International Journal of Computer Science and Applications,

Vol. 15, No. 1, pp. 33 – 44, 2018

A BLOCKCHAIN OF IMAGE COPYRIGHTS USING ROBUST IMAGE FEATURES AND LOCALITY-SENSITIVE HASHING*

ALEKSEI ZHUVIKIN

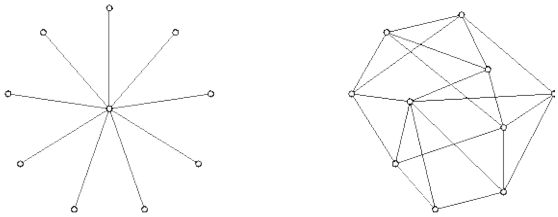
Department of Secured Communication Systems,
The Bonch-Bruevich St. Petersburg State University of Telecommunications,
61 Moika, St. Petersburg 191186, Russia,
zhuvikin@ya.ru

The paper presents the blockchain scheme which allows to build image copyright registry. First of all, the aim of the proposed blockchain is to be an authoritative ledger of image copyrights which contents are protected under some distribution constraints and, secondly, to provide such copyrights over the network in which they are used. Due to the use of robust image feature vector the scheme has an ability to detect images with the same content and reject attempts to add duplicates even if JPEG compression has been applied. In order to achieve fast copyright retrieving by given image content the locality-sensitive hashing scheme is used. Thanks to the decentralized structure of a blockchain there is no single point of failure. On the other hand, newly added blocks with image copyrights are secured because of irreversible nature of the blockchain. Two-step publishing algorithm of new image copyright is used in order to prevent copyright steal. Experimental results show that the introduced image feature vector has robustness to the content-preserving operations while being sensitive to small content changes. All these properties qualify the proposed scheme as recommendable to use instead of services with single trusted authority.

Keywords: Blockchain; Smart Contract; Image Features; Locality-Sensitive Hashing; JPEG; Image Copyright.

1. Introduction

Nowadays it is fairly common to confirm the owner of content copyrights through a trusted authoritative third-party. In the image processing field, photo, clip-art and digital content marketplaces are widely used by photographers, designers and artists. Such services contain copyrighted images submitted by a lot of image owners and are distributed under certain licenses. Authoritative service tracks any owners' copyright violation attempts of and forbids the submitting of the image which was already submitted by someone else. Being quite successful in the current state of affairs the single administrator model has still significant disadvantages. The content owner must trust the administrator of such service. In other words, administrator is a trusted third-party. First of all, this implies a significant risk of image appropriation for the image authors from dishonest services. Secondly, in the trust-centric approach it is more likely to have a single point of failure which makes all copyrights ledger infrastructure vulnerable.



a) **b)**

Fig. 1. The schemes of trusted third-party model (a) and decentralized model of blockchain (b).

Another approach of proving that a person has rights on some digital asset is to use a blockchain. The term blockchain was established in 2008 by a person or an unknown group known as Satoshi Nakamoto [1]. The first implemented public blockchain was built as a core of decentralized digital currency system known as Bitcoin. The information about users' assets and transactions between them is recorded in blockchain by so-called miners. A blockchain system consists of data records known as blocks. These blocks contain all necessary ledger information. Each new block has a link to the previous one and is added to the end of blockchain by the miners. Miners are the honest members of the blockchain network. All network nodes are performing hard calculations in order to earn rewards for the newly calculated blocks. The safety of this system is provided by the so-called proof-of-work principle [2]. The aim of the algorithm is to achieve the blockchain consensus between members and requires miners to find a solution of a difficult problem which complexity is adjusted dynamically making the block generation frequency almost constant. However, the proof-of-work principle is going to consume more and more electrical energy. Recently it was calculated that miners of Bitcoin network in total will consume the same amount of electricity as Denmark in 2020. This fact makes the proof-of-work principle ecologically not acceptable in the near future. This consideration leads researchers to new algorithm known as proof-of-stake [3]. The new algorithm allows to achieve distributed consensus as well but has significant advantages comparing to the proof-of-work method. In this scheme the creator of the next block is chosen randomly or by means of various combinations such as e.g. age of the network member. The new algorithm hasn't worked in the large practical blockchain yet. However, one of the world leading blockchain system called Ethereum is planned to be switched over to proof-of-stake in its next hard fork procedure.

