computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

THE METHOD OF FINDING THE SPAM IMAGES BASED ON THE HASH OF THE KEY POINTS OF THE IMAGE

Nikolay I. Korsunov [1], Dmitri A. Toropchin [2]

[1] Department of Mathematic and Software Information Systems,
The National Research University "Belgorod State University", Belgorod, Russia, 308024,
Korsunov@intbel.ru
2) Department of Applied Informatics and Information Technology,
The National Research University "Belgorod State University", Belgorod, Russia, 308033,
Re1aps@rocketmail.com

Abstract: The article deals with the problems of the existing image recognition algorithms and also presents a method allowing to minimize the drawbacks of existing algorithms, and even to surpass them in certain moments. The proposed method is based on the description of the key points of the image by using perceptual hash. As a demonstration of application of the developed algorithm it is proposed to make the spam filter for images. The purpose of the spam filter is to separate, to divide the images from the collection into certain classes. The core of the spam filter is the developed method. In its conclusion the article contains information that shows the accuracy and performance of the proposed method in comparison with existing methods.

Keywords: perceptual hash; image descriptor; image recognition; image search.

1. INRODUCION

Due to the rapid development of both hardware and software technologies in the information sphere the question of duplication of information is of great importance nowadays. It is a serious problem for the data center (DC).

The purpose of article is to present the fuzzy duplicate images which are distributed mainly through the Internet. According to studies of such search giants like Google and Yandex it is known that more than 50% of all images containing in the Internet are duplicates. It should be said that the fuzzy duplicate image is a serious problem for the data center, because of the following disadvantages:

- although data center may be applied for huge data volume it is necessary to indicate that when the fuzzy duplicate image is used, the center has to process the same information for several times. As a rule it's not effective, because computing power of the data center falls dramatically;
- besides the system doesn't know that multiple copies of the same image are stored in it, so it can't provide a place for storage properly;
- often methods of search of fuzzy duplicate images work very inefficiently, as a result the data center has to work more intensively. So user may not be satisfied with the results of the method that causes usually a new iteration of the search.

Nowadays there are many methods for searching of fuzzy duplicate images [1]. These methods generally use main ways for image recognition which are based on the allocation of image features, on the construction of the image descriptors, on the construction of binary fingerprint [2].

Each of them allows to determine fuzzy duplicate images, but if considering these methods deeper the following conclusion can be made about their disadvantages [3]:

- require a huge time-consuming;
- have a closed architecture ;
- are difficult to implement and scale;
- place heavy demands on hardware and software;
- use the system resources inefficiently;
- have a significant error in the results obtained.

2. THE AIMS AND THE PROBLEMS

The aim of the presented research is improving the efficiency of the recognition process of fuzzy duplicate images by introduction of perceptual hash which is used for construction of descriptors point.

Four types of duplicate images are considered here [3]:

- exact duplicates of images which do not differ in any one bit;
- thumbnail duplicates differing in size;
- floor duplicates – image with inscriptions, image cropping, images with color correction and minor changes;
- advanced floor duplicates – pictures with heavily modified colors or proportions, fragments, etc.

Taking into account all the above it's necessary to point out that the proposed method should ensure pattern recognition of all types of image duplicates and should be easily scalable if a new type of duplicates appears.

To achieve this aim the following problems have been solved [4]:

1. construction of a mathematical model to estimate similarity duplicate images;
2. development of a method for recognition of fuzzy duplicate images;
3. development of the structure of the special software that implements the methods and algorithms for recognition of fuzzy duplicate images;
4. experimental test of the developed method.

To solve the listed problems the recognition method of fuzzy duplicate images based on an image alignment points and calculation of the perceptual hash is proposed. As a result of combining the two approaches a descriptor is constructed, on the basis of similarities of which the decision about the similarity of images is made.

However, construction of descriptors using the existing methods has the following disadvantages [5]:

- the minimum size of the descriptor is 128 KB, so it's impossible to store the sufficient number of descriptors in RAM for quick sampling;
- the large size of descriptors doesn't allow to use effectively the database for their storing;
- the speed of constructing a vector descriptor is quite slow, even when using the parallel algorithms;
- it's difficult to process the image collection in real time;
- there is no way of constructing descriptors that could solve all the above problems.

