DOI: https://doi.org/10.15276/aait.05.2022.10
UDC 004.932.2

Methodology for image retrieval based on binary space partitioning and perceptual image hashing

Mykola A. Hodovychenko[1)]
ORCID: https://orcid.org/0000-0001-5422-3048; nick.godov@gmail.com. Scopus Author ID: 57188700773
Svitlana G. Antoshchuk[1)]
ORCID: https://orcid.org/0000-0002-9346-145X; asg@opu.ua. Scopus Author ID: 8393582500
Varvara I. Kuvaieva [1)]
ORCID: https://orcid.org/0000-0002-9350-1108; vkuvayeva@gmail.com. Scopus Author ID: 57203618134
[1)] Odessa National Polytechnic University, 1, Shevchenko Ave. Odessa, 65044, Ukraine

ABSTRACT

The paper focuses on the content-based image retrieval systems building. The main challenges in the construction of such systems are considered, the components of such systems are reviewed, and a brief overview of the main methods and techniques that have been used in this area to implement the main components of image search systems is given. As one of the options for solving such a problem, an image retrieve methodology based on the binary space partitioning method and the perceptual hashing method is proposed. Space binary partition trees are a data structures obtained as follows: the space is partitioned by a hyperplane into two half-spaces, and then each half-space is recursively partitioned until each node contains only a trivial part of the input features. Perceptual hashing algorithms make it possible to represent an image as a 64-bit hash value, with similar images represented by similar hash values. As a metric for determining the distance between hash values, the Hamming distance is used, this counts the number of distinct bits. To organize the base of hash values, a vp-tree is used, which is an implementation of the binary space partitioning structure. For the experimental study of the methodology, the Caltech-256 data set was used, which contains 30607 images divided into 256 categories, the Difference Hash, P-Hash and Wavelet Hash algorithms were used as perceptual hashing algorithms, the study was carried out in the Google Colab environment. As part of an experimental study, the robustness of hashing algorithms to modification, compression, blurring, noise, and image rotation was examined. In addition, a study was made of the process of building a vp-tree and the process of searching for images in the tree. As a result of experiments, it was found that each of the hashing algorithms has its own advantages and disadvantages. So, the hashing algorithm based on the difference in adjacent pixel values in the image turned out to be the fastest, but it turned out to be not very robust to modification and image rotation. The P-Hash algorithm, based on the use of the discrete cosine transform, showed better resistance to image blurring, but turned out to be sensitive to image compression. The W-Hash algorithm based on the Haar wavelet transform made it possible to construct the most efficient tree structure and proved to be resistant to image modification and compression. The proposed technique is not recommended for use in general-purpose image retrieval systems; however, it can be useful in searching for images in specialized databases. As ways to improve the methodology, one can note the improvement of the vp-tree structure, as well as the search for a more efficient method of image representation, in addition to perceptual hashing.

Keywords: Content-based image retrieval; binary space partitioning; perceptual hashing; vantage-point tree; discrete cosine transform; discrete wavelet transform

***For citation*: Hodovychenko M. A., Antoshchuk S. G., Kuvaieva V. I. Methodology for image retrieval based on binary space partitioning and perceptual image hashing. *Applied Aspects of Information Technology.* 2022; Vol. 5 No. 2: 136–146. DOI: https://doi.org/10.15276/aait.05.2022.10.**

INTRODUCTION

With the general popularity of digital devices with built-in cameras and the rapid development of Internet technology, billions of people are online to share and view photos.

Universal access to digital photos and the Internet makes many image retrieval systems relevant. Image retrieval aims to obtain relevant visual documents for a text or visual request from a large image database.

Although the field of image search has been extensively researched since the early 90s [1], it has still attracted a lot of attention from the multimedia community in the last ten years, due to the problem of scalability and the rising of new search methods and algorithms.

Traditional image search systems organize visual data based on the images metadata, such as content descriptions, tags, and comments.

Because textual information may be inconsistent with visual content, content-based image retrieval (CBIR) rises as the best approach, and progress has been made in recent years.

There are two fundamental problems in the field of CBIR: the intention gap and the semantic gap.

Intention gap refers to the difficulties faced by the user when trying to accurately express the expected visual content. The semantic gap stems from the difficulty in describing high-level semantic

concepts using low-level visual cues [2, 3]. Bridging these gaps requires a great deal of effort on the part of both academia and industry.

From the 90s, comprehensive research was conducted on image retrieval task. In the beginning of the century, the rising of new techniques and algorithms has led to another field of study in the content-based image retrieval industry. Two works have led to significant progress in the field of visual search based on content in big image databases. The first is the appearance of invariant local visual features SIFT [4]. SIFT features have shown great descriptive and discriminatory capabilities to capture visual content in a variety of studies. They can well capture invariance to rotation conversion and scaling and are resistant to changes in lighting. The second work is the introduction of the Bag-of-Visual-Words (BoW) model [5]. Originating in the field of information search, the BoW model makes a compact image representation based on quantization of local features and easily adapts to the classic inverted file indexing structure for scalable image retrieval.

There are three major challenges in the field of retrieving content-based images:

- image representation;
- image organization;
- measuring the similarity of images.

Existing algorithms can be classified based on their contribution to these three key problems.

The problem of image representation stems from the fact that the internal problem of visual search based on content is the comparison of images. For ease of comparison, the image is converted into some function space. The motivation is to achieve implicit alignment to eliminate the influence of background and potential transformations or changes, keeping the internal visual content separate.

