Image Retrieval with Fisher Vectors of Binary Features

Yusuke Uchida and Shigeyuki Sakazawa
KDDI R&D Laboratories, Inc.
Saitama, Japan
$\{$*ys-uchida, sakazawa*$\}$*@kddilabs.jp*

***Abstract*—Recently, the Fisher vector representation of local features has attracted much attention because of its effectiveness in both image classification and image retrieval. Another trend in the area of image retrieval is the use of binary feature such as ORB, FREAK, and BRISK. Considering the significant performance improvement in terms of accuracy in both image classification and retrieval by the Fisher vector of continuous feature descriptors, if the Fisher vector were also to be applied to binary features, we would receive the same benefits in binary feature based image retrieval and classification. In this paper, we derive the closed-form approximation of the Fisher vector of binary features which are modeled by the Bernoulli mixture model. In experiments, it is shown that the Fisher vector representation improves the accuracy of image retrieval by 25% compared with a bag of binary words approach.**

***Keywords*-Fisher vector, Bernoulli mixture model, binary feature, image retrieval**

I. Introduction

With the advancement of both stable interest region detectors [1] and robust and distinctive descriptors [2], local feature-based image or object retrieval has attracted a great deal of attention. In local feature based image retrieval or recognition, each image is first represented by a set of local feature vectors $X = \{x_1, \cdots, x_t, \cdots, x_T\}$. A set of feature vectors X is then encoded into a fixed length vector in order to calculate (dis)similarity between sets of feature vectors. The most frequently used method is a bag-of-visual words (BoVW) representation [3], where feature vectors are quantized into visual words (VWs) using a visual codebook, resulting in a histogram representation of VWs.

Recently, the Fisher vector representation [4] has attracted much attention because of its effectiveness. The Fisher vector is defined by the gradient of log-likelihood function normalized with the Fisher information matrix. In [4], features vectors are modeled by the Gaussian mixture model (GMM) and a closed form approximation is first proposed for the Fisher information matrix of a GMM. Later, the performance of the Fisher vector is improved in [5] by using power and ℓ_2 normalization. Because the Fisher vector can represent higher order information than the BoVW representation, it is shown that it can outperform the BoVW representation in both image classification [5] and image retrieval tasks [6]–[8].

Table I
Position of this paper.

Feature type	BoVW	Fisher Vector
Continuous	[3]	[4]
Binary	[15]	This paper

Another trend in the area of image retrieval is the use of binary features such as ORB [9], FREAK [10], and BRISK [11]. Binary features are one or two orders of magnitude faster than SIFT or SURF in detection and description, while providing comparable performance. These binary features are especially suitable for mobile visual search or augmented reality on mobile devices [12]. While the Fisher vector is widely applied to continuous feature descriptors (e.g., SIFT) which can be modeled by the GMM, to the best of our knowledge, there has been no attempt to apply the Fisher vector to the recent binary features referred to above for the purpose of image retrieval. Considering the significant performance improvement in terms of accuracy in both image classification and retrieval by the Fisher vector of continuous feature descriptors, if the Fisher vector were also to be applied to binary features, we would receive the same benefits in binary feature based image retrieval and classification. In this paper, we derive the closed-form approximation of the Fisher vector of binary features which are modeled by the Bernoulli mixture model. Table I shows the position of this paper. In experiments, we evaluate the effectiveness of both the Fisher vector of binary features and their associated normalization approaches. The proposed Fisher vector representation of binary features is general and not restricted to image features; it is also expected to be applied to other modalities such as audio signals [13], [14].

The rest of this paper is organized as follows. In Section 2, recent binary features are briefly introduced. In Section 3, the BoVW and Fisher vector image representations are described. In Section 4, we derive the Fisher vector representation of binary features. In Section 5, the effectiveness of the Fisher vector of binary features is confirmed. Our conclusions are presented in Section 6.

II. Local binary features

Recently, binary features such as ORB [9], FREAK [10], and BRISK [11] have attracted much attention [16]. Binary features are one or two orders of magnitude faster than SIFT or SURF features in extraction, while providing comparable performance to SIFT and SURF. Resulting binary feature vectors are more compact than SIFT or SURF feature vectors. In this section, recent binary features are briefly introduced.