An important question is how much the image should differ from the copyrighted one in order to be recognized as the image the copyright is not distributed on. The main criteria of such classification have to determine if both images have the same content. An image content actually means the set of objects, contours, gradients and scene depicted on the given image. The most difficult task is to distinguish if there were content-preserving modifications like JPEG compression, resizing, slight color and brightness corrections or this is an image with a different content.

In order to find compact image content description a large amount of techniques have been proposed in the literature [4], [5], [6]. Unfortunately, a lot of image content description algorithms tend to be computationally expensive i.e. complexity is rarely going below $O(n^2)$. This fact makes a real-time searching of images with the same content in the large databases difficult. Nevertheless, there are probabilistic algorithms which allow a user to retrieve the similar images by a given query in constant time. These algorithms are known as locality-sensitive hashing (LSH) techniques [7]. For example, Google's VisualRank algorithm uses color histograms and shape analysis with conjunction of local image descriptors in order to calculate the image features [8]. And then LSH method is used for efficient representation and retrieval of image results which are relevant to the query. As we show below, the retrieving of the similar images is not the only benefit of LSH application in the image processing field. In the proposed scheme we use LSH for fast copyright retrieving from blockchain by the given image content.

In this paper we present a novel algorithm of a public image copyrights ledger based on blockchain by using robust image features and LSH. The contents of paper are organized as follows. Section 2 of the paper presents the main structure of the considered blockchain. The main properties of central finite differences and proposed image feature vector algorithm are covered in Section 3. The usage of LSH technique for retrieving the queried copyright information is described in Section 4. The evaluation of ability to retrieve the image copyright information from blockchain is presented in Section 5 and is followed by Section 6 which contains the conclusions.

2. Proposed blockchain structure

The principle mechanism of any blockchain network is an algorithm of communication between distributed nodes which allows them to reach a consensus on the data stored in the network. Such algorithms are to solve the truth conflicts over all the honest network nodes. Well-known CAP theorem states that for any distributed data store it is impossible to provide consistency, availability and partition tolerance [9] simultaneously. This has an impact on the blockchain network which has a partitioned data for the latest transactions between its nodes. A consistency of data is achieved only after the network nodes reach a consensus.

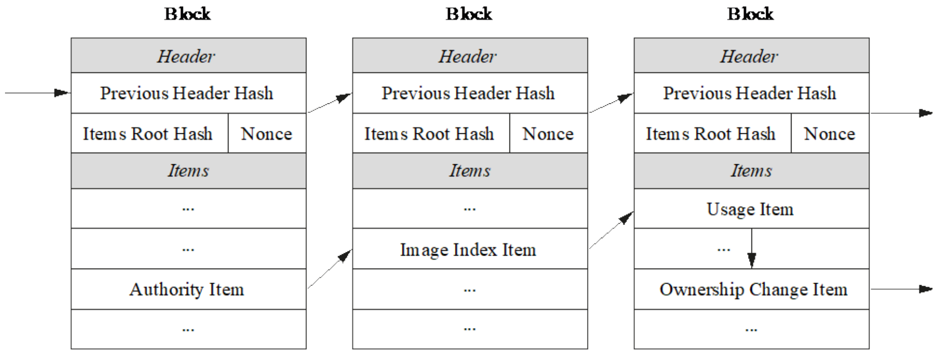


Fig. 2. The blocks structure of the proposed scheme.

