Feature Fusion based Hashing for Large Scale Image Copy Detection

Lingyu Yan[1], Hefei Ling[2], Dengpan Ye[3], Chunzhi Wang[1], Zhiwei Ye[1], Hongwei Chen[1]

[1] *School of Computer, Hubei University of Technology*
Wuhan,43000,China
E-mail: yanranyaya@hust.edu.cn, chunzhiwang@vip.163.com, weizhiye121@163.com, chw2001@sina.com
[2] *School of Computer Science and Technology, Huazhong University of Science and Technology*
1037, Luoyu Road
Wuhan,43000,China
E-mail: lheifei@hust.edu.cn
[3] *School of Computer, Wuhan University,*
16,Luojiashan Road,
Wuhan,43000,China E-mail: yedp@whu.edu.cn

Received 20 October 2013

Accepted 11 March 2015

Abstract

Currently, researches on content based image copy detection mainly focus on robust feature extraction.However, most of existing approaches use only a single feature to represent an image for copy detection, which is often insufficient to characterize the image content. Besides, with the exponential growth of online images, it's urgent to explore a way of tackling the problem of large scale. In this paper, we propose a feature fusion based hashing method which effectively utilize the correlation between two feature models and efficiently accomplish large scale image copy detection. To accurately map images into the Hamming space, our hashing method not only preserves the local structure of individual feature but also globally consider the local structures for all the features to learn a group of hash functions. The experiment results show that the proposed method outperforms the state-of-the-art techniques in both accuracy and efficiency.

Keywords: Content Based Copy Detection, Feature Fusion, Kernel Canonical Correlation Analysis, Neighborhood Structure Preserving Hashing

1. Introduction

Since software for editing digital content is easily accessible, digital images may subject to different kinds of attack like scale change, cropping, resolution or contrast change, etc., yielding different image copies preserving the main semantic content of the original image, causing infringements to the copyright of original works. As a result, copyright protection has become an urgent problem to be solved. To satisfy the necessity of protecting the copyright of some digital images, content based image copy detection is proposed, which performs the detection by processing content of raw multimedia, ignoring partial data and avoiding embedding digital watermarks into the original works and hence is an effective way to trace unmarked content after being distributed.

Traditional copy detection(CBCD) always extract only one single feature to represent an image, which is always insufficient to characterize the content. Simultaneously utilizing multiple features which are complementation to each other is intuitively helpful for image representation. For example, local feature is less robust to the changes in cropping and global feature is sensitive to changes in contrast, brightness, scale, rotation, and so on. It is difficult, if not impossible, to find a single visual feature which is robust to all types of variations. Therefore, it is particularly important to exploiting multiple features for CBCD.

Apart from that, since current image copy detection always involves large scale of images, even globe feature results in large size of feature data, let alone local feature which contains much more information and requires larger payload size on both computation cost and storage cost. In such scenario, constructing efficient indexing structures to facilitate fast similarity search over large scale datasets is also important for online copy detection. Researchers has introduced hashing into the field of content based image copy detection, but existing algorithms still have several drawbacks like sematic loss, coding efficiency.

In this paper, we propose a feature fusion based hashing method that is able to effectively maximize the correlation between two feature models and efficiently accomplish large scale image copy detection. The main idea are as follows:(1)adopts Kernel Canonical Correlation Analysis(KCCA) to maximize the correlation between two kind of features; (2)adopts machine learning to construct an objective function and minimize it to generate neighborhood structure preserving hashing(NSPH) and apply corresponding hash function to the two features. The remainder of this paper is organize as follows. In section 2, we review the state-of-the-art of feature fusion and hashing. In section 3, We present the feature fusion algorithm and then construct an image copy detection framework. And then, in section 4, extensive experiments are conducted to demonstrate the effectiveness and efficiency of the proposed algorithm. Finally, we draw a conclusion in section 5.

2. Related Work

2.1. Feature Fusion

Employing multiple features for multimedia applications aims to explore the way of fusing multiple futures derived from the same object. Typically, we can classify the current fusion methods into two categories according to when the features are fused.

One category combines the different search result-lists derived from different features, which are called late fusion. In[1], D. F. Hsu etc. proposed Combinatorial Fusion Analysis (CFA) to fuse the rank lists of several score derived from calculating the statistic information of the original feature representation. Similarly, D. R. Kisku etc.[2] computed the matching scores of both local area feature (nose and mouth) and globe feature (whole face) and used the Dempster-Shafer Theory to determine whether the query face image belongs to a person or not. In [3], a voting procedure is used to get final classification result based on the results of nearest neighbor classifiers applied to each module. The above late fusion strategies did not consider the individual character of different feature models, especially the inner correlation between them, making them not adequate for realistic application.