A. Detection

Most of the local binary features employ fast feature detectors. The ORB feature utilizes the FAST [17] detector, which detects pixels that are brighter or darker than neighboring pixels based on the accelerated segment test. The test is optimized to reject candidate pixels very quickly, realizing extremely fast feature detection. In order to ensure approximate scale invariance, feature points are detected from an image pyramid. The FREAK and BRISK features adopt the multi-scale version of the AGAST [18] detector. Although the AGAST detector is based on the same criteria as FAST, the detection is accelerated by using an optimal decision tree in deciding whether each pixel satisfies the criteria or not.

B. Description

Local binary features extract binary strings from patches of interest regions instead of extracting gradient-based high-dimensional feature vectors like SIFT. Many methods utilize binary tests in extracting binary strings. The BRIEF descriptor [19], a pioneering work in the area of binary descriptors, is a bit string description of an image patch constructed from a set of binary intensity tests. Consider the t-th smoothed image patch p_t, a binary test τ for d-th bit is defined by:

$$x_{td} = \tau(p_t; a_d, b_d) = \begin{cases} 1 & \text{if } p_t(a_d) \geq p_t(b_d) \\ 0 & \text{else} \end{cases}, \quad (1)$$

where a_d and b_d denote relative positions in the patch p_t, and $p_t(\cdot)$ denotes the intensity at the point. Using D independent tests, we obtain D-bit binary string $x_t = (x_{t1}, \cdots, x_{td}, \cdots, x_{tD})$ for the patch p_t. The ORB feature employs a learning method for de-correlating BRIEF features under rotational invariance. Although the BRISK and FREAK features use different sampling patterns from BRIEF, they are also based on a set of binary intensity tests. These binary features are designed so that each bit has the same probability of being 1 or 0, and bits are uncorrelated.

In addition to these methods which extract binary features directly, there are many methods which encode continuous feature descriptors (e.g., SIFT) into compact binary codes [20]–[24]. By using these methods, an image can also be represented as a set of binary features.

III. Image representations

In local feature based image retrieval or recognition, each image is first represented by a set of local features $X = \{x_1, \cdots, x_t, \cdots, x_T\}$. A set of features X is then encoded into a fixed length vector in order to calculate (dis)similarity between sets of features. In this section, two encoding methods are introduced.

A. Bag-of-Visual Words

The BoVW framework is the de-facto standard way to encode local features into a fixed length vector. In the BoVW framework, feature vectors are quantized into VWs using a visual codebook, resulting in a histogram representation of VWs. Image (dis)similarity is measured by L_1 or L_2 distance between the normalized histograms. Although it was first proposed for an image retrieval task [3], it is now widely used for both image retrieval [25]–[28] and image classification [29], [30]. In [15], the bag-of-visual words approach is also applied to binary features.

B. Fisher kernel and Fisher vector

Fisher kernel is a powerful tool for combining the benefits of generative and discriminative approaches [31]. Let $X = \{x_1, \cdots, x_t, \cdots, x_T\}$ denote the set of T local feature vectors extracted from an image. We assume that the generation process of X can be modeled by a probability density function $p(X|\lambda)$ whose parameters are denoted by λ. In [31], it is proposed to describe X by the gradient G_λ^X of the log-likelihood function, which is also referred to as the Fisher score:

$$G_\lambda^X = \frac{1}{T} \nabla_\lambda \mathcal{L}(X|\lambda), \quad (2)$$

where $\mathcal{L}(X|\lambda)$ denotes the log-likelihood function:

$$\mathcal{L}(X|\lambda) = \log p(X|\lambda). \quad (3)$$

The gradient vector describes the direction in which parameters should be modified to best fit the data [4]. A natural kernel on these gradients is the Fisher kernel [31], which is based on the idea of natural gradient [32]:

$$K(X, Y) = G_\lambda^X F_\lambda^{-1} G_\lambda^Y. \quad (4)$$

F_λ is the Fisher information matrix of $p(X|\lambda)$ defined as

$$F_\lambda = \mathrm{E}_X[\nabla_\lambda \mathcal{L}(X|\lambda)\ \nabla_\lambda \mathcal{L}(X|\lambda)^{\mathrm{T}}]. \quad (5)$$

Because F_λ^{-1} is positive semidefinite and symmetric, it has a Cholesky decomposition $F_\lambda^{-1} = L_\lambda^{\mathrm{T}} L_\lambda$. Therefore the Fisher kernel is rewritten as a dot-product between normalized gradient vectors $\mathcal{G}_\lambda^X$ with:

$$\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X. \quad (6)$$

The normalized gradient vector $\mathcal{G}_\lambda^X$ is referred to as the Fisher vector of X [5].