The proposed blockchain consists of a consecutive data blocks and is depicted on Figure 2. Each block except for the first one has a link to the previous block. The block is divided into a *Header* and several *Items* sections. *Header* section contains hash of the previous block, root hash of items and the nonce parameter. The latter is used in previously mentioned proof-of-work algorithm and is calculated by the miners in order to meet specific requirements. The *Items* section could contain one or more items of the following types:

- *Authority Item* - an initial data structure of any chain of image owning. This item contains public key hash of an image owner and the blind signature of the image content signed by owner's private key.
- *Image Index Item* - a data which can appear in the blockchain only after block with corresponding *Authority Item* is successfully added to the blockchain. The item contains hash of its *Authority Item*, locality-sensitive hash index of the image content and blinding factor which is used in the previous item. The procedure of its construction will be discussed below.
- *Usage Item* - a structure with information on who is able to use the digital asset. It contains hash of the previous item, public key hash of the image user as well as author's signature.
- *Ownership Change Item* - an item appeared when transfer of copyright to the next owner is needed. The structure is similar to the *Usage Item* but it contains next owner's public key hash instead of user's one.

The chain of *Authority*, *Image Index*, *Usage* and *Ownership Change* items is shown on Figure 3.

The next most usual use case of image copyright lifecycle is a process of sharing photographer's photo with advertising agency. Let the photographer be Alice and the advertising agency editor be Bob. Bob wants to use Alice's photo in his latest advertising banner. When Alice is going to get some revenue for her photographing she performs the following two-step publishing algorithm. Firstly, Alice calculates robust image feature vector which will be described in details in the next section. Then she performs locality-sensitive hashing of the image feature vector and signs it by her private key. After that

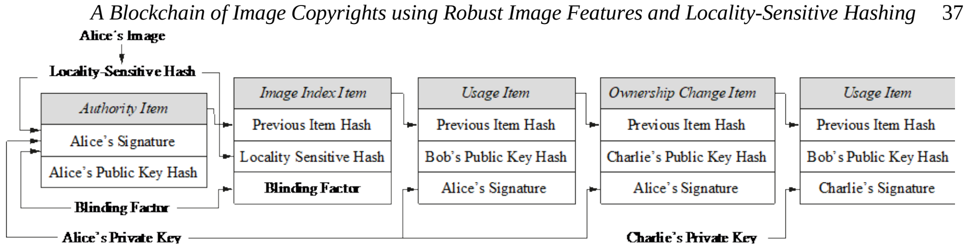


Fig. 3. The chain of image copyright authority, index, usage and ownership change items.

Alice generates random integer so-called *blinding factor* and multiplies it with the signature [10]. The resulted blind signature is added to the new *Authority Item* by Alice and broadcasted to all blockchain nodes called miners. Each node after receiving the Alice's item adds it to the other unprocessed items and tries to calculate the next block with random nonce parameter and given difficulty. This is the step where the proof-of-work algorithm is performed. When the next block is calculated by one of the nodes it is broadcasted to the other nodes then verified and added to the end on the blockchain.

Alice is periodically checking blockchain to be sure if her *Authority Item* is successfully added to the network by the sufficient number of nodes. Then she constructs *Image Index Item* with LSH and *blinding factor* used in the *Authority Item* and broadcasts it to the nodes. All nodes check across the blocks if received copyright data in the Image Index is unique. As miners trying to calculate new blocks all the time the *Image Index Item* will also be processed and added to the blockchain. From the moment when *Authority Item* is added to the blockchain by the sufficient number of nodes it is impossible for any member to steal the image copyright due to irreversibility of blockchain. So from this point forward publishing of image copyright details is safe because it can be verified that Alice is the real author of the image by blocks history traversing.

When Bob plans to use photo he wants to be aware that its usage will not violate anybody's copyrights. Bob calculates LSH and checks *Image Index Items* in the blockchain in order to find the current copyright owner. When Alice's copyright is found Bob pays to Alice some fees. Then Alice constructs new *Usage Item* for Bob and publishes it in the blockchain. After verifying that *Usage Item* appeared in the network Bob is able to use it legally. But after a while Charlie is interested in Alice's photo and would like to be a new copyright owner. He pays the copyright price of the Alice's photo and she publishes new *Ownership Change Item* using Charlie public key hash. Now Charlie is a rightful owner of the image and all subsequent *Usage Items* will be signed by him.

