https://doi.org/10.15407/intechsys.2025.01.003
UDC 004.8 + 004.032.26

O.D. GOLTSEV, PhD (Engineering), Senior Researcher, Head of the Department,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
https://orcid.org/0000-0002-2961-0908
root@adg.kiev.ua

O.O. HOLTSEV, PhD Student,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
https://orcid.org/0000-0002-1846-6648
rcwolf@adg.kiev.ua

I.V. SUROVTSEV, DSc (Engineering), Senior Researcher, Head of the Department,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
https://orcid.org/0000-0003-1133-6207
igorsur52@gmail.com

ALGORITHM FOR CALCULATING THE SIMILARITY BETWEEN HISTOGRAMS FOR TEXTURE SEGMENTATION

Introduction. *An algorithm for calculating the similarity degree between multidimensional histograms is presented. The proposed algorithm was intended for texture segmentation of images using histograms as texture features. The need to develop such a special algorithm is justified by the fact that the methods for estimating the similarity/difference measure between multidimensional vectors described in the literature provide such measures that are not very suitable for solving the texture segmentation task. The main peculiarity of the proposed algorithm is that when calculating the similarity value, it considers not only the corresponding histogram components, but also takes into account their nearest neighboring components. Due to this, the algorithm more adequately evaluates the similarity of histograms. The proposed*

Cite: Goltsev O.D., Holtsev O.O., Surovtsev I.V. Algorithm for Calculating the Similarity Between Histograms for Texture Segmentation. *Information Technologies and Systems*, Kyiv, 2025, Vol. 1 (1), 03—23. https://doi.org/ 10.15407/intechsys.2025.01.003

algorithm was implemented as a computer program as an integral part of the image segmentation model. The effectiveness of the histogram comparison algorithm was indirectly confirmed by the results of texture segmentation of the image segmentation model in experiments on processing various images, including natural landscapes.

Methods. *The task of calculating the similarity between histograms is considered. A special algorithm is proposed because the analogical methods described in the literature are not very suitable for solving the texture segmentation task. The main peculiarity of the algorithm is that it takes into account as the corresponding histogram components as their nearest neighboring components. Due to this, the algorithm more adequately evaluates the similarity of histograms. The algorithm was implemented as a computer program. The effectiveness of the algorithm is indirectly confirmed by the results of texture segmentation of the image segmentation model in experiments on processing various images, including natural landscapes.*

Purpose. *The goal of this work is to develop an efficient algorithm for assessing the similarity of histograms, such as brightness histograms and orientation histograms of the texture windows. The algorithm is based on the idea of taking into account not only the corresponding components of both histograms, but also the components of their immediate environment.*

Results. *The main advantage of the proposed algorithm, compared to popular methods of calculating similarity/difference between objects (vectors), is that the range of similarity between the compared histograms (from complete similarity to complete difference) is 100%, while popular methods can offer several times smaller ranges of similarity percentage.*

Conclusions. *The proposed algorithm provides a wide range of similarity between the compared histograms which is 100% (from complete similarity to complete difference), while popular methods can offer several times smaller ranges of similarity percentage. The algorithm was implemented as a computer program as a component of a model that solves the problem of segmenting a visual image into homogeneous texture areas. It is worth noting that the proposed histogram comparison algorithm calculates the similarity measure between histograms very quickly, since it uses only simple operations.*

The effectiveness of the algorithm for texture segmentation of images into homogeneous texture areas is confirmed by the results in the experiments on natural image processing. The results obtained in the experiments demonstrate the effectiveness of the algorithm and show that the algorithm performs correct (from a human point of view) texture segmentation of a wide range of images. Thus, the effectiveness of the key operation of the segmentation algorithm, the histogram comparison algorithm, is indirectly confirmed.

Keywords: *Image processing, Similarity of histograms, Texture features, Texture segmentation.*

Introduction

This paper considers the problem of evaluating the similarity between histograms, that are used as texture features for the task of dividing an image into texture segments. The problem of texture segmentation of images is a key one for the analysis of natural visual scenes of various natures such as landscapes, satellite photographs, medical images etc.

The problem of texture segmentation of images has different complexity depending on the amount of available information about the image being processed. For example, solving this problem is greatly simplified if the number of texture segments present in the image is known. The task is also significantly simplified if samples of texture fragments that need to be extracted from the image are provided. Using this information, the parameters of the segmentation algorithm can be adjusted accordingly by means of training, for example. This approach belongs to the category of supervised learning, is currently dominant, and presented in a significant number of publications [1–10].

Another approach to the texture segmentation problem implies that neither the number of texture segments nor the texture fragment samples of these segments are given. Instead, the segmentation algorithm extracts texture regions without any additional information or training, but using some universal texture features. This approach falls to unsupervised texture segmentation, [11–21].

Textures may be divided into fine-grained and coarse-grained [13]. Image areas representing objects with discontinuities in depth, material, illumination etc., correspond to coarse texture segments, while fine-grained textures are characterized by a smooth change in image brightness and fine granularity. Texture segmentation of images can be performed both by selecting areas of coarse texture and by selecting homogeneous segments of fine-grained texture. The presented approach belongs to the latter one. It is worth noting that the discrimination of textures to "coarse" or "fine-grained" is, to a large extent, rather relative. Indeed, by increasing the degree of image resolution, fine-grained segments may turn into coarse ones. In the presented work, the size and image resolution are fixed.

The ability to assess the similarity between objects is one of the foundations of both natural and artificial intelligence. However, the concept of "similarity" is an intuitive concept that does not have a strict formal definition. Nevertheless, the literature describes a variety of mathematical methods and algorithms designed to determine the measure of similarity between objects.

Calculation of a similarity measure between objects was used to solve the similarity search (proximity search, best match retrieval) problem, the purpose of which is to identify database objects that are similar to a query object. Also such search is called the nearest neighbor search or the nearest neighbor problem. Similarity measures also are used in many data analysis, statistics, machine learning and other problems related to artificial intelligence. Many methods have been developed to measure the similarity between such objects as vectors, sequences, trees, and graphs [23–28].

The closest to the topic of this work are methods for determining the similarity/dissimilarity of vectors of large dimensions. Distance is a measure of the "dissimilarity" or "difference" of vectors — for example, the Euclidean distance, Manhattan distance, Hamming distance, Minkowski distance, Mahalanobis distance, Bulldozer's distance (earth mover's distance, EMD [22–28]). There are also direct measures of vector similarity, the larger values of which correspond to more similar vectors (for example, the cosine of an angle or the scalar product of vectors [24–28]).

