*Journal of Information Hiding and Multimedia Signal Processing* ©2022 ISSN 2073-4212
**Ubiquitous International** Volume **13**, Number **2**, June **2022**

# Robust Image Hashing Based on Multi-view Feature Representation and Tensor Decomposition

Qiuchen Shang, Ling Du*, Xiaochao Wang

School of Computer Science and Technology
Tianjin Key Laboratory Autonomous Intelligence Technology and Systems
Tiangong University, No. 399, BinShuiXi Road, XiQing District, Tianjin, China
shangstacey@163.com; duling@tiangong.edu.cn; wangxiaochao18@163.com

Xiujie Zhao

Chinasoft Excellency IT Co.,LTD
Floor 2, Building 4, Binhai Service Outsourcing Park, TEDA, Tianjin
zhao_xj@chinasofti.com

Received January 2022; revised March 2022

Abstract. *This paper proposes a Robust Image Hashing Based on Multi-view Feature Representation and Tensor Decomposition. The algorithm effectively utilizes complementarity and independence between multiple views, and combines the characteristics of multiple views to construct a binary robust hash code. The input image is first prepossessed by bilinear interpolation and mean filtering. Next, the multi-view features, i.e. structure features, edge features and color features are extracted. The correlation between the multiple views is explored to construct a high-order tensor. Then, the popular algorithm Tucker Decomposition (TD) is applied to decompose the tensor into core tensor and factor matrices. Finally, a robust hash code is constructed by using factor matrices. The experimental results of this paper show that the algorithm can resist a large number of content preservation operations and has strong discrimination. The ROC curve obviously demonstrates the performance advantage of the proposed algorithm compared with other algorithms.*
**Keywords:** Image hashing, Multi-view, Tucker decomposition, Binary sequence

1. **Introduction.** Image hashing is a compact representation of digital images based on content. It has been widely used for image retrieval[1], image watermarking[2] and other fields. Image hashing technology is a type of single-item mapping from a multimedia data set to a perceptual summary set. Perceptual hashing technology implies two main characteristics to be satisfied, namely robustness and discrimination. (1)Robustness[3]: this means that the distance between the image hash under the content preserving operation and the hashing of the original image should be small enough. (2)Discrimination[4]: this means that the Hamming distance between different images should be very large. In general, robustness and discrimination are contradictory. Improving robustness results in loss of discrimination and vice versa. Therefore, robust hashing algorithms should achieve a good balance between robustness and discrimination. In recent years, researchers have devoted themselves to studying image hashing and proposed many hashing algorithms. Ghouti et al.[5] used Quaternion Fourier Transform(QFT) to process non-overlapping image blocks while calculating the average frequency of each image block. Finally, through the binarization operation, the average frequency energy is compared with the threshold

to generate a binary hash. Hosny et al.[6] proposed an image hashing method based on Gaussian-Hermite invariant moments. By calculating the Gaussian-Hermite moment of the grayscale image, and then extracting the invariants of different orders moment to form the perceptual hash. Evans et al.[7] proposed a framework for constructing image perceptual hash based on feature points. First, the end-stopped image local feature point detector was used to extract the stable feature points in the image, and the probability quantization method was used to binarize the obtained local features to generate the final perceptual hash. Davarzani et al.[8] extracted CSLBP features from non-overlapping block in the original gray image. It proposed a combination of Singular Value Decomposition(SVD) and centrosymmetric local binary patterns robust image hashing algorithm. Naoe et al.[9] proposed an image hashing key generation model based on neural network. This method took the observation signal output by the trained neural network as the image hash value, and realized the efficient image hash content recognition. Cui et al.[10] found that histogram-based hashing algorithms have satisfactory performance for various geometric deformations. They extracted the gray levels from the filtered image and used the average pixel gray value to find groups of pixels to create a histogram. They finally constructed a hash sequence by comparing different groups of pixels. Yu et al.[11] used the DCT-based visual model to extract human visual sensitive features, and then they calculated the GGD parameters through a modified generalized Gaussian distribution. Finally, they compared different parameter values to generate a hash sequence.

Tensor decomposition can well solve dimensionality reduction, sparse data filling and implicit relationship exploring. This paper studies image hashing based on multi-view feature representation and tensor decomposition. The proposed method explore the correlations between various views, so as to reduce the information redundancy between different views to enhance the robustness of the algorithm. Each view is merged into a high-order tensor, and the matrix norm is used as the data reconstruction error constraint to ensure that the robust characteristics are consistent between different views. The objective function is optimized using the augmented iterative direction minimization strategy. Then, factor matrices are generated by Tucker Decomposition. Finally, since the factor matrix can fully express the internal structure of the original tensor while ensuring robustness, we construct hash sequences with factor matrices and encoded them into a compact public binary code space. Experiments are carried out with open image databases. The authors compare the performance with some recently popular algorithms. The proposed algorithm achieves reasonable performance in terms of hash length and execution time.