In [4], the generation process of feature vectors (SIFT) are modeled by the GMM, and the diagonal closed-form approximation of the Fisher vector is derived. Then, the performance of the Fisher vector is significantly improved in [5] by using power normalization and ℓ_2 normalization. The Fisher vector framework has achieved promising results and is becoming the new standard in both image classification [5] and image retrieval tasks [6]–[8].

While the Fisher vector is widely applied to continuous feature descriptors (e.g., SIFT) which can be modeled by the GMM, to the best of our knowledge, there has been no attempt to apply the Fisher vector to recent binary features such as ORB [9] for the purpose of image retrieval. In this paper, we derive the closed-form approximation of the Fisher vector of binary features which are modeled by the Bernoulli mixture model, and evaluate the effectiveness of both the Fisher vector of binary features and their associated normalization approaches.

IV. Fisher vector for binary features

In this section, we model binary features with the Bernoulli distribution, and derive the Fisher vector representation of binary features.

A. Bernoulli mixture model

Let $x_t \in \{0,1\}^D$ denote a D-dimensional binary feature out of T binary features $X = \{x_1, \cdots, x_t, \cdots, x_T\}$ extracted from an image. In modeling binary features, it is straightforward to adopt a single multivariate Bernoulli distribution. However, although many binary descriptors are designed so that bits of resulting binary features are uncorrelated [9], there are still strong dependencies among bits. Therefore, a single multivariate Bernoulli component will be inadequate to cope with the kind of complex bit dependencies that often underlie binary features. This drawback is overcome when several Bernoulli components are adequately mixed. In this paper, we propose to model binary features with a multivariate Bernoulli mixture (BMM). The use of the BMM instead of a single multivariate Bernoulli distribution will be justified in the experimental section.

Let $\lambda = \{w_i, \mu_{id}, i = 1..N, d = 1..D\}$ denote a set of parameters for a multivariate Bernoulli mixture model with N components, and x_{td} represents the d-th bit of x_t. Given the parameter set λ, the probability density function of T binary features X is described as:

$$\begin{aligned} p(X|\lambda) &= \prod_{t=1}^{T} p(x_t|\lambda), \\ p(x_t|\lambda) &= \sum_{i=1}^{N} w_i p_i(x_t|\lambda), \\ p_i(x_t|\lambda) &= \prod_{d=1}^{D} \mu_{id}^{x_{td}} (1-\mu_{id})^{1-x_{td}}. \end{aligned} \tag{7}$$

In order to estimate the values of the parameter set λ, given a set of training binary features $x_1, \cdots, x_s, \cdots, x_S$, the expectation-maximization (EM) algorithm is applied [33]. In the expectation step, the occupancy probability $\gamma_s(i)$ (or posterior probability $p(i|x_s, \lambda)$) of x_s being generated by the i-th component of BMM is calculated as

$$\gamma_s(i) = p(i|x_s, \lambda) = \frac{w_i p_i(x_s|\lambda)}{\sum_{j=1}^{N} w_j p_j(x_s|\lambda)}. \tag{8}$$

In the maximization step, the parameters are updated as

$$S_i = \sum_{s=1}^{S} \gamma_s(i), \; w_i = S_i/S, \; \mu_{id} = \frac{1}{S_i} \sum_{s=1}^{S} \gamma_s(i) x_{sd}. \tag{9}$$

In our implementation, parameter w_i is initialized with $1/N$, and μ_{id} is with uniform distribution $U(0.25, 0.75)$.

B. Deriving the Fisher vector of the BMM

In this section, we derive the Fisher vector of the BMM. In order to calculate the Fisher vector $\mathcal{G}_\lambda^X$ in Eq. (6), the Fisher score G_λ^X in Eq. (2) and the Fisher information matrix F_λ in Eq. (5) should be calculated. In this paper, we consider only the Fisher vector w.r.t. the parameter μ_{id}, because the Fisher vector w.r.t. the weight parameter w_i does not contribute to the performance [4]. The derivation of the Fisher vector w.r.t. w_i is the same as that of GMM [4].