Among the various variations of component-wise comparison methods described in the literature, the simplest measure of vector difference is the Euclidean distance [22, 24, 25]. The Euclidean distance does not take into account the dependencies between all vector components: only corresponding components (i.e., components with the same serial numbers (indices)) are compared. The cross-component distance of quadratic form, also known as the Mahalanobis distance, allows one to determine the similarity between all vector components [14].

However, all mentioned above methods are difficult to use for comparing histograms when solving the problem of texture segmentation.

At the first glance, the concept of a histogram is identical to the concept of a vector. However, this is not entirely true. The fact is that the values of the vector components are generally independent of each other and can take any values. Unlike a vector, the values of the histogram components are interconnected in such a way that the sum of the values of all the histogram components (with any changes in its shape) remains constant. This means that if the height of some histogram column (due to changes in external conditions) increases, this is necessarily accompanied by a decrease in the heights of other histogram columns (always neighboring ones), so that the sum of the heights of all its columns remains unchanged. Such interdependence of the histogram components provides an opportunity for some improvement of the method for calculating the similarity between histograms, compared to the methods for calculating such similarity between arbitrary vectors.

In accordance with the title of the paper, the goal of this work is to develop an efficient algorithm for assessing the similarity of histograms, such as brightness histograms and orientation histograms of the texture windows. The algorithm is based on the idea of taking into account not only the corresponding components of both histograms, but also the components of their immediate environment.

This paper is a continuation, development and revision of the paper [29], which was devoted to solving the same problem. The main difference between this work and [29] is a significant change in the algorithm for calculating the similarity of the corresponding components of histograms. The similarity of the corresponding components was determined by measuring the size of the intersection of the compared histogram columns [29]. The disadvantage of this approach is the uneven calculation of the similarity measure when comparing pairs of histogram columns with significantly different heights. In the present paper, this disadvantage is eliminated by introducing a method for determining the similarity of the corresponding histogram columns by calculating the ratio of their heights, which does not depend on the absolute values of the column heights.

The paper is organized as follows. Section 1 Introduction. Section 2 provides descriptions of such texture features as brightness and orientation histograms. Section 3 gives an overview of the whole segmentation procedure. In Section 4 we try to substantiate the relevance of the paper task. Section 5 contains the actual description of the histogram comparison algorithm. Section 6 shows a comparison between some experimental histograms. Section 7 is devoted to discussion and conclusions.

Histograms as Texture Features

In the field of pattern recognition, it is well known that the choice of features used for recognition has a decisive effect on the recognition results. Accordingly, the results of texture segmentation also strongly depend on how well the selected set of features describes the textures.

In this work, grayscale photographs of some physical objects or natural scenes, such as landscapes, are used as images. The main type of features used in our work is the brightness histogram for the following reasons. The brightness histogram adequately represents the distinctive texture features of a homogeneous segment of fine-grained texture, providing an ambiguous, but very informative description of it. The histogram remains invariant when changing the coordinates of the texture window inside the homogeneous texture segment. At the same time, in all natural images, the histograms of different texture segments differ significantly from each other.

The brightness histogram adequately represents the distinctive peculiarities of the texture of a homogeneous fine-grained segment, remaining invariant when changing the coordinates of the texture window inside the segment. The histogram provides, although not entirely unambiguous, but very informative description of the texture of a homogeneous segment. Of course, it is possible to artificially construct different texture regions with the same brightness histograms. However, in natural images, the histograms of different texture segments differ significantly from each other, and it is almost impossible to find adjacent texture segments with the same histograms in natural images.

Any grayscale image is actually an integer matrix, each element of which represents the brightness value of the corresponding pixel of the image. In such images, brightness values range from 0 to 255.

To evaluate the texture characteristics of different areas of an image, texture windows of the same size are used, which cover the entire image (with overlaps). Currently, the use of texture windows to evaluate texture characteristics is the most common method. Texture characteristics measured in texture windows serve to evaluate the similarity and/or difference between the corresponding areas of the image.

The histogram of brightness of all pixels of the texture window consists of 256 columns, in accordance with the range of brightness of the grayscale image. The height of each column of the histogram represents the number of pixels of the texture window that have the corresponding brightness values. The maximum height of the columns is equal to the number of pixels in the texture window. In the experiments, texture windows of 15×15 pixels in size are used. Accordingly, the maximum height of the column of the brightness histogram is 225.

The second histogram, used in the texture segmentation algorithm as a texture feature is the orientation histogram of all pixels in the texture window. This histogram is calculated based on filtering the image using the Scharr filter [30]. The orientation histogram is necessary to distinguish between texture segments that contain differently oriented strokes. In the experiments, the orientation histogram consists of eight columns.

General Description of the Texture Segmentation Algorithm

In [15–18], a texture segmentation algorithm is described that extracts all homogeneous fine-grained texture segments sequentially, in an iterative process, starting with the more homogeneous and ending with the less homogeneous segments. In each iteration, first of all, the starting point (grain) is detected, which belongs to the most homogeneous texture segment present in the image. Subsequently, this grain is expanded by successively adding to it the surrounding image pixels.

A set of texture features is extracted from the initial point and its immediate surroundings and stored to perform the procedure of extracting this homogeneous texture segment. This set of features is considered as characteristic (typical, representative) for this segment and is used for successive comparison with the surrounding areas of the image. If the set of features extracted from the surrounding area (pixel) coincides with the characteristic one, then this area is attached to the initial point of the extracted texture segment. The process of such successive comparison (with subsequent attachment of pixels) continues until the boundaries of the homogeneous texture segment are reached. Thus, the segmentation procedure is performed based on the region growing method [19–21].

Algorithm for Comparing Low-dimensional Histograms

According to the above description of the segmentation algorithm, the most important procedure of the algorithm is the comparison between the characteristic (representative) texture features extracted from the initial points of the segment with the texture features extracted from the texture windows surrounding the initial point.

Since the main texture features used in this work to solve the segmentation problem are the brightness histogram and the orientation histogram, the key operation of the segmentation algorithm is to calculate the similarity value between these histograms extracted from two adjacent texture windows.

