Hybrid Hashing Method For Similar Vehicle Image Search

A.Fedotov[1], K.Tasov[2]
[1]CitySoft Labs, Moscow, Russia
[2]Bauman Moscow State Technical University, Russia
andrfedotov@gmail.com

***Abstract*— The novel hybrid method of a hash image calculation that can be applied in a search for similar vehicle images is proposed in this paper. The main novelty of the method described herein is the combination of two hashing types: the visual and semantic hash of the image. The method is based on SIFT and DCT algorithms. We use frontal vehicle images to test the method accuracy. The experimental results indicate that the proposed algorithm has the practical application of image search in the vehicle identification systems based on license plate recognition. We show that method is a novel in this area. The proposed method is also applicable for use in other problem domains.**

***Index Terms*— Feature Point; Image Search; Perceptual Image Hashing; RANSAC; SIFT descriptor; Vehicle.**

I. Introduction

The sheer number of vehicles on the streets requires the complex collaboration of all participants in the traffic. Unfortunately, traffic statistics in megacities show that many vehicles are stolen almost every day. In order to combat this criminal activity, many cities have implemented vehicle identification systems based on license plate recognition, which rely on extensive databases of license plate images. Manually processing such huge dataset is impossible. For different reasons such as replacement or removing of a license plate from a car, vehicle search by license plate is not possible in this database. Therefore, we need to provide another kind of search that relies on vehicle image. The method must be accurate and fast in order to have practical application.

We have access to various cameras sending the frontal grayscale images of the vehicles. There are two typical vehicle images: images with a horizontal bottom black strip with detailed information, such as capture date, street, and vehicle velocity, and images without the black strip.

In this work, we describe the method that combines the advantages of the perceptual (visual) and semantic hashing to refine image search.

The remainder of this paper is organized as follows. Section 2 provides an overview of related research and approaches. Section 3 describes our method in more detail. Section 4 provides experimental results of the proposed method, along with performance analysis. Finally, some conclusions about the method are made in Section 5 before offering some suggestions for future work.

II. Related work

Similar image retrieval is a well-known problem in computer vision, as it is encountered in, for example, web image search, photo labeling, mobile geolocation, etc. The problem of image search is often solved by content-based image retrieval (CBIR) systems.

Image hashing eliminates the need for exhaustive searches, decreasing computational complexity.

Our requirements include: compact hash representation (short hash size), the fast comparison between hash vectors, preserving the semantic content of the image in the hash, and ignoring the noise in the images.

We roughly separate hashing methods into three categories: the simplest methods, perceptual hashing methods, and feature-based methods, which are described elsewhere [1]. In addition, some methods are based on learning using labeled training data. We do not consider these approaches because the data labeling process is costly and overly complex for real-time handling.

The simplest methods based on the extraction of primitive features, such as colour, texture, edge histograms, and shape of image elements [2]. There are examples of implementations in the existing algorithms[3][4][5]. The GIST algorithm[6] and algorithms using discrete Cosine transform (DCT), such as pHash algorithm, exhibit one of the good characteristics. However, these algorithms do not take into consideration the semantic part of the image. Moreover, GIST descriptor has the size of 1280 float numbers. In our case, such size is excessive.

The DCT expresses a function or signal (a sequence of finitely many data points) in terms of a sum of sinusoids with different frequencies and amplitudes. The DCT uses only cosine functions, while, e.g., the discrete Fourier transform (DFT) uses both cosines and sines. There are eight different standard variations of the DCT. Various properties of the DCT can be utilized to create perceptual image hash functions. Low-frequency DCT coefficients of an image are mostly stable under image manipulations because most of the signal information tends to be concentrated in a few low-frequency components of the DCT[7]. We implemented the DCT-based algorithm (pHash) before developing the proposed method. The results are shown in the Results section.

A vehicle with the same license plate may be captured with a different view (projective distortion), brightness, and noise. These primitive features are non-stable to these distortions, and cannot be used for hashing.

Algorithms of perceptual hashing allow representing the entire image content in a file of small size. There are different algorithms, such as Locality Sensitive Hashing (LSH [8]), Spectral Hash (SH [9]), methods based on Fast Johnson-Lindenstrauss Transform (FJLT [10]) and others. These algorithms use pseudo randomization techniques for hashing. SH achieves better performance than LSH. FJLT [10] shares the low distortion characteristics of a random projection process. These algorithms are more robust to distortions compared to algorithms using primitive features but still cannot store semantic part of the image.