Today the basic perceptual algorithms are considered to be:

- Simple hash [6] – the gist of this algorithm is to display the average value of the low image frequency. The high frequencies provide detailed elaboration in the images but low frequencies show a structure. The obtained hash is poorly resistant to scaling, to stretching, it changes brightness, contrast, etc. However, its main advantage is operating speed.
- Discrete Cosine Transform Based Hash [7] – this algorithm is based on DCT. Its main advantage is that it is resistant to the small rotations, to blurring, to image compression. The speed of comparing hashes, because of their small size, is very high.
- Radial Variance Based Hash [8] – the idea of this algorithm is to convert the ray vector variance based on the Radon transform. Then, DCT is applied to the ray vector variance and hash is calculated. In fact it's an advanced construction of hash with the help of DCT.
- Marr-Hildreth Operator Based Hash [9] – this algorithm works on the basis of the contours of the image. The size of the obtained hash is large enough, but comparing of two hashes takes less time than Radial Variance Based Hash. This algorithm is sensitive to the rotation of the image, but it is resistant to scaling, to darkening and to compression. This algorithm shows best results if the image is too bright.

Based on the above, for the construction of hash key points it was decided to use the Discrete Cosine Transform Based Hash with some modifications that will be considered further.

3. METHOD PRINCIPLES

Construction of the descriptor begins with identifying key points of the image Q having dimension $m \times n$.

The key point of the image is called such point of the image which is more likely could be found on another image of this object [10].

Finding key points allows to achieve invariance to displacement, to rotation, to scaling, to changing of the brightness and camera position. The main moment in the detection of critical points is the construction of pyramid of Gaussians and Gaussians differences [11].

The result is an image Q' with highlighted key points. Using received key points we can construct vector V which is a coordinate of key points.

Add items to a vector occurs in succession beginning with the key point with the lowest coordinate and ending the key point with the highest coordinate.

$$V = \{I'[i, j], \ldots I'[i', j']\} \quad (1)$$

where i, j – coordinates of the first key point, i', j' – coordinates of the last key point of the image Q'.

To describe the key points it is proposed to use the perceptual hash. The data using to generate the hash are considered as a source of random numbers, so that the same data will provide the same result, but different data – different result. Perceptual hashes may be compared with each other and make a conclusion about the degree of difference between the two sets of data [12].

To construct the hash specify the range of permissible errors S finding of key point. For that the size of S, defined on a grid of pixels Q' must be separable for all points. When setting error come from separability of areas of inclusion of key points.

$$S_1 \,\&\, S_2 = \varnothing \tag{2}$$

For the error of 1% on the 256x256 grid experimentally set the S size – 7x7. After specifying the areas of key points construct the perceptual hash, placed into descriptor.

For each element of vector V (coordinates of key points) find the region S and placed into the vector K in the same manner as the coordinates of key points.

$$K = \{S_1[i,j],\ldots,S_n[i,j]\}, \tag{3}$$

where S is the region taken about the key point, $[i,j]$ – region size.

Next, need to find the hash of taken fields. To do that it's necessary to carry out the following conversions over each element of the vector K 1]-[3]:

1) find the average color value Mc of pixels $S_n[i,j]$.

$$Mc = \frac{\sum Sn[i,j]}{i+j} \tag{4}$$

2) construct binary vector prints K' of each region S from K, each element S assign zero or one depending on Mc. If

$$S[i,j] > Mc \geq S[i,j], \tag{5}$$

then $S[i,j] = 0$ or 1, where i,j – the brightness value pixel of the region S.

$$K' = \{S_1'[0,1,0,\ldots,0,1,0],\ldots,S_n'[0,1,0,\ldots,0,1,0], \tag{6}$$

where the elements S' calculated from the formula (5), (6).

3) to eliminate bitwise comparison use positional system for comparing; besides, such method is convenient for storage. For that the bit sequence converts into hexadecimal notation. Each S' of vector K' converts into the element vector descriptor D.

$$D = \{S_1' -> HEX,\ldots,S_n' -> HEX\} \tag{7}$$

That is, as a result of conducted conversions, form the vector descriptor (7) with a description of features of image Q.