As mentioned above, the algorithm presented in [29] is revised in this work. In [29], the similarity of the corresponding components of the compared histograms was performed by measuring the intersection size of their values. Let us consider an example of two pairs of histogram columns: in one pair the columns have values of 100 and 80 units, and in the other pair they have values of 10 and 8 units. It should be evident that the similarity between considered pairs of columns is the same. However, when comparing a pair of histogram columns by measuring the size of their intersection, the intersection size of the first pair is 80 units, while the second pair is 8. Therefore, if we calculate the similarity of these pairs by measuring the intersection sizes, we should normalize the sizes 80 and 8, if we want to determine the similarity of all corresponding histogram columns by the same algorithm. Thus, the disadvantage of this approach is the uneven calculation of the similarity measure when comparing pairs of histogram columns with different heights. In this paper this disadvantage is eliminated by introducing a method for determining the similarity of two corresponding histogram columns by calculating the ratio of their heights, which does not depend on the absolute values of their heights.

The main subject of this work is the algorithm for calculating the measure of similarity of multidimensional histograms, which is presented below. However, let us first consider the algorithm for comparing histograms consisting of a small number of components. Such, in particular, is the orientation histogram, consisting in this work of 8 components.

Let two histograms P and D be given, each consisting of the same number of components (columns) ($i = 1, 2, \ldots, N$). The task is to determine the value of their similarity. Let us denote the height of the i-th column of the histogram P by P_i, and the height of the i-th column of the histogram D by D_i. Let us denote the ratio of the corresponding heights of the columns with index i by R_i. These designations refer to both low-dimensional and multidimensional histograms.

he algorithm for calculating the similarity measure of low-dimensional histograms is as follows.

Let us assume that all components of both compared histograms have non-zero values. The simplest and the most obvious algorithm for calculating the similarity measure of two such histograms is based on a component-wise comparison of the corresponding histogram columns, i.e. columns with the same indices i. For each such component-wise comparison, the maximum and minimum heights of the corresponding columns $H_i^{\max}$ and $H_i^{\min}$ are first determined. Namely, if $P_i > D_i$, then $H_i^{\max} = P_i$ and $H_i^{\min} =$ $= D_i$; if $D_i > P_i$, then $H_i^{\max} = D_i$ and $H_i^{\min} = P_i$. Then, the ratio $R_i =$ $= H_i^{\min}/H_i^{\max}$ is calculated, which is expressed as a number in the range 0—1. Let us denote the similarity between the histograms, calculated on the basis of a component-wise comparison of their columns, by R^0, it is calculated by the formula

$$R^0 = \sum_{i=0}^{N} \left(H_i^{\min} / H_i^{\max} \right), \tag{1}$$

where $i = 0, 1, 2, \ldots, N$.

Thus, the degree of similarity between histograms is determined by the sum of the ratios of the heights of the corresponding components of the histograms.

However, in reality, many components of histograms may have zero values. In this case, the calculation of the similarity measure is performed by a slightly more complex procedure, which is described by **Algorithm 1** in a (C++ like) the pseudo-code format.

Algorithm 1. Calculation of the similarity between two histograms based on component-wise comparison

```
float $R^0 = 0$;
int $Q = 0$;
for ($i = 0$; $i < N$; $i$++)
{
 if ($H_i^{\max}$ = = 0)
     continue;
 else
 {
      $Q$++;
      $R^0 = R^0$ + float $H_i^{\min}/H_i^{\max}$;
 }
}
$R^0 = R^0/Q$.
```

The similarity value R^0, calculated according to **Algorithm 1**, is a number in the range 0–1. When multiplied by 100, this number turns into a percentage of similarity, in the range 0–100%, where 100% means complete similarity.

As follows from **Algorithm 1**, when calculating the similarity of histograms, only such pairs of corresponding columns are taken into account in which at least one of the columns has a non-zero height. Let us note that with an increase in the number of pairs of columns in which only one of them has a non-zero height, the value of the Q parameter increases, which ultimately leads to a decrease in the similarity value.

Also, from **Algorithm 1** it follows that if both compared components of the histogram have zero values, then these components are not taken into account at all when calculating the similarity. The rationale for ignoring paired zero components of histograms is as follows.

The maximum difference between the brightness histograms of two texture windows occurs when one texture window is located on the black area of the image RGB = {0, 0, 0}, and the second window is on the white: RGB = {255, 255, 255}. In this case, in the histogram of the black texture window, only one component with index $i = 0$ has a non-zero (maximum) value. And in the histogram of the white texture window, the only non-zero column (of maximum height) corresponds to the component with index $i = 255$. These two histogram columns have the same height, equal to the number of pixels in the texture window. At the same time, the similarity between such histograms should obviously be zero. At the same time, all the other 254 corresponding components of both histograms (excluding components with indices $i = 0$ and $i = 255$) have the same zero

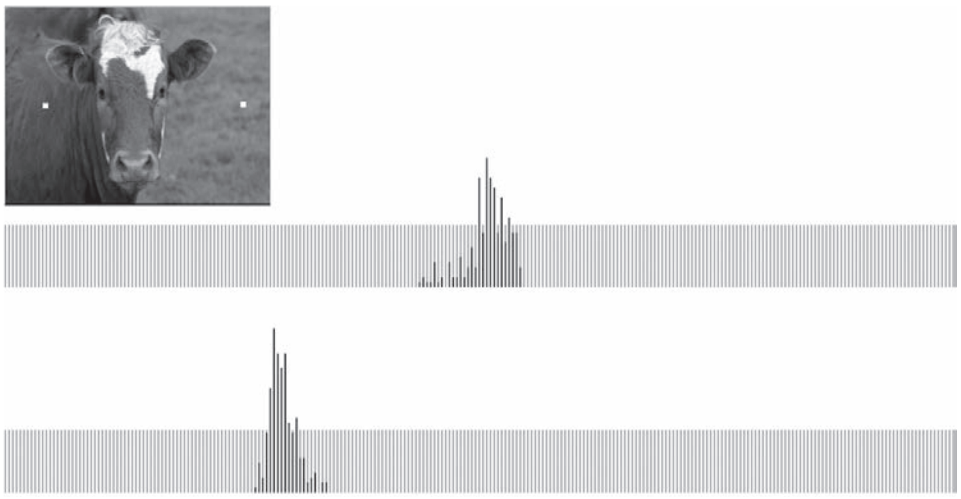

Fig. 1. Brightness histograms of two texture windows, one of which belongs to the "grass" texture (the upper one), and the other — to the "cow" texture (the lower one)

value. So, if we take into account the similarity of these 254 corresponding zero components in any way, even the most minimal way, and add it to the total similarity value, then the degree of similarity of the histograms under consideration will no longer be equal to 0%, which contradicts the initial intuitive assessment of their similarity.