The other category integrates the features and their correlations at stage of generating features for search, which is called early fusion. X. Wan[4] constructed Feature Interaction Graph (FIG), in which every feature is considered as a node and nodes are linked with edges which represent the Intra-type correlation between features of the same model and the Inter-type correlation between features of different models. S. Gundimada[5] tried to linearly construct a cost function in the form of weighted sum, and minimized it to get the final feature representation which is relatively similar with the inner structures of both two feature models.

2.2. Hashing

Hashing is a technique that generates compact hash code from the feature data to represent the main contents, and therefore can be used for image copy detection.Early hashing methods like LSH[15], Multi-probe LSH[16], Kernelized LSH[17] have the bottleneck of large space cost. To make the hash code more efficient and accurate, a few data-aware hashing methods have been proposed by introducing the machine learning tricks into the field of hashing to enhance the effectiveness of hash codes.

Tang etc.[6] develop a global method using non-negative matrix factorization (NMF), which first convert image into a fixed-sized pixel array and then generate secondary image by rearranging pixels and applying NMF to produce a feature-bearing hash code, after that, the fingerprint is coarsely quantized into binary string and scrambled to generate the image fingerprint.

Similarity Sensitive Coding (SSC)[10] and Forgiving Hashing (FgH)[11] adopts boosting approach, they first train AdaBoost classifiers with similar pairs of items as positive examples (and also non-similar pairs of items as negative examples in SCC), and then take the output of all (decision stump) weak learners on a given document as its binary code.

In addition, spectral hashing (SpH)[12] and self-taught hashing (STH)[13] are proposed and are considered as the state-of-the-art works[14]. However, these two methods both suffer overfitting problem since the operations of generating hash codes for traing data and hash function for test data are independently handled, which will lead to poor generalization ability.

On the whole, though hashing based method is considered as the effective approach to approximate nearest neighbor search applications like content based copy detection, semantic loss issue remains the big challenge in the area of hashing. Motivated by this, we propose a new machine learning based hashing algorithm and optimize it with future fusion to achieve superior semantic preservation.

3. Feature Fusion Hashing for Image Copy Detection

3.1. Framework of Image Copy Detection

We propose an image copy detection framework illustrated in Figure 1. It is composed of two stages called *offline stage* and *online stage*, the key point of which is to accomplish both *correlation maximization* and *similarity preservation*.

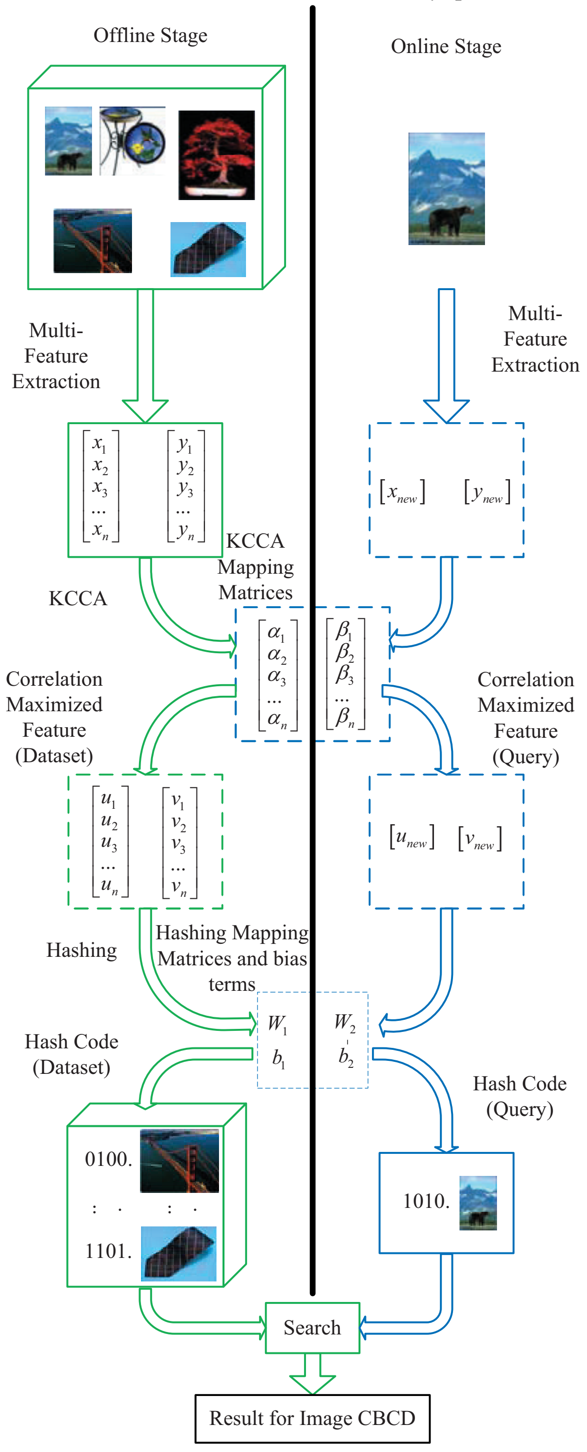
Offline Stage
Online Stage
Multi-Feature Extraction
Multi-Feature Extraction
$x_1, x_2, x_3, \dots, x_n$
$y_1, y_2, y_3, \dots, y_n$
$[x_{new}]$ $[y_{new}]$
KCCA
KCCA Mapping Matrices
$\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$
$\beta_1, \beta_2, \beta_3, \dots, \beta_n$
Correlation Maximized Feature (Dataset)
Correlation Maximized Feature (Query)
$u_1, u_2, u_3, \dots, u_n$
$v_1, v_2, v_3, \dots, v_n$
$[u_{new}]$ $[v_{new}]$
Hashing
Hashing Mapping Matrices and bias terms
W_1 b_1
W_2 b_2
Hash Code (Dataset)
Hash Code (Query)
0100.
1101.
1010.
Search
Result for Image CBCD