Letting $G_{\mu_{id}}^X$ denote the Fisher score w.r.t. the parameter $\mu_{id} \in \lambda$, $G_{\mu_{id}}^X$ is calculated as:

$$\begin{aligned} G_{\mu_{id}}^X &= \frac{1}{T} \frac{\partial \mathcal{L}(X|\lambda)}{\partial \mu_{id}} = \frac{1}{T} \sum_{t=1}^{T} \frac{\partial \mathcal{L}(x_t|\lambda)}{\partial \mu_{id}} \\ &= \frac{1}{T} \sum_{t=1}^{T} \frac{1}{p_i(x_t|\lambda)} \frac{\partial p_i(x_t|\lambda)}{\partial \mu_{id}}. \end{aligned} \tag{10}$$

Considering that x_{td} in Eq. (7) can only be 0 or 1, we get:

$$\frac{\partial p_i(x_t|\lambda)}{\partial \mu_{id}} = (-1)^{1-x_{td}} \prod_{e=1, e \neq d}^{D} \mu_{ie}^{x_{te}} (1-\mu_{ie})^{1-x_{te}}. \tag{11}$$

Finally we obtain:

$$G_{\mu_{id}}^X = \frac{1}{T} \sum_{t=1}^{T} \gamma_t(i) \frac{(-1)^{1-x_{td}}}{\mu_{id}^{x_{td}} (1-\mu_{id})^{1-x_{td}}}, \tag{12}$$

where $\gamma_t(i)$ is the occupancy probability defined in Eq. (8).

Then, we derive the approximate Fisher information matrix of the BMM under the following three assumptions [4]: (1) the Fisher information matrix F_λ is diagonal, (2) the number of binary features x_t extracted from an image is constant and equal to T, and (3) the occupancy probability $\gamma_s(i)$ is peaky; there is one index i such that $\gamma_s(i) \approx 1$ and that $\forall j \neq i$, $\gamma_s(j) \approx 0$.

As we assume the Fisher information matrix is diagonal, Eq. (5) is approximated as $F_\lambda \approx \mathrm{diag}(F_{\mu_{11}}, \cdots, F_{\mu_{ND}})$, where $F_{\mu_{id}}$ denotes the Fisher information w.r.t. μ_{id}:

$$F_{\mu_{id}} = \mathrm{E}\left[\left(\frac{\partial \mathcal{L}(X|\lambda)}{\partial \mu_{id}} \right)^2 \right]. \tag{13}$$

Then, with the (2) and (3) assumptions, we approximately obtain:

$$F_{\mu_{id}} = T w_i \left(\frac{\sum_{j=1}^{N} w_j \mu_{jd}}{\mu_{id}^2} + \frac{\sum_{j=1}^{N} w_j (1-\mu_{jd})}{(1-\mu_{id})^2} \right). \tag{14}$$

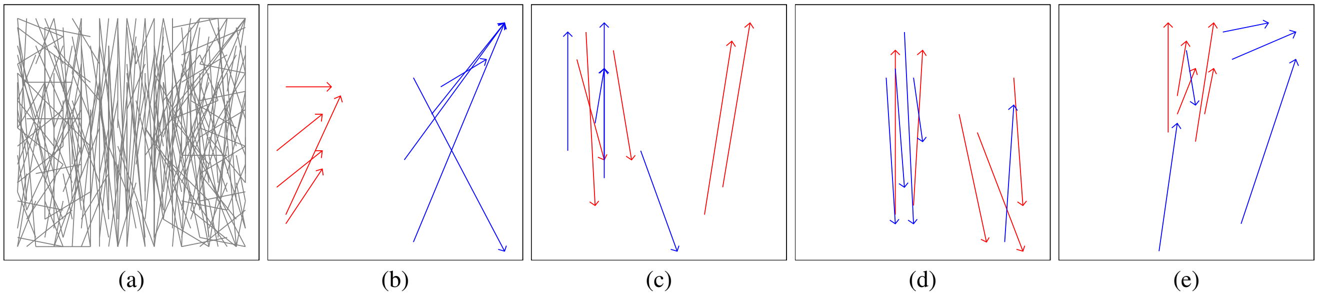

(a) (b) (c) (d) (e)

Figure 1. (a) all point pairs of 256 binary tests used in the ORB descriptor. (b)-(e) five tests corresponding to the bits with the top five probabilities μ_{id} of being 1 (red) and 0 (blue). Four components out of $N = 32$ components are shown.