In the experiments conducted in this paper, texture windows of 15 × 15 = 225 pixels were used. The brightness histogram of each such texture window (in contrast to the low-dimensional orientation histogram), which, in the case of a grayscale image, consists of 256 components (0–255), is quite variable. This paper presents an algorithm designed to calculate the similarity value between multidimensional variable histograms, which is somewhat more complex than the one described above. The need to develop such an algorithm is argued as follows.

Figure 1 shows an example of two brightness histograms belonging to different textures. These are histograms of two texture windows measuring 15 × 15 pixels, the centers of which are marked with white squares in the original image. Namely, the upper histogram (No. 1) belongs to the texture window located on the "grass" texture, and the lower histogram (No. 2) corresponds to the texture window related to the "cow" texture segment.

As can be seen from Figure 1, the histograms of these texture windows differ significantly from each other. They differ both in amplitude and, above all, in that the non-zero components of both histograms do not intersect, but are located in different positions within the histograms. From the point of view of common sense, similar histograms should have significant intersections between corresponding columns, and dissimilar ones should have small intersections, in the limiting case equal to zero. In Fig. 1, there is no such intersections at all. For the purposes of texture segmentation, the histograms presented in Fig. 1 should preferably be considered completely different, with a similarity measure equal to zero.

However, popular methods for calculating similarity/difference between vectors, such as, for example, the Euclidean distance ([22], [24], [25]), the Bulldozer distance (earth mover's distance, EMD [23], [24]), the Mahalanobis distance [14], give not zero, but significant values of similarity between such histograms.

Among the various variants for component-wise comparison of vectors, the Euclidean distance is the simplest measure of similarity/dissimilarity between them. The Euclidean distance — Dist, between multi-component vectors is calculated by the formula:

$$Dist = \sqrt{\sum_{i=0}^{N}\left(P_i - D_i\right)^2}, \tag{2}$$

where $i = 0, 1, 2, ..., N$.

In particular, the Euclidean distance between the histograms shown (Fig. 1) is 89.24. To convert this number into a percentage of similarity between the histograms, we use the reasoning considered above.

As mentioned above, the maximum difference (distance) between the brightness histograms of two texture windows occurs when one texture window is located on the black area of the image RGB = {0, 0, 0}, and the second window is on the white area RGB = {255, 255, 255}. In this case, for texture windows of 15 × 15 = 225 pixels, the maximum Euclidean distance (i.e., the maximum difference value) between these vectors according to Eq. (2) is $\sqrt{\left(225^2 + 225^2\right) = 318.2}$. The minimum Euclidean distance, naturally, is zero, i.e., this is a complete, 100% similarity. Accordingly, the intermediate percentages of similarity between the histograms are calculated as follows.

If the Euclidean distance between the histograms (Fig. 1) is 89.24, then the percentage difference between them is determined by the value (89.24/318.2)100% = 28.05%. Accordingly, the percentage similarity between the histograms is (100 – 28.05) = 71.95%. Thus, there is a significant difference between the desired percentage similarity — 0% and the percentage similarity calculated based on the Euclidean distance — 71.95%.

As mentioned above, the process of texture segmentation involves a sequential expansion of the area occupied by a homogeneous texture segment. Such expansion is performed based on the results of comparison of texture features, the main of which is the brightness histogram. If, in the process of texture segmentation, the percentage of similarity calculated based on the Euclidean distance (i.e. 71.95%) is used to compare the histograms shown (Fig. 1), then with such a large value of the percentage of similarity, the segmentation algorithm would have difficulties while separating the areas occupied by the "cow" and "grass" textures (Fig. 1). A completely different situation occurs if the percentage of similarity between these histograms is 0%. It should be obvious that in this case it will be easier for the segmentation algorithm to find the boundary between the indicated texture areas.

The above considerations justify the relevance of developing a specialized algorithm for measuring the similarity between multi-component histograms for the task of texture segmentation, presented in this paper.

The main advantage of this algorithm is that the range of similarity values between the compared histograms (from complete similarity to absolute difference) is 100%, while the popular methods of measuring similarity between vectors mentioned above provide a much smaller range of similarity percentage values. In particular, in the example considered above, the range of similarity measure variation for the Euclidean distance is 100 – 72 = 28%. In practice, the range of similarity percentage values from 0 to 100% of the proposed algorithm allows for more accurate detection of boundaries between adjacent texture segments.

The Algorithm of the Histogram Comparison

When comparing histograms that consist of a small number of components and, therefore, are not variable, it is quite sufficient to use Algorithm 1 to determine the similarity between them. However, for multi-component histograms, such as texture window brightness histograms, the sum of the ratios of the heights of the corresponding histogram columns does not adequately reflect the similarity between the histograms. The fact is that the texture window brightness histogram is very susceptible to changing its shape with small fluctuations in the environment. That is, the values of some of the brightness histogram components can change with minor local illumination fluctuations, small shifts or rotations of the image. Accordingly, the similarity between the compared texture window histograms will also change.

When comparing such histograms in the process of solving the texture segmentation task, it would be highly desirable to have a degree of similarity between the histograms that is weakly dependent on small changes in their shape. The algorithm presented below, which is insensitive (robust) to small changes in their shape, is intended precisely for comparing such variable histograms. The algorithm is based on the obvious idea of taking into account not only the corresponding components of both histograms, but also the components of their immediate surroundings. The meaning of taking into account the immediate surroundings of each histogram component can be explained by the following reasoning.

In case of local changes in the illumination of any object zone in the image within the texture window, the brightness histogram of this window will change. Let us consider two texture windows with histograms P and D. Let the value of the i-th component $P_i > 0$ in the histogram P. Let us assume that in case of a small local change in the illumination of the window P (while the window D remains in the same lighting conditions), some change in the value of the i-th component of the histogram P occurs, i.e. the height of the P_i column changes. In this case, since the heights of all the columns in each histogram are interconnected, the heights of the columns of the immediate environment of the i-th component of the histogram P necessarily change. Moreover, the changes in the heights of the P_i column and the columns of its immediate environment will be oppositely directed. Namely, if the height of P_i increases, the height of the P_{i-1} and P_{i+1} columns will decrease and vice versa.

Then, when calculating the similarity of the histograms P and D, if we compare only the corresponding columns P_i and D_{i}, then due to the fluctuation of illumination the size of the similarity between the histograms also changes. If, however, when calculating the similarity of the histograms, we take into account the components of the immediate environment of the components of both histograms, then the change in the value of similarity between the histograms will be compensated, at least to some extent.