LSH is applicable for hashing high-dimensional data, and its main disadvantage stems from the inefficiency of the hash codes. Since the hash functions in LSH are randomly generated and independent of the data, it is not very efficient. Usually, it needs long codes in each hash table to achieve an acceptable accuracy [11].

Semantic hashing utilizes image content applicable to a specific domain. Major semantic hashing methods apply high-level characteristics of image such descriptors of feature points and provide the most stable results.

There are several semantic hash algorithms, such as Radial-Shape Context Hashing (RSCH), Angular Shape Context Hashing (ASCH) [12], and Bilinear Projection-based Binary Codes (BPBC) [13].

In our previous work [14], we showed that RSCH and ASCH are applicable for decision-making pertaining to our problem. However, both are insufficient for achieving satisfactory results. Therefore, we implemented our improved method for resolving the problem domain.

III. The proposed method

We address a specific problem domain: the image depicts the frontal part of the vehicle. We apply feature-point extraction to the images. First, we cut off the horizontal strip and localize the bounding box of the vehicle using the mirror symmetry. Second, we use the extracted vehicle feature points obtained in Step 1. Matching of the vehicle images is performed with good results using SIFT descriptors.

In addition, our aim is storing the semantic part of the image and achieving fast search. Therefore, we utilize feature-based hashing methods to generate compact binary hash codes. Previous research [14] has shown that SIFT descriptors yield stable results when applied to vehicle detection.

The feature-based methods require long computation time in case of the large data set, but they allow to compute semantic similarity between images.

When building an image index, two features must be taken into consideration: (a) it should be based on the semantic similarity and (b) should ensure a fast search of similar images in the database. This can be achieved by using a hash function for images, which is based on some appearance criteria. The idea of hashing is to compute a vector that has fewer dimensions in comparison with the original image, but retains its semantics (1):

$$y = H(x) = [h_1(x), h_2(x), \dots, h_k(x)]. \qquad (1)$$

Strictly, the desirable properties of a hash function H can be defined as follows: perceptual robustness, uniqueness, compactness, and pair-wise independence for perceptually different images [15].

A typical vehicle image contains approximately 200 feature points. Standard approach requires calculating the hash vector for every feature descriptor in order to determine the image hash. The proposed method overcomes these disadvantages as follows.

The proposed method starts with image preprocessing. After that, we calculate the float hash vector using the descriptors of the feature points before finally calculating the binary hash (see Figure 1).

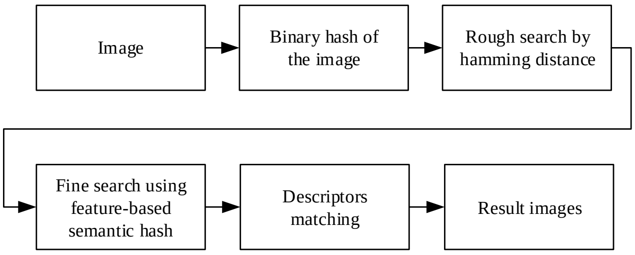
Image
Binary hash of the image
Rough search by hamming distance
Fine search using feature-based semantic hash
Descriptors matching
Result images

Figure 1: Pipeline of binary hash formation

Image preprocessing is performed in two phases.

First, the horizontal bottom black strip with detailed information is localized if it exists. It is determined by means of detection of continuous horizontal black line with the threshold for black colour. Such image has two typical horizontal lines in the bottom part.

Second, the invariant descriptors are calculated. We chose SIFT (Scale Invariant Feature Transform) [16] descriptors because previous research [14] demonstrated that such descriptors give stable results when used in vehicle detection.

Further, SIFT descriptors are used for their multiple orthogonal projections into orthogonal hyperplanes [17] and binary hash results are obtained using DCT binarization.

Let $D_{n \times m}$ denote the image descriptor matrix, where n is the number of the feature points and, m is the descriptor length. n depends on the image content, m is a fixed value depending on the detector algorithm (for example $m = 128$ for SIFT, $m = 64$ for SURF). Let k be the number of hyperplanes. It is also a trade-off between low size and high performance. Hash is determined by the mapping $\mathbb{R}^{n \times m} \to \mathbb{R}^{1 \times km}$. Let hyperspace be given by the orthogonal unit vector $\widehat{u_l} \in \mathbb{R}^n, l \in \{1, \dots, k\}$. Then, every hash vector component $h_l \in \mathbb{R}$ can be calculated as follows:

$$h_l = \bigoplus_{i = 1}^{m} \left(\sum_{j=1}^{n} D(j,i)\, \widehat{u_l}(j) \right), \qquad (2)$$

where $\oplus$ is the concatenation operator, $D(j,i)$ – (j,i) element of the image descriptor matrix. Hence, the final signature H is defined as:

$$H = \bigoplus_{l = 1}^{k} h_l. \qquad (3)$$

Initialization of every $\widehat{u_l}, l \in \{1, \dots, k\}$ occurs by the pseudorandom generation of k mutually orthogonal unit hyperplanes: $\left((\widehat{u_1} \perp \widehat{u_2}) \perp \cdots \perp \hat{u}_{k-1}\right) \perp \widehat{u_k}$.

The experiments illustrate that the appropriate accuracy can be achieved when $k = 3$ [17]. We conduct our initial experiment as a means of determining the appropriate k for our application domain. As a result the hash vector has a length of $W = 384$ float numbers for all SIFT image descriptors. Using projection yields the following advantage: images with similar descriptor matrices will result in similar signatures according to the projection properties.

In line with our requirements, our aim is to reduce the vector size and perform a fast binary search using Hamming distance. Therefore, we apply perceptual hashing using binarization by DCT coefficients:

1) Calculate sequence of DCT coefficients from the float vector H;

2) Calculate $mean$ of the sequence;

3) Take the first T elements ($T \ll W \ll nm$) of the sequence and binarize it: $b_i = 1$ if $c_i > mean$, $b_i = 0$ otherwise, $i = \{1, \ldots, T\}$. Therefore, the final binary signature B is calculated as:

$$B = \bigoplus_{i=1}^{T} b_i. \tag{4}$$

For the initial experiment, we take T = 64, which is the same as an unsigned long type (8 bytes).

Image search is performed in reverse order.

Our method for image search uses the hierarchical approach, thus eliminating a significant amount of unnecessary information (see Figure 2).

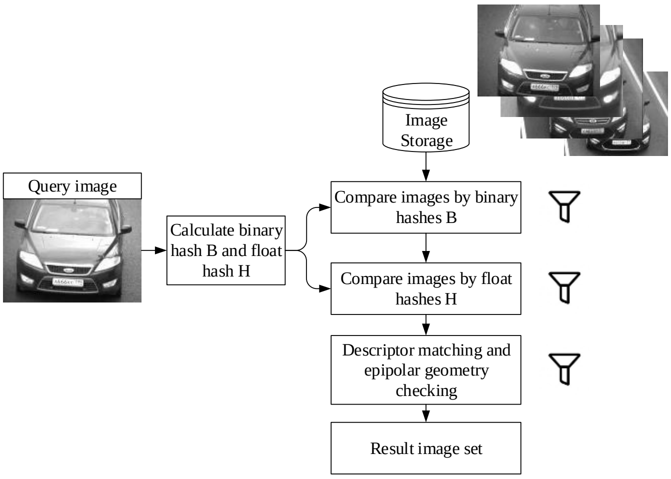
Image Storage
Query image
Calculate binary hash B and float hash H
Compare images by binary hashes B
Compare images by float hashes H
Descriptor matching and epipolar geometry checking
Result image set

Figure 2: Flowchart of similar image search

4) Coarse search using binary hash vector B using fast Hamming distance (HD) comparison: $Q_C = \{I_i \mid HD(H(Q), H_i(I)) \leq d, i = \{1, \ldots, S\}\}$, where S – size of the initial set of the images;

5) Fine search using feature-based semantic hash H based on the L_1 norm. As shown elsewhere [17] , using L_1 norm yields marginally better results compared to the L_2 norm. Thus, we reorder hashes by the norm: $Q_F = \{I_j \mid sorted\ by\ L_1(H(Q), H_j(Q_C)), j = \{1, \ldots, S\}\}$, $|Q_F| = |Q_C|$;

6) Matching images using feature point descriptors with verification by RANSAC (Random Sample Consensus) algorithm (set Q_M: $|Q_M| \leq |Q_F|$). Thus, Q_M contains the result set of similar images for the query image sorted in descending order of inlier count. This step allows us to perform epipolar geometry checking.

Upon completion of the aforementioned steps, the set Q_M contains the resulting image set.

IV. THE RESULTS

We evaluate the method efficiency using an image dataset obtained from the traffic management centre of the Moscow Government. Hence, we utilize grayscale images for validating the experimental results. The mean count of feature points is equal to approximately 200 features.

