low-latency MVS, we extract around 500 SURF features per image, which takes about 1 second on a mobile device [2].

B. Aggregation Type

Let $\mathbf{c}_1, \cdots, \mathbf{c}_k$ be the set of d-dimensional visual words. As illustrated in the toy example of Fig. 2(b) with $k = 3$, after each descriptor in an image is quantized to the nearest visual word, a set of vector WRs will surround each visual word. Let $\mathrm{V}(\mathbf{c}_i) = \{\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \cdots, \mathbf{v}_{i,N_i}\}$ represent the set of N_i separate WRs around the i^{th} visual word. To aggregate the WRs, several different approaches are possible:

- Sum aggregation: This is the approach used by VLAD [17]. Here, the aggregated WR for the i^{th} visual word is $\mathbf{S}_i = \sum_{j=1}^{N_i} \mathbf{v}_{i,j}$. Note that $\mathbf{S}_i \in \Re^d$.
- Mean aggregation: We normalize the sum of WRs by the cardinality of $\mathrm{V}(\mathbf{c}_i)$, so the aggregated WR becomes $\mathbf{S}_i = 1/N_i \cdot \sum_{j=1}^{N_i} \mathbf{v}_{i,j}$.

- Median aggregation: This is similar to mean aggregation, except we find the median along each dimension, i.e., $\mathbf{S}_i(n) = \text{median}\left\{\mathbf{v}_{i,j}(n) : j = 1, \cdots, N_i\right\}\ n = 1, \cdots, d.$



(a) (b)

Fig. 3. (a) ROCs for Vocabulary Tree (VT) and Word Residuals (WRs) with three aggregation methods. (b) ROCs for VT and WRs with and without outlier rejection.

Next, let $\mathbf{S}$ be the concatenation of aggregated WRs: $\mathbf{S} = [\mathbf{S}_1\ \mathbf{S}_2\ \cdots\ \mathbf{S}_k] \in \Re^{kd}$. A normalized image signature $\bar{\mathbf{S}}$ is formed as $\bar{\mathbf{S}} = \mathbf{S}/\left|\left|\mathbf{S}\right|\right|_2$. To compare two normalized image signatures $\bar{\mathbf{S}}_q$ and $\bar{\mathbf{S}}_d$, we compute their Euclidean distance $\left|\left|\bar{\mathbf{S}}_q - \bar{\mathbf{S}}_d\right|\right|_2$, or equivalently the inner product $\left\langle \bar{\mathbf{S}}_q, \bar{\mathbf{S}}_d \right\rangle$.

The ROCs of the three aggregation methods are shown in Fig. 3(a) for $k = 128$ visual words, along with the ROC for a VT with $k = 1$ million words. The same 64-dimensional SURF features are used for each method. The sum-aggregated WR, which is a version of VLAD without PCA or PQ, has a performance gap compared to the VT in this MVS setting. The mean-aggregated WR, which requires just one additional division per visual word compared to the sum-aggregated WR, performs substantially better. Furthermore, the mean-aggregated WR performs slightly better than the median-aggregated WR, which is more expensive to compute.

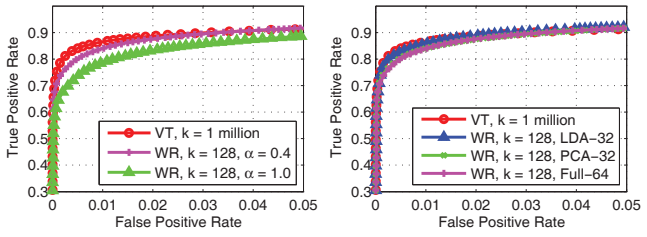
C. Outlier Feature Rejection

Some features that lie close to the boundary between two Voronoi cells reduce the repeatability of the aggregated residuals. Consider the feature that lies very near the boundary between the Voronoi cells of $\mathbf{c}_1$ and $\mathbf{c}_3$ in Fig. 2(b). Even a small amount of noise can cause this feature to be quantized to $\mathbf{c}_3$ instead of $\mathbf{c}_1$, which would significantly change the composition of $\text{V}(\mathbf{c}_1)$ and $\text{V}(\mathbf{c}_3)$ and consequently the aggregated residuals $\mathbf{S}_1$ and $\mathbf{S}_3$.