Fig. 1. proposed Framework for Image Copy Detection.

In offline stage, we first extract two kind of features derived from the same images, which are respectively denoted by $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{n \times p_1}$ and $Y = [y_1, y_2, \dots, y_n] \in \mathbb{R}^{n \times p_2}$ (n is the number of training images, p_1 and p_2 are the dimensionality of the two features respectively). Then the features are projected into two subspaces using Kernel Canonical Correlation Analysis, obtaining the correlation-maximized feature representations matrices, which are respectively denoted by $U = [u_1, u_2, \dots, u_n] \in \mathbb{R}^{n \times p}$ and $V = [v_1, v_2, \dots, v_n] \in \mathbb{R}^{n \times p}$ (p is the dimensionality of the correlation-maximized features), and get two corresponding mapping matrices $\alpha \in \mathbb{R}^{n \times p}$ and $\beta \in \mathbb{R}^{n \times p}$. After that, the third step maps the features into binary codes denoted by $H = [h_1, h_2, \dots, h_n]^T \in \{1, -1\}^{n \times p}$ (c is the length of the hash codes), which meets the requirements of similarity preservation.

In online stage, for a query image, features x^q and y^q are extracted through the same methods in the online stage. Then the features are projected into the same subspaces using the two transform matrices α and β learned in the offline stage and get correlation-maximized features u_{new} and v_{new}. Finally, u_{new} and v_{new} are mapped into binary code through the same method of the offline stage.

3.2. Correlation Maximization

Canonical correlation Analysis (CCA) [7] is a method of maximizing the correlating linear relationships between two multidimensional variables, which can be applied in information based tasks and in our natural selection. It can be seen as using two views of the same object to extract two similar representations, making it feasible to fuse two features and preserve as much semantic information as possible, the ideal case of which is the two result representations are totally the same.

In an attempt to increase the flexibility of the feature selection, kernelization of CCA (KCCA) is proposed to overcome the limitation of CCA, which maps the hypotheses to a higher-dimensional feature space and brings in improvement to the results.

Given two matrices X and Y, we define a new coordinate for Y by choosing a mapping direction w_x and projecting X onto that direction. Similarly, we choose a mapping direction w_y for Y. Then we get new matrices U and V:

$$\begin{aligned} U &= {w_x}^T X \\ V &= {w_y}^T Y \end{aligned} \quad . \tag{1}$$

To maximize the correlation between X and Y, we need to choose proper w_x and w_y to maximize the Pearson's Correlation Coefficient:

$$\rho = \frac{\mathrm{cov}(U,V)}{\sigma_U \sigma_V} = \frac{{w_x}^T C_X X w_y}{\sqrt{{w_x}^T C_X X w_x {w_y}^T C_Y Y w_y}} \quad . \tag{2}$$

Using the definition of the covariance matrix, we can rewrite the covariance matrix C using the data matrices (of vectors) X and Y, which have the sample vector as rows, we obtain:

$$\begin{aligned} C_{XX} &= X^T X \\ C_{XY} &= X^T Y \\ C_{YY} &= Y^T Y \end{aligned} \quad . \tag{3}$$

And the mapping directions can be rewritten as the projection of the data onto two direction α and β:

$$\begin{aligned} w_x &= X^T \alpha \\ w_y &= Y^T \beta \end{aligned} \quad . \tag{4}$$