In fact, the problem of image representation is a fundamental problem of computer vision in the field of image comprehension. There is a saying: “An image is worth a thousand words”. However, identifying these “words” is a non-trivial task. Usually, images are represented by one or more visual features. The presentation is expected to be descriptive and discriminatory in order to distinguish between similar and dissimilar images. More importantly, it is also expected that such a description will be invariant to various transformations, such as translation, rotation, resizing, lighting change, and so on.

In multimedia search engines, the visual database is usually very large. Organizing a large database to effectively identify relevant query characteristics is a non-trivial task. Inspired by success in information retrieval, many existing content- and system-based visual search algorithms use the classic inverted file structure to index a large visual database for scalable search.

Meanwhile, some hashing-based methods for indexing in a similar perspective are also proposed. To achieve this goal, visual codebook training and quantization of functions on high-dimensional visual features with built-in spatial context are performed to further enrich the discriminatory possibilities of visual representation.

Ideally, the similarity between the images should reflect the relevance in the semantics, which, however, is complicated by the internal problem of “semantic gap”. Conventionally, the similarity of images in the field of content search is formulated on the basis of the similarity of visual features with some scheme of coefficients. In addition, the formulation of image similarity in existing algorithms can also be considered as a comparison of different matching kernels [6].

Thus, **the purpose of this study** is to develop a methodology for content-based image retrieval that addresses three key problems of image searching: image representation, image organization and measuring the similarity of images.

1. RELATED WORK

Consider the works that are devoted to various aspects of CBIR systems building.

Search request. The beginning of any CBIR system is a user request to search for an image. The quality of the query has a significant impact on the search results.

The most intuitive query formation is an image query. That is, the user has an example image, and he would like to get more or better images with approximately the same or similar meaning.

Thus, most of the work in the field of CBIR that focuses on the search query question explores the query format in the form of a sample image [5, 7], [8].

Also, it is worth mentioning the works that suggest using the sketch provided by the user as the request format [9, 10]. This query format can be effective when searching for certain types of images (such as patterns or clip art).

Image representation. A key challenge in the field of CBIR is how to efficiently assess the similarity of images to each other. Direct pixel-by-pixel comparison of images is inefficient, so the developers of CBIR systems need to choose such visual features that will effectively solve the problem of indexing and comparing images in a reasonable time.

The most important issues in image representation are feature extraction, visual codebook genera-

tion, spatial context embedding, and feature aggregation.

Traditionally, local and global visual features are distinguished. Initially, global features were used for CBIR systems to describe an image in terms of color [11, 12], shape [13, 14], texture [15, 16], and structure [17, 18].

With the advent of SIFT features [4], local features have become a popular tool for image representation [19, 20], [21],

Recently, the most popular is the search for visual features using deep neural networks [22, 23], [24]. This method is used both to search for global features and to search for local ones [25].

To ensure the compactness and speed of operation of CBIR systems, it is necessary to form a visual code book from the extracted features - a set of words that characterize a particular feature. For this, the Bag of Words [5], VLAD [26], or the Fisher vector [27] model can be used.

The efficiency of CBIR systems can be increased by taking into account information about the spatial context of the image [28]. For example, some feature extraction algorithms attempt to take into account the spatial relationship between local image features [29, 30].

After the formation of the visual code book, it is necessary to quantize the features in order to match each feature with a certain word. To do this, it could be used the nearest neighbor method [31, 32] or its variant in the form of an approximate search for nearest neighbors [33].

Organization of images. Since image search time plays a key role in the success of a CBIR system, efficient organization of the image database is an important task. In the field of CBIR, as a rule, two indexing methods are used – inverted indexing [5, 34] and hashing [35].

Measurement of image similarity. In multimedia search systems, search results are sorted based on the relevance of the found images to the sample. To calculate the relevance index, methods for calculating the distance between feature vectors [36] or voting methods for matching features in vectors [37] are used.

2. METHODOLOGY

The proposed methodology is based on the use of the perceptual hashing method for generating an image feature vector, methods for comparing perceptual hash codes as a way to measure the similarity of images, and binary space partition trees as a way to organize an image database.

Consider separately each of the points of the methodology and define specific algorithms and methods for implementing these aspects.

2.1. Perceptual hashing and algorithms for its implementation

The principle of perceptual hashing of images is based on the extraction of certain stable or invariant features from an image to create a hash value. Such hash value has the following important property: two completely different images will have uncorrelated hash values, while two visually similar images (i.e., perceived by the human eye as similar) will have highly correlated hash values.

With regard to the problem of hashing images, the most popular methods are methods based on the analysis of spatial characteristics and methods based on the frequency characteristics of the image.

Popular methods based on the analysis of spatial characteristics include the hashing algorithm based on the difference or Difference Hash. Techniques based on the frequency characteristics of an image include hashing based on the discrete cosine transform or P-Hash, and hashing based on the wavelet transform or W-Hash.

The Difference Hash method calculates the difference in values between adjacent pixels in an image to generate a hash value.

The algorithm of the method can be described as follows:

1) reduce the image size to get rid of high frequencies and image details;

2) convert image to grayscale format;

3) calculate the difference between adjacent pixels;

4) build a bit string with a length of 64 bits. The bit takes the value “0” if the left pixel is greater than the right and the value “1” if the left pixel is greater than the middle value.

The P-Hash method uses a discrete cosine transform of the second type to determine the low frequencies of the image.

The algorithm of the method can be described as follows:

1) reduction of the image to the size of 32×32 pixels;

2) converting the image to grayscale;

3) applying type 2 DCT to create a 32×32 matrix. Each element of the matrix is a color value;

4) obtained 32×32 matrix is reduced to an 8×8 size. The submatrix is taken from the upper left corner;

5) calculate the average value of all elements of the resulting submatrix;

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

6) generate a binary vector, where "0" means that the corresponding element of the submatrix is less than the mean value and "1" means that the corresponding element of the submatrix is greater than the mean value.