We can remove this type of "outlier feature" by exploiting the fact that for a given visual word, its outlier features are those farthest away from the visual word. By discarding every feature whose distance is above the C^{th} percentile on a distribution of distances, we can effectively remove most of the outlier features. Note that the C^{th} percentile level is different for the various visual words, because the distance distributions are different. Experimentally, we found that $C = 90$ is the best value. Using outlier rejection, a significant improvement in ROC can be observed in Fig. 3(b).

D. Power Law

Applying a power law to the WRs has been found helpful for reducing peaky components which are difficult to match



(a) (b)

Fig. 4. (a) ROCs for VT and WRs with and without power law. (b) ROCs for VT and WRs with no transform (full 64 dimensions per word), with PCA (32 dimensions per word), and with LDA (32 dimensions per word).

[18]: $\mathbf{S}_{\text{PL}} = \left[\mathbf{S}(1)^\alpha \cdots \mathbf{S}(kd)^\alpha\right], \alpha \in [0, 1]$. An L_2 normalization follows the power law to generate a normalized image signature. Experimentally, we found the optimal value for the exponent is $\alpha = 0.4$ for SURF features. Fig. 4(a) shows the positive improvement in the ROC when we apply a power law.

E. Classification-Aware Dimensionality Reduction

Since the WR dimensionality is proportional to the size of the database, we want to reduce the dimensionality as much as possible, without adversely impacting retrieval performance. VLAD [17] and CFV [18] both use principal component analysis (PCA) for dimensionality reduction. In contrast, we develop a classification-aware method for reducing the dimensionality using linear discriminant analysis (LDA). We define the problem as follows for each visual word separately:

$$\begin{aligned} \mathbf{S}_j &= \text{aggregated WR from image } j \\ J_{\text{M}} &= \{(j_1, j_2) : \text{images } j_1 \text{ and } j_2 \text{ are matching}\} \\ J_{\text{NM}} &= \{(j_1, j_2) : \text{images } j_1 \text{ and } j_2 \text{ are non-matching}\} \end{aligned}$$

$$\underset{\mathbf{w}}{\text{maximize}} \frac{\sum_{(j_1,j_2)\in J_{\text{NM}}} \left\langle \mathbf{w}, \mathbf{S}_{j_1} - \mathbf{S}_{j_2} \right\rangle^2}{\sum_{(j_1,j_2)\in J_{\text{M}}} \left\langle \mathbf{w}, \mathbf{S}_{j_1} - \mathbf{S}_{j_2} \right\rangle^2} \tag{1}$$

The objective in Eq. (1) is to maximize the ratio of inter-class variance to intra-class variance by varying the projection direction $\mathbf{w}$. This problem is similar to that defined in [23] for reducing the dimensionality of feature descriptors, except here we are concerned with reducing the dimensionality of WRs. Eq. (1) can be solved as a generalized eigenvector problem, with the following solution:

$$\begin{aligned} &\mathbf{R}_{\text{NM}}\mathbf{w}_i = \lambda_i \mathbf{R}_{\text{M}}\mathbf{w}_i \qquad i = 1, 2, \cdots, d_{\text{LDA}} \\ &\mathbf{R}_\theta = \sum_{(j_1,j_2)\in J_\theta} \left(\mathbf{S}_{j_1} - \mathbf{S}_{j_2}\right)\left(\mathbf{S}_{j_1} - \mathbf{S}_{j_2}\right)^T \quad \theta \in \{\text{M}, \text{NM}\} \end{aligned} \tag{2}$$

We retain the d_{LDA} most energetic components after projection. In Fig. 4(b), we plot the ROC for (1) WR without any transform, with 64 dimensions/word, (2) WR with PCA, with 32 dimensions/word, and (3) WR with LDA, also with 32 dimensions/word. PCA performs similarly as the case with no transform, while LDA outperforms the two other WR schemes. With LDA, we can reduce the image signature's dimensionality in half, while actually boosting the retrieval performance.