2. **Method.** This section mainly introduces the implementation steps of the algorithm in this paper, as shown in Figure 1. The image hashing algorithm proposed first preprocesses the input image with bilinear interpolation followed by multi-view feature extraction. Then, the authors construct the obtained feature matrix into a third-order tensor. Finally, the Tucker Decomposition (TD) is employed to decompose the tensor, so as to construct the hash sequence.

2.1. **Preprocessing.** After the digital image is attacked by the content preserving operation, the content of the image remains unchanged. Therefore, the human eye judges that the original image and the attack image are visually similar images. In order to further minimize the impact of content preserving operations on the image, the image needs to be preprocessed.

2.1.1. *Bilinear interpolation.* Image scaling is one of the most vulnerable content preserving operations. If the input image is processed by bilinear interpolation, the image hash

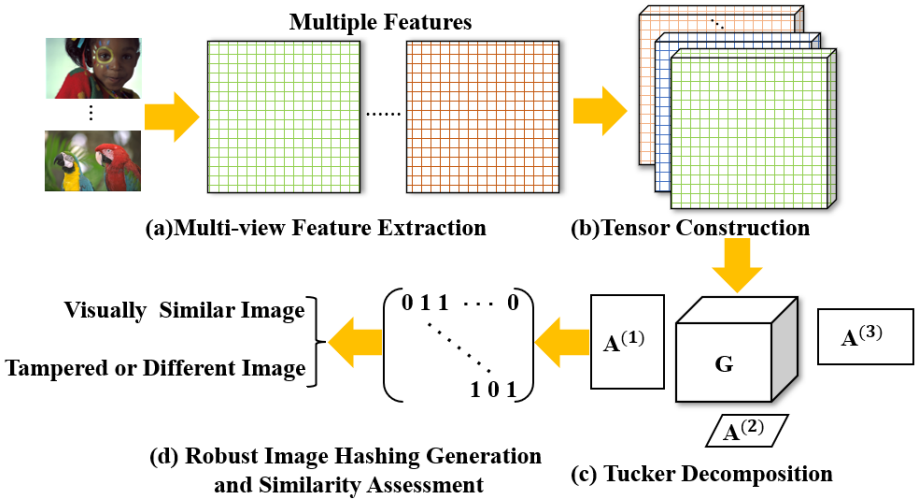


FIGURE 1. Framework of proposed image hashing method.

algorithm will be robust to image scaling. In addition, bilinear interpolation can damage the high-frequency components and blur the image contour. This algorithm uses bilinear interpolation to adjust the original input image $F(x, y)$ to $F^{'}(x, y)$ with size $S \times S$.

2.1.2. *Image feature extraction.* Different perspective information describes the same object. Correspondingly, the analysis and presentation results from different perspectives should be consistent or relevant. Each perspective contains to a certain extent unique information that other views do not possess, that is, complementarity or independence. Therefore, it is necessary to investigate how to effectively use multi-view features. For the attacks of different content preserving operations, multi-view feature selection is very important to enhance the robustness of hash coding for different content preserving operations. The authors extract three types of features from images[13], considering the structural features ($\mathbf{F}^{(1)}$), edge features($\mathbf{F}^{(2)}$) and color features ($\mathbf{F}^{(3)}$). After that, the authors divide $2 \times 2$ non-overlapping blocks and calculate the average of each block to realize mean filtering. The authors randomly select N blocks with size $b \times b$ to generate $b^2 \times N$ normalized matrix $\mathbf{F}$.

$$\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N]\,. \tag{1}$$

Here, the authors use $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N] \in \mathbb{R}^{b^2 \times N}$ to represent the data matrix composed of all data vectors. Each column vector in the matrix corresponds to a data point. The example of preprocessing result is shown in Figure 2.

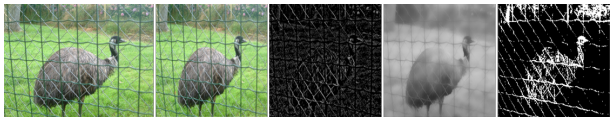

FIGURE 2. An example of preprocessing results.

2.2. **Tensor Construction.** In order to solve the problem of a large dimension, we use the principal structure $\mathbf{Z}$ to construct tensor. For the multi-view data $\mathbf{F}$ obtained in the previous section. The relationship between the original matrix and its principal structure representation is $\mathbf{F} = \mathbf{FZ} + \mathbf{R}$. $\mathbf{R}$ represents the reconstruction error matrix. Here $\mathbf{R}$ is

defined as follows:

$$\|\mathbf{R}\|_{2,1} = \sum_{i=1}^{d} \sqrt{\sum_{j=1}^{n} |\mathbf{R}_{i,j}|^2}. \tag{2}$$

As a direct and essential expression of multidimensional structural data, tensor can fully preserve the essential structural characteristics of multidimensional data. Therefore, the generation of perceptual hash value is a summary representation derived from tensor.