Let $x[m], m = 0, \dots, N-1$ be a sequence of real signals of length N.

Then the type 2 discrete cosine transform will look like this:

$$\mathrm{X}[n] = \sqrt{\frac{2}{N}} * \sum_{m=0}^{N-1} \left(x[m] * \cos\left(\frac{(2m+1) * n\pi}{2N} \right) \right), \quad (1)$$

where $n = 0, \dots, N-1$.

Difference hash (1) allows to determine the relative direction of the gradient

Wavelet hashing (or W-Hash) is a frequency domain hashing method that uses a discrete wavelet transform. It is based on image analysis in the wavelet domain with the preservation of temporal information. The algorithm of the method is described in detail in [16].

2.2. Binary space partitioning trees

A binary space partitioning tree is a geometric data structure obtained using a recursive partitioning method called binary space partitioning on a set of input objects: the space is partitioned by a hyperplane into two half-spaces, then each half-space is recursively partitioned until each subproblem contains only the trivial part of the input features. The concept of binary space trees originated from the field of computer graphics in the seventies. This framework was originally designed to help efficient hidden surface removal algorithms for moving viewpoints but has since found widespread use in many areas of computational and combinatorial geometry.

In the field of image search, as a rule, two types of binary space partition trees are used: kd-tree and vp-tree. Based on the works [24, 25], within the framework of this technique, it was decided to use a vp-tree as a binary space partition tree. It is assumed that the use of the vp-tree will achieve better image search time compared to the kd-tree.

The way a vp-tree works is to partition the search space using the relative distances between the vp-point and its children. Thus, it is easy to calculate the distance between a point and the interface to which it belongs.

Let us give a formal description of the operation of a vp-tree. Let there be a metric space (S, d) and its finite subset $S_d \subset S$, which is the image base. Since the range of values of any metric can be normalized to the interval [0,1] without affecting the nearest neighbor ratio, only that metric can be considered.

It is necessary to consider the case of binary partitioning. Let some point v from the set S_d be taken as a vantage point. For some other point $p \in S_d - \{v\}$, the distances from v are calculated and the median value μ is chosen among the distances. Next, the entire data set S_d is divided into two parts according to the viewpoint and the median: a subset S_l is created, which contains points whose distances from v are less than the median value, and a subset S_r is created, which contains points whose distances from v are greater than the median value.

Let it be necessary to find the nearest neighbors for some point q, for which the distance from q is less than some threshold σ. It turns out that if $d(v,q) \le \mu - \sigma$, then it is necessary to examine only the subset S_l, and $if\ d(v,q) > \mu + \sigma$, then it is necessary to examine only the set S_r.

This observation is based on the triangle inequality: if $d(v,q) \le \mu - q$ then for each $p \in S_r$ the following will be true: $d(q,p) \ge |d(v,p) - d(v,q)| > |\mu - (\mu - \sigma)| = \sigma$, i.e. $d(p,q) > \sigma$. This means that a subset may be excluded from the search.

And if $d(v,q) > \mu + \sigma$, then for each $p \in S_l$ there is $d(q,p) > |d(v,q) - d(v,p)| > |\mu + \sigma| = \sigma$. Therefore, $d(q,p) > \sigma$ and the subset S_r can be excluded from the search. This statement is shown in Fig. 1.

In this way, one half of the search space can be effectively reduced if the following conditions are met: the power of the subset S_l is approximately equal to the power of S_r and $d(v,q)$ does not satisfy the following two-sided inequality:
$\mu - \sigma < d(v,q) \le \mu + \sigma$.

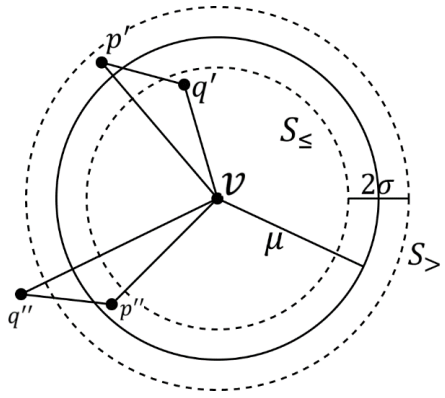


Fig. 1. **Data partitioning and search**
Source: **compiled by the authors**

2.3. Methods for comparing perceptual hash values

To compare hash values of images, its need to select a function that returns some numerical value

based on some mathematical operations. The Hamming distance and the peak of the cross-correlation function should be considered in more detail.

Hamming distance. The Hamming distance determines the number of distinct positions between two binary sequences.

Suppose that A is an alphabet of finite length, $x = x_1, \dots, x_n,\ y = y_1, \dots, y_n$ are binary sequences (vectors). The Hamming distance Δ between x and y can be defined as

$$\Delta(x, y) = \sum_{x_y \neq y_i} 1, i = 1, \dots, n. \quad (2)$$

This method of comparing hash values is used by the P-Hash method. The hash occupies 8 bytes, so the Hamming distance (2) lies in the interval $[0,64]$.

The smaller the value of Δ, the more similar the images are. To facilitate comparison, the Hamming distance can be normalized using the length of the vectors

$$\Delta_n(x, y) = \frac{1}{n} \sum_{x_i \neq y_i} 1, i = 1, \dots, n. \quad (3)$$

Normalized Hamming distance (3) is used in Simple Hash and Marr-Hildreth Operator Based Hash algorithms. The Hamming distance lies in the interval $[0,1]$ and the closer to 0, the more similar the images are.