F. Fast Score Computation and Discriminative Weighting

Following LDA, each component of the projected WR is binarized to +1 or -1 depending on the sign. As in [18], this binarization creates a compact image signature that just requires at most $k \cdot d_{\text{LDA}}$ bits. Another benefit of the binarization is fast score computation. The inner product $\left\langle \bar{\mathbf{S}}_q, \bar{\mathbf{S}}_d \right\rangle$ can be closely approximated by the following expression:

$$\frac{1}{\left\|\mathbf{S}_q^{\text{bin}}\right\|_2 \left\|\mathbf{S}_d^{\text{bin}}\right\|_2} \sum_{\text{i visited in common}} \underbrace{C\left(\mathbf{S}_{q,i}^{\text{bin}}, \mathbf{S}_{d,i}^{\text{bin}}\right)}_{C_i} \tag{3}$$

where $C\left(\mathbf{S}_{q,i}^{\text{bin}}, \mathbf{S}_{d,i}^{\text{bin}}\right) = d_{\text{LDA}} - 2H\left(\mathbf{S}_{q,i}^{\text{bin}}, \mathbf{S}_{d,i}^{\text{bin}}\right)$ is the binary correlation, $H(\mathbf{A}, \mathbf{B})$ is Hamming distance between binary vectors $\mathbf{A}$ and $\mathbf{B}$, and $\mathbf{S}_q^{\text{bin}}$ and $\mathbf{S}_d^{\text{bin}}$ are the binarized WRs for the query and database images, respectively. Since Hamming distance can be computed very quickly using bitwise XOR and POPCOUNT, the score in Eq. 3 can be efficiently calculated.

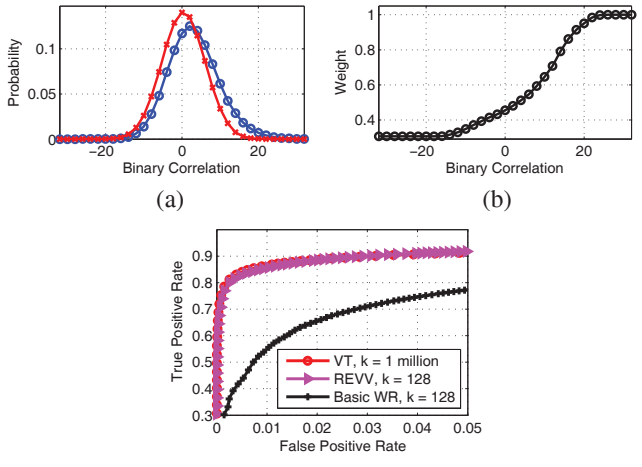


Fig. 5. (a) Distributions of binary correlation per visual word, for matching (blue o) and non-matching (red x) image pairs. (b) Weights for different binary correlation values. (c) ROCs for VT, REVV, and basic WR scheme.

Finally, we apply a discriminative weighting based on correlations computed between binarized signatures. Fig. 5(a) plots two distributions: (1) the distribution $p_{\text{M}}(C)$ of binary correlation per visual word for matching images pairs, and (2) the analogous distribution $p_{\text{NM}}(C)$ for non-matching image pairs. It can be observed that on average matching image pairs exhibit higher binary correlation values, and we design a weighting function $w(C)$ as follows that exploits this property:

$$w(C) = \frac{p_{\text{M}}(C)}{p_{\text{M}}(C) + p_{\text{NM}}(C)} \tag{4}$$

Assuming $\Pr\{\text{match}\} = \Pr\{\text{non-match}\}$, then $w(C)$ is exactly equal to $\Pr\{\text{match}|C\}$. Using this weighting function, which is plotted in Fig. 5(b) the score changes from Eq. (3) to:

$$\frac{1}{\left\|\mathbf{S}_q^{\text{bin}}\right\|_2 \left\|\mathbf{S}_d^{\text{bin}}\right\|_2} \sum_{\text{i visited in common}} w(C_i) \cdot C_i \tag{5}$$

The weighting function effectively rewards observations with higher binary correlation values.