Please refer to Appendix for the derivation. Finally, the Fisher vector $\mathcal{G}_\lambda^X$ is obtained with the concatenation of normalized Fisher scores $F_{\mu_{id}}^{-1/2} G_{\mu_{id}}^X$ $(i = 1..N, d = 1..D)$.

The Fisher vector is further normalized with power normalization and ℓ_2 normalization [5]. Given a Fisher vector $z = \mathcal{G}_\lambda^X$, the power-normalized vector $f(z)$ is calculated as

$$f(z) = \text{sign}(z)|z|^\alpha. \qquad (15)$$

In experiments, we set $\alpha = 0.5$ as recommended in [5]. After the power normalization, ℓ_2 normalization is performed to $f(z)$, resulting in the final Fisher vector representation of the set of binary features.

V. Experiment

In the experiments, the Stanford mobile visual search dataset[1] is used. It contains camera-phone images of products, CDs, books, outdoor landmarks, business cards, text documents, museum paintings and video clips. While it includes eight classes of images, we use general CD class images in this paper. These images consist of 100 reference images and 400 query images. In the experiments, because some query images are too large (10M pixels), all images are resized so that the long sides of images are less than 640 pixels, keeping the original aspect ratio.

As an indicator of retrieval performance, mean average precision (MAP) [25], [27] is used. For each query, a precision-recall curve is obtained based on the retrieval results. Average precision is calculated as the area under the precision-recall curve. Finally, the MAP score is calculated as the mean of average precisions over all queries.

We adopt the ORB [9] descriptor as a binary feature because of its efficiency. On average, 900 features are extracted from 4 scales. The parameter set λ is estimated with the EM algorithm using one million ORB binary features extracted from the MIR Flickr collection[2].

First, we investigate a clustering results performed in the estimation of the parameter set λ of the BMM with $N = 32$ components. Figure 1 (a) represents all point pairs of 256 binary tests used in the ORB descriptor. Figure 1 (b)-(e) represent five tests corresponding to the bits with the top five probabilities μ_{id} of being 1 (red) and 0 (blue) in four components out of $N = 32$ components. It implies that some bits of the ORB descriptor are highly correlated and that the BMM successfully captures this correlation. The result justifies the use of the BMM instead of single multivariate Bernoulli distribution to model binary features.

Then, the performance of the Fisher vector of binary features is evaluated in terms of image retrieval accuracy. Dissimilarity between two images is defined by the Euclidean distance between the BoVW or the Fisher vector representations of the images. The following five methods are compared: (1) bag of binary words approach (BoBW) [15], (2) Fisher vector without normalization (FV), (3) Fisher vector with ℓ_2 normalization (L2 Norm), (4) Fisher vector with power normalization (P Norm), and (5) Fisher vector with both power and ℓ_2 normalization (P+L2 Norm). For BoBW, a visual codebook with 1024 centroids is used.