The algorithm for calculating the similarity of histograms taking into account their immediate environment is as follows.

Let M denote the radius of the nearest neighborhood of each i-th component of the histograms. In the process of calculating the similarity of histograms taking into account their nearest neighborhood, the operations described below are performed between each i-th component of the histogram P (i.e., P_i) with each of the ($2M$ + 1) components of the histogram D. That is, to perform the operations described below, the i-th component of the histogram P is related to the component of the histogram D shifted by a certain number of columns to the right or left relative to the value of i. If we denote the shift value by m, then the component P_i is successively related to the components D_j, where $j = i+(-)\ m$, $m = 0, 1, 2, 3, \ldots, M$.

The mentioned above operations performed between each component P_i and component D_j are as follows. First of all, the height of the column D_j is multiplied by some reduction factor K_m, $K_m < 1$. Let us denote the height of the j-th column of the histogram D after such multiplication as D_j^m, then $D_j^m = K_m\, D_j$. If a local fluctuation in the illumination of the texture window P occurs, which leads to a change in the height of the column P_i, a change in the heights of the columns of its immediate environment P_j, $j = i+(-)\ m$, necessarily occurs. The greater the difference $m = |i - j|$, the smaller the change in the height of the column P_j. By introducing the reduction factors K_m, we tried to introduce the same relationship between the columns P_i and D_j^m. That is, the larger $m = |i - j|$, the less influence the component D_j^m should have on the degree of similarity between the histograms.

The reducing factors $K_0, K_1, K_2, \ldots, K_m, \ldots., K_M$ are related to each other as follows: $K_0 > K_1 > K_2 > K_3 > \ldots K_m > \ldots > K_M$, where $K_0 = 1$.

Then, between the considered components P_i and D_j^m, the maximum and minimum values $H_i^{\max}$ and $H_i^{\min}$, are determined, namely, if $P_i > D_j^m$, then $H_i^{\max} = P_i$ and $H_i^{\min} = D_j^m$; if $D_j^m > P_i$, then $H_i^{\max} = D_j^m$ and $H_i^{\min} = P_i$.

The ratio $R_i^j = H_i^{\min}/H_i^{\max}$ is calculated, which determines the measure of similarity between the heights of the considered columns P_i and D_j^m.

For each index i, the values of R_i^j are calculated for all j: from $j = i - M$, to $j = i + M$. Then, among all calculated $(2M + 1)$ values of R_i^j, the maximum is determined – $R_i^j(\max)$:

$$R_i^j(\max) = \underset{j = i - 2M}{\overset{j = i + 2M,}{\mathrm{MAX}}} (R_i^j), \tag{3}$$

Let us denote the similarity value between the histograms P and D, calculated taking into account the immediate surroundings of the histogram columns, by R^M, which is determined by summing the values of $R_i^j(\max)$ over all indices i, similar to **Algorithm 1**. The procedure is described below by **Algorithm 2** also a (*C*++ like) the pseudo-code format.

Algorithm 2. Calculation of the similarity between two histograms taking into account the immediate neighborhood of the histogram columns

```
float  $R^M = 0$;
int  $Q = 0$;
for ($i = 0$; $i < N$; $i$++)
{
 if  ($P_i$ = = 0)
          continue;
 else
 {
          $Q$++;
          $R^M = R^M + R_i^j(\max)$;
 }
}
$R^M = R^M/Q$.
```

The radius of the nearest environment of the histogram components M depends on the size of the histograms N and on the initially specified tendency of the algorithm. The larger the radius M, the more pronounced the tendency of the algorithm to consider the compared histograms similar. Conversely, with a decrease in M, the tendency to consider the histograms dissimilar increases.

For the task of texture segmentation, it is advisable to express the similarity between the histograms of texture windows as a percentage. That is, if the similarity percentage is close to 0%, then the compared texture windows should obviously be classified as different textures. Conversely, if the similarity percentage approaches 100%, then these texture windows should be classified as belonging to the same texture. Let us denote the similarity percentage of histograms by S, $S = R^M\,100\%$. So, the measure of similarity of histograms S is expressed by a number lying in the range 0–100%.

When assessing the similarity of texture characteristics of different image areas (texture windows), an important texture characteristic is the average illumination level of the texture window. If the compared texture windows have the same average illumination level, the algorithm described above adequately assesses the degree of their texture similarity based on the similarity of the histograms. However, if the compared windows belonging to the same texture have significantly different average illumination levels, the algorithm can assess their similarity as zero. The fact is that the histograms of texture windows with different average illumination are shifted relative to each other by the value of the difference in their illumination. And, therefore, there may be no intersections between their columns.

Thus, if the task is to evaluate the level of texture similarity of two texture windows regardless of their illumination, then, obviously, the histograms should be shifted relative to each other by the difference between their average illuminations.

Experimental Comparison of Similarity Measures of Brightness Histograms

To illustrate the properties of the proposed algorithm for measuring the percentage of similarity between histograms, the following experiments were conducted.

In the first series of experiments, pairs of texture windows belonging to different textures were considered. Figure 1 shows an example of two such texture windows located on different textures, the centers of which are shown as white squares in the figure. As can be seen from Figure 1, the histograms of these texture windows differ significantly from each other, mainly because the non-zero components of both histograms are located in inappropriate positions within the histograms. From the point of view of common sense and from the functional point of view (for solving the problem of texture segmentation), these histograms should be classified as completely different. And, accordingly, the algorithm proposed in this paper gives an estimate of the percentage of similarity between them equal to zero.

At the same time, the Euclidean distance calculated by Eq. (2) is 89.24. According to the reasoning given in Section 4, the percentage of similarity between the histograms under consideration, determined based on the Euclidean distance, is 71.95%.

In the second series of experiments, pairs of texture windows belonging to the same texture were considered. In Figure 2, the centers of two such texture windows are also marked with white squares. The corresponding brightness histograms were built, which are also shown (Fig. 2). As can be seen from Figure 2, the non-zero components of both histograms occupy approximately the same position inside the histograms, and the corresponding columns have significant intersections.

