Selective Aggregated Descriptors for Robust Mobile Image Retrieval

Jie Lin *, Zhe Wang †, Yitong Wang †, Vijay Chandrasekhar *, Liyuan Li *
* Institute for Infocomm Research, A*STAR, Singapore
E-mail: {lin-j,lyli,vijay}@i2r.a-star.edu.sg
† Institute of Digital Media, the School of EE & CS, Peking University, Beijing, China
E-mail: {zhew,wangyitong}@pku.edu.cn

***Abstract*—Towards low latency query transmission via wireless link, methods have been proposed to extract compact visual descriptors on mobile device and then send these descriptors to the server at low bit rates in recent mobile image retrieval systems. The drawback is that such on-device feature extraction demands heavy computational cost and large memory space. An alternate approach is to directly transmit low quality JPEG compressed query images to the server, but the lossy compression results in compression artifacts, which subsequently degrade feature discriminability and deteriorate the retrieval performance. In this paper, we present selective aggregated descriptors to address this problem of mobile image retrieval on low quality query images. The proposed mechanism of selective aggregation largely reduces the negative impact of noisy features caused by compression artifacts, enabling both low latency query transmission from mobile device and effective image retrieval on the server end. In addition, the proposed method allows fast descriptor matching and less storage of visual descriptors for large database. Extensive experiments on benchmark datasets have shown the consistent superior performances of the proposed approach over the state-of-the-art.**

I. Introduction

Smart phones and Tablet PCs have shown great potentials in mobile image retrieval [3][20], thanks to the integrated functionality of high resolution color camera and broadband wireless network connection. Many mobile image retrieval applications (such as Google Goggles [1] and Amazon Flow [2]) have been developed for retrieving similar images containing a rigid object in a large set of database images, given a query image of that object (such as CD/book cover, poster, logo, landmark, etc). In general, most mobile image retrieval systems follow the client-server architecture. A captured query is sent through the wireless network to the server, where image retrieval is conducted to identify the relevant images from an image database stored on the server.

To reduce delivery latency for better user experience, the upstream query data is expected to be as small as possible, especially for unstable or limited bandwidth wireless connection. Recent works have proposed to extract compact visual descriptors of query images on the mobile device, and then send such descriptors over a wireless link at low bit rates (see Fig. 1(a)). The compact descriptors extraction [8][11][10][19] follows a typical pipeline: statistics of local invariant features (such as SIFT [12] and SURF [13]) are aggregated to form a fixed-length vector representation, which is subsequently compressed into compact descriptors.

Aside from low latency query delivery, such on-device descriptors extraction may demand heavy computational cost as well as memory footprint, making it impractical to work with mobile devices that have limited processor power and RAM. One example is local feature detection and description (such as SIFT [12]) on mobile device typically takes a few seconds. Another example is locally aggregated descriptors (such as Bag-of-Words (BoW) [15][22]) often involves a large visual vocabulary containing 0.1-1 million visual words. The vocabulary would cost hundreds of megabytes to be loaded in the limited RAM of mobile device.

As smart phones are hardware friendly to support fast JPEG image compression at extremely low cost, an alternate approach is to directly transmit JPEG compressed query images over a wireless link, the subsequent descriptor extraction and matching are performed on the server side (see Fig. 1(b)). One can reduce the compressed image size by decreasing the JPEG quality factor, however, the lossy compression introduces compression artifacts appeared in low quality images. It is noticed that compression artifacts would degrade the discriminability of detected local features, and deteriorate the retrieval and matching performance (see Section IV). As shown in Fig. 2, the number of inlier matches (i.e., true positive matches) is largely reduced with the quality of query image (from 100 to 5). Thus, it is crucial to consider a mechanism that incorporates the selection of informative local features into descriptor extraction.

In this paper, we present selective aggregated compact descriptors to address the problem of low quality image retrieval. Our contributions are three fold. First, state-of-the-art locally aggregated descriptors [9][8] unfairly assume that all local features extracted from an image contribute equally to the subsequent aggregation stage, resulting in suboptimal retrieval accuracy. We propose a selective aggregation to reduce the negative impact of noisy local features on the aggregated descriptors. Second, we propose to model the characteristics of match/non-match keypoint pairs for selecting informative local features, with the observation that true positive match keypoints are statistically associated with informative local features (see Fig. 2). Third, the proposed approach enables both low bit rate query transmission (e.g., 6 KB per image) and effective image retrieval. In addition, the proposed method

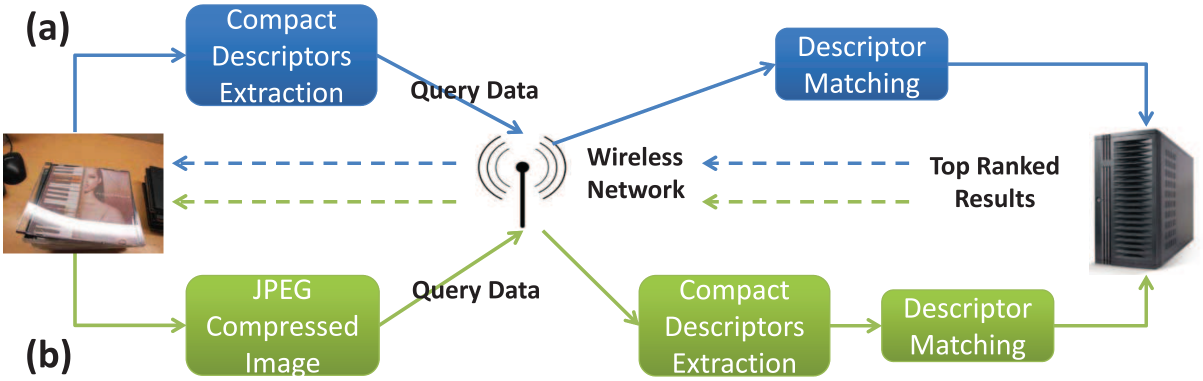

Fig. 1. Low bit rate mobile image retrieval frameworks: (a)Extracting and compressing visual descriptors on the mobile device, and sending the compact codes over a wireless link (Top), and (b)Transmitting highly compressed JPEG query images, and subsequent descriptors extraction and matching are performed on the server (Bottom).