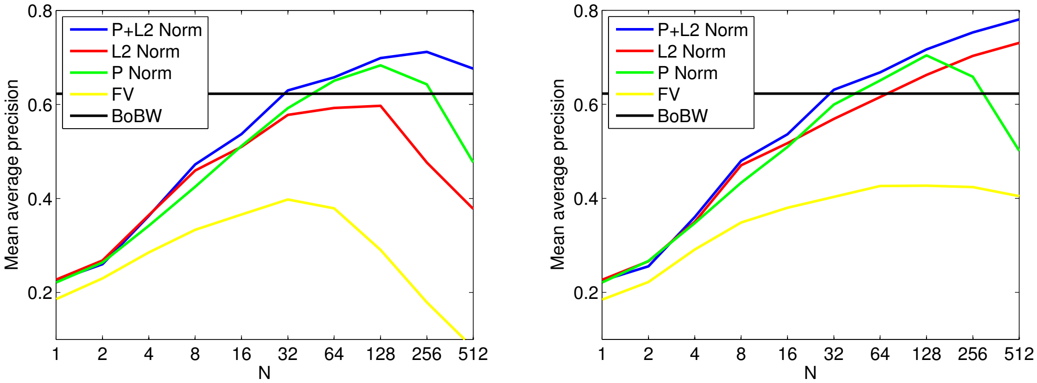
Figure 2 shows a comparison of the Fisher vector and BoVW representations applied to binary features, where the Fisher information matrix is assumed to be the identity matrix $\text{diag}(1, \cdots, 1)$ in Figure 2 (a), while the approximate Fisher information matrix derived in this paper is used in Figure 2 (b). The accuracy of the Fisher vector without any normalization (FV) is disappointing compared with the BoBW framework. A little surprisingly, even if the Fisher information matrix is approximated to the identity matrix, the accuracy is improved from 0.623 (BoBW) to 0.712 (P+L2 Norm $N = 256$) in Figure 2 (a). If the proposed Fisher information matrix is adopted, the accuracy is further improved from 0.712 to 0.781 (P+L2 Norm $N = 512$) as shown in Figure 2 (b). This is because the Euclidean metric is not an appropriate metric in the parameter space. We can also see that the accuracy improves as the number N of components increases, which is consistent with the case of SIFT+GMM [8].

VI. Conclusions

In this paper, we derived the closed-form approximation of the Fisher vector of binary features which are modeled by the Bernoulli mixture model. The effectiveness of the

[1] http://www.stanford.edu/~dmchen/mvs.html

[2] http://press.liacs.nl/mirflickr/



(a) $F_\lambda \approx \mathrm{diag}(1, \cdots, 1)$ (b) $F_\lambda \approx \mathrm{diag}(F_{\mu_{11}}, \cdots, F_{\mu_{ND}})$

Figure 2. Comparison of the Fisher vector and BoVW representations applied to binary features.

Fisher vector of binary features was confirmed. There were some interesting observations such as that the performance of the Fisher vector without power and ℓ_2 normalization is very poor, while the Fisher vector with power and ℓ_2 normalization outperforms the BoBW framework even if the Fisher information matrix is approximated by the identity matrix. In future, we will apply the Fisher vector of binary features to image classification problems. We are also interested in exploring the scalability of the Fisher vector of binary features for large-scale image retrieval. We also expect that the proposed Fisher vector representation can also be successfully applied to other modalities such as audio signals.

Appendix

We derive the Fisher information matrix under the following three assumptions: (1) the Fisher information matrix F_λ is diagonal, (2) the number of binary features x_t extracted from an image is constant and equal to T, and (3) the occupancy probability $\gamma_s(i)$ is peaky. From Eq. (13), we get:

$$\begin{aligned} F_{\mu_{id}} &= \mathrm{E}\left[\left(\frac{\partial\mathcal{L}(X|\lambda)}{\partial\mu_{id}}\right)^2\right] = \mathrm{E}\left[\left(\sum_{t=1}^{T}\frac{\partial\mathcal{L}(x_t|\lambda)}{\partial\mu_{id}}\right)^2\right] \\ &= \sum_{t=1}^{T}\mathrm{E}\left[\left(\frac{\partial\mathcal{L}(x_t|\lambda)}{\partial\mu_{id}}\right)^2\right] \\ &\quad +2\sum_{1\le t<s\le T}\mathrm{E}\left[\frac{\partial\mathcal{L}(x_t|\lambda)}{\partial\mu_{id}}\right]\mathrm{E}\left[\frac{\partial\mathcal{L}(x_s|\lambda)}{\partial\mu_{id}}\right]. \quad (16) \end{aligned}$$

If the parameter set λ is estimated with maximum-likelihood estimation, we have:

$$\mathrm{E}\left[\frac{\partial\mathcal{L}(x_t|\lambda)}{\partial\mu_{id}}\right] = 0. \quad (17)$$

Using the value of the Fisher score in Eq. (12), we get:

$$\begin{aligned} \mathrm{E}\left[\left(\frac{\partial\mathcal{L}(x_t|\lambda)}{\partial\mu_{id}}\right)^2\right] &= \int_{x_t} p(x_t|\lambda)\frac{\gamma_t^2(i)}{(\mu_{id}^{x_{td}}(1-\mu_{id})^{1-x_{td}})^2}dx_t \\ &= \int_{x_{td}=1} p(x_t|\lambda)\frac{\gamma_t^2(i)}{\mu_{id}^2}dx_t \\ &\quad + \int_{x_{td}=0} p(x_t|\lambda)\frac{\gamma_t^2(i)}{(1-\mu_{id})^2}dx_t. \quad (18) \end{aligned}$$