To assess the similarity of images, use Hamming distance [13] between reference image D and image duplicate D', the result placed into the vector similarity H.

$$H = \{D_{i=0}^1 - D_{i=0}^2,\ldots,D_{i=n}^1 - D_{i=n}^2\}, \tag{8}$$

where D_1, D_2 – indices elements descriptor of the reference image and duplicate images, i – index of element D_1, D_2.

Next determine the similarity coefficient k^*. $k^* = 0$ indicates 100% similarity. To determine the partial similarity experimentally it found that k^*: $0 \leq k^* \leq 10$, if k^* is less than 10 – the images are partly similar, if more than 10 – the images are not similar.

For classification convenience formed weighted vector of similarity H' of vector H based on k^*. To each element H assigned its weight depending on its similarity.

During the experiment, it revealed that k^* is given by (9) on the basis of the weighted coefficients vector H'

$$\begin{cases} partlySimilar, H'[i] = 1 \\ similar, H'[i] = 2 \\ notSimilar, H'[i] = 0 \end{cases} \tag{9}$$

Each element of the vector H' corresponds to the coordinates of the key points of vector V.

4. DEVICE OF THE SPAM FILTER ON THE BASIS OF THE PROPOSED METHOD

In the first section it is explained what the fuzzy duplicate images are as well as the actual problems connected with processing are considered. This section gives an example of elimination of duplicate fuzzy images by adapting the developed algorism as a base algorism for spam filter.

Suppose that there is a service that is attacked by spams very often. The concept "the duplicate image" replaced by another term "spam" [14] further. In most cases the moderators of this service when

detecting spam-images check up whether there are images having the similar spam-fragment in the database. So their task is to find such images, then delete them and also to block loading of this image to the server. The proposed spam filter using the foregoing method, allows to automate the process completely and reduce the work of the spam moderators to a minimum.

Spam filter device is quite simple: plurality of patterns are supplied to the input of the filter. The plurality of patterns represent a set of pairs: the way to the spam image and perceptual hash of key points of the image.

The gist of plurality of patterns is that the filter constructs a hash values for all the spam images among the patterns and when searching a spam compares hashes of key points among the patterns with hashes of the key points of scanned image. If the Hamming distance between the hashes is sufficiently small, at least for one pattern, the filter declares a corresponding image as a spam.

As a result of work of spam filter the images are divided into three groups:

1. Certainly spam
2. Probably spam.
3. Just do not spam

The group "possible spam" means that the proposed algorithm compares two hashes, and the "similarity" returns in the exit: the distance obtained with the help of certain metrics. There is a range of "similarities" for the proposed algorithm, that contains both spam and not spam. Because of this reason this group has been introduced.

When using the database need not to process all the images each time that greatly increases the speed of the work of the proposed method.

The algorithm of the work of the proposed spam filter can be presented by several steps:

1. Addition of spam images into the plurality of patterns.
2. Search spam images for plurality of patterns.
3. Division of the obtained results into groups
   a. certainly spam
   b. probably spam
   c. just do not spam.
4. In the case that:
   a. deleting images from the database
   b. moderator checks the spam-image manually
   c. spam-images are added into the plurality of patterns.
5. The cycle is executed until the images are in the plurality of patterns.

5. TESTING METHOD

The test consists of a set of photos of different sizes and different quality. Photos are not linked any additional information (such as annotations, tags or other context).

The test simulates the task of finding images in private collections of amateur photographers. For fast implementation of common methods and approaches of computer vision [15] Computer Vision library Open CV was used [16].

Collection is a subset of the collection of Flickr [17] and contains 20,000 pictures taken indoors and outdoors, including portraits, landscapes, city scenes and other types of photos. The dimension of the images does not exceed 500 pixels (the typical size of 500x375).

Of the hardware – Intel Core i7 with 16Gb RAM.

In order to test the proposed method the key points perceptual hash was described by all known hash algorithms.

Divide the collection of images into 2 classes: spam (class 1) and non-spam (class 2). There are 5 different work outcomes of filter (Table 1). To evaluate the quality of the obtained results, the following metrics were used [18]:

False Accept Rate (FAR). FAR is the number of all the false-positive alarms (errors of the 2nd kind). False-positive alarm arises if the filter determines the image as a spam, which, in fact, is not spam.