In the proposed method, the low-rank tensor constraint is introduced into image hashing. Through low-rank tensor constraints, the complementarity and consistency between different views can be well explored. At the same time, tensor as a generalization of the matrix, the authors use the tensor kernel norm to constrain the Low-rank tensor $\mathbf{Z}$. The nuclear norm of tensor $\mathbf{Z}$ is:

$$\|\mathbf{Z}\|_* = \sum_{t=1}^{T} \xi_t \left\|\mathbf{Z}_{(t)}\right\|_*, \tag{3}$$

where $\mathbf{Z} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_t}$ is a t-order tensor. $\|\mathbf{Z}\|_*$ is tensor nuclear norm. Its function is to constrain low rank matrices. If rank $(\mathbf{Z})$ is non-convex, the tensor nuclear norm is the linear combination of nuclear norm of $\mathbf{Z}_{(t)}$. $\xi_t$ is constant, which satisfies $\sum_{t=1}^{T} \xi_t = 1$. After introducing the low-rank tensor constraints, the objective function is as follows:

$$\begin{aligned} \min_{\mathbf{Z}^{(v)},\mathbf{R}^{(v)}} & \|\mathbf{Z}\|_* + \lambda \|\mathbf{R}\|_{2,1} \\ s.t.\ \mathbf{F}^{(v)} =& \mathbf{F}^{(v)}\mathbf{Z}^{(v)} + \mathbf{R}^{(v)}, v = 1, 2, ..., V, \\ \mathbf{R} =& [\mathbf{R}^{(1)}; \mathbf{R}^{(2)}; ...; \mathbf{R}^{(v)}], \end{aligned} \tag{4}$$

$\mathbf{Z}^{(v)} \in \mathbb{R}^{N \times N}$ corresponds to the principle structure representation of each view in $\mathbf{F}^{(v)} \in \mathbb{R}^{b^2 \times N}$. Through the definition of tensor kernel norm in Formula (3), the objective function is optimized as follows:

$$\min_{\mathbf{Z}^{(v)},\mathbf{R}^{(v)}} \|\mathbf{R}\|_{2,1} = \sum_{t=1}^{T} \gamma_t \left\|\mathbf{Z}_{(t)}\right\|_*, \tag{5}$$

here $\gamma_t = \frac{\xi_t}{\lambda} > 0$. By setting the size of $\gamma_t$, the constraint strength of the low-rank tensor can be adjusted. Under the constraint of low-rank tensor, the objective function is convex, so it can only ensure local convergence and will be affected by local minimum. ALM-ADM algorithm[14] solves this problem by optimizing the objective function using an iterative direction minimization strategy. This method makes different sets of variables separable in objective function, so the authors can mine higher-order correlations between different views. The method[15] is based on convex optimization, which is faster and more efficient in restoring known structures. By introducing auxiliary variables, the objective function is transformed into a convex optimization problem, as follows:

$$\min_{\mathbf{Z}^{(v)},\mathbf{R}^{(v)},\mathbf{G}_t} \|\mathbf{R}\|_{2,1} + \sum_{t=1}^{T} \gamma_t \|\mathbf{G}_t\|_*, s.t.\ \mathbf{P}_t\mathbf{z} = \mathbf{g}_t, t = 1, 2, ..., T, \tag{6}$$

where $\mathbf{z}$ and $\mathbf{g_t}$ are obtained by vectorizing tensor $\mathbf{Z}$ and matrix $\mathbf{G}$ respectively, and $\mathbf{P}$ is the alignment matrix. $\mathbf{G}$ has low rank characteristics, so tensor $\mathbf{Z}$ also has low rank characteristics. The constraint of $\ell_{2,1}$ norm is applied to matrix $\mathbf{R}$, making the same column of matrix $\mathbf{R}^{(1)}, \mathbf{R}^{(2)}, ..., \mathbf{R}^{(v)}$ have the same sparse characteristics. It is solved accurately by minimizing the convex optimization of the combination of kernel norm and

kernel function. The convex optimization problem is solved by the ALM-ADM method. The following is the solution process:

$$\mathcal{L}_{\mu>0}(\mathbf{Z}^{(1)},...,\mathbf{Z}^{(V)}) = \|\mathbf{R}\|_{2,1}+\sum_{t=1}^{T}(\gamma_t \|\mathbf{G}_t\|_*+\mu\Phi(\alpha_t,\mathbf{P}_t\mathbf{z}-\mathbf{g}_t))+\sum_{v=1}^{V}\mu\Phi(\mathbf{Y}_v^T,\mathbf{F}^{(v)}-\mathbf{F}^{(v)}\mathbf{Z}^{(v)}-\mathbf{R}^{(v)}), \tag{7}$$