Our samples of $M = 5000$ images include 10 vehicle models (500 images per model): Chevrolet Cruze, Mitsubishi Lancer Evolution, Ford Focus I, Ford Focus II, Ford Focus III, Ford Mondeo III, Ford Mondeo IV, Lada Samara-2 VAZ-2114, Lada Priora (Lada 2170), and Mercedes-Benz W220.

For quality assessment of our method, we calculated precision and recall, as these are well-known statistical metrics. We find similar images for all vehicle models.

In our case, precision P is the ratio of correct images to the total number of images (in other words, it determines how precise the recall is):

$$P = \frac{TP}{TP + FP}. \tag{5}$$

Recall R is the ratio of the number of correctly retrieved images to the total number of all correct images:

$$R = \frac{TP}{TP + FN}. \tag{6}$$

Here, TP (true positives) is the number of correctly identified images in the positive set, i.e., count of the correct images determined as correct. FP (false positive) means that, although the image is incorrect, the method incorrectly determined it as a correct image. Similarly, FN (false negative) is the number of incorrectly identified images in the positive set. Recall is also called true positive rate.

For the purpose of the present study, recall is the preferred term, and our aim is to obtain a high value of recall. Recall reflects the ability to identify all relevant images in a given set. We compare results of our method with those yielded by the pHash algorithm. We use perceptual hashing for our purposes, focusing on vehicles that are similar in colour and shape (perceptual hashing) and apply it for semantic hashing.

We determine initial parameters of the method using our original developed technique from our previous research [14].

We take each image for each of 10 vehicle models when calculating the metrics. Then we calculate metrics from received data for average hash length T ($T \in \{8,16,32,64,128,256,512\}$). Summary of the results is shown in Table 1. Here, the old method denotes the pHash algorithm, and the new method is the proposed method.

As shown in Table 1, pHash metric values are too low and yield unstable values, while our method improves and stabilizes the precision and recall values.

We also compared our method with the results obtained from popular and well-known image search engines, such as Google Images and Tineye. We took the first 100 images in the result set from Google Images for assessment because these were deemed the most relevant. We performed the same experiment as that used for compiling Table 1. However, Google Images did not give any similar images for vehicle models (except two images for Chevrolet Cruze and Ford Focus I). Tineye found no images. Only one true positive image was found for Chevrolet Cruze. Thus, precision is close to 0 for these image search engines, and the results obtained from these image search engines are not appropriate for our purposes.

Table 1
The metric values for each vehicle model

Method	Model number	1	2	3	4	5
Old	Precision	0.1352	0.0341	0.2142	0.0888	0.0922
	Recall	0.1526	0.0546	0.0546	0.1656	0.0783
New	Precision	0.2371	0.2342	0.2212	0.3451	0.3114
	Recall	0.8894	0.9254	0.6035	0,6977	0.8801
Method	Model number	6	7	8	9	10
Old	Precision	0.0642	0.2688	0.0515	0.2391	0.3161
	Recall	0.0772	0.0564	0.1528	0.0528	0.1094
New	Precision	0.4307	0.2951	0.2327	0.2894	0.2787
	Recall	0.9244	0.8936	0.8350	0.9495	0.8935

Note. Model numbers: 1 – Chevrolet Cruze, 2 – Mitsubish Lancer Evolution, 3 – Ford Focus I, 4 – Ford Focus II, 5 - Ford Focus III, 6 – Ford Mondeo III, 7 – Ford Mondeo IV, – Lada Samara-2 VAZ-2114, 9 – Lada Priora (Lada 2170) 10 – Mercedes-Benz W220.

We suppose that the results yielded by image search engines are inappropriate because these are general-purpose applications. In our case, we deal with a specific domain. Nonetheless, in terms of computational time, they provide fast results (each search query by image was performed in under 1 second).

Moreover, we changed the bit length of hash and displayed the obtained results in Figure 3. The best results are achieved when the number of bits is $T = 128$, which is expected, as the dimension of SIFT descriptor is also 128 elements. Descriptors of Ford Mondeo car models are correlated as these are visually similar vehicles. In addition, we applied pHash algorithm (marked in Figure 3 as Old). As can be seen from Figure 3, the recall of the old method has low values, confirming that the DCT-based algorithm is suitable for visual similarity by colour and shape.

Overall, the average precision of the proposed method is equal to 0.2876, and the recall is equal to 0.8492. Note that values of recall and precision have small variance, confirming that our method provides predictable results.