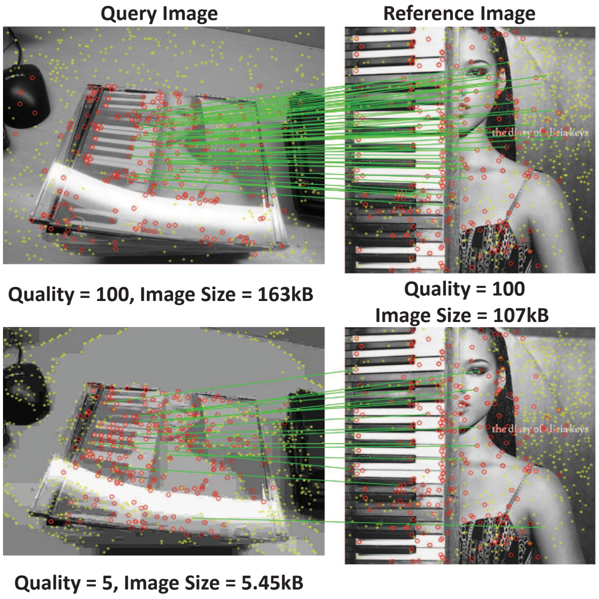
Query Image
Reference Image
Quality = 100, Image Size = 163kB
Quality = 100
Image Size = 107kB
Quality = 5, Image Size = 5.45kB

Fig. 2. Negative effect of lossy JPEG compression on local invariant feature matching. A highly compressed query image (Quality = 5) degrades the discriminability of detected features, which subsequently reduces the number of inlier matches (highlighted in green). The informative local features detected by the proposed method are denoted in red, while the noisy ones are in yellow.

allows fast descriptor matching as well as light storage of compact descriptors extracted from large scale database images.

Our extensive experiments on benchmark datasets (such as UKbench) combined with 1 million distractor images have shown the consistent superior performances of the proposed approach over the state-of-the-art [9][8][10]. For example, mean average precision (mAP) is improved from 33.9% to 69.4% on Graphics dataset (JPEG quality $Q = 30$ for query images), compared to the compressed Fisher vector [8].

II. Related Work

Local invariant features like SIFT [12] and SURF [13] cannot meet the requirement of compactness, as the size of local descriptors usually exceeds the size of raw image itself. There are a lot works on compressing local descriptors [7][6]. For instance, Chandrasekhar et al. [6] proposed a Compressed Histogram of Gradient (CHoG), which adopts Gagie tree coding to compress each local feature into approximate 60 bits. Suppose that 1,000 keypoints are detected per image, the overall feature size is approximately 8KB.

Recent work stepped forward to compute statistics of local features and then aggregate them to form a fixed-length vector representation, which is subsequently compressed for efficient storage and transmission. The Bag-of-Words (BoW) representation [14][15][25] is the most widely adopted method for this purpose. Each local feature from an image is quantized to its closest visual word in a visual vocabulary. BoW accumulates the number (0-order statistics) of local features assigned to each visual word. Chen et al. [21] and Ji. et al. [5][4] proposed to further compress the quantized BoW histogram.

Recently, the Fisher Vector (FV) [8] extends the BoW by computing higher-order statistics of the distribution of local features, e.g., Gaussian Mixture Model (GMM). Specifically, FV aggregates the gradient vector of each local feature's likelihood w.r.t. the GMM parameters (mean or variance) for each Gaussian. Jegou et al. [9] proposed a non-probabilistic FV, namely, the Vector of Locally Aggregated Descriptors (VLAD), to aggregate residual vectors (difference between local feature and its nearest visual word). Both FV and VALD can be compressed into compact binary codes [8][9] for fast Hamming distance computation. Chen et al. [10] introduced the Residual Enhanced Visual Vector (REVV), where linear discriminant analysis (LDA) is employed to reduce the dimensionality of VLAD, followed by sign binarization to generate compact codes. Lin et al. [19] further improved the compactness of these binary aggregated descriptors by progressively coding informative sub-vectors, which achieves comparable retrieval accuracy to the raw FV or VLAD.

III. Aggregating Informative Local Features

In this section, we introduce the selective aggregated compact descriptors. We first give an overview of the proposed method in III-A. In III-B, we formulate the selective aggregation as a keypoint ranking problem, and implement the keypoint ranking by employing likelihood ratio test with Gaussian Mixture Models (GMM) that fit the empirical distribution of match/non-match keypoint pairs. Finally, we discuss how to estimate the parameters of the GMM distribution in III-C.

A. Overview

We illustrate the pipeline of locally aggregated compact descriptors in Fig. 3. Three stages including feature coding, aggregation and compression, are usually adopted to generate a compact descriptor.

Feature coding. Let $\mathbf{I} = \{(\mathbf{z}_t, \mathbf{x}_t)\}_{t=1}^{T}$ denote a collection of d-dimensional local features $\mathbf{x}$ and their detected keypoints $\mathbf{z}$ in image $\mathbf{I}$. In this work, we focus on the SIFT feature ($d = 128$), and the keypoint $\mathbf{z} = (\eta, \theta, \upsilon, \xi)$ is of four dimensions, where η, θ, υ and ξ denote scale, orientation, peak value in scale space and the distance from a keypoint to the image center, respectively. The goal of feature coding is to embed local features $\mathbf{x}$ in a visual vocabulary space based on a encoder r:

$$r : \mathbf{x} \in \mathbb{R}^d \rightarrow r(\mathbf{x}) \in \mathbb{R}^d. \tag{1}$$

Specifically, the BoW approach obtains a codebook $\mathbb{Q}$ by k-means clustering, where $\mathbb{Q} = \{\mathbf{q}_1, ..., \mathbf{q}_K\}$ is comprising of K visual words, and the encoder r quantizes each local feature to its nearest visual word $\mathbf{q}_{1NN}$ from $\mathbb{Q}$:

$$r(\mathbf{x})_{BoW} = \mathbf{q}_{1NN}. \tag{2}$$

The VLAD encodes each local feature to its residual error:

$$r(\mathbf{x})_{VLAD} = \mathbf{x} - \mathbf{q}_{1NN}. \tag{3}$$

The FV [16] extends the discrete k-means clustering to probability GMM clustering. We denote the GMM codebook as: $\mathbf{q}_k = \{\omega_k, \mu_k, \sigma_k^2\}, k = 1, ..., K$, where ω_k, μ_k and σ_k^2 are the weight, mean vector and variance vector of the kth Gaussian (visual word), respectively. In this work, we derive the codes as the gradient of local feature's likelihood w.r.t. the mean μ_k of each Gaussian:

$$r(\mathbf{x})_{FV} = \gamma(k)\sigma_k^{-1}(\mathbf{x} - \mu_k), \tag{4}$$

where $\gamma(k) = \omega_k p_k(\mathbf{x}) / \sum_{l=1}^{K} \omega_l p_l(\mathbf{x})$ denotes the probability of local feature $\mathbf{x}$ being assigned to the kth Gaussian.