here, $\mu > 0$ is a penalty factor. $\alpha_t$ is a Lagrange multiplier. The authors use the iterative minimization strategy to update $\mathbf{Z}^{(v)}$ iteratively. $\Phi(\mathbf{X},\mathbf{C}) = \frac{1}{2}\|\mathbf{C}\|_{\mathbf{F}}^2 + \langle\mathbf{X},\mathbf{C}\rangle$. $\langle\mathbf{X},\mathbf{C}\rangle$. Here,$\langle\mathbf{X},\mathbf{C}\rangle$ represents the inner product of the matrix. $\mathbf{Z}$ is solved by using imprecise ALM algorithm and iterative direction minimization strategy, which is shown as follows:

$$\begin{aligned}\mathbf{Z}^{(v)^*} &= \arg\min_{\mathbf{Z}^{(v)}} \sum_{t=1}^{T}\mu\Phi(\alpha_t,\mathbf{P}_t\mathbf{z}-\mathbf{g}_t)+\mu\Phi(\mathbf{Y}_v^T,\mathbf{F}^{(v)}-\mathbf{F}^{(v)}\mathbf{Z}^{(v)}-\mathbf{R}^{(v)}) \\ &= \arg\min_{\mathbf{Z}^{(v)}} \sum_{t=1}^{T}\mu\Phi(\Omega^v(\alpha_t)^T,\Omega^k(\mathbf{P}_t\mathbf{z}-\mathbf{g}_t))+\mu\Phi(\mathbf{Y}_v^T,\mathbf{F}^{(v)}-\mathbf{F}^{(v)}\mathbf{Z}^{(v)}-\mathbf{R}^{(v)}),\end{aligned} \tag{8}$$

where $\mathbf{g_t}$ is obtained after vectorization of matrix $\mathbf{G_t}$. $\Omega^{(v)}(\cdot)$ defines selection and alignment operations. It will select the elements corresponding to the v-th view to form the corresponding matrix. By $\Psi(\cdot)$, the authors combine the principle structure representation corresponding to each view into high-order tensor $\mathbf{Z}$, which dimension is $N \times N \times V$.

2.3. **Tucker Decomposition.** In this paper, the authors decompose the third-order (v=3) tensor to get three factor matrices, so as to construct the perceptual hash sequence. For the third-order tensor $\mathbf{Z}$, the Tucker decomposition definition is as follows:

$$\mathbf{Z} \approx \mathbf{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} = [\![\mathbf{G};\mathbf{A},\mathbf{B},\mathbf{C}]\!], \tag{9}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are the factor matrices. $\mathbf{G}$ represents the core tensor. Lathauwer et al.[16]computed the Tucker Decomposition by higher-order SVD. The method is better known as HOSVD. The HOSVD is a great origin for the iterative ALS algorithm[17]. Kroonenberg et al.[18] proposed an algorithm to compute a Tucker decomposition for three-way arrays. The target function is as follows:

$$\min_{\mathbf{G},\mathbf{A}^{(1)},\mathbf{A}^{(2)},\mathbf{A}^{(3)}} \left\|\mathbf{Z}-[\![\mathbf{G};\mathbf{A}^{(1)},\mathbf{A}^{(2)},\mathbf{A}^{(3)}]\!]\right\|, \tag{10}$$

$\mathbf{G}$ represents the core tensor, $\mathbf{A}^{(n)}$ $(n = 1, 2, 3)$ is orthogonal factor matrices obtained by Tucker decomposition. The internal structure of the original tensor can be reflected by $\mathbf{A}^{(n)}$ $(n = 1, 2, 3)$, the authors can get the hash sequence through matrix $\mathbf{A}^{(n)}$.

2.4. **Robust Image Hashing.** For the factor matrix $\mathbf{P} = \left[\mathbf{A}^{(1)},\mathbf{A}^{(2)},\mathbf{A}^{(3)}\right]$, the authors calculate the mean and standard deviation($\delta_i$) of each eigenvector. Subsequently, the hash sequence $H(i)$ is obtained by binarization operation. $\delta_m$ is the mean of vector.

$$H(i) = \begin{cases} 0, & \delta_i < \delta_m \\ & ,i = 1,2\cdots,h. \\ 1, & otherwise \end{cases} \tag{11}$$

In the proposed image hashing methods, Hamming distance $D(H_1, H_2)$ is used to measure similarity between two hashes:

$$D = \sum_{i=1}^{h} |H_1(i) \oplus H_2(i)|, \tag{12}$$

where $H_1(i)$ and $H_2(i)$ are the i-th elements of $H_1$ and $H_2$. When the input image pairs are visually similar, the Hamming distances between them are smaller than the predefined threshold value. Otherwise, they are different image pairs.