Substituting Eq.(3) and(4) into Eq.(2) we get:

$$\rho = \frac{\alpha^T X X^T Y Y^T \beta}{\sqrt{\alpha^T X X^T X X^T \alpha \beta^T Y Y^T \beta}} \quad . \tag{5}$$

Let $K_X = XX^T$ and $K_Y = YY^T$ be the kernel matrices corresponding to the two features. We get the transformation of Eq.(5):

$$\rho = \frac{\alpha^T K_X K_Y \beta}{\sqrt{\alpha^T {K_X}^2 \alpha \beta^T {K_Y}^T \beta}} \quad . \tag{6}$$

The above maximization problem can be transformed to be a optimization problem:

$$\begin{aligned} &\arg max\, \alpha^T K_X K_Y \beta \\ s.t. \quad & \alpha^T {K_X}^2 \alpha = 1 \\ & \beta^T {K_Y}^T \beta = 1 \end{aligned} \quad . \tag{7}$$

By solving the corresponding Lagrangian problem of Eq.(7), the optimization problem can be solved by calculating a generalized eigenproblem to get w_x,w_y and corresponding U, V.

3.3. *Neighborhood Structure Preserving Hashing(NSPH)*

To achieve fast similarity search, hashing is always used for indexing. A good semantic hashing should ensure the similarity preservation to guarantee its effectiveness, which means that semantically similar feature data should be mapped into similar codes in hamming space and vice verse. Here we construct an objective function and optimize it to achieve the similarity preservation.

Given a feature matrix $Z = [z_1, z_2, \ldots, z_n] \in \mathbb{R}^{n\times p}$ (n is the number of training images, p is the dimensionality the feature), we want to get hash codes denoted by $H = [h_1, h_2, \ldots, h_c]^T \in \{1,-1\}^{n\times c}$ (c is the length of the hash codes). To exploit the local similarity structure of a feature, we construct a $n \times n$ similarity matrix S as

$$S_{ij} = \begin{cases} 1 & ,\text{if} \quad z_i \in N_k(z_j) or z_j \in N_k(z_i) \\ 0 & ,\text{otherwise} \end{cases} \quad . \tag{8}$$

where $N_k(z)$ represents the set of k-nearest-neighbor of feature vector z.

The Hamming distance between two binary codes h_i and h_j (corresponding to features z_i and z_j) is given by the number of bits that are different between them. To ensure the similarity preservation, we seek to minimize the weighted average Hamming distance which represents the semantic loss

$$\sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij} \left\| h_i - h_j \right\|^2 \quad . \tag{9}$$

For the convenience of implementation for large scale dataset, we need to find a way to transform new feature data to binary hash code. Here we adopt the linear transformation as the hash function for simplicity of implementation and optimization, which is defined as:

$$h_c(x_i) = {w_c}^T Z_i + b_i \quad . \tag{10}$$

Then we can learn hash mapping matrix W and bias term b by minimizing the empirical error of the hash function. Because the constraint $H = [h_1, h_2, \ldots, h_c]^T \in \{1,-1\}^{n\times c}$ makes the objective function to be an NP-hard problem, we relax that constraints to make the problem computationally solvable. After that, we construct a joint frame work which aims to minimize semantic loss and empirical error simultaneously. The final objective function turns out to be:

$$\begin{aligned} &\arg min \sum_{i,j=1}^{n} S_{ij} \left\| h_i - h_j \right\|^2 \\ &\qquad + \phi(\left\| Z^T W + \mathbf{1}b - H \right\|_F^2 + \gamma \|W\|_F^2) \\ &s.t. YY^T = I \end{aligned} \quad . \tag{11}$$

Where $\|W\|_F^2$ is a regularization function, the constraint $YY^T = I$ is imposed to avoid trivial solution. ϕ and γ are parameters.What should be noted is minimizing the empirical error term $\left\| Z^T W + \mathbf{1}b - H \right\|_F^2$ not only give a way to transform new feature data to binary hash code, but also mean globe information on training data, helping prevent the problem of overfitting and make the learning more robust.