The Euclidean distance between the histograms calculated by Eq. (2) is 40.64. Since the maximum Euclidean distance between the brightness histograms (in this work) is 318.2 (see Section 4), the percentage of dif-

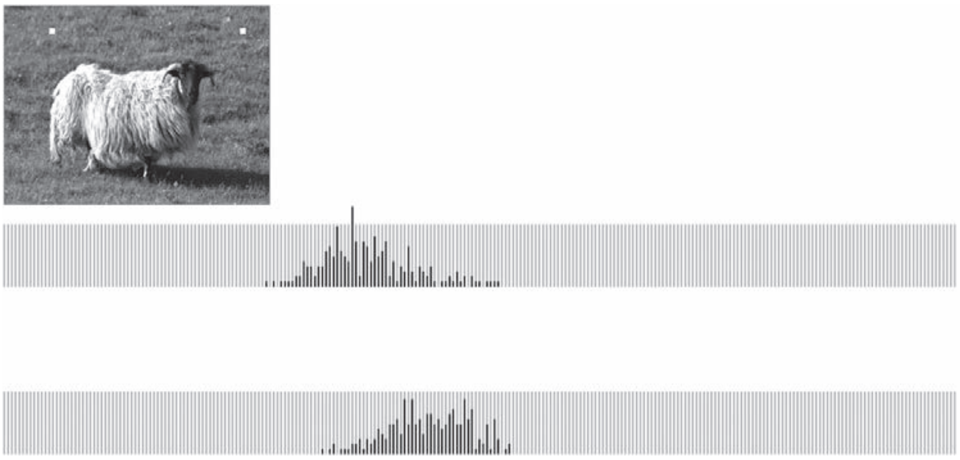

Fig. 2. Histograms of two texture windows belonging to the same "grass" texture

ference between the histograms based on the Euclidean distance is determined by the value (40.64 / 318.2) 100% = 12.77%. Accordingly, the percentage of similarity is 100 – 12.77 ≈ 87%.

The similarity value of the histograms presented in Figure 2, calculated by means of a component-wise comparison R^0 according to **Algorithm 1** (without taking into account the immediate surroundings of the histogram components) is 37%.

Then, the similarity value of the histograms was calculated taking into account the immediate surroundings of the histogram components. In the process of such calculating, the reducing coefficients K_1, K_2, K_3,…, K_m ,…, K_M were used (Section 5). The radius of the immediate surroundings of each column M was limited by 7. The coefficients K_1, K_2, K_3,…, K_m, …, K_7 were calculated using the formula $K_m = (1 – 0.1\ m)$. Using such coefficients led to the following result. The percentage of similarity of the histograms turned out to be equal to 67%.

Thus, compared to R^0, the percentage of similarity R^M calculated using the algorithm proposed in this paper increases by approximately 45%. (This percentage was obtained as a result of averaging in a series of similar experiments.)

This result is predictable. Indeed, in accordance with **Algorithm 2**, taking into account neighboring components of the compared histograms leads only to an increase in the degree of similarity between them, but in no case to a decrease. Moreover, with an increase in the radius of the nearest environment M, the percentage of similarity between the histograms only grows.

The following conclusion can be drawn from the experiments. For the algorithm proposed in this paper, the percentage of similarity between the histograms of different textures (such as "cow" and "grass") is 0%, and the percentage of similarity between the histograms of the same textures (such as "grass" and "grass") is 67%. That is, the range of change in the percentage of similarity from different to the same textures is approximately 70%. At the same time, for the algorithm based on the Euclidean distance, the percentage of similarity between the histograms of different textures (such as "cow" and "grass") is 72%. And the percentage of similarity for the same textures (such as "grass" and "grass") is 87%. That is, the range of change in the percentage of similarity from different to the same textures is only 15%. So, the advantage of the proposed algorithm is obvious for solving the problem of texture segmentation of images.

Discussion

A simple algorithm for measuring the similarity between histograms is presented. The proposed algorithm is intended for application, first of all, to texture segmentation of images using brightness histograms of texture windows as texture features. This algorithm was used in the texture segmentation models described in [15–18].

The texture segmentation algorithm sequentially extracts all areas of homogeneous fine-grained texture present in any image. As the main features characterizing the texture features of different areas of the image, the algorithm uses a brightness histogram and an orientation histogram built in texture windows covering the entire analyzed image. The texture similarity of different areas of the image is determined by comparing their histograms, using the algorithm described above.

In order to correctly estimate the similarity value of histograms used as texture features and characterizing the texture peculiarities of different areas of an image, it turned out to be appropriate to develop a special algorithm, presented in this paper. The need to develop such a special algorithm is justified by the fact that the methods for estimating the similarity/difference measure described in the literature provide such similarity/difference measures of histograms that are not very suitable for solving the problem of texture segmentation (Section 4).

The main advantage of the proposed algorithm, compared to popular methods of calculating similarity/difference between objects (vectors), is that the range of similarity between the compared histograms (from complete similarity to complete difference) is 100%, while popular methods can offer several times smaller ranges of similarity percentage.

It should be noted that, in contrast to the component-wise comparison of histograms, a characteristic feature of the proposed algorithm is the fact that this algorithm gives, in general, higher percentages of similarity.

It is well known that "texture" is an intuitive concept that does not have a formal or generally accepted definition [31]. Thus, the problem of texture segmentation belongs to the category of fuzzy problems, these statements are also true for the concept of "similarity". It follows that all methods and algorithms proposed for solving the problem of texture segmentation and the problem of calculating similarity are heuristic. Accordingly, the algorithm for determining the similarity measure between histograms proposed in the article is also heuristic.

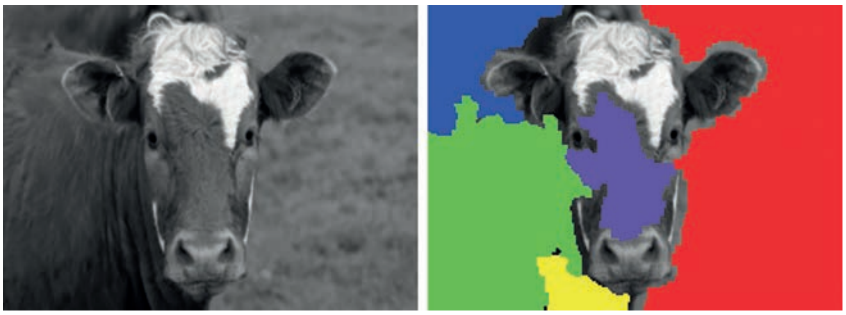

Fig. 3. Results of texture segmentation of a grayscale image (left part of the figure). The right part of the figure shows the 5 largest texture segments extracted by the model (in different colors)

The algorithm for calculating the similarity measure between histograms was implemented as a computer program as a component of a model that solves the problem of segmenting a visual image into homogeneous texture areas. It is worth noting that the proposed histogram comparison algorithm calculates the similarity measure between histograms very quickly, since it uses only simple operations.