## 3. Experiment.

### 3.1. Experimental Setting.

3.1.1. *Datasets.*
**KodakImageDataset**: The Kodak Image dataset contains 24 full color images, and they are available in two formats, respectively $768 \times 512$ and $512 \times 768$.
**CopydaysDataset**: The INRIA Copydays dataset is a set of images made up exclusively of our personal holiday photos. It contains 157 color images ranging in size from $1200 \times 1600$ to $3008 \times 2000$.
**HolidaysDataset**: This dataset contains 500 queries and 991 corresponding similar images with a total of 1,491 images. The operation methods include rotation, viewpoint and illumination changes, blurring, etc. Its size ranges from $1600 \times 1200$ to $3264 \times 2448$.
**RTDDataset**: It is a class of manual tamper datasets with high resolution. The color images are uncompressed $1920 \times 1080$ unit8 TIFF.

3.1.2. *Baselines.* The authors compare the proposed algorithm with some latest methods, such as image hashing algorithm based on Tucker Decomposition (TD)[19], Ring Partition and Invariant Vector Distance (RPIVD)[20], Quaternion Singular Value Decomposition (QSVD)[12], Quaternion Fourier Transform (QFT)[13] and Feature Point (FPHash) [7]. TD was used to compute hashes by orthogonal factor matrices. RPIVD combined ring partitions and invariant vector distances into the image hashes by calculating statistical information. Ghouti et al.[5] used the quaternion singular value decomposition method to represent the general matrix, and then calculated the hash sequence. Pun et al.[13] processed non-overlapping image blocks while calculating the average frequency of each image block to calculate hashes. FPHash used an end-stop wavelet transform to detect visually meaningful feature points to construct the image hash.

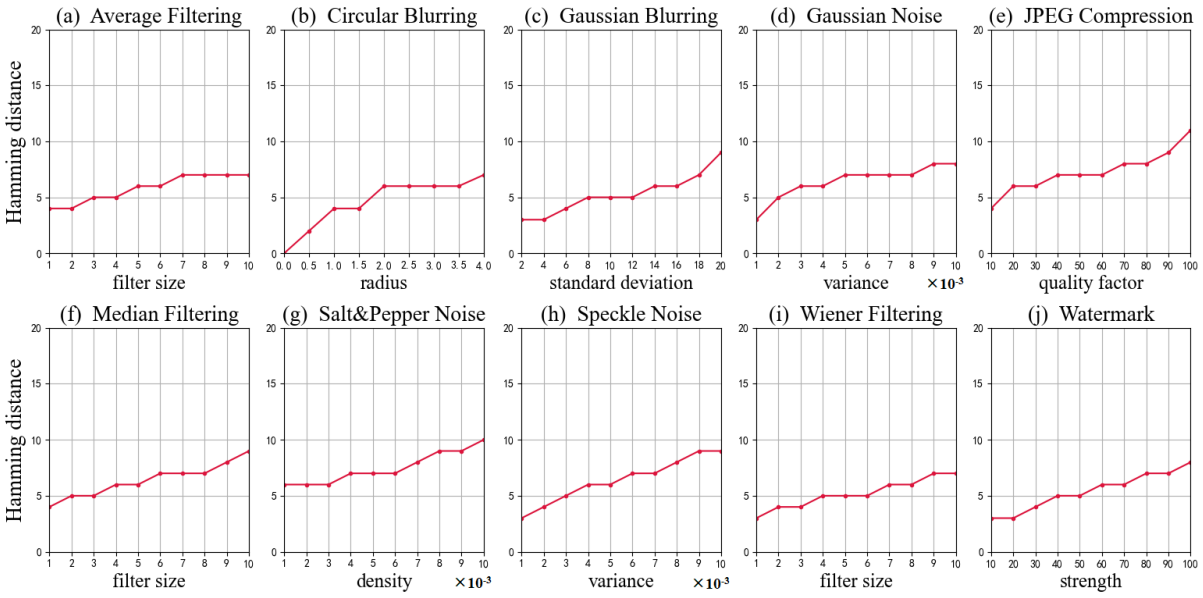


FIGURE 3. Average Hamming distance distribution under different content preserving operations.