It is necessary to consider the peak function of the cross-correlation function. We define the correlation between two signals as:

$$r_{xy}(T) = \int_{-\infty}^{+\infty} x(t) y(t + T) dt, \quad (4)$$

where $x(t)$ and $y(t)$ – are two continuous functions of real numbers.

The function $r_{xy}(T)$ determines the offset of these two signals with respect to time T. The variable T determines how far the signal is shifted to the left. If the signals $x(t)$ and $y(t)$ are different, the function (4) is called cross-correlated.

It is necessary to determine the normalized cross-correlation function (NCF).

Let x_i and y_i, where $i = 0, \dots, N-1$ – are two sequences of real numbers, and N – the length of both sequences. NCF with delay d is defined as:

$$r_d = \frac{\sum_i (x_i - m_x)(y_{i-d} - m_y)}{\sqrt{\sum_i (x_i - m_x)^2} \times \sqrt{\sum_i (y_i - m_y)^2}}, \quad (5)$$

where m_x and m_y denote the mean value for the respective sequence.

The peak of the cross-correlation function (5) is the maximum value of the r_d function that can be reached in the interval $d = 0, \dots, N$.

PCF is used to compare hash values in the Radial Variance Based Hash algorithm. The value of PCF lies within $[0,1]$. The larger its value, the more similar the images are.

2.4. Proposed image retrieve methodology algorithm

The proposed image search technique consists of two stages:

1) the stage of creating a database of images hash values;

2) the stage of searching for images on request.

The stage of creating a database can be divided into the following steps:

1) uploading a corpus of images;

2) obtaining a hash values for each image from the corpus;

3) building a vp-tree for the received hash values.

The image retrieve stage can be broken down into the following steps:

1) uploading an image as a search query;

2) calculation of the hash values for the user's image;

3) search in the vp-tree for images with a closest distances of hash values;

4) obtaining a list of similar images based on the search results in the vp-tree.

It should be noted that for step 3 at the stage of image search, either a certain number of images with minimum distances between hash codes can be specified, or a threshold value of the distance can be set and all images whose hash codes have a distance less than or equal to this threshold can be retrieved.

3. EXPERIMENTS

An experimental study of the proposed methodology was carried out in the Google Colab environment in the Colab Pro version [38]. The server accelerator Python 3 based on Google Compute Engine ((TPU)) was used as a runtime environment. The Caltech-256 Object Category Dataset [39] was used as the image dataset. This set contains 30607 images divided into 256 categories.

The hashing methods chosen were P-Hash (based on the discrete cosine transform), Difference Hash (based on the difference between adjacent pixels), and W-Hash (using the discrete wavelet transform).

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

3.1. Building a vp-tree from hash values

Let's conduct an experimental study of the first stage of the technique, which is to build a vp-tree with image hashes. The key steps at this stage are getting a collection of hashes and building a vp-tree from this collection.

Consider the operation of obtaining a collection of hashes. For each hashing method, the time to create a collection of 30607 hashes was measured. The measurement results are shown in Figure 2.

As expected, the hash time increases linearly. To determine the regression function for each hashing method, we can use the linear regression method.

For Difference Hash the regression function looks like $y = 0.004x + 1.2371$, for P-Hash the regression function looks like $y = 0.005x + 1.3960$. For W-Hash, the regression function is $y = 0.011x + 2.2947$. For all regression functions, the coefficient of determination was more than 0.99.

The fastest hashing operation is performed by the Difference Hash method, the P-Hash method works, on average, 15 % slower; the Wavelet Hash method works 168 % slower. It is obvious that the differences in the operating time are primarily due to the use of the discrete cosine transform and the discrete wavelet transform).

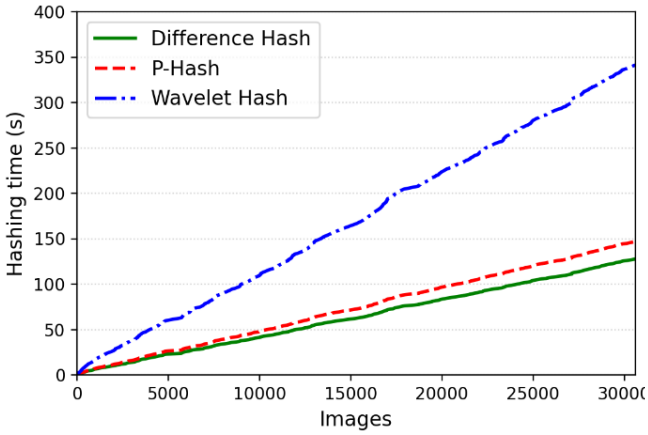


Fig. 2. **Comparison of hash times for different methods**
Source: **compiled by the authors**

Next, we need to consider the process of building a vp-tree from a collection of hash values. To do this, three vp-trees were built, each of which contains hash values obtained by a certain hashing method.

As a result of the experiments, it was revealed that when using various hashing methods, the time to build a tree does not change significantly, so the P-Hash method was chosen to illustrate the time spent on building a vp-tree.

The vp-tree creation time was measured with the number of hashes in the range from 1 to 30607 with a step of 1000 elements. The measurement results are shown in image 3.

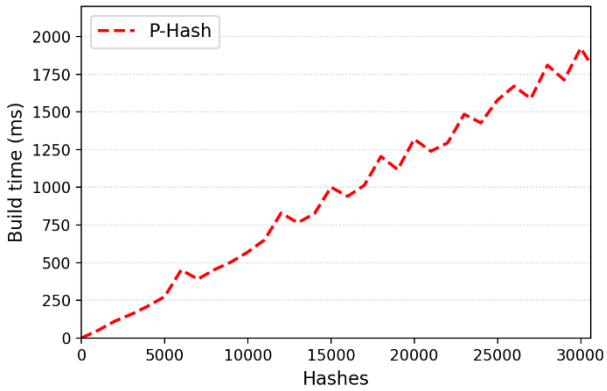


Fig. 3. **Time to build a vp-tree depending on the number of hash values**
Source: **compiled by the authors**

3.2. Image retrieve by query image

The original image used as the search query was taken from the dataset and presented in Fig. 4.