Feature aggregation accumulates the feature codes of local descriptors into a fixed-length vector representation for an image. State-of-the-art approaches usually employ average pooling to aggregate the feature codes for each visual word:

$$g(k) = \sum_{\mathbf{x} \in \mathbb{X}_k} f(r(\mathbf{x})), \tag{5}$$

where $\mathbb{X}_k$ represents the subset of local features in image $\mathbf{I}$ that are assigned to the kth visual word. $f(\cdot)$ denotes an operation on the feature codes $r(\mathbf{x})$. In the case of BoW, $f(\cdot)$ simply refers to the occurrences of each visual word in an image:

$$g(k)_{BoW} = \sum_{\mathbf{x} \in \mathbb{X}_k} 1. \tag{6}$$

Thus, the dimensionality of BoW representation is K.

While the VLAD and FV directly accumulate the residual vectors:

$$g(k)_{VLAD} = \sum_{\mathbf{x} \in \mathbb{X}_k} r(\mathbf{x})_{VLAD}, \tag{7}$$

$$g(k)_{FV} = \sum_{\mathbf{x} \in \mathbb{X}_k} r(\mathbf{x})_{FV}. \tag{8}$$

Finally, the VLAD and FV $\mathbf{g}$ are formed by concatenating the sub-vectors $\mathbf{g} = [g(0), ..., g(K)]$ of all visual words and is therefore Kd-dimensional.

Feature compression aims to compress high dimensional aggregated descriptors $\mathbf{g}$ into binary codes, which supports ultra-fast Hamming distance computation (XOR operation and bit count) as well as light storage of features for large scale image retrieval. For instance, the Compressed Fisher vector (CFV) [8] proposed to quantize each dimension of the FV representation into a single bit based on a sign function. Formally, we project each element g of descriptors $\mathbf{g}$ to 1 if $g > 0$; otherwise, 0:

$$sgn(g) = \begin{cases} 1 & \text{if } g > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Problem definition From Eq. 5, we observe that existing approaches unfairly assume that all local features contribute equally to the aggregation stage. As shown in Section IV, it significantly degenerates the retrieval accuracy, especially for lower quality JPEG queries. To address the problem, we propose a selective aggregation that injects a weight term $w(\mathbf{z})$ associated with local features $\mathbf{x}$ into Eq. 5:

$$g(k) = \sum_{\mathbf{x} \in \mathbb{X}_k} w(\mathbf{z}) f(r(\mathbf{x})), \tag{10}$$

where $w(\mathbf{z})$ is defined over the detected keypoints $\mathbf{z}$. In this work, we model the term $w(\mathbf{z})$ as a keypoint ranking function based on likelihood ratio test, which determines whether the corresponding local features $\mathbf{x}$ are involved in aggregation or not.

B. Keypoint Ranking for Selective Aggregation

From the matching point of view (see Fig. 2), informative local features tend to be associated with true positive match keypoints between images. Thus, to fulfill the selective aggregation of local features, we propose to learn $w(\mathbf{z})$ from the perspective of patch-level keypoint matching. In particular, we employ likelihood ratio test to accomplish the selection of local features for subsequent aggregation [18]:

$$w(\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{H}_0)}{p(\mathbf{z}|\mathbf{H}_1)}, \tag{11}$$

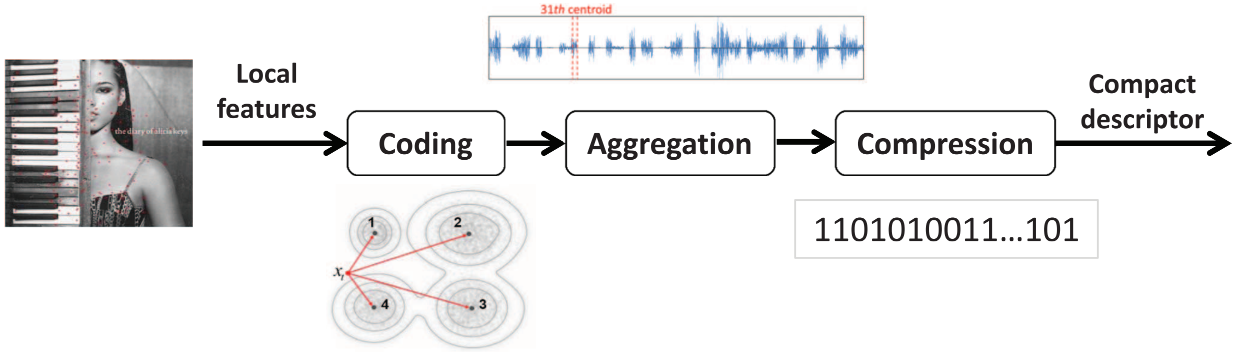

Fig. 3. The extraction pipeline of locally aggregated compact descriptors.

where hypothesis $\mathbf{H}_0$ and $\mathbf{H}_1$ represent whether keypoint $\mathbf{z}$ would be a correct match or not, respectively. $p(\mathbf{z}|\mathbf{H}_i), i = 0, 1$ is the probability density function for hypothesis $\mathbf{H}_i$, also referred to as the likelihood of the hypothesis $\mathbf{H}_i$ given a keypoint sample $\mathbf{z}$. The likelihood functions can be learned from match and non-match keypoint pairs, using the keypoints' characteristics $\mathbf{z}$ (e.g., scale, orientation, etc). During a test, we compute the values of $p(\mathbf{z}|\mathbf{H}_0)$ and $p(\mathbf{z}|\mathbf{H}_1)$ for a given keypoint $\mathbf{z}$, and predict the likelihood ratio of $\mathbf{z}$ being correctly matched or not using Eq. 11. We note that $w(\mathbf{z}) \to \infty$ if $p(\mathbf{z}|\mathbf{H}_0) \to 1$ and $p(\mathbf{z}|\mathbf{H}_1) \to 0$, which is the required objective.