TABLE 1. Hamming Distance statistics under different attacks.

| Operation | Max | Min | Mean | BER |
|---|---|---|---|---|
| Average Filtering | 29 | 0 | 8.51 | 0.033 |
| Circular Blurring | 30 | 0 | 7.30 | 0.028 |
| Gaussian Blurring | 22 | 0 | 8.72 | 0.034 |
| Gaussian Noise | 35 | 0 | 9.13 | 0.035 |
| JPEG Compression | 30 | 0 | 9.52 | 0.037 |
| Median Filtering | 30 | 0 | 8.54 | 0.033 |
| Salt&Pepper Noise | 30 | 0 | 9.84 | 0.028 |
| Speckle Noise | 29 | 0 | 9.18 | 0.035 |
| Wiener Filtering | 30 | 0 | 7.47 | 0.029 |
| Watermark | 30 | 0 | 7.37 | 0.028 |

3.2. **Perceptual Robustness.** In order to verify the robustness of the proposed hashing algorithm. The authors choose two open image databases: the Kodak Image database and the INRIA Copydays database with a total of 181 color images. The authors add 10 common content-preserving operations to each image for robust attacks, including Average Filtering, Circular Blurring, Gaussian Blurring, Gaussian Noise, JPEG Compression, Median Filtering, Salt&Pepper Noise, Speckle Noise, Wiener Filtering and Watermark. There are 10 parameters set for each operation, so the total number is $10 \times 10 = 100$. Obviously, the total number of similar image pairs is $181 \times 100 = 18{,}100$. The authors calculate the Hamming distance between similar images to evaluate the similarity of each pair of images. Figure 3 shows the average Hamming distances under different content-preserving operations. The x-axis and y-axis represent the parameter value of each content preserving operation and the average hash distance of each parameter, respectively. In general, the average Hamming distance under all the content-preserving operations is not greater than 15. In addition, in order to analyze the robust performance of the hashing algorithm more comprehensively. Under the default parameter settings, Table 1 show the statistics of hamming distances under various numerical operations in the INRIA Holidays dataset. The minimum value of all numerical operations is 0, the maximum value is 35, the mean value is lower than 10, and the Bit Error Ratio (BER) is no more than 0.037. This implies that the algorithm in this paper is resistant to most numerical operations.

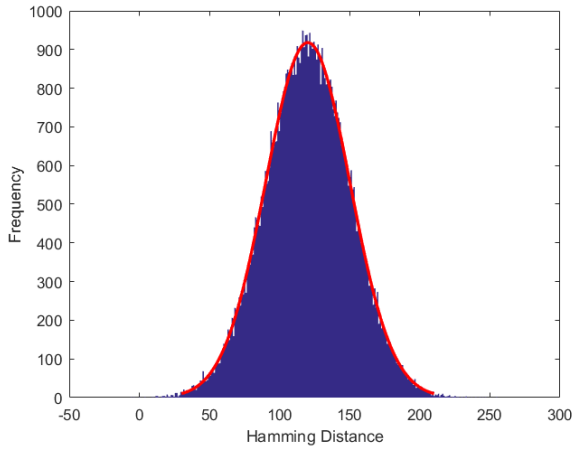


FIGURE 4. Distribution of hash distances between different images.

3.3. **Discriminative Capability.** To evaluate the ability of hash differentiation, the authors choose the RTD open image dataset. The RTD dataset contains 220 color images

TABLE 2. Detection performances under different thresholds.

| Threshold | Robustness | Discrimination |
|---|---|---|
| 5 | 86.53% | 99.36% |
| 10 | 88.45% | 99.17% |
| 25 | 95.21% | 98.95% |
| 30 | 98.43% | 98.73% |
| 35 | 98.81% | 98.15% |
| 40 | 99.60% | 97.80% |
| 45 | 99.81% | 96.18% |
| 50 | 99.95% | 93.20% |

of 1920 ×1028 sizes. The authors generate the hashes for 220 color images, and the authors calculate the Hamming distance between the hash of each image and the hash of the other images. Then the authors get a total of 220 × (220-1)/2 = 48,180 results. The distribution of all Hamming distances is as shown in Figure 4. The x-axis and the y-axis represent the Hamming distance and statistics, respectively. The authors found that most Hamming distances are much greater than 30. This shows that the hashing algorithm proposed has good discriminability. If the authors choose a threshold of 30, i.e., Th=30, then 99.18% of the image pairs will be correctly identified. In fact, a smaller threshold can boost distinguishing ability, but it will reduce robustness. Similarly, a larger threshold will improve robustness, but will reduce discrimination. In practical applications, a suitable threshold can be selected according to different requirements. The results of robustness and discrimination under different thresholds are listed in Table 2, when Th=30, the algorithm achieves the best compromise effect. The larger the Th, the stronger the robustness. The smaller the Th, the stronger the discrimination.