After applying the weighting, we obtain the ROC for REVV plotted in Fig. 5(c), where it can be seen that REVV with $k = 128$ visual words performs very close to the VT with $k = 1$ million words. For comparison, the basic WR scheme (black curve from Fig. 3(a)) is also included in the plot. With our new enhancements, REVV significantly outperforms the basic WR scheme in the MVS setting. In the next section, we will see that REVV has similar retrieval performance to the VT in large-scale tests, while requiring significantly less memory to store the database.

IV. Experimental Results

A. Large-Scale Retrieval Performance

We test the retrieval performance of REVV versus the VT on two data sets: (1) the Stanford YouTube Data Set [24], where the database consists of 1 million keyframes taken from over 2,000 YouTube video clips, and the query set contains 1,224 viewfinder frames captured by camera phones, and (2) the Stanford MVS Data Set [25], where the database consists of 1,200 labeled "clean" images of various objects and 1 million distractor images, and the query set contains 3,300 images of the same objects taken with different camera phones.

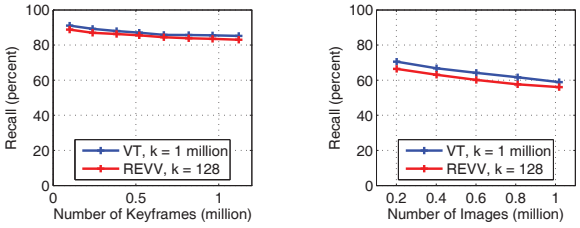


Fig. 6. Recall for (a) Stanford YouTube and (b) Stanford MVS Data Sets.

Fig. 6 plots the recall versus database size for both data sets. For the YouTube Data Set, REVV with $k = 128$ words achieves recall within 2 percent relative to that of the VT with $k = 1$ million words. Similarly, for the MVS Data Set, REVV performs comparably as the VT, achieving recall within 3 percent relative to that of the VT. As database size increases, the recall rates of the two schemes closely track one another.

B. Memory Usage

Fig. 7 plots the memory usage per database image for three schemes: (1) VT with an uncompressed index, (2) VT with a compressed index [14], and (3) REVV. Memory usage is generally lower for the YouTube Data Set versus the MVS Data Set because of temporal correlations between keyframes and fewer features per image. Index compression for VT yields $5-6\times$ memory savings relative to the VT with uncompressed index. In contrast, REVV provides far greater savings: memory usage is shrunk $24-26\times$ from that of the VT with uncompressed index.

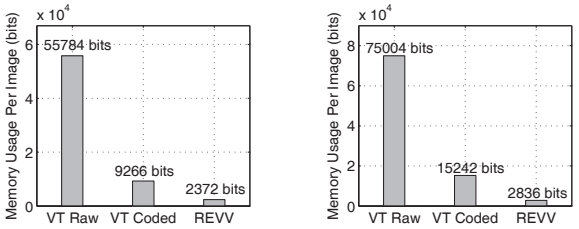


Fig. 7. Memory usage per database image for (a) Stanford YouTube and (b) Stanford MVS Data Sets.

Suppose a smartphone has 256 MB of RAM, 64 MB are used by the operating system, and 128 MB are consumed by other applications. An MVS application would then have 64 MB available. If we use 64-dimensional SURF descriptors, a VT with $k = 1$ million leaf nodes requires 70 MB [15] and thus would not fit in our 64 MB budget. In contrast, if we employ REVV with the same SURF features, we would require just 264 KB to store $k = 128$ centroids and $d_{\text{LDA}} = 32$ eigenvectors per centroid, provided we again use 8 bits per dimension. On top of that, each database image's REVV signature consumes just 0.35 KB.