Fig. 4. **Source image for search query**
Source: **compiled by the authors**

First, we need to consider the search time for k nearest images in the vp-tree. For this experiment, k similar images were searched for the original image query, where k ranged from 1 to 30606 images. During the experiment, it was found that the hashing method used does not significantly affect the search time for k nearest images, so Fig. 5 shows a graph of the increase in search time with an increase in the number of nearest images only for the P-Hash method.

Next, we need to make sure that the search technique allows finding the same source image in the database. As a result of the experiment, all three vp-trees found an identical image in the database. It should be noted that the search in the vp-tree built using the D-Hash method took 50.57 milliseconds,

while the trees built using the P-Hash and W-Hash methods completed in 31.33 milliseconds and 20.36 milliseconds, respectively. This may indicate that the tree structure for the D-Hash method turned out to be less balanced.

Next, we need to study how well the hashing method performs in the case of image modification. Using an image editor, two billiard balls were removed from the original image, after which a search was made for the most similar image.

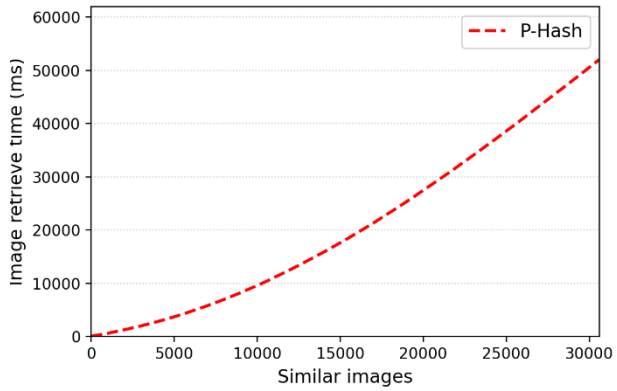


Fig. 5. **Search time graph depending on the number of similar images**
Source: **compiled by the authors**

All three hash methods found an unmodified image. It should be noted that according to the Hamming distance metric, for the D-Hash method, the distance between the modified and the original image was 3 units, for the P-Hash method, the distance was 4 units, and for the W-Hash method, the distance was 0 units. It can be concluded that the hashing method based on the wavelet transform is more resistant to minor image modifications. The retrieve time was 52.38 milliseconds for the D-Hash method, 39.51 milliseconds for the P-Hash method, and 20.97 milliseconds for the W-Hash method.

Next, it is necessary to study the robustness of the hashing method to image compression. The original image was compressed to 1 % quality in JPEG format, after which the most similar image was retrieved. As a result, all vp-trees found the desired image, but for the P-Hash method, the Hamming distance was 2, while for the other methods this distance turned out to be zero. From this we can conclude that the P-Hash hashing method turned out to be more sensitive to image compression.

Another important experiment is to study the robustness of hashing methods to image blurring. For this experiment, the original image was passed through a filter that performs “boxed blur” (blurring the image based on the average color value of adjacent pixels). This filter has a “kernel size” parameter, which is used to find the average color value. The larger the kernel size, the blurrier the resulting image will be.

During the experiment, the size of the blur kernel was gradually increased until the vp-tree, built on the basis of the particular hashing method, found the wrong closest image.

The W-Hash method turned out to be the most sensitive to blurring and, with a kernel size of 53×53; it began to retrieve the wrong image. The D-Hash method stopped coping with a kernel size of 73x73. The most resistant to blur was the P-Hash method, which stopped coping with a kernel size of 84×84. This shows the effectiveness of the cosine transform operation, which allows ignoring the small details of the image that are lost when it is blurred.

Another experiment is the resistance of hashing methods to image noise. During the experiment, the original image was noised with a Gaussian distribution noise. All three hashing methods turned out to be resistant to noise and correctly found the desired image even at a very strong level of noise.

The last important experiment is the stability of the hashing method to image rotation. The original image was rotated counterclockwise by a certain number of degrees. The rotation degree value was incrementally increased until an image search in a certain vp-tree returned the wrong most similar image.

The most sensitive to rotation was the W-Hash method, which stopped coping when the image was rotated by 5 degrees. The D-Hash method failed at 6 degrees, while the P-Hash method failed to find the correct image at 8 degrees of rotation.

CONCLUSION AND FUTURE WORK

In this paper, we propose a method for searching for images by a sample using the binary space partitioning method and the perceptual hashing method.

For the experiments, a vp-tree was used as a structure that implements the technique of binary space partitioning and three methods of perceptual hashing – the D-Hash method, based on comparing the values of adjacent image pixels, the P-Hash method, which uses a discrete cosine transform, and the W-Hash that applies the Haar wavelet transform.

As a result of the experiments, it was not possible to establish the unambiguously best hashing method: each of the tested methods had its own advantages and disadvantages.

In particular, the D-Hash method turned out to be the fastest hashing method; however, the inefficient structure of the constructed tree leads to a longer search time for an image in the tree. In addition, the D-Hash method coped well with minor

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

image modifications, proved to be resistant to image compression, and showed good efficiency in blurring and noisy images.

The W-Hash method turned out to be the slowest when performing the hash acquisition operation, but the structure of the resulting vp-tree turned out to be the most efficient of the three methods considered. The discrete wavelet transform operation allows this hashing method to be resistant to image modification and compression. However, this method did the worst with blurring and rotating the image.