To get the an optimal solution of the objective function, we need to first minimize the objective function with respect to W and b. Set the derivative of Eq. (11) with respect to b to zero, we have

$$\begin{aligned} &\mathbf{1}^T (Z^T w + \mathbf{1}b - H) = 0 \\ &\Rightarrow b = \frac{1}{n}(\mathbf{1}^T H - \mathbf{1}^T Z^T W) \end{aligned} \quad . \tag{12}$$

Set the derivative of Eq. (11) with respect to W to zero, we have

$$Z(Z^T W + \mathbf{1}b - H) + \gamma W = 0 \quad . \tag{13}$$

Using the obtained result of Eq. (12), we transform Eq. (13) to be as follows.

$$\begin{aligned} &ZZ^T W + Z\mathbf{1}(\frac{1}{n}(\mathbf{1}^T H - \mathbf{1}^T Z^T W)) - XH + \gamma W = 0 \\ &\Rightarrow W = (ZAZ^T + \gamma I)^{-1} ZAH \end{aligned} \quad . \tag{14}$$

where $A = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, it is obvious that $A = A^T$ and

$A = AA^T$. So we get

$$\begin{aligned} & Z^TW + \mathbf{1}b - H \\ & = Z^TW + \mathbf{1}\frac{1}{n}(\mathbf{1}^TH - \mathbf{1}^TZ^TW) - H \\ & = AZ^TW - AH \\ & = AZ^T(ZAZ^T + \gamma I^{-1})ZAH - AH \\ & = (AZ^T(ZAZ^T + \gamma I^{-1})ZA - A)H \end{aligned} \quad . \qquad (15)$$

Let $B = (ZAZ^T + \gamma I)^{-1}$, it is obvious that $B = B^T$, so $W = BZAH$ and $Z^TW + \mathbf{1}b - H = (AZ^TBZA - A)H$,then we transform the last part of the objective function as follows by applying the theory of trace optimization[8]:

$$\begin{aligned} & \|Z^TW + \mathbf{1}b - H\|_F^2 + \gamma\|W\|_F^2 \\ & = trH^T(AZ^TBZA - A)^T(AZ^TBZA - A)H + \\ & \quad \gamma tr(W^TW) \\ & = trH^T(AZ^TBZAAZ^TBZA - 2AZ^TBZA + A)H + \\ & \quad \gamma tr(H^TAZ^TBBZAH) \\ & = trH^T(\gamma AZ^TBBZA + AZ^TBZAZ^TBZA \\ & \qquad - 2AZ^TBZA + A)H \\ & = trH^T(AZ^TB(\gamma I + ZAZ^T)BZA \\ & \qquad - 2AZ^TBZA + A)H \\ & = trH^T(AZ^TBB^{-1}BZA - 2AZ^TBZA + A)H \\ & = trH^T(AZ^TBZA - 2AZ^TBZA + A)H \\ & = trH^T(A - AZ^TBZA)H \\ & = trH^TCH \end{aligned} \quad . \qquad (16)$$

Where $C = A - AZ^TBZA$. Apart from that, the first part of the objective function can be transformed to be following formula:

$$\sum_{i,j=1}^{n} S_{ij}\left\|h_i - h_j\right\|^2 = tr(H^T(N - S)H) \quad . \qquad (17)$$

Where $N_{ii} = \sum_{j=1}^{n} S_{ij})$ and other elements are zero. Combine the Eq. (16) and Eq. (17), the objective function turn out to be:

$$\arg\min_{YY^T=I} tr(H^T(N - S + \phi C)H) \quad . \qquad (18)$$

Then H can be obtained by the c eigenvectors of $(L + \phi B)$ corresponding to the c smallest eigenvalues. Once we got the hash code H of the correlation-maximized two features U and V in continuous domain, we concatenate and binarize them to get the feature fusion hash code $\widehat{H}$.

4. Experiments

4.1. Experiment Setting

Our training image dataset consists of 46,735 images, where 1000 images come from the COREL 1000 image database[*], 15,128 Images are from the CEA CLIC database[†] 30,607 images come from the Caltech 256 database[‡]. All the three image databases are commonly used in the research of image processing, image retrieval and computer vision, and the last image database is used to test our method on real world images.

We randomly select 1000 images from the image dataset to be query images, and the rest are treated as non-copies. Using Stirmark, every query image is modified to generate 100 copies in the way of[9]. After that, the 100,000 copies are inserted into the image dataset for query image to search their copy.Fig. 2 shows some sample images and corresponding image copies. The list of alterations of all attacks is as follows:

1. SPA (15): signal processing attacks, including colorizing (3), changing saturation(3), changing intensity(3), median filtering(3),Gaussian filtering, sharpening, and frequency mode Laplacian removal(FMLR).