False Reject Rate (FRR) – is the number of all false-negative alarms (errors of the 1st kind). False-negative alarm arises if the filter determines the image as not spam, which actually is spam.

Table 1. Table of outcomes

Result of work	Class 1	Class 2
The positive alarm	True positive	False-positive (errors of the 2nd kind)
The negative alarm	False-negative (errors of the 1st kind)	True negative
Uncertain alarm	"Possible spam" group in which there are both classes of images	

It was decided to limit the amount of the group "Undetected images" the 1% of the entire collection. Results are presented in Table 2. In the lines 1 and 2 the share of the spam images from their total number, caught into a group, is shown in percentages.

In the line "Undetected images" it is shown the number caught into this group of images and the number of images that were spams among them after manual checking.

In the line "False-positive alarm" it is shown the probability of error as a percentage of entire database.

The result of the comparison of two hashes is "similarity" s. If s is greater than a selected threshold T, the images are considered to be perceptually similar. Thanks to variation of the threshold possible to increase the number of errors 1 and decrease the number of errors 2, and vice versa.

Furthermore, an additional group "possible spam" was introduced. This group caught the images with such similarities s that $T1 < s < T2$, where $T1$ and $T2$ are the thresholds that $T1 < T < T2$. Group "possible spam" allows to reduce significantly the number of errors 1 and 2.

In the Fig. 1, T – threshold, s – the similarity of the two hashes, $p\ (s \mid H0)$ – the probability that the similarity s image is actually spam, $p\ (s \mid H1)$ – the probability that the similarity s image is not spam.

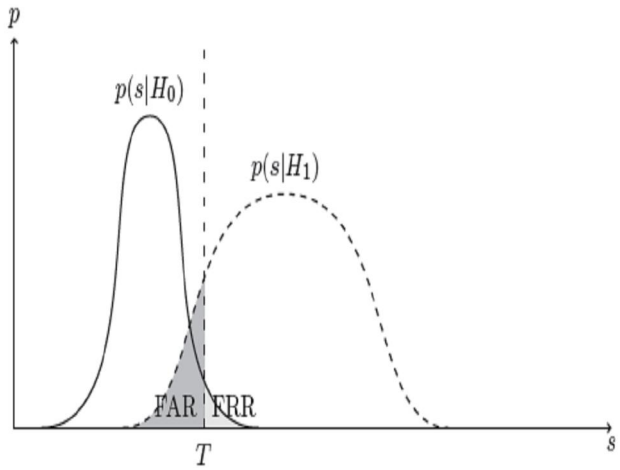

Fig. 1 – FAR and FRR metrics

In the course of the experiments the best thresholds similarity for each algorithm were revealed, minimizing false-positive and false-negative alarms.

It was decided to limit the power of the group "possible spam" the 1% of the database capacity. The results are shown in Table 2.

Table 2. The results of the working methods

Method\ Alarm	SH	DCT	RV	MH	Proposed method
True positive	71%	79%	82%	75%	89,5%
Undetected images	3%	11%	17%	8%	4%
False positive	0%	0,7%	0%	1%	0%
False negative	26%	10%	1%	17%	5,5%

In the 1st and 2nd lines share of spam images as a percentage of their total number, caught into a group is shown.

In the line "probably spam" it is shown the amount of images caught into this group and the amount of images that were spams among them after manual checking.

In the line for a false-positive alarm it is shown the probability of error as a percentage of all database.

Thus, it becomes possible to avoid the false-positive alarms almost for all methods. Thanks to the group "undetected images", the number of false-negative alarms is minimized for all methods.

Analysis of the effectiveness of the method proved that the time needed to construct descriptor with the help of proposed method is much less than the time required for one of the known methods (Table 3) [19].

Table 3. Time processing a single image

Method	SH	DCT	RV	MH	Proposed method
Time (seconds)	0.15	0.27	0.41	0.93	0.32

The results showed that Simple Hash algorithm is the fastest, being ahead of the rest of algorithms, but its accuracy leaves much to be desired. For algorithms SH, DCT, MH and for the proposed method also comparison functions of the hashes values (Hamming distance) work much faster compared with the calculation of hash , so in these cases it's not worth worrying about the cardinality of the plurality of patterns.

