2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing

A Novel Progressive Transmission in Mobile Visual Search

Qi Zhang Zhiyang Li Yegang Du Wenyu Qu
College of Information Science and Technology
Dalian Maritime University
Dalian 116023, P.R.China
Email: qizhangonly@gmail.com

Xiaoxuan Zhang
Logistics Department
Unit 91550
Dalian 116023, P.R.China
Email: xinhua0830@163.com

***Abstract*—How to reduce the transmission latency is a main concern in the context of Mobile Visual Search (MVS). Transmitting extremely compacted visual descriptor in a progressive manner is the start-of-art solution. In this paper, we present a novel MVS system following the client-server architecture. To reduce the transmission latency, the inquiry image is represented by a set of hash bits, which are then progressively transmitted. In the server side, all images are indexed by their hash bits, similar as the classic Bag-of-Word (BoW) model. Owe to the merit of the proposed system, the IDF weight of the hash bits are encoded into a sparse vector which retained in the mobile client, and provides a transmission order of the inquiry hash bits. The hash bit with lower IDF weight will be more discriminative, which should have higher priority during the transmission. As far as we know, this work is the first one attempting to transmit the hash bits in a proper progressive manner in MVS. Extensive experiments have been done on the public Stanford MVS database, demonstrating that the proposed progressive transmission strategy achieves higher recognition rate compared to other strategies, when delivering the same amount of data.**

***Index Terms*—ITQ; Bag of Hash Bits; Mobile Visual Search**

I. Introduction

In recent years, smart phones and tablets have evolved into powerful electronic products in daily life. Those mobile devices usually equip with high-resolution camera, color displays and high-performance CPU, connected to broadband wireless networks, which provide a perfect platform for mobile visual search (MVS) [1]. To this end, many of MVS applications follow the client-server architecture [2]. In a typical scenario, the mobile client takes a picture and sends the query image or usually its features in a compressed format to the server via a limited bandwidth wireless networks. In the server end, the received features are represented via Bag-of-Word methods [3], and are matched with the images with common words, which have been indexed by their words offline [4]. Then the server sends the result with high similar score back to the mobile client. The typical mobile visual search system flow could be summarized as shown in Fig. 1. Since the increasing performance of the mobile devices and high computation ability of the server, user experience greatly depends on the waiting time consumed by the transmission on the wireless network. Thus, a main concern in most existing mobile visual search is how to reduce the transmission latency which generally depends on the amount of the data transmitted.

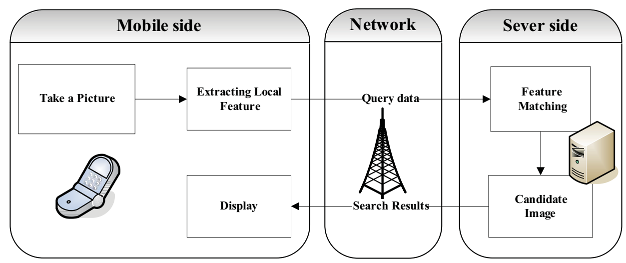

Fig. 1. Mobile visual search

As images are usually represented by a set of image descriptors in a MVS system, one straightforward way to reduce transmission latency is compressing the image descriptors [5]. Local feature like SIFT [6], PCA-SIFT [7] or SURF [8] is a popular choice, but those features are somehow large for transmitting on the network. The most popular way nowadays is to compress the local features by some coding method, such as CHoG [9], Hash Bits [10] and so on. The data transmitted on the network is very compact and much smaller than the previous methods. The other popular way to reduce transmission latency is exploiting progressive transmission scheme in mobile visual search [11]. Local descriptors such as SURF [12] and CHoG are firstly extracted on the mobile client side, and then are sorted by their importance [9] [13]. Those descriptors with more importance have higher priority. Sometimes the descriptors are sorted in a group manner to increase the final recognition rate [14]. The server begins matching when the several descriptors arrive, and terminates data transmission until the query image is matched correctly. Since the progressive transmission schemes mostly obtain the correct match in the former transmission, these methods can significantly reduce the amount of data transmission [15]. However, as far as we know, there is no work about sorting hash bits in MVS system.