The P-Hash method shows resistance to modification, blurring and rotation of the image, which is due to the discrete cosine, transform operation.

In general, this technique cannot be recommended for commercial use in image retrieval systems; however, it may be useful in some specialized image retrieval tasks. The technique does not allow overcoming the problem of the semantic gap but allows searching for images when external similarity means semantic similarity.

As ways to further improve the methodology, one can indicate the use of the mathematical apparatus of balanced search trees to build a vp-tree, the selection of more efficient image representation methods, or the search for a more efficient perceptual hashing method.

REFERENCES

1. Rui, Y., Huang, T. S., Ortega, M. & Mehrotra, S. “Relevance feedback: a power tool for interactive content-based image retrieval”. *IEEE Transactions on Circuits and Systems for Video Technology*. 1998; Vol.22 No.5: 644–655. DOI: https://doi.org/10.1109/76.718510.

2. Alzubi, A., Amira, A. & Ramzan, N. “Semantic content-based image retrieval: A comprehensive study”. *Journal of Visual Communication and Image Representation*. 2015; Vol.32 No.1: 20–54. DOI: https://doi.org/10.1016/j.jvcir.2015.07.012.

3. Lin, Z., Ding, G., Hu, M. & Wang, J. “Semantics-preserving hashing for cross-view retrieval”. *IEEE Conference on Computer Vision and Pattern Recognition*. 2015; Vol.1 No.1: 3864–3872. DOI: https://doi.org/10.1109/CVPR.2015.7299011.

4. Lowe, D. G. “Distinctive image features from scale invariant keypoints”. *International Journal of Computer Vision*. 2004; Vol.60 No.2: 91–110. DOI: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

5. Sivic, J. & Zisserman, A. “Video Google: A text retrieval approach to object matching in videos”. *IEEE Conference on Computer Vision and Pattern Recognition*. 2003; Vol.1 No.1: 1470–1477. DOI: https://doi.org/10.1109/ICCV.2003.1238663.

6. Tolias, G., Avrithis, Y. & Jgou, H. “To aggregate or not to aggregate: selective match kernels for image search”. *International Conference on Computer Vision (ICCV)*. 2013; Vol.1 No.1: 111–123. DOI: https://doi.org/10.1109/ICCV.2013.177.

7. Nister, D. & Stewenius, H. “Scalable recognition with a vocabulary tree”, *IEEE Conference on Computer Vision and Pattern Recognition*. 2006; Vol.2 No.1: 2161–2168. DOI: https://doi.org/10.1109/CVPR.2006.264.

8. J´egou, H., Douze, M. & Schmid, C. “Improving bag-of-features for large scale image search”. *International Journal of Computer Vision*. 2010; Vol.87 No.3: 316–336. DOI: https://doi.org/10.1007/s11263-009-0285-2.

9. Cao, Y., Wang, H., Wang, C., Li, Z., Zhang, L. & Zhang, L. “Mindfinder: interactive sketch-based image search on millions of images”. *ACM International Conference on Multimedia*. 2010; Vol.1 No.1: 1605–1608. DOI: https://doi.org/10.1145/1873951.1874299.

10. Xiao, C., Wang, C., Zhang, L. & Zhang, L. “Sketch-based image retrieval via shape words”. *ACM International Conference on Multimedia Retrieval*. 2015; Vol.1 No.1: 571–574. DOI: https://doi.org/10.1145/2671188.2749360.

11. Wang, J. & Hua, X. “Interactive image search by color map”. *ACM Transactions on Intelligent Systems and Technology*. 2011; Vol.3 No.1: 1–23. DOI: https://doi.org/10.1145/2036264.2036276.

12. Polyakova, M. V. & Nesteryuk, A. G. “Improvement of the color text image binarization method using the minimum-distance classifier”. *Applied Aspects of Information Technology*. 2021; Vol.4 No.1: 57–70. DOI: https://doi.org/10.15276/aait.01.2021.5.

13. Cao, Y., Wang, C., Zhang, L. & Zhang, L. “Edgel index for largescale sketch-based image search”. *IEEE Conference on Computer Vision and Pattern Recognition*. 2011; Vol.1 No.1: 761–768. DOI: https://doi.org/10.1109/CVPR.2011.5995460.

14. Xie, J., Fang, Y., Zhu, F. & Wong, E. “Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval”. *IEEE Conference on Computer Vision and Pattern Recognition*. 2015; Vol.1 No.1: 1275–1283. DOI: https://doi.org/10.1109/CVPR.2015.7298732.

15. Park, M., Jin, J. S. & Wilson, L. S. “Fast content-based image retrieval using quasi-gabor filter and reduction of image feature dimension”. *IEEE Southwest Symposium on Image Analysis and Interpretation*. 2002; Vol.1 No.1: 178–182. DOI: https://doi.org/10.1109/IAI.2002.999914.

16. Wang, X., Zhang, B. & Yang, H. “Content-based image retrieval by integrating color and texture features*”. Multimedia Tools and Applications*. 2014; Vol.68 No.3: 545–569. DOI: https://doi.org/10.1007/s11042-012-1055-7.

17. Wang, B., Li, Z., Li, M. & Ma, W. “Large-scale duplicate detection for web image search”. *IEEE International Conference on Multimedia and Expo*. 2006; Vol.1 No.1: 353–356. DOI: https://doi.org/10.1109/ICME.2006.262509.

18. Ruvinskaya, V. M. & Timkov, Y. Y. “Deep learning technology for videoframe processing in face segmentation on mobile devices”. *Herald of Advanced Information Technology*. 2021; Vol.4 No.2: 185–194. DOI: https://doi.org/10.15276/hait.02.2021.7.