6. CONCLUSION

The obtained results show that the proposed algorithm [20] is the best to be used for description of the key points. It has the highest accuracy among the considered perceptual hash algorithms and is not inferior to them in speed. It can be seen from testing the algorithm was successfully used as a basis for the spam filter to search spam images. Based on the foregoing, it can be said that the tasks set out in this article have been resolved, and the aims have been achieved.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education and Science of the Russian Federation, project 14.578.21.0138.

7. REFERENCES

[1] *Computer vision. The modern approach*, Moscow, Williams Publishing House, 2004. – 928 p. (in Russian).

[2] L. Shapiro, G. Stockman, *Computer Vision*, Moscow, Bean, Laboratory Knowledge, 2006, 752 p. (in Russian).

[3] U. V. Vizilter, S. U. Zheltov, A.V. Bondarenko, M. V. Ososkov, A. V. Morzhin, *Image Processing and Analysis Tasks in Machine Vision: Lectures and Practical Classes*, Moscow, Fizmatkniga, 2010, 672 p. (in Russian).

[4] V. I. Vasiliev, *The Problem of Learning Pattern Recognition. The Principles, Algorithms, Implementation*, Kiev, Vyshchaya School, 1989, 64 p. (in Russian).

[5] D. V. Sorokin, A. S. Krylov, "A projection local image descriptor," *Pattern Recognition and Image Analysis*, Springer, Vol. 22, No. 1, pp. 380-385, 2012.

[6] Simple and DCT perceptual hash-algorithms [Online] www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html

[7] Zeng Jie, "A novel block-DCT and PCA based image perceptual hashing algorithm," *International Journal of Computer Science Issues*, Vol. 10, Issue 1, No 3, January 2013.

[8] F. X. Standaert, F. Lefebvre, G. Rouvroy, B. M. Macq, J. J. Quisquater, J. D. Legat, "Practical evaluation of a radial soft hash algorithm," in *Proceedings of the IEEE International Symposium on Information Technology, Coding and Computing (ITCC)*, April 2005, Vol. 2, pp. 89-94.

[9] D. Marrand, E. Hildret, *Theory of Edge Detection*, 1979, pp. 187-215.

[10] A. S. Krylov, D. V. Sorokin, D. V. Yurin, E. V. Semeikina, "Use of color information for keypoints detection and descriptors construction," *Lecture Notes in Computer Science, Intelligent Science and Intelligent Data Engineering*, Springer, Vol. 7202, pp. 389−396, 2012.

[11] J. Russel, R. Cohn, *Difference of Gaussians*, 2012, 76 p.

[12] M. Egmont-Petersen, D. de Ridder, H. Handels, "Image processing with neural networks – a review," *Pattern Recognition*, Vol. 35, Issue 10, pp. 2279-2301, 2002.

[13] R. V. Hemming. *Numerical Methods for Scientists and Engineers*, Science, 1972, 400 p.

[14] https://en.wikipedia.org/wiki/Spamming

[15] G. Bradski and A. Kaehler, *Learning open CV,* First Edition, O'Reilly Media, Inc., September 2008.

[16] Open Computer Vision Library [Online] http://opencv.org/

[17] www.flickr.com

[18] *Encyclopedia of Biometrics*, S. Z. Li, A. K. Jain (eds.), Springer, 2009, 1445 p.

[19] C. Zauner, *Implementation and Benchmarking of Perceptual Image Hash Functions*, PhD Thesis, 2010.

[20] N. I. Korsunov, D. A. Toropchin, "Recognition method of near-duplicate images based on the perceptual hash and image key points using," in *Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and applications (IDAACS'2015)*, Warsaw, Poland, 24-26 September 2015, Vol. 1, pp. 261-265.

Nikolay Korsunov, *Doctor of Technical Sciences, Professor of the Department of Mathematic and Software Information Systems at the National Research University "Belgorod State University" / "BelSU", Belgorod, Russia.*

Research interests: Development of methods and algorithms of computer vision.

Dmitri Toropchin, *Post-graduate student at the Department of Applied Informatics and Information Technology at the National Research University "Belgorod State University" "BelSU", Belgorod, Russia.*

Research interests: Development of methods and algorithms of computer vision.