Currently, such algorithm parameters as the texture window size, the similarity threshold by which similar texture windows are distinguished from dissimilar ones, and the radius of the nearest neighbor *M* are selected heuristically, and the correctness of their choice is verified experimentally — by comparing the results of image segmentation with human common sense.

The effectiveness of the algorithm for texture segmentation of images into homogeneous texture areas is confirmed by the results in the experiments on natural image processing. The results obtained in the experiments demonstrate the effectiveness of the algorithm and show that the algorithm performs correct (from a human point of view) texture segmentation of a wide range of images [15–18]. Thus, the effectiveness of the key operation of the segmentation algorithm, the histogram comparison algorithm, is indirectly confirmed.

Figure 3 is presented as an example, which demonstrates the results of the texture segmentation of a landscape image (upper part of the figure).

The lower part of the figure shows the five largest homogeneous texture segments extracted by the algorithm.

Conclusions

An algorithm for calculating the similarity between histograms for texture segmentation is proposed. A characteristic feature of the proposed algorithm is the fact that this algorithm gives high percentages of similarity. The algorithm implemented as a computer program as a component of a model that solves the problem of segmenting a visual image into homogeneous texture areas. The effectiveness of the algorithm is confirmed by the results in the experiments on natural image processing.

REFERENCES

1. Kussul E.M., Rachkovskij D.A., Baidyk T.N. On image texture recognition by associativeprojective neurocomputer. *ANNIE'91 Conf., Intelligent Engineering Systems through Artificial Neural Networks,* St. Louis, 1991, 453–458.
2. Kussul E.M., Baidyk T.N., Lukovitch V.V., Rachkovskij D.A. Adaptive neural network classifier with multifloat input coding. *6th Int. Conf. "Neuro-Ni mes 93",* Nimes, 1993, 209–216.
3. Kussul E.M, Baidyk T.N., Wunsch D.C. *Neural Networks and Micro Mechanics*. Springer, New York, 2010, 210 p. https://doi.org/10.1007/978-3-642-02535-8
4. Goltsev A. An assembly neural network for texture segmentation. *Neural Networks,* 1996, Vol. 9 (4), 643–653. https://doi.org/10.1016/0893-6080(95)00136-0
5. Goltsev A., Wunsch D.C. Inhibitory connections in the assembly neural network for texture segmentation. *Neural Networks*, 1998, Vol. 11 (5), 951–962. https://doi.org/10.1016/S0893-6080(98)00053-7
6. Goltsev A.D. *Neural networks with the assembly organization*. Naukova Dumka, Kyiv, 2005, 200 p. [In Russian]
7. Melendez J., Puig D., Garcia M.A. Multi-level pixel-based texture classificati on through efficient prototype selection via normalized cut. *Pattern Recognition,* 2010, Vol. 43 (12), 4113–4123. https://doi.org/10.1016/j.patcog.2010.06.014
8. Freixenet J., Munoz X., Raba D., Marti J., Cuff X. Yet another survey on image segmentation. *7th European Conf. on Computer Vision (ECCV),* Copenhagen, 2002, 408–422. https://doi.org/10.1007/3-540-47977-5_27
9. Rousson M., Brox T., Deriche R. Active unsupervised texture segmentation on a diffusion based feature space. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'03): June 16 – June 22 2003,* Madison, 2003, Vol. 2, 699–704. https://doi.org/10.1109/CVPR.2003.1211535
10. Clausi D.A., Deng H. Design-based texture feature fusion using Gabor filters and cooccurrence probabilities. *IEEE Transactions on Image Processing,* 2005, Vol. 14 (7), 925–936. https://doi.org/10.1109/TIP.2005.849319
11. Wei H, Bartels M. Unsupervised segmentation using Gabor wavelets and statistical features in LIDAR data analysis. *18th Int. Conf. on Pattern Recognition (ICPR'06): August 20 – August 24 2006*, Hong Kong, 2006, Vol. 1, 667–670. https://doi.org/10.1109/ICPR.2006.1145
12. Yang A.Y., Wright J., Ma Y., Sastry S.S. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding,* 2008, Vol. 110 (2), 212–225. https://doi.org/10.1016/j.cviu.2007.07.005
13. Sebe N., Lew M.S. Texture features for content-based retrieval. In: M.S. Lew (Ed.), *Principles of Visual Information Retrieval.* Advances in Computer Vision and Pattern Recognition: Springer, 2001, 51–85. https://doi.org/10.1007/978-1-4471-3702-3_3
14. Hafner J.L., Sawhney H.S., Equitz W., Flickner M., Niblack W. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. PAMI,* 1995, Vol. 17 (7), 729–736. https://doi.org/10.1109/34.391417
15. Goltsev A., Gritsenko V. Algorithm of sequential finding the characteristic features of homogeneous texture regions for the problem of image segmentation. *Cybernetics and Computer Engineering,* 2013, Vol. 173, 25–34. [In Russian]
16. Goltsev A., Gritsenko V., Kussul E., Baidyk T. Finding the texture features characterizing the most homogeneous texture segment in the image. *Lecture Notes in Computer Science*, Springer, 2015, Vol. 9094, Part I, 287–300. https://doi.org/10.1007/978-3-319-19258-1_25
17. Goltsev A., Gritsenko V., Husek D. Extraction of homogeneous fine-grained texture segments in visual images. *Neural Network World,* 2017, Vol. 27, 447–477. https://doi.org/10.14311/NNW.2017.27.024
18. Goltsev A., Gritsenko I.V., Husek D. Segmentation of visual images by sequential extracting homogeneous texture areas. *Journal of Signal and Information Processing,* 2020, Vol. 11, 75–102. https://doi.org/10.4236/jsip.2020.114005