2. JPEG (12): JPEG compression with quality factors ranging from 90% to 10%.

[*]http://wang.ist.psu.edu/docs/related.shtml
[†]http://www.irit.fr/RFIEC/CLIC/CLIC kernel/CLIC kernel.zip
[‡]http://vision.caltech.edu/Image Datasets/Caltech256/

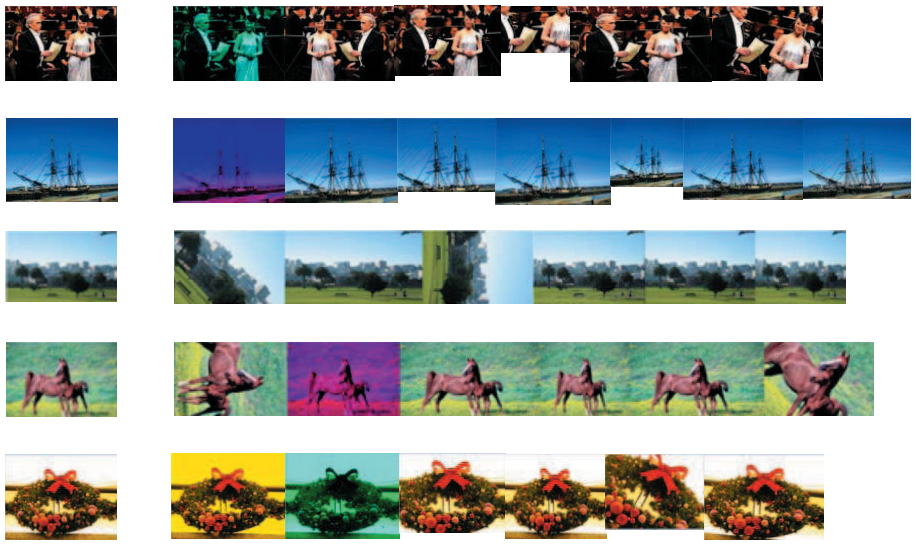

Figure 2: Sample images and corresponding sample image copies.

3. GLGT (3): general linear geometric transform.
4. CAR (7): change of the aspect ratio.
5. LR (5): line removal.
6. RC (16): rotation + cropping.
7. Scaling (5): scaling with factors ranging from 0.5 to 2.0.
8. Cropping (9): cropping the image by a percentage ranging from 1% to 75%.
9. Shearing (6): apply affine warp on both x- and y axes.
10. RRS (16): rotation + rescaling.
11. RB (1): random bending.
12. Flipping (1).
13. Seam carving (4).

4.2. Performance of the Proposed Method

Before the experiment, we need to define some equation for evaluation of the performance. Let R_P be the number of true copies correctly assigned to the positive class, F_P is the number of false copies incorrectly assigned to the positive class, and R_N the number of true copies incorrectly rejected by the positive class. The precision and recall are defined as:

$$precision = \frac{R_P}{R_P + F_P}, recall = \frac{R_P}{R_P + R_N} \quad . \quad (19)$$

Then F1-measure, also called F1-score, is the harmonic mean of precision and recall:

$$F_1 - measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad . \quad (20)$$

To evaluate the performance of the proposed approach, several experiments are done. HSV and Local Binary Pattern(LBP) of all the images are extracted, which are globe feature and local feature respectively and can be considered as complementation to each other.To exhibit the performance of the proposed approach, we conduct a series of experiments among our method, STH, and SpH to test the accuracy, efficiency and effectiveness of our method. The parameters of our proposed method are set as follows: (1) k is set to be 5 when construct the similarity matrix; (2) ϕ, γ are tuned from $10^{-6}, 10^{-3}, 10^{0}, 10^{+3}, 10^{+6}$, and finally set as $10^{0}, 10^{-5}$ respectively.

We first evaluate the nearest neighbor search performance on GIST1M, which contains 1 million 384-d GIST features. Following the search strategy of Hamming ranking commonly adopted in many hashing methods, we evaluate the recall at the first N Hamming neighbors, the performance of which is shown in fig. 3. Our proposed method outperforms all other methods.

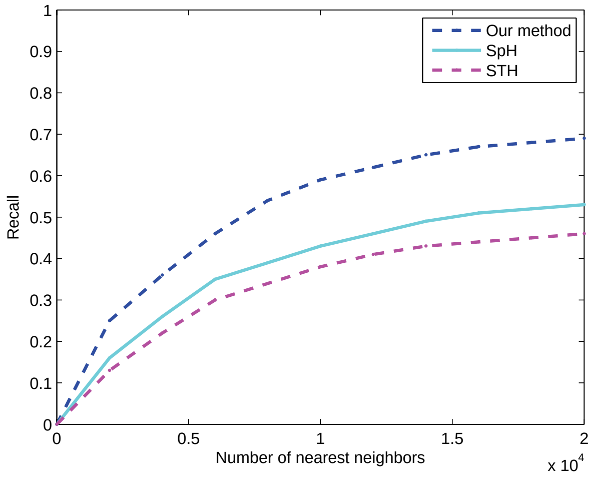

Fig. 3. Nearest neibhor search performance of hashing methods on GIST1M.