Using the assumption that the occupancy probability $\gamma_t(i)$ is peaky, we approximate $\gamma_t^2(i)$ as $\gamma_t(i)$ in Eq. (18). Finally, using the following equations,

$$\begin{aligned} \int_{x_{td}=1} p(x_t|\lambda)\gamma_t(i)dx_t &= w_i\sum_{j=1}^{N} w_j\mu_{jd}, \\ \int_{x_{td}=0} p(x_t|\lambda)\gamma_t(i)dx_t &= w_i\sum_{j=1}^{N} w_j(1-\mu_{jd}), \quad (19) \end{aligned}$$

we obtain:

$$F_{\mu_{id}} = Tw_i\left(\frac{\sum_{j=1}^{N} w_j\mu_{jd}}{\mu_{id}^2} + \frac{\sum_{j=1}^{N} w_j(1-\mu_{jd})}{(1-\mu_{id})^2}\right). \quad (20)$$

References

[1] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *IJCV*, vol. 60, no. 1-2, pp. 43–72, Nov. 2005.

[2] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *TPAMI*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[3] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. of ICCV*, 2003, pp. 1470–1477.

[4] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. of CVPR*, 2007.

[5] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. of ECCV*, 2010, pp. 143–156.

[6] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Proc. of CVPR*, 2010, pp. 3384–3391.

[7] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. of CVPR*, 2010, pp. 3304–3311.

[8] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *TPAMI*, vol. 34, no. 9, pp. 1704–1716, 2012.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. of ICCV*, 2011, pp. 2564–2571.

[10] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Proc. of CVPR*, 2012, pp. 510–517.

[11] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Proc. of ICCV*, 2011, pp. 2548–2555.

[12] X. Yang and K. Cheng, "Ldb: An ultra-fast feature for scalable augmented reality on mobile devices," in *Proc. of ISMAR*, 2012, pp. 49–57.

[13] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. of ISMIR*, 2002, pp. 107–115.

[14] X. Anguera, A. Garzon, and T. Adamek, "Mask: Robust local features for audio fingerprinting," in *Proc. of ICME*, 2012.

[15] D. Gálvez-López and J. D. Tardós, "Real-time loop detection with bags of binary words," in *Proc. of IROS*, 2011, pp. 51–58.

[16] J. Heinly, E. Dunn, and J.-M. Frahm, "Comparative evaluation of binary features," in *Proc. of ECCV*, 2012, pp. 759–773.

[17] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proc. of ICCV*, 2005, pp. 1508–1515.

[18] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Proc. of ECCV*, 2010.

[19] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proc. of ECCV*, 2010, pp. 778–792.

[20] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. of NIPS*, 2009.

[21] J. Wang, S. Kumar, and S. F. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proc. of CVPR*, 2010, pp. 3424–3431.

[22] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. of CVPR*, 2011, pp. 817–824.

[23] M. Ambai and Y. Yoshida, "Card: Compact and real-time descriptors," in *Proc. of ICCV*, 2011.

[24] Y. Lee, J. Heo, and S. Yoon, "Quadra-embedding: Binary code embedding with low quantization error," in *Proc. of ACCV*, 2012.

[25] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. of CVPR*, 2006, pp. 2161–2168.

[26] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. of CVPR*, 2007, pp. 1–8.

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

[28] Y. Uchida, K. Takagi, and S. Sakazawa, "An alternative to idf: Effective scoring for accurate image retrieval with non-parametric density ratio estimation," in *Proc. of ICPR*, 2012.

[29] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. of CVPR*, 2006, pp. 2169–2178.

[30] Y. Jiang, C. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proc. of CIVR*, 2007, pp. 494–501.

[31] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proc. of NIPS*, 1998, pp. 487–493.

[32] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[33] A. Juan and E. Vidal, "Bernoulli mixture models for binary images," in *Proc. of ICPR*, 2004, pp. 367–370.