In this paper, we choose a set of Hash Bits (HB) to represent each image and build a Bag of Hash Bits (BoHB) in the server side. More specifically, SIFT features are firstly extracted from each image and then encoded into a set of M bits (M is tens

978-1-4799-5079-9/14 $31.00 © 2014 IEEE
DOI 10.1109/DASC.2014.53

in the experiment) by ITQ hash method. The obtained bits are similar as the words used in the BOW method. Thus, all the database images can be indexed by those bits. The inquiry image only matches with database images with common bits in the BoHB scheme.

Meanwhile, an observation is that the IDF weight of these bits in BoHB scheme can be encoded into a sparse vector V. Each weight is indexed by its decimal value of the bit. The mobile client can download V and retain it with low memory. Upon obtaining inquiry bits of a captured image, the IDF weights of these bits are read from V and provide a transmission order of these bits. Generally speaking, bits with higher IDF weight will be more discriminative, which should have higher priority in the transmission queue.

The main contribution of our work are summarized as follows:

- We present a novel mobile visual search scheme based on Bag of Hash Bits (BoHB) [16][17], which is different from the classic Bag of Words. Compared to SIFT and SURF, hash bits is a kind of lighter descriptors, which can significantly reduce the amount of data transmission from the mobile to the server.
- To further reduce transmission latency, we propose a novel progressive transmission method based on the IDF weight of each hash bit. The IDF weights are represented by sparse vector, being retained in the mobile client with low memory.
- Extensive experiments have been done on the public Stanford MVS database, demonstrating that the proposed progressive transmission BOHB strategy achieves higher recognition rate compared to other strategies, when delivering the same amount of data.

The rest of this paper is organized as follows. We give an overview of the proposed search system in Section II. Section III details our progressive transmission based on Bag of Hash Bits and TF-IDF used in MVS. Section IV exhibits and analyzes our experiments and Section V concludes this paper.

II. System Framework

Our proposed progressive transmission method based on BoHB and TF-IDF weights follows the traditional client-server architecture. Fig. 2 shows the overall workflow of our system. As we will see, the progressive transmission can reduce transmission latency, raise the transmission rate and achieve low memory and computation.

In the server-side, three steps are accomplished offline. Firstly, the SIFT features are extracted from all the reference images in the database. Here, S is used to represent the set of these SIFT features. Secondly, iterative quantization (ITQ) method [18] is exploited to hash S into a Bits set B. It has been proven that ITQ method handles large data set more accurately. To find the proper projection matrix P and rotational transformation R, all the SIFT features in S are chosen as the training set. Thirdly, encoding the feature set S into the bits set B with the obtained P and R, and indexing

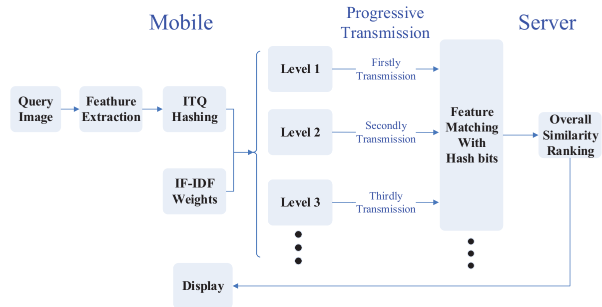

Fig. 2. The overview of our system framework

all the images by the hash bits in B. Then the IDF weight of each Hash Bit is computed, which measures its frequency of occurrence in the images. The IDF weights are encoded into a sparse vector V, with each is indexed by its decimal value of the corresponding bit. A well known fact in BoW research is that words with lower IDF weights will have better ability to distinguish between categories.

In the client-side, the first task is downloading the parameters P, R and V from the server. Although this task is a little time consuming, it only needs to be carried out offline, which means the process would not produce any latency during retrieval. When the mobile user captures an image and starts a mobile visual search task, SIFT features are extracted from the inquiry image and then are encoded into a set of hash bits with the downloaded P and R. After that, hash bits are transmitted to the server in a queue sorted by their IDF weights which are stored in V. Those hash bits of lower TF-IDF weights have higher priority, and will be preferentially transmitted.

The server begins the matching process when several hash bits arrive and terminates data transmission if the inquiry image is matched correctly. Matching test will only performed on the reference images sharing the arrived hash bits. Usually, there is no need to transmit all the hash bits to get the right feature matching. We will show this in the experiment section. That is to say, our transmission mechanism can greatly reduce the amount of transmission data, and thus cause low transmission latency.