In statistics, the hypothesis test has been widely applied to test if a certain statistical model fits the samples [23]. In this wok, we first train a universal model $p(\mathbf{z}|\lambda_{\mathbf{H}_1})$ with parameters $\lambda_{\mathbf{H}_1}$ for hypothesis $\mathbf{H}_1$ over the entire keypoint set $\mathbf{B}_{H_1}$ detected from training images. Rather than independently learning the model $p(\mathbf{z}|\lambda_{\mathbf{H}_0})$ for hypothesis $\mathbf{H}_0$, we adopt Bayesian adaptation to derive $\lambda_{\mathbf{H}_0}$ smoothly by updating the well-trained parameters $\lambda_{\mathbf{H}_1}$ of the universal model using the match keypoint set $\mathbf{B}_{H_0}$ generated from match image pairs. Bayesian adaptation is a popular modeling approach in speech and speaker recognition [23], which is able to harvest sufficient prior knowledge about the distribution of keypoints via the universal model.

Once the weight $w(\mathbf{z})$ is computed for each keypoint detected from an image, we propose to rank $w(\mathbf{z})$ of the detected keypoints in descending order, and then produce a binary decision for each local feature $\mathbf{x}$. Specifically, if $w(\mathbf{z})$ is in the top τ highest weight values, the corresponding descriptor $\mathbf{x}$ is adopted in aggregation (i.e., $w(\mathbf{z}) = 1$); otherwise, it is discarded (i.e., $w(\mathbf{z}) = 0$).

C. Parameter Estimation

Constructing the training keypoint set $\mathbf{B}_{H_1}$ and $\mathbf{B}_{H_0}$. Let $\Omega = \{\langle \mathbf{I}_n^l, \mathbf{I}_n^r \rangle\}_{n=1}^N$ denote N match image pairs, $(\mathbf{Z}_n^e, \mathbf{X}_n^e) = \{(\mathbf{z}_{nm}^e, \mathbf{x}_{nm}^e)|e \in \{l, r\}, m = 1...M_n\}$ denote a collection of detected keypoints $\mathbf{z}_{nm}^e$ and the corresponding descriptors $\mathbf{x}_{nm}^e$ extracted from image $\mathbf{I}_n^e$. The entire keypoint set $\mathbf{B}_{H_1} = \{\mathbf{z}_t|t = 1...B_1, \mathbf{z}_t \in \mathbf{Z}_n^e\}$.

We employ a distance ratio test [12] to compute match keypoint pairs $\mathbf{D}_n = \{\langle \mathbf{x}_{nd}^l, \mathbf{x}_{nd}^r \rangle | d = 1...D_n\}$ from $\langle \mathbf{X}_n^l, \mathbf{X}_n^r \rangle$, which may remove many false matches from background clutter. Subsequently, a geometric consistency check like RANSAC [22] is applied to divide $\mathbf{D}_n$ into inliers $\hat{\mathbf{D}}_n = \langle \hat{\mathbf{X}}_n^l, \hat{\mathbf{X}}_n^r \rangle = \{\langle \hat{\mathbf{x}}_{nd}^l, \hat{\mathbf{x}}_{nd}^r \rangle | d = 1...\hat{D}_n\}$ and outliers $\mathbf{D}_n \setminus \hat{\mathbf{D}}_n$. The inliers $\hat{\mathbf{D}}_n$ are finally considered as true positive matches. Finally, we construct the match keypoint set $\mathbf{B}_{H_0} = \{\mathbf{z}_t|t = 1...B_0, \mathbf{z}_t \in \hat{\mathbf{Z}}_n^e\}$, where $\hat{\mathbf{Z}}_n^e$ denotes the keypoints associated with $\hat{\mathbf{X}}_n^e$. Note that $\mathbf{B}_{H_1}$ contains both match and non-match keypoints, while $\mathbf{B}_{H_0}$ is a subset of $\mathbf{B}_{H_1}$.

Estimating model $p(\mathbf{z}|\lambda_{\mathbf{H}_1})$. Given the training set $\mathbf{B}_{H_1}$, we adopt a GMM model to learn the distribution of keypoint features $\mathbf{z}$ as:

$$p(\mathbf{z}|\lambda_{\mathbf{H}_1}) = \sum_{c=1}^{C} \tilde{\omega}_c p_c(\mathbf{z}), \tag{12}$$

where $\lambda_{\mathbf{H}_1} = \{\tilde{\omega}_c, \tilde{\mu}_c, \tilde{\sigma}_c^2\}_{c=1}^C$, C denotes the number of Gaussian components. The covariance matrices are assumed to be diagonal and the variance vector is denoted as $\tilde{\sigma}_c^2$. We learn the parameters $\lambda_{\mathbf{H}_1}$ by maximizing the likelihood of $\mathbf{B}_{H_1}$.

Estimating model $p(\mathbf{z}|\lambda_{\mathbf{H}_0})$. Given the match keypoint set $\mathbf{B}_{H_0}$ and the universal model $p(\mathbf{z}|\lambda_{\mathbf{H}_1})$, we perform Bayesian adaptation in twin-stage iteration. The first step is identical to the expectation step of EM algorithm, which uses B_0 keypoint samples $\mathbf{z}$ from $\mathbf{B}_{H_0}$ to calculate the sufficient statistics about the GMM parameters of weight, mean and variance:

$$n_c = \sum_{b=1}^{B_0} \gamma_b(c), \tag{13}$$

$$E_c(\mathbf{z}) = \frac{1}{n_c} \sum_{t=1}^{B_0} \gamma(c)\mathbf{z}, \tag{14}$$

$$E_c(\mathbf{z}^2) = \frac{1}{n_c} \sum_{t=1}^{B_0} \gamma(c)\mathbf{z}^2, \tag{15}$$

where $\gamma(c) = \frac{\tilde{\omega}_c p_c(\mathbf{z})}{\sum_{\tilde{c}=1}^{C} \tilde{\omega}_{\tilde{c}} p_{\tilde{c}}(\mathbf{z})}$ denotes the soft assignment of keypoint $\mathbf{z}$ to Gaussian c.