Let us note that we do not focus on how payments are implemented but they can also be implemented by means of blockchain. There are practically implemented systems of so-called smart contracts which allows to do so. One of the most popular is the Ethereum blockchain [11]. In the current paper we will focus only on image copyrights managing which can be easily integrated into the existing blockchain systems regardless of what type of consensus algorithm they use. We use LSH of robust image feature vector as an image description used in the copyright. The details of both techniques are considered in the next sections.

3. Robust image feature vector

In order to represent image content in such a way it can be easily added to the blockchain we calculate *image feature vector* first. An image feature vector also known as a perceptual image hash [5] is a compact representation of the image content and have to satisfy the following properties.

- Robustness to the content-preserving modification like JPEG compression.
- Low computation complexity.
- Low collision probability with the feature vectors of another image contents

In order to meet the given requirements, we use *Central finite differences* (CFD) as a fast and detailed image contents descriptor. We will show later that it has acceptable robustness as well as content description detalization. CFD of the first order is function $\{0, 1, \ldots, n_x\} \times \{0, 1, \ldots, n_y\}$ that is defined [12] as:

$$\delta_x(x, y) = \frac{1}{2}\big(I(x+1, y) - I(x-1, y)\big), \tag{1}$$

$$\delta_y(x, y) = \frac{1}{2}\big(I(x, y+1) - I(x, y-1)\big). \tag{2}$$

In order to decrease the noise at CFD after content-preserving manipulations it is proposed a convolution of the image luminance $(I(x, y) \mid (x, y) \in \{0, 1, \ldots, n_x\} \times \{0, 1, \ldots, n_y\})$ with a two-dimensional Gaussian-wise filter having pulse response:

$$h(i, j) = \begin{cases} \frac{1}{2\pi\sigma^2} exp\left(\frac{\left(i - \frac{n}{2}\right)^2 + \left(j - \frac{n}{2}\right)^2}{2\sigma^2}\right) & if\ 1 \le i, j\ \le n \\ 0 & otherwise \end{cases}. \tag{3}$$

where σ^2 characterizes the filtering parameter and n is the size of the Gaussian-wise window. After the two-dimensional convolution $h ** I$ we get

$$\tilde{I}(x, y) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} h(i, j) I(x - i, y - j). \tag{4}$$

The corresponding CFD's $(\tilde{\delta}_x\ (x, y))_{x,y}$, $(\tilde{\delta}_y\ (x, y))_{x,y}$ are obtained as in (1)-(2) with replacing $I(x, y)$ by $\tilde{I}(x, y)$. Let us define

$$\tilde{\delta}(x, y) = \sqrt{\tilde{\delta}_x(x, y)^2 + \tilde{\delta}_y(x, y)^2}$$

and let us consider the matrix $\mathbf{G} = [\tilde{\delta}(x, y)]_{x,\,y}$ (It is worth to note that the computation of $\tilde{I}(x, y)$ by (3) can be simplified when the CFD's are presented as a trivial convolution).

The elements of the matrix **G** are changing slightly after content-preserving manipulations but to be robust to image resizing it is needed to make this matrix be fixed size. We decrease the size of this matrix by means of so called *average downsampling* technique [13] with parameters s and t depending on original image size, corresponding divisors of n_x, n_y, for horizontal and vertical directions: $\forall(k, m) \in \{1, \ldots, n_x/s\} \times \{1, \ldots, n_y/t\}$ as follows

$$d_\Delta(k,m) = \frac{1}{st}\sum\left\{\tilde{\delta}(i,j) \mid s(m-1) < i \leq sm \ \& \ t(k-1) < j \leq tk\right\}$$