III. Methodology

A. Iterative Quantization (ITQ)

Upon obtaining the SIFT features S of all the reference images, an important task is how to hash it into the bit set B. The ideal B should preserve the pair-wise distance of the original SIFT features. Aiming at this, many hash methods are proposed in the literature, such as PCA-Hash, Spectral Hash (SH) [19], Iterative Quantization (ITQ) [18] and so on. Compared to PCA and SH, ITQ has a smaller error when encoding each direction with the same number of bits, which finds a optimal orthogonal transformation to reduce quantization error by utilizing the orthogonal Procrustes problem [20] and eigenvector discretization. It can be used both with unsupervised data embedded such as PCA, and supervised data embedded such as canonical correlation analysis (CCA). In this paper, we hash the obtained SIFT features S into Bit set B following the ITQ method. Specifically speaking, we first perform principal component analysis (PCA) to reduce the dimensionality of the SIFT features. After that, a projection matrix P is obtained, with columns are the eigenvectors of the first M largest eigenvalues. A natural encoding method is to project the SIFT data S by matrix P, and then encode each direction into bit by predefined threshold. However, the variance of the SIFT features in each principal direction is different, when it is measured by the corresponding eigenvalue, encoded each direction with the same number of bits, it is bound to produce poor performance. Thus, following the ITQ method, we try to find the optimal rotating matrix R which can preserve the locality structure of the projected data, so as to minimize the quantization loss Q which can be calculated by Eq. 1.

$$Q(B, R) = ||B - SPR||^2 \tag{1}$$

In practice, a random rotation R is chosen, because it has been proven to work better in our experiment. Then we refine R in an iterative quantization method [18]. Although it is somehow time consuming, it is performed offline. After obtaining a proper rotational matrix R, we hash the SIFT features S by the following Eq. 2.

$$B = \mathrm{sgn}(SP) \tag{2}$$

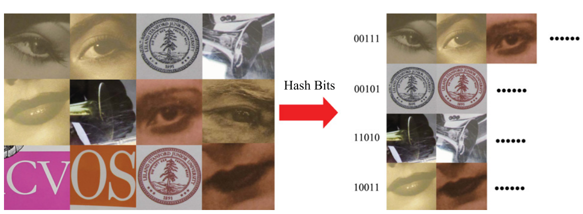

Fig. 3. Bag of Hash Bits

B. Bag of Hash Bits

In this section, we discuss the presented novel mobile visual search scheme based on Bag of Hash Bits (BoHB), motivated by the classic Bag of Words (BoW) scheme. As all the images are represented by a Bag of Hash Bits, a natural idea is that images can be indexed by hash bits as depicted in Fig. 3. Thus, instead of quantizing the features into visual words as BoW did, we consider the bits as words and index images directly. When the inquiry image represented by a set of hash bits is received, it only matches with reference images sharing common hash bits, following the classic BoW method. Compared to BoW, the proposed BoHB method has several unique advantages. 1) The hash bits is a kind of lighter descriptor, which can obviously decrease the amount of data transmission from mobile to serve and thus makes low transmission latency. 2) For matching local features, although accuracy is lost to some extent when the number of the bins is small, the matching process is much more quickly. In the case of hash bits, it is no need to compare with all the bits to find the same one. 3) The IDF weight of each hash bits can be encoded into a sparse vector. The vector is easily to retain and address, which can be stored in the mobile side and provide a transmission order of the hash bits.

IV. Experiment and performance analysis

In this section, we evaluate the performance of the proposed system via image retrieval experiments on Stanford data set. The code is implemented in MATLAB 2012a with some parts written in C with a MEX interface. The SIFT features are extracted and then are hashed into bits based on VLFeat toolbox and ITQ toolbox, respectively. In the serve side, we built the Bag of Hash Bits model, and indexed all the images by the hash bits, following the framework of Caltech Large Scale Image Search toolbox [21], which provides an implementation of classic BOW model.

The Stanford data set is public a database for MVS system, which consists of 8 categories, including book covers, business cards, cd covers, dvd covers, landmarks, museum paintings, print and video frames. All the eight categories except landmarks contain two kinds of images: 400 query images and 100 reference images. Query images are taken on the reference images by different kinds of mobile phones. Since the landmarks category is different from other categories, we only do experiments on the 7 categories. For the experiments, reference images are regarded as standard, and each query image is retrieved to find its corresponding reference image. To make it fair, all the images in the database are converted to gray-scale format with a fixed size (width 480 pixels), keeping the original image aspect ratio.