The second step is to apply the above sufficient statistics from $\mathbf{B}_{H_0}$ to update the parameters $\{\tilde{\omega}_c, \tilde{\mu}_c, \tilde{\sigma}_c^2\}$. The adapted parameters $\lambda_{\mathbf{H}_0} = \{\hat{\omega}_c, \hat{\mu}_c, \hat{\sigma}_c^2\}_{c=1}^{C}$ is derived as follows:

$$\hat{\omega}_c = \alpha_c^w n_c / B_0 + (1 - \alpha_c^w)\tilde{\omega}_c, \tag{16}$$

$$\hat{\mu}_c = \alpha_c^s E_c(\mathbf{z}) + (1 - \alpha_c^s)\tilde{\mu}_c, \tag{17}$$

$$\hat{\sigma}_c^2 = \alpha_c^t E_c(\mathbf{z}^2) + (1 - \alpha_c^t)(\tilde{\sigma}_c^2 + \tilde{\mu}_c^2) - \hat{\mu}_c^2, \tag{18}$$

where α_c^w, α_c^s and α_c^t are adaptation coefficients to control the impact of universal model on parameters updating. For example, when α_c^s is large, the statistics $E_c(\mathbf{z})$ from matched keypoints tend to dominate in Eq. 17. In this work, we define the coefficients $\alpha_c^\rho, \rho \in \{w, s, t\}$ as the ratios $\alpha_c^\rho = \frac{n_c}{n_c + \pi^\rho}$, where π^ρ is a constant relevance factor for parameter ρ and n_c is defined in Eq. 13.

IV. Experimental Results

A. Datasets and evaluation metrics

To evaluate the performance of the proposed approach, we carry out retrieval experiments over public available datasets, including heterogeneous categories of objects and scenes (see Table 1).

The **Graphics** dataset depicts 5 product categories including CDs, DVDs, books, text documents and business cards. There are 1,500 queries and 1,000 reference images. The queries are captured by mobile phones under widely varying lighting conditions with foreground or background clutter.

The **Painting** dataset contains 400 queries and 100 reference images for museum paintings, including history, portraits, landscapes and modern-art.

The **Frame** dataset is an image set of 500 video frames, containing diverse content like movies, news reports and sports. There are 400 queries taken by mobile phone from laptop, computer and TV screens to include typical specular distortions.

The **Landmark** dataset consists of 3,499 queries and 9,599 reference images collected from landmarks and buildings from the world.

The **UKbench** dataset contains images of 2,550 objects. Each one has 4 images taken from different viewpoints.

Query image compression. Each query image is compressed by JPEG compression with quality factor $Q = \{5, 10, 15, 20, 30, 50, 100\}$. The compression artifacts become stronger as the image quality decreases. Color images are converted to gray ones to save bits, as local features are extracted from luminance component. For each compression factor, 8,349 query images are generated.

For large scale experiments. We use a dataset *FLICKR1M* containing 1 million distractor images randomly downloaded from Flickr website. This image set is merged with the reference images to evaluate the accuracy and efficiency over a large-scale.

Evaluation measures. For all experiments we use the mean Average Precision (mAP) to measure the search accuracy. mAP is defined as follows:

$$mAP = \frac{1}{N_q}\sum_{i=1}^{N_q}\left(\frac{\sum_{r=1}^{N} P(r)}{\#\ relevant\ images}\right). \tag{19}$$

where N_q is the number of queries; N the number of relevant images for the i^{th} query; $P(r)$ is the precision at rank r. We deploy both the SIFT feature extraction (see Fig. 1(a)) and JPEG compression scheme (see Fig. 1(b)) on a HTC DESIRE G7 smart phone. This application is used to evaluate the extraction time and memory cost on mobile client, as well as the transmission delay of JPEG compressed query images over a WLAN link.

TABLE I
Statistics of the image datasets in the experiments.

Dataset	# query images	# database images
Graphics	1,500	1,000
Painting	400	100
Frame	400	100
Landmark	3,499	9,599
UKbench	2,550	7,650
FLICKR1M	–	1,000,000
In Total	8,349	1,018,449

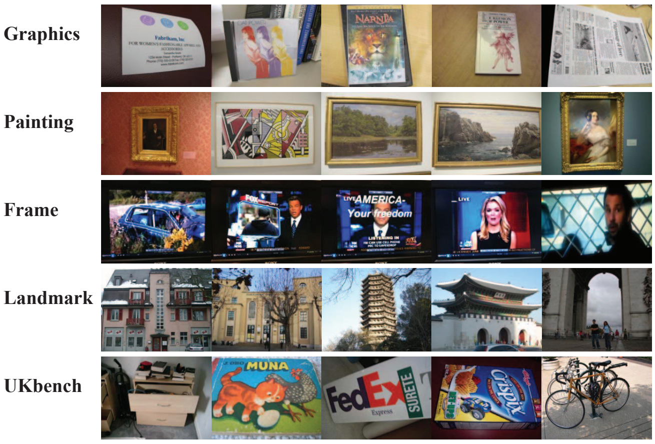

Fig. 4. Sample images from different test datasets.

B. Experiment setup

All the images are resized with reduced resolutions (max side ≤ 640 pixels). SIFT features are extracted by the VLFeat library. We employ independent image sets for all the training stages. Specifically, the MIRFLICKR25000 dataset is used to train all the vocabularies (i.e. GMM and k-means codebook). To obtain training keypoint set $\mathbf{B}_{H_1}$ and $\mathbf{B}_{H_0}$, we use the match/non-match image pairs from Oxford building and Caltech building datasets.