19. Lowe, D. G. “Object recognition from local scale-invariant features”. *IEEE International Conference on Computer Vision*. 1999; Vol.2 No.1: 1150–1157. DOI: https://doi.org/10.1109/ICCV.1999.790410.

20. Matas, J., Chum, O. & Pajdla, T. “Robust widebaseline stereo from maximally stable extremal regions”. *Image and Vision Computing*. 2004; Vol.22 No.10: 761–767. DOI: https://doi.org/10.1016/j.imavis.2004.02.006.

21. Rosten, E., Porter, R. & Drummond, T. “Faster and better: A machine learning approach to corner detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2010; Vol.32 No.1: 105–119. DOI: https://doi.org/10.1109/TPAMI.2008.275.

22. Krizhevsky, A. & Hinton, G. E. “Using very deep autoencoders for content-based image retrieval”. – Available from: http://www.cs.toronto.edu/~fritz/absps/esann-deep-final.pdf. – [Accessed February 2021].

23. Simonyan, K. & Zisserman, A. “Very deep convolutional networks for large-scale image recognition”. – Available from: https://arxiv.org/pdf/1409.1556.pdf. – [Accessed February 2021].

24. Berezsky, O. M. & Liashchynskyi, P. B. “Comparison of generative adversarial networks architectures for biomedical images synthesis”. *Applied Aspects of Information Technology*. 2021; Vol.4 No.3: 250–260. DOI: https://doi.org/10.15276/aait.03.2021.4.

25. Xie, L., Hong, R., Zhang, B. & Tian, Q. “Image classification and retrieval are ONE”, *ACM International Conference on Multimedia Retrieval*. 2015; Vol.1 No.1: 3–10. DOI: https://doi.org/10.1145/2671188.2749289.

26. J´egou, H., Douze, M. & P´erez, P. “Aggregating local descriptors into a compact image representation”. *IEEE Conference on Computer Vision and Pattern Recognition*. 2010; Vol.1 No.1: 3304–3311. DOI: https://doi.org/10.1109/CVPR.2010.5540039.

27. Perronnin, F., Liu, Y. & Poirier, H. “Large-scale image retrieval with compressed fisher vectors”. *IEEE Conference on Computer Vision and Pattern Recognition*. 2010; Vol.1 No.1: 3384–3391. DOI: https://doi.org/10.1109/CVPR.2010.5540009.

28. Zhou, W., Li, H. & Tian, Q. “Large scale partial-duplicate image retrieval with bi-space quantization and geometric consistency”. *IEEE International Conference Acoustics Speech and Signal Processing*. 2010; Vol.1 No.1: 2394–2397. DOI: https://doi.org/10.1109/ICASSP.2010.5496205.

29. Zhang, S., Tian, Q. & Li, S. “Descriptive visual words and visual phrases for image applications”. *ACM International Conference on Multimedia*. 2009; Vol.1 No.1: 75–84. DOI: https://doi.org/10.1145/1631272.1631285.

30. Zhang, S., Tian, Q. & Gao, W. “Generating descriptive visual words and visual phrases for large-scale image applications”. *IEEE Transactions on Image Processing*. 2011; Vol.20 No.9: 2664–2677. DOI: https://doi.org/10.1109/TIP.2011.2128333.

31. Bentley, J. L. “K-d trees for semidynamic point sets”, *Sixth annual symposium on Computational geometry*. 1990; Vol.1 No.1: 187–197. DOI: https://doi.org/10.1145/98524.98564.

32. Silpa-Anan, C. & Hartley, R. “Localization using an image map”. – Available from: https://www.araa.asn.au/acra/acra2004/papers/silpa-anan.pdf. – [Accessed February 2021].

33. Indyk, P. & Motwani, R. “Approximate nearest neighbors: towards removing the curse of dimensionality”. – Available from: https://theoryofcomputing.org/articles/v008a014/v008a014.pdf. – [Accessed February 2021].

34. Baeza-Yates, R. & Ribeiro-Neto, B. “Modern information retrieval”. – Available from: http://web.cs.ucla.edu/~miodrag/cs259-security/baeza-yates99modern.pdf. – [Accessed February 2020].

35. Tang, J., Li, Z. & Zhao, R. “Neighborhood discriminant hashing for large-scale image retrieval”. *IEEE Transactions on Image Processing*. 2015; Vol.24 No.9: 2827–2840. DOI: https://doi.org/10.1109/TIP.2015.2421443.

36. Qin, D., Wengert, C. & Van Gool, L. “Query adaptive similarity for large scale object retrieval”. *IEEE Conference on Computer Vision and Pattern Recognition*. 2013; Vol.1 No.1: 1610–1617. DOI: https://doi.org/10.1109/CVPR.2013.211.

37. Xie, H., Gao, K. & Liu, Y. “Pairwise weak geometric consistency for large scale image search”. *ACM International Conference on Multimedia Retrieval*. 2011; Vol.1 No.1: 42–68. DOI: https://doi.org/10.1145/1991996.1992038.

38. “Welcome To Colaboratory”. – Available from: https://colab.research.google.com. – [Accessed June 2020].

39. “Caltech-256 Dataset”. – Available from: https://authors.library.caltech.edu/7694/. – [Accessed June 2020].