Now, we describe the parameters used in our system. The SIFT are extracted with all the parameters in the default value providing by VLFeat toolbox. In the ITQ method, all the SIFT extracted from reference images in each categories are used as training features. The projections are iterated 50 times to train a proper projection matrix. Upon obtaining the projection matrix, each training feature is projected and hashed into M bits. $M = 30$ is chosen as the default value in the experiments.

A. Parameter evaluation and analysis

One important parameter of our progressive BoHB schemes is M, the length of each bit. Generally speaking, bit with larger M will be more discriminative, since it is more informative and generate larger vocabulary. However, it will mean greater amount data to be transmitted leading to higher transmission latency. To find a proper M of Stanford data set, we perform the experiments as follows.

For each categories, we regard the query images as images captured by mobile terminal, and find the best match in the reference images which are maintained in the server side. The ratio of the right matches is exploited to evaluate the recognition accuracy. The recognition experiments are performed following a classic client-server model. That is to say, all the

TABLE I
VOCABULARY SIZE AND RECOGNITION ACCURACY ($M = 20$)

Saliency level	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
book covers	50.50%	65.10%	75.74%	76.73%	79.21%	80.45%	82.18%	81.19%	79.46%	79.21%
business cards	22.25%	32.00%	37.00%	44.00%	48.25%	50.50%	46.75%	41.50%	39.50%	38.50%
cd covers	34.75%	47.75%	56.00%	64.00%	67.50%	70.00%	69.25%	67.75%	65.00%	61.50%
dvd covers	42.25%	58.00%	66.00%	73.00%	74.50%	76.50%	75.75%	76.25%	72.75%	70.00%
museum paintings	32.69%	45.88%	53.02%	61.54%	65.11%	66.48%	69.78%	67.03%	63.46%	59.07%
print	9.75%	15.50%	21.00%	24.25%	25.25%	27.25%	24.00%	23.25%	25.50%	26.00%
video frames	44.75%	65.50%	74.50%	81.00%	85.75%	83.75%	81.75%	78.25%	74.50%	68.00%
Average	33.85%	47.10%	54.75%	60.65%	63.65%	64.99%	64.21%	62.17%	60.02%	57.47%

TABLE II
VOCABULARY SIZE AND RECOGNITION ACCURACY ($M = 30$)

Saliency level	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
book covers	56.93%	74.50%	82.67%	85.15%	85.64%	86.14%	86.14%	88.37%	87.38%	87.13%
business cards	31.25%	43.75%	50.75%	56.50%	63.50%	66.75%	66.00%	62.50%	59.00%	55.50%
cd covers	42.25%	56.25%	65.25%	71.25%	75.25%	77.50%	79.75%	78.00%	77.75%	72.25%
dvd covers	50.25%	62.25%	70.00%	77.00%	81.75%	82.50%	82.50%	84.25%	85.25%	82.00%
museum paintings	32.42%	43.68%	54.40%	62.91%	67.03%	71.43%	72.80%	76.65%	73.90%	65.93%
print	13.50%	24.00%	29.50%	32.25%	33.25%	35.00%	35.75%	34.25%	33.00%	33.50%
video frames	47.75%	71.00%	77.75%	83.50%	85.50%	86.50%	87.50%	87.25%	85.25%	77.75%
Average	39.19%	53.63%	61.47%	66.94%	70.28%	72.26%	72.92%	73.04%	71.65%	67.72%

TABLE III
VOCABULARY SIZE AND RECOGNITION ACCURACY ($M = 40$)

Saliency level	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
book covers	61.39%	77.23%	82.92%	85.15%	86.39%	87.38%	87.38%	87.62%	87.62%	88.86%
business cards	36.50%	41.50%	49.50%	53.00%	57.75%	59.00%	63.00%	64.25%	63.50%	61.25%
cd covers	45.00%	61.50%	68.50%	73.50%	77.00%	78.75%	78.50%	79.50%	79.50%	75.75%
dvd covers	49.75%	65.50%	72.75%	75.75%	80.25%	83.25%	86.00%	85.25%	86.00%	83.25%
museum paintings	35.99%	49.45%	56.04%	62.91%	69.78%	73.90%	75.00%	79.95%	80.22%	71.98%
print	14.75%	27.25%	30.50%	34.00%	35.00%	36.00%	35.75%	36.50%	36.00%	36.00%
video frames	54.50%	69.75%	79.75%	85.00%	87.25%	88.00%	88.00%	88.75%	86.25%	83.00%
Average	42.55%	56.03%	62.85%	67.04%	70.49%	72.33%	73.38%	74.55%	74.16%	71.44%

hash bits are transmitted to the server side, and match with the images with common hash bits. For comparison, we choose M as 20, 30, 40, respectively.