C. Baselines

(1) *Bag-of-Words (BoW)* [15]: We adopt hierarchical k-means clustering to train a vocabulary tree with depth 5 and branch factor 10, resulting in a 10^5 visual words. Inverted

index file is build up to implement efficient search. (2) *Residual Enhanced Visual Vector (REVV)* [10]: Chen et al. applied dimensionality reduction and sign binarization to compress the VLAD representation. (3) *Compressed Fisher vector (CFV)* [8]: The work in [8] employed sign binarization to quantize raw FV signature vector, which outperformed the state-of-the-art, such as Locality Sensitive Hashing and Spectral Hashing. (4) *Product Quantized SIFT (PQ-SIFT)* [17]: PQ-SIFT is the state-of-the-art compact local descriptor [17], following the pipeline in Fig. 1(a) which extracts and quantizes raw SIFT features by a product quantization technique [24]. (5) *Selective Aggregated BoW (BoW_SA)*: the proposed selective aggregation scheme combined with BoW. (6) *Selective Aggregated REVV (REVV_SA)*: the proposed selective aggregation scheme combined with REVV. (7) *Selective Aggregated CFV (CFV_SA)*: the proposed selective aggregation scheme combined with CFV.

D. Impact of parameter τ

We first study the impact of the number τ of aggregated local features for CFV aggregation (Quality $Q = 15$). As shown in Fig 5, the retrieval performance in terms of mAP over all test datasets is consistently improved when τ increased from 100 to 300, and the mAP rapidly decreases as the number of selected features increased from 300 to CFV_All (i.e., standard CFV). The optimal τ for all test datasets is about 300. For instance, on the Graphics dataset, $\tau = 300$ yields the best mAP 65.8%. To make clear the advantage of the selective aggregation scheme, we produce the results of CFV aggregation by randomly sampling 300 SIFT descriptors from each query (see Fig 5). We can see that the mAP of CFV_Rand is even much worse than standard CFV, e.g. 24.3% *vs.* 42.4% on Landmark dataset. This demonstrates the power of selective aggregation as well.

E. Compression factor analysis

Fig. 6 shows the retrieval mAP vs. JPEG compression factors over different datasets. Firstly, the selective aggregation significantly outperforms the state-of-the-art over all datasets at all compression factors. For $Q = 20$, BoW, REVV and CFV yield mAP 40.1%, 24.1% and 36.0% on average over all datasets, while the BoW_SA, REVV_SA and CFV_SA have achieved better mAP 59.1%, 62.8% and 64.2%, respectively. This gain may be attributed to the fact that the SA scheme discards less informative local features. Secondly, as the quality factor Q increases, all methods improve the performance progressively. For example, mAP is improved from 65.4% with $Q = 20$ to 72.7% with $Q = 50$ for CFV_SA on Painting dataset . More importantly, we observe that the SA scheme at low quality (e.g., $Q = 20$) performs better than the baselines at high quality (e.g., $Q = 100$). For instance, on Frame dataset, BoW_SA with $Q = 20$ outperforms BoW with $Q = 100$ (mAP 72.8% vs. 63.8%) by ∼10 times query size reduction (∼16KB vs. ∼170KB). The results demonstrated that the selective aggregation provides a trade-off between query size and search accuracy. Fig. 7 shows several groups of visualized retrieval performances using CFV_SA with comparisons to CFV with $Q = 50$.

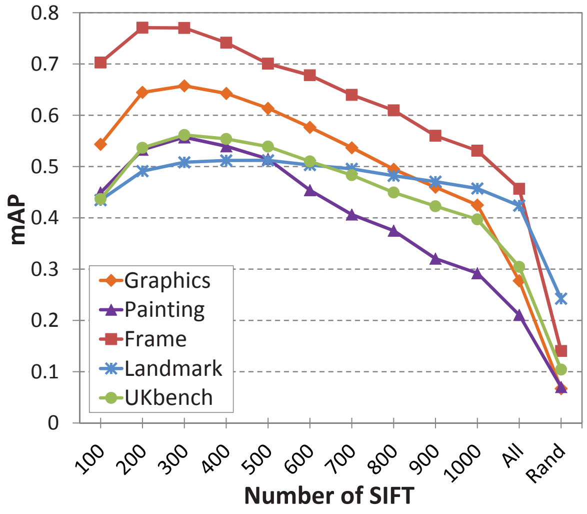

Fig. 5. Influence of the number of selected SIFTs for CFV aggregation on different datasets, combined with the 1 million distractor set FLICKR1M (Quality $Q = 15$ for query images).

TABLE II
COMPARISON OF THE PROPOSED CFV_SA ($Q = 20$) AND THE STATE-OF-THE-ART PQ-SIFT, IN TERMS OF MAP (%) ON DIFFERENT DATASETS, COMBINED WITH THE 1 MILLION DISTRACTOR SET FLICKR1M.

Dataset	CFV_SA	PQ-SIFT
Graphics	**67.4**	62.6
Painting	65.4	**72.8**
Frame	**78.8**	69.6
Landmark	**51.3**	42.8
UKbench	**58.2**	37.8

F. Comparison with the state-of-the-art

Table 2 compares the performance of CFV_SA ($Q = 20$) with PQ-SIFT [17][24] for comparable query size transmission on mobile client (i.e., ∼16KB). The CFV_SA obtains better mAP than PQ-SIFT over all datasets except the Painting dataset. For instance, CFV_SA achieves a much better mAP 58.2% on UKbench, while PQ-SIFT reports 37.8%. This is probably due to (1) PQ-SIFT causes considerable quantization error of local features and degenerates the subsequent retrieval performance; (2) the selective aggregation is able to reduce the negative impact of lossy JPEG compression.

TABLE III
MEMORY AND COMPUTATION TIME COMPARISON BETWEEN JPEG COMPRESSION AND SIFT EXTRACTION ON A HTC SMART PHONE BY AVERAGING COSTS FROM 1000 VGA SIZE QUERY IMAGES.