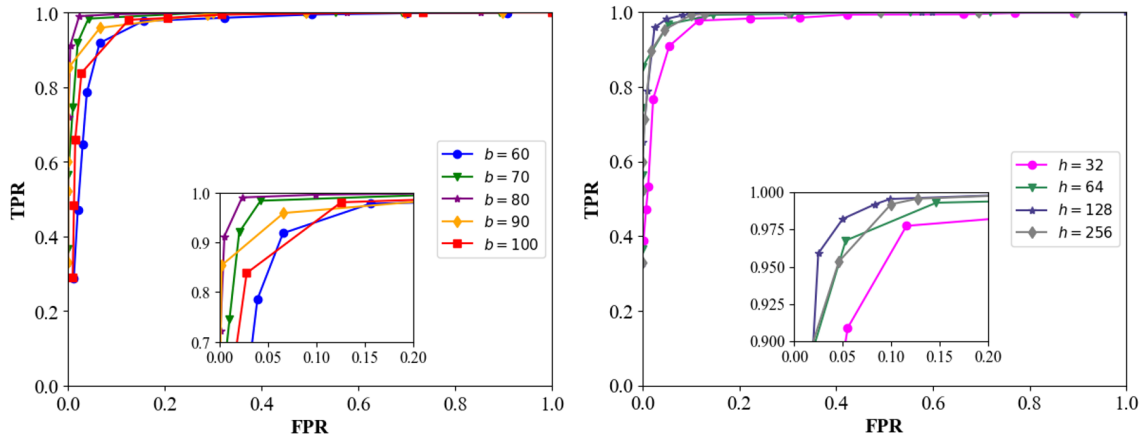


FIGURE 5. The ROC curve under different b and h values respectively.

3.4. **Dominant Parameter Discussion.** In order to verify the algorithm's trade-off effect on robustness and discrimination, the parameters involved need to be adjusted in the experiment. The ROC curve reflects the influence of different parameter settings on the performance of the algorithm. These parameters are: the size of overlapping blocks (b) and the length of hash bits (h). In the experiments, other parameters remain unchanged. The authors use the Holidays database and RTD database to verify the performance of the algorithm. Firstly, the authors verify the influence of the overlap

block size on the performance of the algorithm. The values of b are set as 60, 70, 80, 90, 100 respectively. As shown in the Figure 5, TPR is the proportion of similar image pairs that are correctly identified. FPR is the proportion of different images that are correctly identified. the authors can find that when b=80, the ROC curve is closer to the upper left corner, which means that the algorithm can better balance robustness and discrimination under this condition. Secondly, the authors verify the impact of hash code length on the performance of the algorithm. Image hashing technology converts huge image data into shorter binary sequences, which greatly simplifies the retrieval and management of digital images. Therefore, different hash bits also affect the algorithm. The authors set the value of h to 32, 64, 128, 256, respectively. As shown in the Figure 5, the authors can find that when h=128, the algorithm achieves the best balance between robustness and discrimination.

3.5. **Performance Comparison.** The authors compare the proposed algorithm with some advanced image hashing algorithms in terms of robustness, discrimination, time and memory, i.e., TD [19], RPIVD [20], QSVD [12], QFT [13], Feature Point [7]. For fair evaluation, the parameters for each compared algorithm are set the same as the original paper. In order to compare the robustness and discrimination of each algorithm. The authors choose a reasonable threshold for each hash algorithm to balance its robustness and discrimination. For a more detailed and theoretical analysis, the authors use the ROC curve. The ROC curve comparison diagram of different hashing algorithms is obtained, as shown in Figure 6. To see more clearly, the authors zoom in on the upper left corner and create a detailed image in the lower right corner. It can be observed that the performance of the algorithm proposed is better than other algorithms.

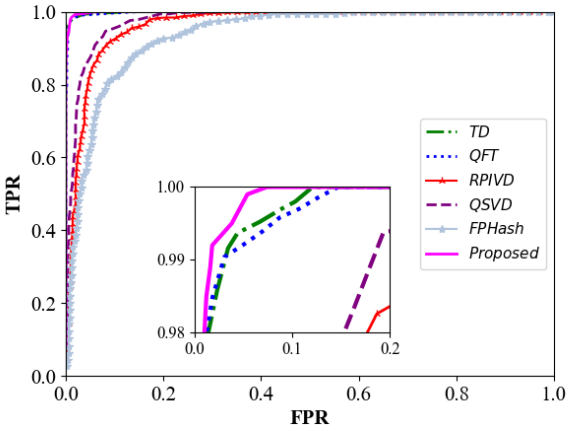


FIGURE 6. ROC curve comparison among different hashing algorithms.

In addition, the authors compare the length of the hash and the time required to generate the hash. The authors record the total time to generate all the image hashes in the experiment. Then calculate the average time to generate a single image hash. Furthermore, all hash algorithms are written in MATLAB 2020a and run on the same configured computer. The results are shown in Table 3. In general, the algorithm in this paper has achieved reasonable performance in terms of hash length and calculation time.