We call matrix $\mathbf{D} = [d(k, m)]_{k,m}$ the *image feature matrix*. Then, let us quantize the values $d(k, m)$ with some step $\Delta \in \mathbb{R}$ as

$$d_\Delta(k,m) = \left\lfloor \frac{d(k,m)}{\Delta} \right\rfloor + 1 \tag{5}$$

where $\lfloor\cdot\rfloor$ is the floor map. For simplicity, we will present these values as a linear array,

$$d_\Delta = \left(d_\Delta(i)\right)_{i=1}^{\frac{n_x n_y}{st}}$$

Resulting image feature vector $\tilde{d}_\Delta$ is robust to content-preserving manipulations and is sensitive to different image content at the same time. The following condition has been taken for image content equality rule

$$\left(I(x,y)\right)_{x,y} \ and \ \left(\tilde{I}(x,y)\right)_{x,y} \ are \ equal \ \Leftrightarrow \ \max_i\left|\tilde{d}_\Delta(i) - d_\Delta(i)\right| \leq 1. \tag{6}$$

Once image feature vector $\tilde{d}_\Delta$ calculated it can be published to the blockchain in *Image Index Item* as an image content descriptor. However, it would be impractical to use image feature directly in the blockchain. The reason is that the performance of checking if there is a copyright for the given image in the ledger will be worsen. On the other hand, there is a LSH technique allowing make the needed search fast and will be considered in the next section.

4. Locality-Sensitive hashing and copyright retrieving

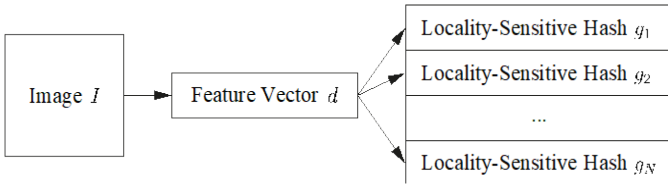


Fig. 4. The scheme of image copyright descriptor calculation.

Locality-sensitive hashing is an algorithm of reducing the dimensionality of high-dimensional data. Locality-sensitive hashing maps data in the way that similar items hashed to the same buckets with high probability. In order to achieve reducing of input data the number of bucket much be relatively small. Unlike cryptographic hash functions the goal of LSH is to maximize the collision probability for similar data [14]. Locality-sensitive hashing technique is widely used in the approximate or exact near neighbor search [15]. Nevertheless, we propose to use locality-sensitive hashing to retrieve the same image content even if it changed by content preserving operations like JPEG compression, resizing and etc. We use locality-sensitive hashing scheme based on α-stable distributions introduced by M. Datar et al. in [14].

In our scenario, an image feature vector is a point of L-dimensional space. Let us denote it by $\vec{d}$, then locality-sensitive hashing functions are given as follows [14]

$$f_i(\vec{d}) = \left\lfloor \frac{\vec{d} \cdot \vec{x}_i + b_i}{w} \right\rfloor, \quad i \in [0, K] \tag{7}$$

where $\lfloor \cdot \rfloor$ is a integer rounding operation, $w \in \mathbb{R}$ is the previously chosen value, $b_i \in \mathbb{R}$ is the randomly generated value, using uniform distribution such that $b_i \in [-w/2, w/2]$. $\vec{x}$ is the vector having the same dimensions as image feature vector $\vec{d}$ but with randomly chosen coordinates using one of the α-stable distributions, for example Gaussian distribution $N(0, 1)$.

Each locality-sensitive function f_i maps input image feature vector $\vec{d}$ to the one integer $f(d(I))$. Let $(f : S \rightarrow U)$ is a family of hashing functions f_i where $i \in [0, K]$, $K \in \mathbb{N}$. Then, let define new family $(g : S \rightarrow U)$ where each g_i is derived by concatenation of randomly chosen functions f_i from family $\boldsymbol{\mathcal{F}}$ as follows

$$\mathcal{G} = ([f_1(d),\ f_2(d),\ \dots, f_k(d)])_{1 < i \leq L,}\ k \in [0, t],\ t \in \mathbb{N} \tag{8}$$

The family of functions $\mathcal{G}$ is called locality-sensitive hashing because the more input image feature vector is close to the other the more likely there will be a collision of one or more hashes g_i. It will be shown in the next section that the probability of collision is increased with increasing the size of the family $\mathcal{G}$. The scheme of image copyright descriptor calculation is shown on Figure 4.

The resulting L locality sensitive hashes are used as copyrighted content details in the *Image Index Items* and published to the blockchain along with link to the *Authority Item*.