Table I, Table II and Table III show the recognition accuracies in each categories under different M. Fig. 4 illustrates the corresponding vocabulary size. Higher M will generate larger vocabulary size and achieve higher recognition accuracy in most cases. However, higher M means greater amount data to be transmitted and will cause higher transmission latency. For Stanford data set, since the recognition accuracy of each category under $M = 40$ only a little more than $M = 30$ as Fig. 5 shows, we choose $M = 30$ as the default case to reduce the amount of data transmitted.

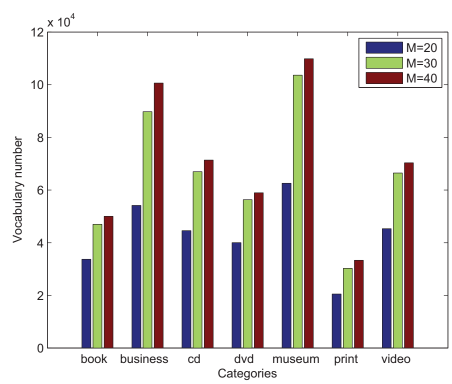

Fig. 4. Vocabulary numbers in each categories under different M

B. Comparisons with the baseline progressive transmission scheme

We compare our proposed progressive BOHB schemes with the baseline scheme, in which the hash bits are transmitted in a random order. To make comparison, in each scheme, the mentioned recognition accuracy is evaluated by using 20%, 40%, 60%, 80% and 100% of the transmission data, respectively. The transmission latency is measured by the amount of data needed to be transmitted. That is to say, the scheme achieves higher recognition accuracy with less amount of data, will cause lower transmission latency.

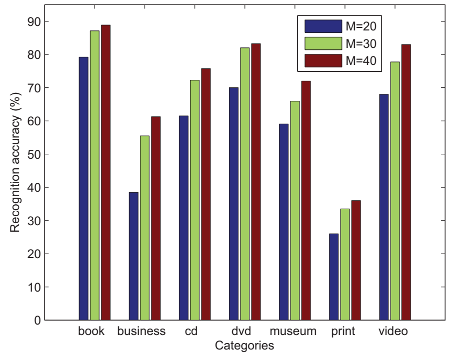

Fig. 5. Recognition accuracy in each categories under different M

Table IV lists the recognition accuracy results on each category for our progressive BOHB scheme under the mentioned transmission cases, respectively. Table V lists the corresponding results of the baseline progressive transmission scheme. From these two tables, we can see that our scheme significantly outperforms the baseline scheme, which means the hash bits with small IDF weight will do favor to the recognition. Fig. 6 illustrates the recognition accuracy of these two schemes in each transmission case.

TABLE IV
OUR PROGRESSIVE BOHB SCHEME

Saliency level	20%	40%	60%	80%	100%
book covers	74.50%	85.15%	86.14%	88.37%	87.13%
business cards	43.75%	56.50%	66.75%	62.50%	55.50%
cd covers	56.25%	71.25%	77.50%	78.00%	72.25%
dvd covers	62.25%	77.00%	82.50%	84.25%	82.00%
museum paintings	43.68%	62.91%	71.43%	76.65%	65.93%
print	24.00%	32.25%	35.00%	34.25%	33.50%
video frames	71.00%	83.50%	86.50%	87.25%	77.75%
Average	53.63%	66.94%	72.26%	73.04%	67.72%

TABLE V
BASELINE PROGRESSIVE TRANSMISSION SCHEME

Saliency level	20%	40%	60%	80%	100%
book covers	71.04%	82.43%	84.90%	87.87%	87.13%
business cards	28.25%	39.50%	47.50%	52.25%	55.50%
cd covers	42.75%	61.25%	69.25%	72.50%	72.25%
dvd covers	53.50%	68.25%	75.25%	80.75%	82.00%
museum paintings	28.02%	36.54%	49.45%	59.07%	65.93%
print	18.25%	31.50%	34.25%	33.75%	33.50%
video frames	57.50%	77.00%	82.25%	80.25%	77.75%
Average	42.76%	56.64%	63.26%	66.63%	67.72%