Complexity	JPEG compression	SIFT
Memory	∼0	∼18MB
Computation time (s)	∼0.1	∼2.2

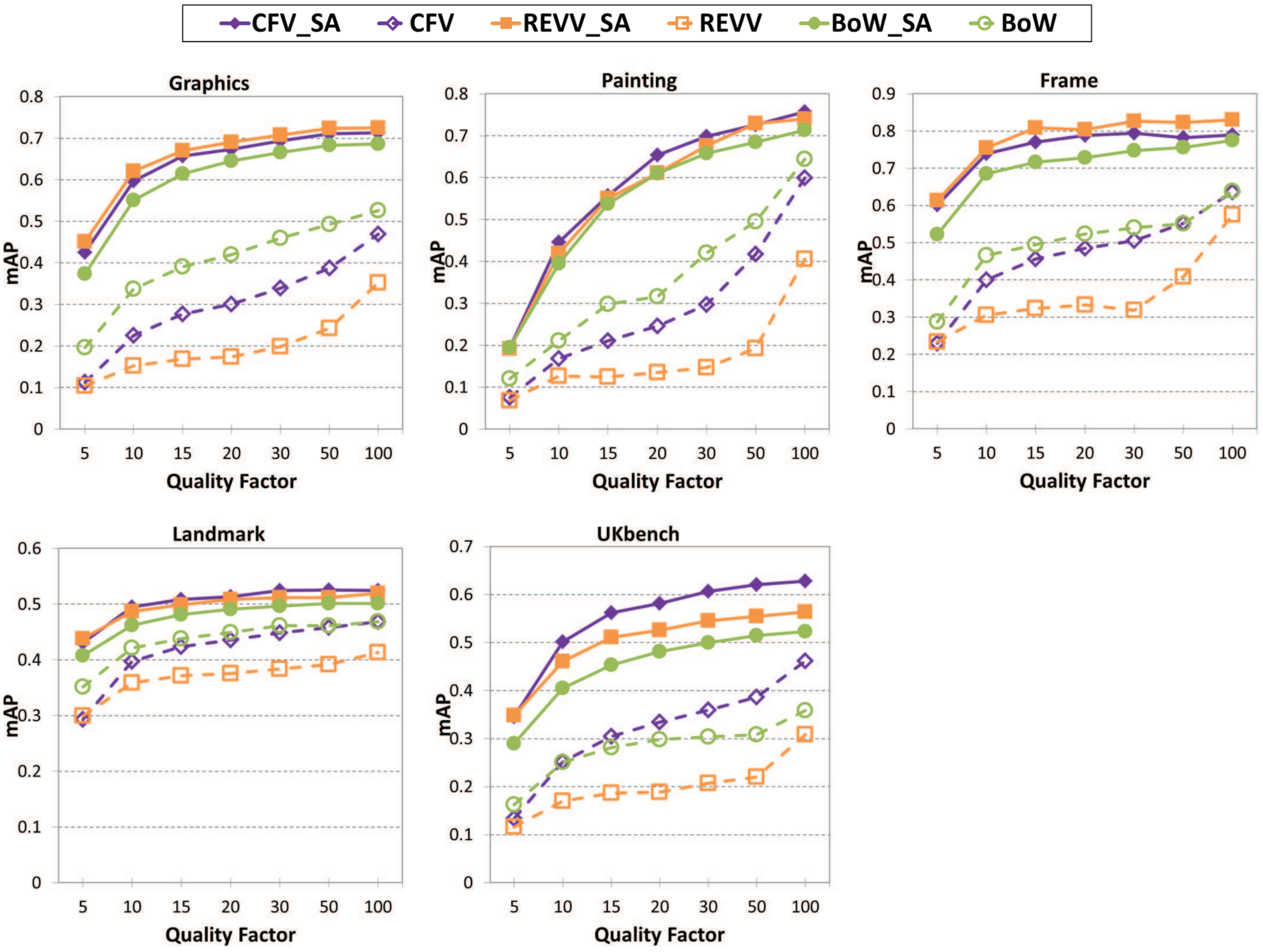

Fig. 6. mAP vs. query image with different JPEG quality factors over various types of datasets, combined with the 1 million distractor set FLICKR1M.

TABLE IV
QUERY IMAGE SIZE VS. TRANSMISSION TIME (VIA WLAN) AT DIFFERENT JPEG QUALITY FACTORS ON A HTC SMART PHONE BY AVERAGING TRANSMISSION TIME FROM 1000 VGA SIZE QUERY IMAGES.

Quality Factor	Image Size (KB)	Upload Time (s)
5	6.65	0.27
10	9.96	0.47
15	12.91	0.50
20	15.53	0.55
30	20.49	0.80
50	28.31	0.97
100	169.36	4.58

G. *Complexity analysis*

Memory and computation time. Table 3 compares the memory and computation (extraction) time between JPEG compression and SIFT extraction on a smart phone. The results show that directly sending a highly compressed JPEG image provides prominent advantages in terms of memory and computation time, compared to extracting local features directly on the mobile phone.

Transmission time. Table 4 reports the delivery time of JPEG compressed query images with different quality factors over a WLAN link. The time required to send the highest quality query image ($Q = 100$) is several times longer than the lower quality ones, which would cost serious energy consumption. Fortunately, the selective aggregation supports low quality query transmission, and battery saving may be expected.

V. CONCLUSION

We have proposed discriminative locally aggregated compact descriptors by informative local feature selection. The selective aggregation is able to reduce the negative effect of compression artifacts that appear in low quality JPEG query images, which enables low latency query delivery as well as power saving on the mobile client. In addition, the selective aggregation scheme supports fast similarity matching of descriptors based on Hamming distance and light storage of large scale database images. Our approach has shown promising retrieval performance over extensive benchmarks.