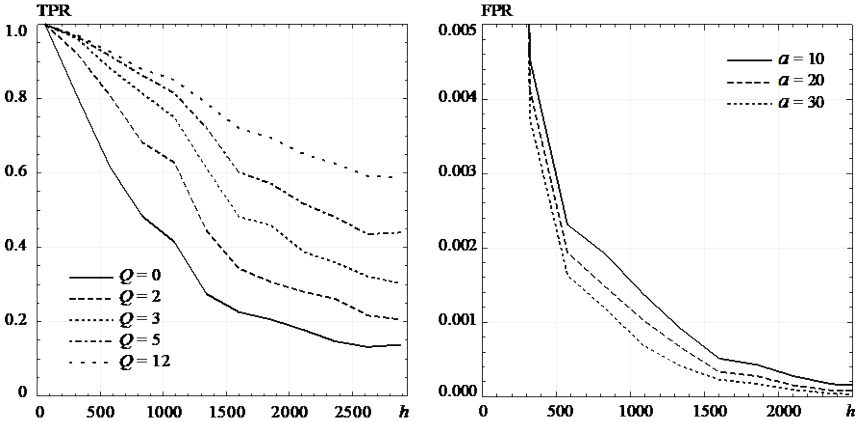


Fig. 5. Dependencies of copyright retrieving TNR and FPR against the number h of hash functions f in the family $\mathcal{F}$ for different JPEG compression quality Q and content modification area size a.

As honestly nodes are always in the consensus about already added blocks, then each of them contains the same set of *Image Index Items*. Let each honest node use hash tables [16] in order to store *Image Index Items* data on its local database. If the *Image Index Items* are stored effectively by each node, then it is possible to retrieve the copyright owner for the particular image in constant time $O(L)$ using L hash tables in the most cases. As it was mentioned before, image contents are assumed to be equal if there is at least one collision in theirs LSHs. This means that duplicate image will be rejected by miners even if there was some content-preserving modifications. In the next section the effectiveness of the proposed scheme will be evaluated.

5. Experimental results

First of all, it is necessary to investigate the sensitivity of the copyright retrieving system to content-preserving manipulations. We have chosen JPEG compression as the widely used content-preserving manipulation for our experiments. We suppose that our copyright retrieving algorithm is robust to such compression if there is at least one collision of LSH hashes calculated before and after compression. We have selected 200 different 512×512 DI having varied content, textures and so forth. Then we compress every of these images by means of JPEG with quality factor $Q \in [0, 12]$ and added to the testing image base.

The property of significant changing for another image content is another requirement which was formulated for image feature vector. In other words, it is necessary to minimize influence of other copyrights which could be possibly retrieved along with needed. In order to verify such opportunity for the proposed blockchain system we arranged the following experiment. It was selected a truly randomly square $(a\times a)$-pixel areas and inside of these areas it was chosen truly random luminance of pixels. The number of these areas was taken at least 50 for each of the images and the number of different typical images as 100. Images with slightly different content were added to the

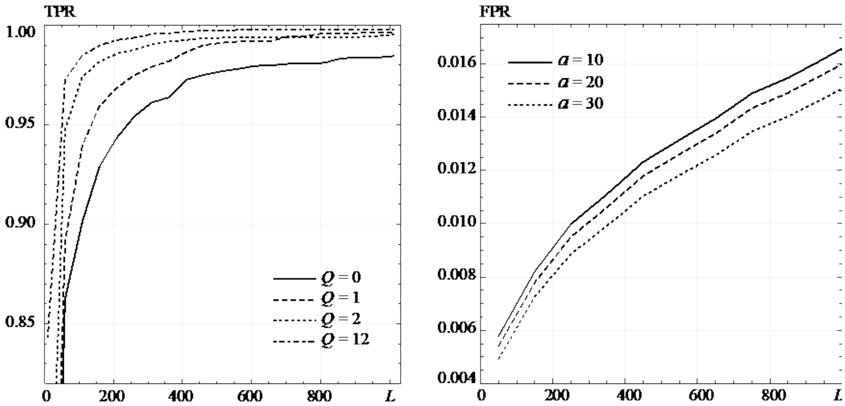


Fig. 6. Dependencies of copyright retrieving TNR and FPR against the length *L* of image feature vector for different JPEG compression quality *Q* and content modification area size *a*.