C. Comparisons with the ordinary transmission

Here, ordinary transmission means all of the features are transmitted to sever, and then matched. Compared to ordinary transmission, the progressive transmission will have lower latency but causes lower recognition rate too for the transmission

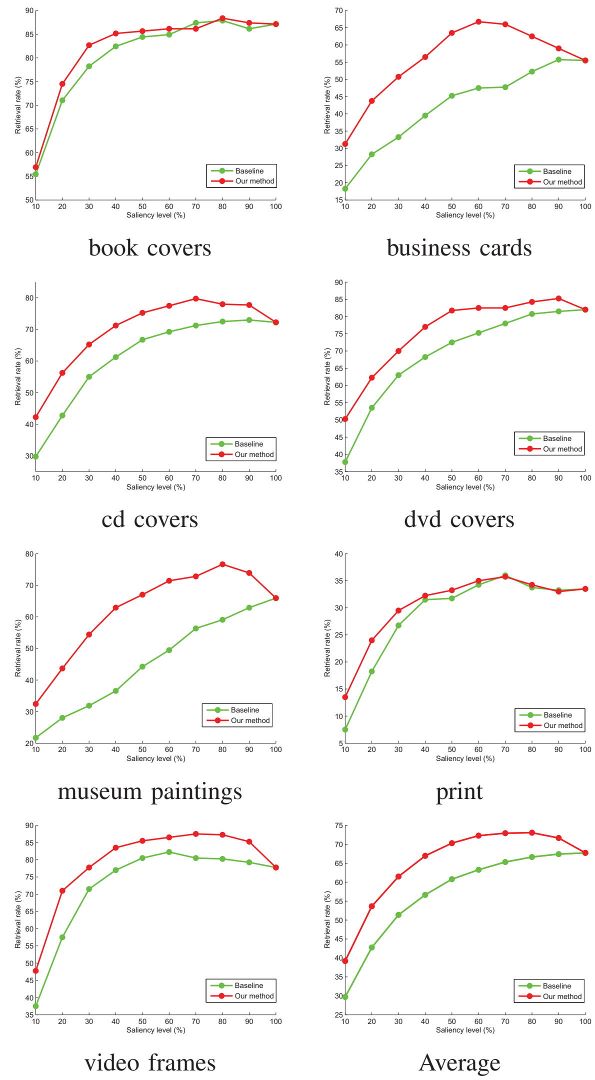

Fig. 6. The contrast between our method and baseline method

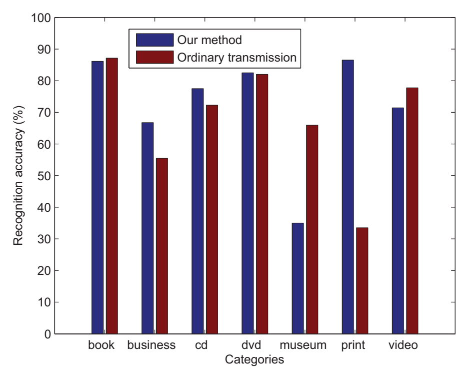

Fig. 7. Our progressive transmission at the 60% saliency level comparisons with the ordinary transmission

lost. As high discriminative features are transmitted in high priority, our progressive transmission can not only reduce the transmission latency, but also improve the final recognition accuracy. We will show this in the following.

Fig. 7 shows our progressive transmission is a little more than ordinary transmission at the 60% saliency level. It means that our progressive transmission can achieve the recognition rate equivalent to ordinary transmission only with half of the data transmitted. That is to say, our method can greatly reduce transmission latency.

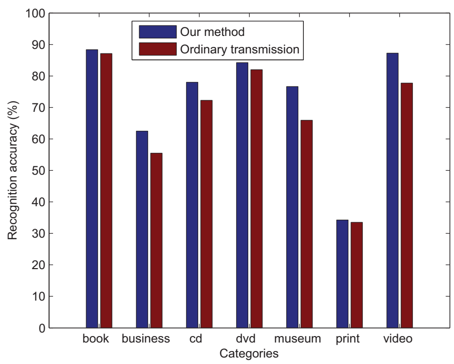

Fig. 8. Our progressive transmission at the 80% saliency level comparisons with the ordinary transmission

Fig. 8 shows our progressive transmission is much better than ordinary transmission at the 80% saliency level. It means that progressive transmission can guarantee higher recognition accuracy than ordinary transmission at the same time of reducing the amount of transmission.