Conflicts of Interest: the authors declare no conflict of interest

Received 21.12.2020
Received after revision 27.02.2021
Accepted 14.03.2021

DOI: https://doi.org/10.15276/aait.05.2022.10
УДК 004.932.2

Методологія пошуку зображень на базі бінарного розбиття простору та перцептивного хешування

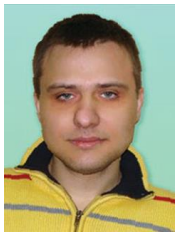
Микола Анатолійович Годовиченко[1)]
ORCID: https://orcid.org/0000-0001-5422-3048; nick.godov@gmail.com. Scopus Author ID: 57188700773
Світлана Григорівна Антощук[1)]
ORCID: https://orcid.org/ 0000-0002-9346-145X; asg@opu.ua. Scopus Author ID: 8393582500
Варвара Ігорівна Куваєва[1)]
ORCID: https://orcid.org/0000-0002-9350-1108; vkuvayeva@gmail.com. Scopus Author ID: 57203618134
[1)] Одеський національний політехнічний ун-т, пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Стаття розглядає питання щодо побудови систем пошуку зображень на основі зображення-зразка. Розглянуто основні виклики, які стоїть на шляху вчених та інженерів при побудові подібних систем, розглянуто складові частини подібних систем і дано короткий огляд основних методів і технік, які використовувалися в цій галузі для реалізації основних складових частин систем пошуку зображень. У якості одного з варіантів розв'язання подібного завдання запропоновано методологію пошуку зображень на основі методу бінарного розбиття простору та методу перцептивного хешування. Дерева бінарного розбиття простору являють собою структуру даних, отриману наступним чином: простір розбивається гіперплощиною на два напівпростори, потім кожен напівпростір рекурсивно розбивається до тих пір, поки кожен вузол не буде містити тільки тривіальну частину вхідних об'єктів. Алгоритми перцептивного хешування дозволяють отримати уявлення зображення у вигляді хеш-значення довжиною 64 біта, при чому схожі зображення будуть представлені схожими хеш-значеннями. У якості метрики визначення відстані між хеш-значеннями використовується відстань Геммінга, яка підраховує кількість різних біт. Для організації бази хеш-значень використовується vp-дерево, що є реалізацією структури бінарного розбиття простору. Для експериментального дослідження методики був використаний набір даних Caltech-256, який містить 30607 зображень, розбитих на 256 категорій, у якості алгоритмів перцептивного хешування були використані алгоритми Difference Hash, P-Hash та Wavelet Hash, дослідження проводилося в середовищі Google Colab. В рамках експериментального дослідження було розглянуто стійкість алгоритмів хешування до модифікації, стиснення, розмиття, зашумлення та повороту зображення. Крім того, було проведено дослідження процесу побудови vp-дерева та процесу пошуку зображень у дереві. В результаті експериментів було встановлено, що кожен з алгоритмів хешування має свої переваги та недоліки. Так, алгоритм хешування, заснований на різниці значень пікселів на зображенні, виявився найшвидшим, але був не надто стійким до модифікації та повороту зображень. Алгоритм P-Hash, заснований на використанні дискретного косинусного перетворення, показав кращу стійкість до розмиття зображень, але виявився чутливим до стиснення. Алгоритм W-Hash, заснований на вейвлет-перетворенні Гаара, дозволив побудувати найбільш ефективну структуру дерева і виявився стійким до модифікації та стиснення зображення. Запропонована методика не рекомендується для використання в системах пошуку зображень загального призначення, однак, може бути корисна для пошуку зображень у спеціалізованих базах. В якості подальших шляхів покращення методики можна відзначити удосконалення структури vp-дерева, а також пошук більш ефективного методу представлення зображення, ніж перцептивне хешування.

Ключові слова: пошук зображень на основі вмісту; бінарне розбиття простору; перцептивне хешування; дерево точки огляду; дискретне косинусне перетворення; дискретне вейвлет-перетворення.

ABOUT THE AUTHORS



Mykola A. Hodovychenko - Candidate of Engineering Sciences, Associate Professor of the Department of Artificial Intelligence and Data Analysis. Odessa National Polytechnic University, 1, Shevchenko Ave. Odessa, 65044, Ukraine
ORCID: https://orcid.org/0000-0001-5422-3048; nick.godov@gmail.com. Scopus Author ID: 57188700773
***Research field*:** Deep learning; data mining; smart cities; video processing; motion tracking; project-based learning; patter recognition

Микола Анатолійович Годовиченко - кандидат технічних наук, доцент кафедри Штучного інтелекту та аналізу даних. Одеський національний політехнічний ун-т, пр. Шевченка, 1. Одеса, 65044, Україна

Svitlana G. Antoshchuk – D.Sc (Eng), Professor, Head of Computer Systems Institute. Odessa National Polytechnic University, 1, Shevchenko Ave. Odessa, 65044, Ukraine
ORCID: https://orcid.org/0000-0002-9346-145X; asg@opu.ua. Scopus Author ID: 8393582500
***Research field*:** Pattern recognition; deep learning; object tracking; face recognition; graphic images formation and processing

Світлана Григорівна Антощук - доктор технічних наук, професор, директор Інституту Комп'ютерних систем. Одеський національний політехнічний ун-т, пр. Шевченка, 1. Одеса, 65044, Україна

Varvara I. Kuvaieva - Candidate of Engineering Sciences, Associate Professor of the Department of Information Systems. Odessa National Polytechnic University, 1, Shevchenko Ave. Odessa, 65044, Ukraine
ORCID: https://orcid.org/0000-0002-9350-1108; vkuvayeva@gmail.com. Scopus Author ID: 57203618134
***Research field*:** Network collective expert estimation

Варвара Ігорівна Куваєва - кандидат технічних наук, доцент кафедри Інформаційних систем. Одеський національний політехнічний ун-т, пр. Шевченка, 1. Одеса, 65044, Україна