After that, we explore the influences of code length on the retrieval performance by tuning the code length c from 4 to 80. Fig.4 shows the comparison results among NSPH(our method), STH, and SpH on HSV feature. Before reaching the peak, the F1-measure of proposed method is higher than SpH and STH. After reaching the peak respectively, the F1-measure of our method and STH decrease gradually while SpH hold its superiority. However, since long binary codes demand more memory and computation, the performance of long code length is not that important, so we set the code length to be 20 for further comparison experiment, and hence the code length of feature fusion hashing is 40.

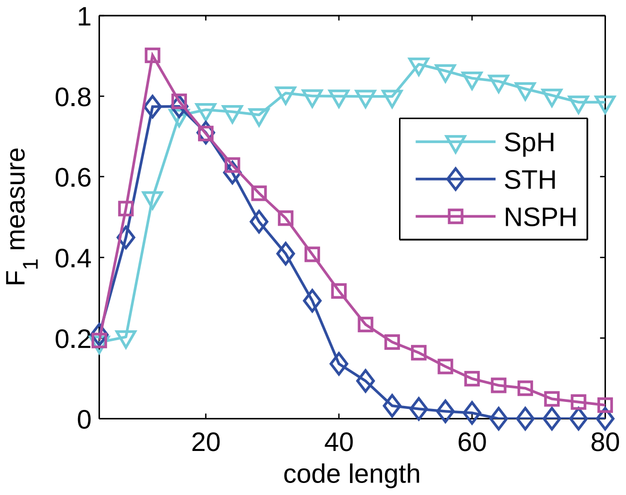

Fig. 4. F1-measure of the comparative algorithm under various code length.

Fig.5 shows the accuracy comparison results among NSPH(our method), STH, and SpH on HSV feature in the form of Precision-Recall curve. It's obvious that NSPH outperforms SpH and STH, which mainly derives from that NSPH effectively decrease semantic loss in the process of hashing.

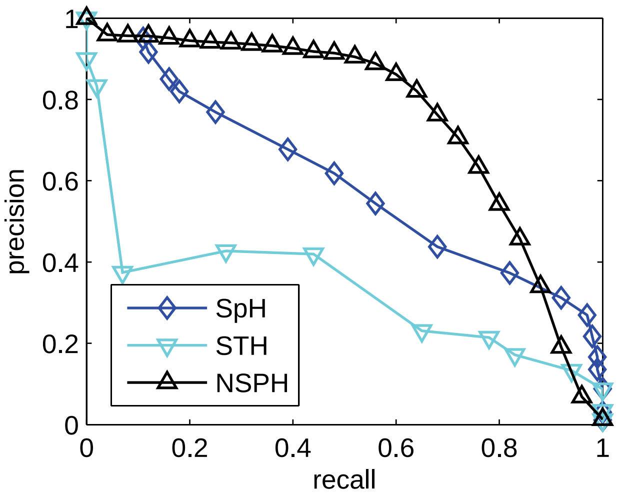

Fig. 5. Precision-Recall curve of the comparative hashing algorithm.

Fig.6 shows the comparison results among feature fusion+NSPH(FF+NSPH), HSV+NSPH, LBP+NSPH, in terms of precision-recall curves for query images. It's obvious that feature fusion hashing outperforms single feature hashing, which mainly comes from that correlation maximization effectively fuse features of two complementary modality.

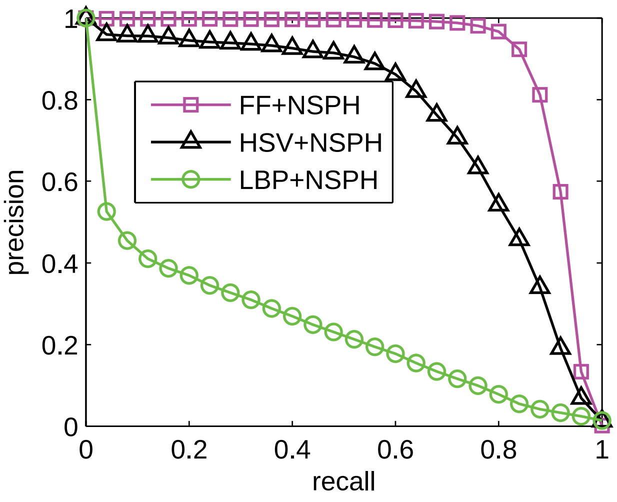

Fig. 6. Precision-Recall curve of feature fusion and single feature.

Table 1. The time cost(ms) of the two method.