19. Fan J.P., Zeng G.H., Body M., Hacid M.S. Seeded region growing: an extensive and comparative study. *Pattern Recognition Letters,* 2005, Vol. 26, 1139–1156. https://doi.org/10.1016/j.patrec.2004.10.010
20. Kamdi S., Krishna, R.K. Image segmentation and region growing algorithm. *International Journal of Computer Technology and Electronics Engineering,* 2012, Vol. 2, 103–107.
21. Peng Q. An image segmentation algorithm research based on region growth. *Journal of Software Engineering,* 2015, Vol. 9, 673–679. https://doi.org/10.3923/jse.2015.673.679
22. Cleary J.G. Analysis of an algorithm for finding nearest neighbors in Euclidean space. *ACM Trans. on Math. Softw.,* 1979, Vol. 5 (2), 183–192. https://doi.org/10.1145/355826.355832
23. Rubner Y., Tomasi C., Guibas L.J. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision.,* 2000, Vol. 40 (2), 99–121.
24. Deza M., Deza E. *Encyclopedia of Distances.* Berlin, Heidelberg, Springer, 2014, 733 p. https://doi.org/10.1007/978-3-662-44342-2
25. Rachkovskij D.A. *Introduction to Fast Similarity Search.* Kyiv, Interservice, 2019, 294 p. [In Russian]
26. Gricenko V.I., Rachkovskij D.A. *Methods of Vector Representation of Objects for Fast Similarity Assessment.* Naukova Dumka, Kyiv, 2019, 312 p. [In Russian]
27. Rachkovskij D.A. *Codevectors: Sparse Binary Distributed Representations of Numerical Data.* Interservice, Kyiv, 2019, 200 p. [In Russian]
28. Rachkovskij D.A. Representation of spatial objects by shift-equivariant similarity-preserving hypervectors. *Neural Computing and Applications*, 2022, Vol. 34 (24), 22387–22403. https://doi.org/10.1007/s00521-022-07619-1
29. Goltsev A., Holtsev O. An algorithm for measuring the similarity of histograms for texture image segmentation. *WSEAS Transactions on Information Science and Applications,* 2024, Vol. 21, 107–115. https://doi.org/10.37394/23209.2024.21.11
30. Jahne B., Scharr H., Korkel S. Principles of Filter Design. In: B. Jahne, H. Scharr, S. Korkel, B. Jahne, H. Haussecker, P. Geissler (Eds.), *Handbook of Computer Vision and Applications*, Vol. 2: Signal Processing and Pattern Recognition, Academic Press, San Diego, 2000, 125–152.
31. Ruiz-Del-Solar J. Neural-based architectures for the segmentation of textures. *Proceedings of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, 2000, Vol. 3, 1080–1083. https://doi.org/10.1109/ICPR.2000.903733

Received 30.09.2024

О.Д. ГОЛЬЦЕВ, канд. техн. наук, старш. наук. співроб., зав. відділу,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
https://orcid.org/0000-0002-2961-0908
root@adg.kiev.ua

О.О. ГОЛЬЦЕВ, аспірант,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
https://orcid.org/0000-0002-1846-6648
rcwolf@adg.kiev.ua

І.В. СУРОВЦЕВ, д-р техн. наук, старш. наук. співроб., зав. відділу,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
https://orcid.org/0000-0003-1133-6207
igorsur52@gmail.com

АЛГОРИТМ ОБЧИСЛЕННЯ ПОДІБНОСТІ МІЖ ГІСТОГРАМАМИ ДЛЯ СЕГМЕНТАЦІЇ ТЕКСТУРИ

Вступ. Наведено алгоритм обчислення ступеня подібності між багатовимірними гістограмами. Запропонований алгоритм призначений для текстурної сегментації зображень з використанням гістограм як текстурних ознак. Необхідність розробки такого спеціального алгоритму обґрунтована тим фактом, що методи оцінки міри подібності / відмінності між багатовимірними векторами, описані в літературі, надають такі міри, які не дуже підходять для вирішення задачі текстурної сегментації. Основною особливістю запропонованого алгоритму є те, що під час обчислення значення подібності він враховує не лише відповідні компоненти гістограми, а й їхні найближчі сусідні компоненти. Завдяки цьому алгоритм більш адекватно оцінює подібність гістограм. Запропонований алгоритм реалізовано у вигляді комп'ютерної програми як складової частини моделі сегментації зображення. Ефективність алгоритму порівняння гістограм опосередковано підтверджується результатами текстурної сегментації моделі сегментації зображення в експериментах з оброблення різноманітних зображень, у тому числі природних ландшафтів.

Методи. Розглянуто задачу обчислення подібності гістограм. Запропоновано спеціальний алгоритм, оскільки аналогічні методи, описані в літературі, не дуже підходять для вирішення задачі текстурної сегментації. Основною особливістю алгоритму є те, що він враховує як відповідні компоненти гістограми, так і їхні найближчі сусідні компоненти. Завдяки цьому алгоритм більш адекватно оцінює подібність гістограм. Алгоритм реалізовано у вигляді комп'ютерної програми. Ефективність алгоритму опосередковано підтверджується результатами текстурної сегментації моделі сегментації зображення в експериментах з обробки різноманітних зображень, у тому числі природних ландшафтів.

Мета. Метою цієї роботи є розробка ефективного алгоритму для оцінки подібності гістограм, таких як гістограми яскравості та гістограми орієнтації вікон текстури. Алгоритм засновано на ідеї врахування не тільки відповідних компонентів обох гістограм, а й компонентів їхнього найближчого оточення.

Результати. Основною перевагою запропонованого алгоритму, порівняно з популярними методами обчислення подібності / відмінності між об'єктами (векторами) є те, що діапазон подібності між порівнюваними гістограмами (від повної подібності до повної відмінності) становить 100%, тоді як популярні методи можуть запропонувати в кілька разів менші діапазони відсотка подібності.

Висновки. Запропонований алгоритм забезпечує широкий діапазон подібності порівнюваних гістограм, який становить 100% (від повної подібності до повної відмінності), тоді як популярні методи можуть запропонувати в кілька разів менші діапазони відсотка подібності. Алгоритм реалізовано у вигляді комп'ютерної програми як складової моделі, що розв'язує задачу сегментації візуального зображення на однорідні текстурні ділянки. Варто зазначити, що запропонований алгоритм порівняння гістограм дуже швидко обчислює міру подібності між гістограмами, оскільки використовує лише прості операції.

Ефективність алгоритму текстурної сегментації зображень на однорідні текстурні області підтверджена результатами експериментів з обробки природних зображень. Результати, отримані в експериментах, демонструють ефективність алгоритму та показують, що алгоритм виконує коректну (з точки зору людини) текстурну сегментацію зображень широкого діапазону. Отже, опосередковано підтверджується ефективність ключової операції алгоритму сегментації — алгоритму порівняння гістограм.

Ключові слова: *оброблення зображень, подібність гістограм, текстурні ознаки, сегментація текстури*.