We have recently developed a landmark recognition application on a Nokia N900 smartphone, which has a 600 MHz ARM processor and 256 MB of total RAM. We store REVV signatures for $10,000$ images of San Francisco landmarks in the phone's RAM. Using the architecture of Fig. 1(b), our application achieves a mean latency of 1.6 seconds per query, which is remarkably fast considering that the entire recognition process occurs on the phone. Thus, we can provide fast responses for real-time mobile augmented reality, without any assistance from an external server.

V. Conclusions

We have developed a new discriminative, compact global image signature called REVV that is optimized for on-device MVS applications. By incorporating improved mean aggregation, outlier feature rejection, dimensionality reduction with LDA, and discriminative correlation weighting, REVV noticeably outperforms existing word residual schemes in the MVS setting. In large-scale tests, REVV attains similar retrieval accuracy as a VT that uses significantly more memory. The compactness of REVV enables the signatures for many database images to be stored in the limited RAM of a mobile device, which greatly improves the quality of visual search results in many applications including large-scale landmark recognition and product recognition.

References

[1] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *ACM Multimedia Information Retrieval*, October 2008, pp. 427–434.

[2] S. S. Tsai, D. Chen, V. Chandrasekhar, G. Takacs, N.-M. Cheung, R. Vedantham, R. Grzeszczuk, and B. Girod, "Mobile product recognition," in *ACM International Conference on Multimedia*, October 2010, pp. 1587–1590.

[3] S. S. Tsai, H. Chen, D. Chen, G. Schroth, R. Grzeszczuk, and B. Girod, "Mobile visual search on printed documents using text and low bit-rate features," in *IEEE International Conference on Image Processing*, September 2011.

[4] H. Chen, S. S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *IEEE International Conference on Image Processing*, September 2011.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.

[7] V. Chandrasekhar, Y. R. G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "Quantization schemes for low bitrate compressed histogram of gradients descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 1–8.

[8] G. Takacs, V. R. Chandrasekhar, S. S. Tsai, D. M. Chen, R. Grzeszczuk, and B. Girod, "Unified real-time tracking and recognition with rotation-invariant fast features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1–8.

[9] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, vol. 2, October 2003, pp. 1470–1477.

[10] D. Níster and H. Stewenius, "Scalable recognition with a vocabulary tree," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2006, pp. II: 2161–2168.

[11] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–7.

[12] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[13] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 316–336, February 2010.

[14] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod, "Inverted index compression for scalable image matching," in *IEEE Data Compression Conference*, March 2010, p. 525.

[15] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, J. P. Singh, and B. Girod, "Tree histogram coding for mobile image matching," in *IEEE Data Compression Conference*, March 2009, pp. 143–152.

[16] X. Zhang, Z. Li, L. Zhang, W.-Y. Ma, and H.-Y. Shum, "Efficient indexing for large scale visual search," in *IEEE International Conference on Computer Vision*, October 2009, pp. 1103 –1110.

[17] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3304–3311.

[18] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3384–3391.

[19] J. Philbin and A. Zisserman, *Oxford Building Data Set*, June 2007. [Online]. Available: http://tinyurl.com/2afuglg

[20] H. Stewenius and D. Nister, *University of Kentucky Benchmark Data Set*, September 2006. [Online]. Available: http://tinyurl.com/ddoht2

[21] D. Chen, S. Tsai, and B. Girod, *Stanford Media Cover Data Set*, September 2009. [Online]. Available: http://tinyurl.com/5rbsf7d

[22] H. Shao, T. Svoboda, and L. V. Gool, *ZuBuD - Zurich Buildings Data Set*, April 2003. [Online]. Available: http://tinyurl.com/6ctwwl6

[23] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, January 2011.

[24] D. Chen, N.-M. Cheung, S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod, "Dynamic selection of a feature-rich query frame for mobile video retrieval," in *IEEE International Conference on Image Processing*, September 2010, pp. 1017–1020.

[25] V. Chandrasekhar, D. Chen, S. Tsai, N.-M. Cheung, H. Chen, G. Takacs, Y. Reznik, R. Vedantham, R. Grzeszczuk, J. Bach, and B. Girod, "The Stanford mobile visual search data set," in *ACM Conference on Multimedia Systems*, February 2011, pp. 117–122.