Method	Time
Original HSV	2130.45
FF+NSPH	36.73
HSV+NSPH	16.19
LBP+NSPH	15.82
HSV+STH	19.35
HSV+SpH	13.98

Table 1 shows average time consumption for a query to search its copies in our image dataset(including both the time of getting hash code and search). Algorithms are run by Matlab on a Windows server with a 2.40 GHz quad-core Intel Xeon CPU and 12 GB of memory. Obviously, our method(FF+NSPH) takes longer time than other hashing methods, but is much more faster than Original HSV data. The phenomenon manly derives from the difference between original feature data and hash code. For hash code, calculating pairwise similarity only need to operate a XOR on the two hash codes and count the number of non-zero bits. However, for original feature data, we need to calculate the Euclidean distance between the two feature data, which is obviously much more time-consuming than the former. Besides, the time cost of those hashing methods are almost the same.

5. CONCLUSIONS AND FUTUREWORK

In this paper we propose a feature fusion based hashing for image copy detection. We perform extensive tests with dataset and our method shows good performance in terms of not only precision and recall but also efficiency. Future work will include apply another image features to see whether our method shows better performance, and further investigating the way of correlation maximization and neighborhood structure preservation that may improve CBCD in aspect of precision and recall, scalability, and so on.

Acknowledgement

Thanks for the funding supported by the National Natural Science Foundation of China (No. 61170135, No. 61202287, No. 61440024, No. 61272453, and the General Program for Natural Science Foundation of Hubei Province in China (No. 2013CFB020).

1. D. F. Hsu and Y. S Chung and B.S. Kristal, "Combinatorial Fusion Analysis: Methods and Practice of Combining Multiple Scoring Systems," *Ideal Group Inc.*, Oakland, USA(2006).
2. D. R. Kisku and M. Tistarelli and J. K. Sing, "Face Recognition by Fusion of Local and Global Matching Scores using DS Theory: An Evaluation with Uni-classifier and Multi-classifier Paradigm," *Proceeding of Computer Vision and Pattern Recognition Workshops*, 60–65 (2009).
3. B. Cui and A. K. H. Tung and C. Zhang and Z. Zhao, "Multiple Feature Fusion for Social Media Applications," *Proceeding of International Conference on Management of Data*, 526–531 (2010).
4. X. Wan and J. Xiao, "Graph-Based Multi-Modality Learning for Topic-Focused Multi-Document Summarization," *Proceeding of International Joint Conference on Artifical Intelligence*, 1586–1591 (2009).

5. S. Gundimada and V. K Asari and N. Gudur, "Face recognition in multi-sensor images based on a novel modular feature selection technique," *Information Fusion*, **11**, 124–132 (2010).
6. Z. Tang and S.Wang and X. Zhang and W.Wei and S. Su, "Robust image hashing for tamper detection using non-negative matrix factorization," *J. Ubiquitous Convergence Technol.*, **2**, 18–26 (2008).
7. D. R. Hardoon and S. Szedmak and J. S.Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Department of Computer Science, Egham, Surrey TW200EX, England*, (2003).
8. Fuhao Zou and Cong Liu and Hefei Ling and HuiFeng and Lingyu Yan and Dan Li, "Least square regularized spectral hashing for similarity search," *Signal Processing*, **93**, 2265–2273 (2013).
9. H. F. Ling and H. Chen and Q. Ma, "Efficient Image Copy Detection Using Multiscale Fingerprints," *IEEE. Multimedia*, **19**, 60–69 (2012).
10. G.Shakhnarovich and P.Viola and T.Darrell, "Fast pose estimation with parameter-sensitive hashing," *Proceedings of the IEEE International Conference on Computer Vision*, 750–757 (2003).
11. D.M.Blei and A.Y.Ng and M.I.Jordan, "Latent Dirichlet allocation," *Machine Learning Research*, **3**, 993-1022 (2003).
12. Y.Weiss and A.Torralba and R.Fergus, "Spectral hashing," *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*, 1753–1760 (2008).
13. D.Zhang and J.Wang and D.Cai and J.Lu, "Self-taught hashing for fast similarity search," *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 18–25 (2010).
14. P.Li and M.Wang and J.Cheng and C.Xu, "Spectral hashing with semantically consistent graph for image indexing," *IEEE Trans.Multimedia*, **15**, 141–152 (2013).
15. M.Datar and P.Indyk and N.Immorlica and V.S.Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," *Proceedings of the Annual Symposium on Computational Geometry*, 253–262 (2004).
16. Q.Lv and W.Josephson and Z.Wang and M.Charikar and K.Li, "Multi-probe LSH: efficient indexing for high-dimensional similarity search," *Proceedings of the 33rd International Conference on Very large databases*, 950–961 (2007).
17. B.Kulis and K.Grauman, "Kernelized locality-sensitive hashing for scalable image search," *Proceedings of the IEEE International Conference on Computer Vision*, 2130–2137 (2009).