We varied projection number k when using the method provided by P. L. N. Carrasco, F. Bonin-Font, G. Oliver-Codina [17]. These authors note that appropriate projection number is $k = 3$ for their purposes. We explored how changing the projection number influences the recall and precision. The results are shown in Figure 3. The best results are achieved with 32 projections. We changed the initial orthogonal projections and obtained the same results. Thus, we assume that the results depend on our domain specifics. Consequently, the increasing of the projection number leads to redundant information. Moreover, we determined that the results are inappropriate when the projection number k is not equal to the power of two. It is noted that in case of $k = 2$ the recall reaches appropriate values almost in accordance with research performed by P. L. N. Carrasco, F. Bonin-Font, G. Oliver-Codina [17] (these authors claim that selecting $k = 3$ provides a good trade-off between signature size and accuracy in the detection of loop closure). In case of increasing k above 64, we obtain decreasing measures. This finding is attributed to information noise, because we store redundant data (in addition, the mean of feature points per image is equal to 200). In our case, $k = 2$ is preferable because of the vector size $W = 256$ (1024 bytes of memory) is less than $W = 4096$ (16384 bytes of memory) with $k = 32$.

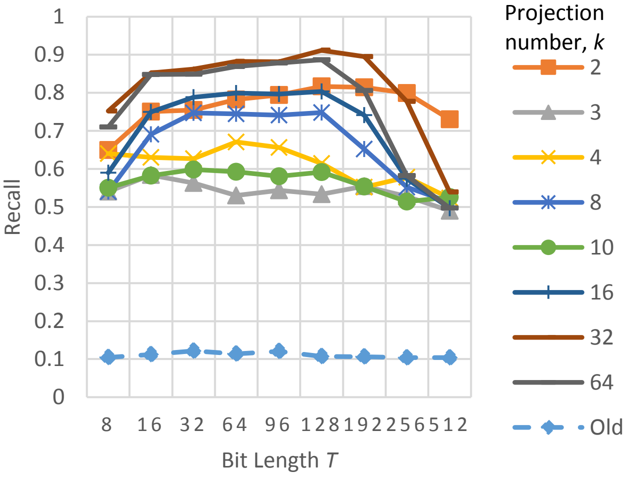

Figure 3: Recall curve

We also performed an experiment using projection, without using binary hash code. The average metric results for all evaluated vehicle models are shown in Table 2.

Table 2
The metric values depending on the projection number

Projection number k	1	2	3	4	5	8
Recall	0.6523	**0.8771**	0.6430	0.6954	0.6295	0.7186
Precision	0.3232	**0.3258**	0.2131	0.2742	0.2410	0.2947
Projection number k	10	16	32	64	128	
Recall	0.6951	0.8829	**0.9379**	0.8273	0.7912	
Precision	0.2831	0.3101	**0.3836**	0.3291	0.2247	

While the metrics in Table 2 are greater than metrics using the binary hash code, higher values are reached with the projection number set to $k = 32$. Moreover, for $k = 2$, the measures reach local maximum.

Unacceptably low precision is explained by the fact that we use perceptual hashing methods that rapidly lose semantic similarity. In construct, the pHash algorithm shows unacceptable metric values for search by vehicle models. Still, its application in visual search (search by colour, shape) is possible.

Therefore, in our case, we reached recall $R = 91{,}11\%$ with $k = 32$ projections (and recall $R = 87{,}71\%$ with $k = 2$ projections) and $T = 128$ bits.

V. Conclusion and future work

We proposed the hybrid method of a hash image calculation for the search for similar vehicle images.

We determined the method parameters and obtained appropriate results. We use our original developed technique for determination of initial parameters of the method.

Our SIFT-based hashing method is applicable for image search using binary codes in the vehicle identification systems based on license plate recognition.

We use initial parameters of the method using our original developed technique

We reach robust recall values using SIFT descriptors and RANSAC verification in the last stage of the proposed method. We further showed that DCT-based pHash algorithm is not applicable for semantic hashing, as confirmed by metrics given in the experimental results. Our method is characterized by increased precision and recall values; in particular, recall values reach the acceptable level.

Although Google Images search service uses state-of-the-art algorithms and extensive resources, we could not use it for our purposes as we focus on a specific domain and cannot access personal or local image databases. Our system has no access to the Internet for image processing. The using third-party services make dependent on the Internet connection, availability, and pricing. However, the using of own method allows controlling server equipment, payload, and flexible changing.