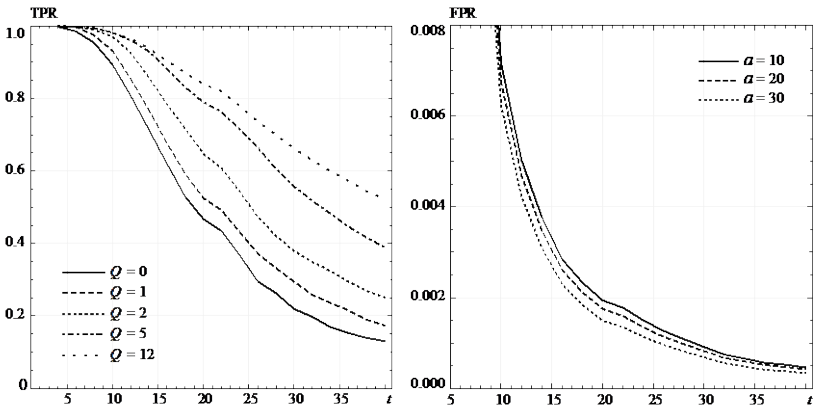


Fig. 7. Dependencies of copyright retrieving TNR and FPR against the number *t* of randomly chosen hash functions *f* concatenated to each of the LSH functions *g* for different JPEG compression quality *Q* and content modification area size *a*.

testing base. It is assumed that two images have different content if there are no LSH hash collisions.

In order to optimize parameters of the proposed system the following value ranges were selected. For the number *h* of hash functions *f* in the family $\mathcal{F}$ the values $h \in [0, 3000]$ were chosen. The results of evaluation are presented at Figure 5, where the dependencies of *True Positive Rate* (TPR) is shown as a function of parameter *h* depending on JPEG compression quality factor *Q* and *False Positive Rate* (FPR) depending on the content modification size *a*. The similar experiments were performed for the other parameters: the length of image feature vector $L \in [0, 800]$ on Figure 6, the number $t \in [0, 40]$ of hash functions *f* concatenated in each LSH function *g* on Figure 7 and the parameter $w \in [0, 16]$ from (7) on Figure 8.

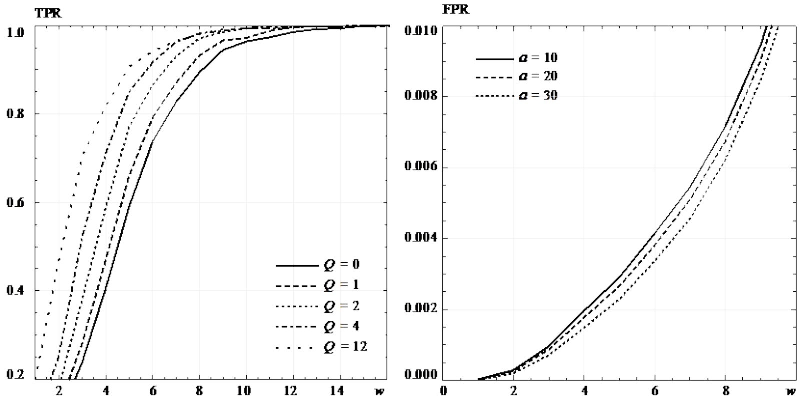


Fig. 8. Dependencies of copyright retrieving TNR and FPR against the parameter *w* for different JPEG compression quality *Q* and content modification area size *a*.

We can see on Figures 5-8 that the greater are *w*, *L*, 1/*h*, 1/*t*, the TPR tends to be close to 1, i. e. the better copyright retrieving opportunity by given image content. However, the FPR mostly has the inverse dependencies. Thus, the proposed system requires optimizing its parameters. During experiments it was derived that the following parameter values have good trade-off between retrieving ability and skipping the copyrights with different contents: $w = 7$, $b = 4$, $h = 320$, $L = 380$.