V. Conclusion

Recently smart phones and tablets provide a excellent platform for mobile visual search. However, due to the limited bandwidth of wireless network, how to reduce the transmission latency is the main concern in the related research. We propose to represent image by hash bits and transmit the hash bits in a progressive manner to address the above issue. The transmission queues are ordered by the IDF weights of these hash bits, which are stored as a sparse vector in the mobile side. Extensive experiments have been done on the public Stanford MVS database, demonstrating that the proposed progressive transmission strategy achieves higher recognition rate to baseline progressive transmission strategy. The future work will be concentrated on increasing the discriminative ability of the hash bits.

Acknowledgment

This work is supported by the National Science Foundation of China under grant nos.of 61173165,61370199,61300187, and the Fundamental Research Funds for the Central Universities under grant nos.of 2013QN044.

References

[1] B. Girod, V. Chandrasekhar, D. M. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham, "Mobile visual search," *Signal Processing Magazine, IEEE*, vol. 28, no. 4, pp. 61–76, 2011.

[2] Y. Du, Z. Li, W. Qu, S. Miao, and S. Wang, "Mvss: Mobile visual search based on saliency," in *High Performance Computing and Communications, 2013. HPCC 2013. 15th IEEE International Conference on*, IEEE, 2013.

[3] T. Li, T. Mei, I.-S. Kweon, and X.-S. Hua, "Contextual bag-of-words for visual categorization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 4, pp. 381–392, 2011.

[4] R. Ji, L.-Y. Duan, J. Chen, L. Xie, H. Yao, and W. Gao, "Learning to distribute vocabulary indexing for scalable visual search," *Multimedia, IEEE Transactions on*, vol. 15, no. 1, pp. 153–166, 2013.

[5] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod, "Residual enhanced visual vector as a compact signature for mobile visual search," *Signal Processing*, vol. 93, no. 8, pp. 2316–2327, 2013.

[6] T.-W. R. Lo and J. P. Siebert, "Local feature extraction and matching on range images: 2.5 d sift," *Computer Vision and Image Understanding*, vol. 113, no. 12, pp. 1235–1250, 2009.

[7] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–506, IEEE, 2004.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, pp. 404–417, Springer, 2006.

[9] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "Chog: Compressed histogram of gradients a low bit-rate feature descriptor," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2504–2511, IEEE, 2009.

[10] X. Liu, J. He, and B. Lang, "Multiple feature kernel hashing for large-scale visual search," *Pattern Recognition*, vol. 47, no. 2, pp. 748–757, 2014.

[11] J. Xia, K. Gao, D. Zhang, and Z. Mao, "Geometric context-preserving progressive transmission in mobile visual search," in *Proceedings of the 20th ACM international conference on Multimedia*, pp. 953–956, ACM, 2012.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[13] W. He, T. Yamashita, H. Lu, and S. Lao, "Surf tracking," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1586–1592, IEEE, 2009.

[14] X. Sun, M. Chen, and A. Hauptmann, "Action recognition via local descriptors and holistic features," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 58–65, IEEE, 2009.

[15] K. Srinivasan, J. Dauwels, and M. R. Reddy, "Multichannel eeg compression: Wavelet-based image and volumetric coding approach," *Biomedical and Health Informatics, IEEE Journal of*, vol. 17, no. 1, pp. 113–120, 2013.

[16] J. He, T.-H. Lin, J. Feng, and S.-F. Chang, "Mobile product search with bag of hash bits," in *Proceedings of the 19th ACM international conference on Multimedia*, pp. 839–840, ACM, 2011.

[17] J. He, J. Feng, X. Liu, T. Cheng, T.-H. Lin, H. Chung, and S.-F. Chang, "Mobile product search with bag of hash bits and boundary reranking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3005–3012, IEEE, 2012.

[18] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 817–824, IEEE, 2011.

[19] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," NIPS, 2008.

[20] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*, vol. 3. Oxford University Press Oxford, 2004.

[21] M. Aly, M. Munich, and P. Perona, "Indexing in large scale image collections: Scaling properties and benchmark," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pp. 418–425, IEEE, 2011.