Thus, we will continue to work on improving our method. This will include using semantic preserving hashing method, such as the state-of-the-art convolutional neural networks, and comparing the results, as well as using deep learning of binary hash codes such as those described by K. Lin, H.-F. Yang, J.-H. Hsiao, C.-S. Chen [18]. Moreover, we will research the hash functions, such as Circulant Binary Embedding (CBE, [19]), Iterative quantization (ITQ,[20]), Spectral Hashing (SH,[9]), and others for applying to our domain. Another direction for future research is developing special SQL database tables containing vehicle images for effective image search. Our planned future work includes the integration of the proposed method into a vehicle handling system.

References

[1] K. Grauman, R. Fergus. Learning Binary Hash Codes for Large-Scale Image Search. Chapter Machine Learning for Computer Vision, Volume 411 of the series Studies in Computational Intelligence, pp. 49-87, 2013.

[2] N. Goyal, N. Singh. A Review on Different Content Based Image Retrieval Techniques Using High Level Semantic Features. International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 2, Issue 7, pp. 4933-4938, 2014.

[3] M. Schneider, S.-F. Chang. A robust content based digital signature for image authentication. International Conference on Image Processing, Vol. III of III, pp. 227-231, 1996.

[4] C. Kailasanathan, R. S. Naini. Image authentication surviving acceptable modifications using statistical measures and k-mean segmentation, IEEE-EURASIP Work. Nonlinear Sig. and Image Processing, Vol. 1, 2001.

[5] R. Venkatesan, S. M. Koon, M. H. Jakubowski, P. Moulin. Robust image hashing. Proc. IEEE Conference on Image Processing, Vol. 3, pp. 664–666, 2000.

[6] Oliva, A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. International Journal of Computer Vision (IJCV), Vol. 42(3), pp. 145-175, 2001.

[7] Zauner. Implementation and Benchmarking of Perceptual Image Hash Functions, Master's thesis, Upper Austria University of Applied Sciences, Hagenberg Campus, 2010.

[8] P. Indyk, R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. Proceedings of the thirtieth annual ACM symposium on Theory of computing, pp. 604-613, 1998.

[9] Y. Weiss, A. Torralba, R. Fergus. Spectral Hashing. Advances in Neural Information Processing Systems 21, pp. 1753-1760, 2008.

[10] X. Lv, Z. J. Wang. An Extended Image Hashing Concept: Content-Based Fingerprinting Using FJLT. EURASIP Journal on Information Security, Volume 2009, Issue 1, 2009.

[11] J. He, R. Radhakrishnan, S.-F. Chang, C. Bauer. Compact hashing with joint optimization of search accuracy and time. Proceedings of Computer Vision and Pattern Recognition, pp. 753-760, 2011.

[12] X. Lv, Z. J. Wang. Perceptual Image Hashing Based on Shape Contexts and Local Feature Points. IEEE Transactions on Information Forensics and Security, Vol. 7, Issue 3, pp. 1081-1093, 2012.

[13] Y. Gong, S. Kumar, H. A. Rowley, S. Lazebnik. Learning Binary Codes for High-Dimensional Data Using Bilinear Projections. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 484-491, 2013.

[14] Fedotov, K. Tassov. Index formation for similar images search of vehicles. Engineer Magazine: science and innovation, No 6 (18). Bauman Moscow State Technical University, 2013.

[15] R. Davarzani, S. Mozaffari, K. Yaghmaie. Perceptual image hashing using center-symmetric local binary patterns. Multimedia Tools and Applications, Vol. 75, Issue 8, pp. 4639-4667, 2016.

[16] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision, Vol. 60, Issue 2, pp. 91-110, 2004.

[17] P. L. N. Carrasco, F. Bonin-Font, G. Oliver-Codina. Global Image Signature for Visual Loop-Closure Detection. Autonomous Robots, pp. 1-15, 2015.

[18] K. Lin, H.-F. Yang, J.-H. Hsiao, C.-S. Chen. Deep Learning of Binary Hash Codes for Fast Image Retrieval. EEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 27-35, 2015.

[19] F. X. Yu, S. Kumar, Y. Gong, S.-F. Chang. Circulant Binary Embedding. Proceedings of The 31st International Conference on Machine Learning, Vol. 32, pp. 946-954, 2014.

[20] Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval. IEEE Transactions on Software Engineering, Vol. 35, Issue 12, pp. 2916-2929, 2013.