6. Conclusion

The article introduces image copyrights blockchain based on robust image feature vector and locality-sensitive hashing. The proposed block structure allows image author to publish the image content descriptors without any risk of being stolen by other members of the blockchain. This is achieved by the two-step copyright publishing algorithm. Firstly, only *Authority Item* which contains blind signature of LSH is published. Then *Image Index Item* with blinding factor as well as LSH hashes are published in the next blocks.

The robust image feature vector calculation is based on image CFDs which are used as an image content descriptor. CFDs have good content selective ability and can be easily computed. In order to make copyright retrieving scheme scalable the LSH algorithm is used to perform probabilistic search in the blockchain. The experiments show that optimized LSH and image feature vector parameters give acceptable values of FNR and keep TPR equal to 1.

The proposed image copyright blockchain algorithm can be easily integrated into existing systems such as Ethereum. The goal of such integration would be the reduction of expenses which is necessary for building up the network. Currently most of image copyright ledgers belong to the small amount of trusted authorities. If the proposed copyright ledger implemented, designers and photographers would never need to trust a third-party in order to safe their image copyrights.

Acknowledgments

Author thanks professor V. Korzhik for his everyday kind attention, help in investigations and fruitful discussions.

References

Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Available at: http://bitcoin.org/bitcoin.pdf, 2008.

Diksha Gupta, Jared Saia, and Maxwell Young. Proof of work without all the work. In Proceedings of the 19th International Conference on Distributed Computing and Net-working, ICDCN'18, pages 6:1-6:10, New York, NY, USA, 2018. ACM.

Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]y. SIGMETRICS Perform. Eval. Rev., 42(3):34-37, December 2014.

Junlin Ouyang, Xingzi Wen, Jianxun Liu, and Jinjun Chen. Robust hashing based on quaternion zernike moments for image authentication. ACM Trans. Multimedia Com-put. Commun. Appl., 12(4s):63:1-63:13, September 2016.

Victoria J. Hodge, John Eakins, and James Austin. Inducing a perceptual relevance shape classier. In Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR'07, pages 138-145, New York, NY, USA, 2007. ACM.

Noureddine Abbadeni. Content representation and similarity matching for texture-based image retrieval. In Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR '03, pages 63{70, New York, NY, USA, 2003. ACM.

Flavio Chierichetti and Ravi Kumar. Lsh-preserving functions and their applications. J. ACM, 62(5):33:1-33:25, November 2015.

Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(11) 1877-1890, Nov 2008.

C. Kong, M. Gao, W. Qian, M. Zhou, X. Gong, R. Zhang, and A. Zhou. Acid encountering the cap theorem: Two bank case studies. In 2015 12th Web Information System and Application Conference (WISA), pages 235-240, Sept 2015.

S. Babu, R. Abdulhammed, K. Elleithy, and S. Babu. Blind digital signature schemes with four particle entanglement states. In 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT), pages 1-6, May 2017.

Y. N. Aung and T. Tantidham. Review of ethereum: Smart home case study. In 2017 2nd International Conference on Information Technology (INCIT), pages 1-4, Nov 2017.

David Eberly. Derivative approximation by finite differences. Technical report, 2008. Neil Anthony Dodgson. Image resampling. Technical Report UCAM-CL-TR-261, University of Cambridge, Computer Laboratory, 1992.

Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the Twentieth Annual Symposium on Computational Geometry, SCG '04, pages 253-262, New York, NY, USA, 2004. ACM.

Ken Tokoro, Kazuaki Yamaguchi, and Sumio Masuda. Improvements of tlaesa nearest neighbour search algorithm and extension to approximation search. In Proceedings of the 29th Australasian Computer Science Conference - Volume 48, ACSC'06, pages 77-83, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

Sailesh Kumar and Patrick Crowley. Segmented hash: An efficient hash table implementation for high performance networking subsystems. In Proceedings of the 2005 ACM Symposium on Architecture for Networking and Communications Systems, ANCS'05, pages 91-103, New York, NY, USA, 2005. ACM.