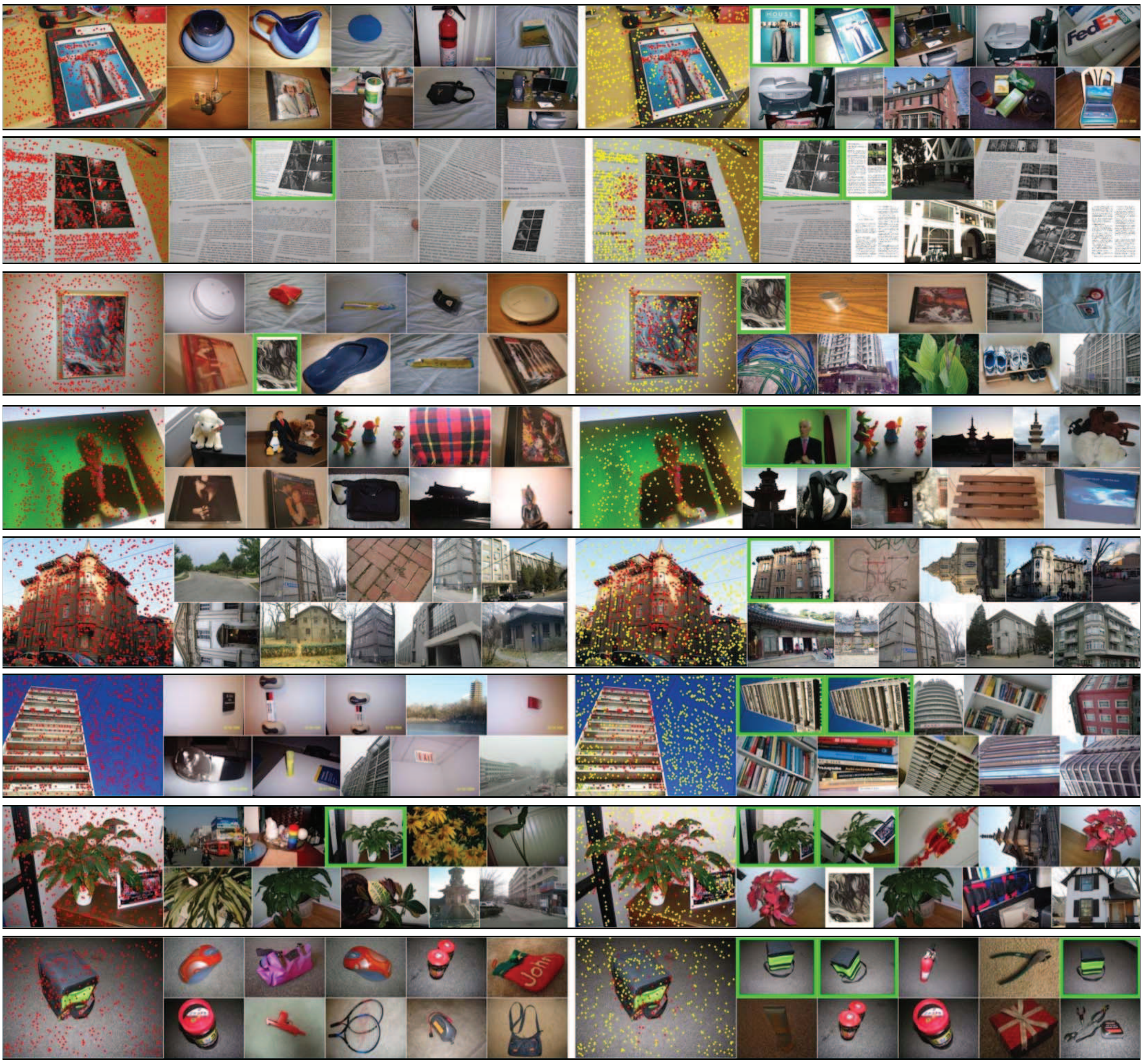

Fig. 7. The retrieval performance of CFV_SA in comparison to CFV [8] with JPEG query quality $Q = 50$. Each line corresponds to a query with top 10 dataset images returned. Left: CFV; Right: CFV_SA. The selected local features are denoted in red in query image, the noisy local features are in yellow. Green boxes indicate relevant image. The query images are randomly chosen from Graphics, Painting, Frame, Landmark and UKbench datasets.

REFERENCES

[1] Google Goggles. www.google.com/mobile/goggles.

[2] Amazon Flow. www.a9.com/whatwedo/mobile-technology/flow-powered-by-amazon.

[3] B. Girod, V. Chandrasekhar, D. Chen and *et al.*. Mobile Visual Search. *IEEE Signal Processing Magazine*, 2011.

[4] R. Ji, L.-Y. Duan, J. Chen and *et al.*. Location discriminative vocabulary coding for mobile landmark search. *IJCV*, 2012.

[5] R. Ji, L.-Y. Duan, J. Chen and *et al.*. Towards Low Bit Rate Mobile Visual Search with Multiple-Channel Coding. *ACM Multimedia*, 2011.

[6] V. Chandrasekhar, G. Takacs, D. Chen and *et al.*. Compressed Histogram of Gradients: A Low-Bitrate Descriptor. *IJCV*, 2012.

[7] V. Chandrasekhar, G. Takacs, D. Chen and *et al.*. Transform coding of image feature descriptors. *VCIP*, 2009.

[8] F. Perronnin, Y. Liu, J. Sanchez and *et al.*. Large-Scale Image Retrieval with Compressed Fisher Vectors. *CVPR*, 2010.

[9] H. Jegou, M. Douze, C. Schmid and *et. al*. Aggregating local descriptors into a compact image representation. *CVPR*, 2010.

[10] D. Chen, S. Tsai and *et al.* Residual enhanced visual vector as a compact signature for mobile visual search. *Signal Processing*, 2012.

[11] H. Jegou, F. Perronnin, M. Douze and *et. al*. Aggregating local images descriptors into compact codes. *PAMI*, 2012.

[12] D. G. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 2004.

[13] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust

features. *ECCV*. 2006.

[14] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. *ICCV*, 2003.

[15] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *CVPR*, 2006.

[16] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. *NIPS*, 1999.

[17] J. Chen, L.-Y. Duan, J. Lin, and *et al.*. On the Interoperability of Local Descriptors Compression. *ICASSP*, 2013.

[18] J. Lin, L.-Y. Duan, T. Huang, and W. Gao. Robust Fisher Codes for Large Scale Image Retrieval. *ICASSP*, 2013.

[19] J. Lin, L.-Y. Duan, Y. Huang, and *et al.*. Rate-adaptive Compact Fisher Codes for Mobile Visual Search. *IEEE Signal Processing Letters*, 2014.

[20] L.-Y. Duan, J. Lin, J. Chen, and *et al.*. Compact descriptors for visual search. *IEEE Multimedia*, 2014.

[21] D. Chen , S. Tsai, V. Chandrasekhar and *et al.*. Tree Histogram Coding for Mobile Image Matching. *DCC*, 2009.

[22] J. Philbin, O. Chum, M. Isard, and *et al.*. Object Retrieval with Large Vocabularies and Fast Spatial Matching. *CVPR*, 2007.

[23] D. Reynolds, T. Quatieri, R. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 2000.

[24] H. Jegou, M. Douze, C. Schmid. Product quantization for nearest neighbor search. *PAMI*, 2011.

[25] H. Jegou, M. Douze, C. Schmid. Improving Bag-of-Features for Large Scale Image Search. *IJCV*, 2010.