4. **Conclusion.** In this paper, the authors propose a robust image hashing algorithm based on multi-view feature representation and tensor decomposition. The algorithm effectively utilizes the complementarity and independence of multiple views, and constructs

a binary robust hash code combining the characteristics of multiple views. The Hamming distance is used to estimate the similarity between two hashes. Experiments were conducted using open image databases. The results show that the proposed algorithm is robust to most numerical operations and achieves strong discrimination. The authors also further compare the performance with some recently popular algorithms. The results show that the proposed algorithm is superior to other algorithms in terms of robustness and discrimination and achieves reasonable performance in terms of hash length and execution time.

TABLE 3. Performance comparison for the running time (second) and hash length.

| **Algorithm** | **Averagetime(s)** | **Hashlength** |
|---|---|---|
| TD [19] | 0.170 | 96 bits |
| RPIVD [20] | 0.203 | 440 bits |
| QSVD [12] | 0.250 | 130 bits |
| QFT [13] | 1.754 | 86 bits |
| FeaturePoint[7] | 0.304 | 960 bits |
| Proposed | 0.451 | 259 bits |

## REFERENCES

[1] S. Unar, X.Y. Wang, C. Zhang. Visual and textual information fusion using kernel method for content based image retrieval. In information Fusion. 44: 176-187, 2018.

[2] C. Wang, X.Y. Wang, Z.Q. Xia, C. Zhang. Geometric correction based color image water marking using fuzzy least squares support vector machine and bessel k form distribution. In Signal Processing. 134: 197-208, 2017.

[3] Z.Q. Huang, H. Lao, Z.J. Tang, X.Q. Zhang. Robust image hashing with multidimensional scaling. In Signal Processing. 137: 240-250, 2017.

[4] Gharde, Nilesh Dilipkumar, Thounaojam, Dalton Meitei, Soni. Robust perceptual image hashing using fuzzy color histogram. In Multimedia Tools and Applications. 77(23): 1-26, 2018.

[5] L. Ghouti, E.H. Khiari, I.H. Laradji. Perceptual hashing of color images using hypercomplex representations. IEEE International Conference on Image Processing. pp. 4402-4406, 2013.

[6] Hosny, I. Walid, M. Khalid, Mohamed, M. Yasmeen. Robust image hashing using exact Gaussian–Hermite moments. In IET Image Processing. 12(12): 2178-2185, 2018.

[7] B.L. Evans, V. Monga. Perceptual image hashing via feature points: performance evaluation and tradeoffs. In IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society. 15(11): 3453-3466, 2006.

[8] R. Davarzani, S. Mozaffari, K. Yaghmaie. Perceptual image hashing using center-symmetric local binary patterns. In Multimedia Tools and Applications. 75(8): 4639-4667, 2016.

[9] K. Naoe, Y. Takefuji. Damageless image hashing using neural network International Conference of Soft Computing and Pattern Recognition. 442-447, 2010.

[10] C. Cui, S. Wang and X.M. Niu. A Robust Depth-Image-Based-Rendering 3D Image Hashing Scheme Based on Histogram Shape, Journal of Information Hiding and Multimedia Signal Processing, 7(3): 543-557, May 2016.

[11] F.X. Yu, Y.Q. Lei, Y.G. Wang, Z.M. Lu. Robust Image Hashing Based on Statistical Invariance of DCT Coefficients, Journal of Information Hiding and Multimedia Signal Processing, 1(4): 286-291, October 2010.

[12] L. Ghouti. Robust perceptual color image hashing using quaternion singular value decomposition. In IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). 2014.

[13] C.M. Pun, C.P. Yan, X.C. Yuan. Multi-scale image hashing using adaptive local feature extraction for robust tampering detection. In Signal Processing. 121: 1-16, 2016.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. In foundations and Trends in Machine Learning. 3(1): 1-122, 2010.

[15] K. Hayashi, H. Kashima, R. Tomioka. Estimation of low-rank tensors via convex optimization. Arxiv 2011.

[16] L.D. Lathauwer, B.D. Moor, J. Vandewalle. A multilinear singular value decomposition. In SIAM. J. Matrix Anal. 21(4): 1253-1278, 2000.
[17] P.M. Kroonenberg. Applied Multiway Data Analysis. In Wiley-Interscience 2008.
[18] P.M. Kroonenberg, J.D. Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. In Psychometrika. 45(1): 69-97, 1980.
[19] L. Chen, Z.J. Tang, S.C Zhang, X.Q. Zhang. Robust image hashing with tensor decomposition. In IEEE Transactions on Knowledge and Data Engineering. 31(3): 549-560, 2019.
[20] X. Li, Z.J. Tang, X.Q. Zhang, S. Zhang. Robust image hashing with ring partition and invariant vector distance. IEEE Transactions on Information Forensics and Security. 11(1): 200-214, 2015.