

УДК 004.9: 519.237.8

O. G. Baybuz, M. G. Sydorova, Y. D. Siniechina, A. D. Siniechina

Oles Honchar Dnipropetrovsk National University

SOFTWARE «PROIMG» OF CONTENT-AWARE CATALOGING, SEARCHING AND EDITING IMAGES

Розроблено математичне та програмне забезпечення каталогізації, пошуку та редагування зображень, що враховує та аналізує їх змістове наповнення для отримання більш якісних та ефективних результатів.

Ключові слова: каталогізація зображень; кластерний аналіз; класифікація; перцептуальні хеші; алгоритм seam carving.

Разработано математическое и программное обеспечение каталогизации, поиска и обработки изображений, учитывающее и анализирующее их содержательное наполнение для получения более качественных и эффективных результатов.

Ключевые слова: каталогизация изображений; кластерный анализ; классификация; перцептуальные хэши; алгоритм seam carving.

The mathematical and software of cataloging, searching and processing of images has been developed, taking into account and analyzing content for obtaining more qualitative and effective results.

Keywords: image cataloging; cluster analysis; classification; perceptual hashes; algorithm seam carving.

Problem statement. A tendency to storage a large amount of graphic information has been observed recently both in various spheres of professional activity and in everyday human life. The need for its systematization, processing and editing makes a task of developing the appropriate software tools actual.

The following tasks are most often appeared while working with images:

- systematization (cataloging) of a large amount of data for convenient storage, quick access, adding new files to existing directories;
- search for similar images and duplicates;
- image resizing, which can be caused by different factors, like viewing in many different display devices with different resolutions in Internet, using in design solutions, etc.;
- removal of unnecessary objects.

It is important to use algorithms which consider and analyze the image content at an acceptable time to achieve better and more efficient results in each of these tasks.

Literature review and problem statement. Since the tasks of systematization and image processing are very relevant, there are various commercial software tools for solving them. However, each of them has both advantages and disadvantages.

Image resizing is a standard tool in many image processing applications. Simple attempts include scaling and cropping, which lead to loss of important features or distortion. More effective resizing can only be achieved by considering the image content and not only geometric constraints. One of these approaches is seam carving [1–2], which allows to change the image size by finding so-called seams of the least important pixels and their further removal or duplication. In addition to resizing this approach is also used to remove unnecessary objects defined by the user.

The problem of program search for identical images (duplicates) is quite often arises in practice. One of the most common approaches of solving the image recognition problem is the use of neural networks. This way, you can find not only full copies of images, but also similar images. However, you need to have a large amount of training data while using this approach to search for a duplicate, and this can be a problem, since it is impossible to know in advance, for which image the duplicate may appear. A bitwise comparison is also not a good idea for a large number of files, as it will require significant computing costs, in addition, images can be modified (have different size, quality, or watermark), remaining the same for human perception. An interesting approach for comparing images is perceptual hash algorithms [3–4], which are based on the analysis of the appearance of the image. Unlike classical hash functions, perceptual algorithms generate hashes that are used to compare input data. The more inputs were similar, the closer Hash values of such algorithms are.

Image cataloging is to group files that are similar in some parameters (keywords, date, geotags, content). A classic task of detecting similar groups of objects is a cluster analysis [5–6].

The aim of this work is to develop effective computing technology based on methods of cluster analysis, classification, perceptual hash algorithms and seam carving; to create modern software for systematizing and editing images which considers the analysis of content; to conduct a research of the quality of the results obtained in different sets of data.

Main material. To accomplish the assigned task, the software was developed in the programming language C#. It consists of two separate modules:

- systematization of images (cataloging by content, searching for duplicates, searching for similar images, determining the most relevant directory when adding new files);
- image editing (content-aware resizing, image-resizing with the main objects preservation (automatic and manual detecting), protecting them from deformation, removing unnecessary objects defined by the user, calculating gradient energy functions and Sobel operator with constructing an energy map based on it).

Let us consider in more detail the basic operations and computational schemes that were laid in their basis.

Image comparison can be both an independent (search for duplicates and similar images) and auxiliary task in clustering and classification methods. The "PROIMG" system implements perceptual hash algorithms [3–4] which show good results. Also, they are fast and do not require the additional data for learning. The Heming metric is used to compare the images with their perceptual hash functions.

aHash (Average hash) algorithm:

1. Reduce the size of the image (8x8 pixels are offered, scaling can be performed without keeping the aspect ratio). The size of the image affects the accuracy of the comparison and the speed of the algorithm.

2. Convert the image to a grayscale using formula $Y = 0.299R + 0.587G + 0.114B$, this will reduce the hash size three times. The Y value is also known as luminance.

3. Calculate the average color: $avg = \frac{\sum_{i=1}^8 \sum_{j=1}^8 img[i, j]}{8 \times 8}$.

4. Compute the bits (each pixel is replaced by zero if it is less than the average value calculated at the third step and by one otherwise) and construct the hash.

pHash (Perceptive hash) algorithm differs from aHash by adding another stage which performs DCT (Discrete Cosine Transform) to separate the image into the parts of "importance" which affect the image quality::

$$F(u, v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} A(i) A(j) \cos\left[\frac{\pi u}{2N} (2i+1)\right] \cos\left[\frac{\pi v}{2M} (2j+1)\right] \cdot f(i, j),$$

where $A(\xi) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{when } \xi = 0 \\ 1, & \text{when } \xi \neq 0 \end{cases}$, M, N – determine the size of the

input matrix (32x32); $f(i, j)$ – pixel intensity value (i, j) , $F(u, v)$ – coefficient from the DCT matrix located in line u and in column v .

The low frequencies are significant for most images which will be located in the upper left corner of the DCT matrix. The DCT matrix is reduced by rejecting the insignificant (high) frequencies. At the result a smaller block (8x8) is obtained.

dHash (Difference hash) algorithm pays attention to the gradient of the image:

1. Reduce the size of the image to 9x8 (in general $(N + 1) \times N$).
2. Convert the image to a grayscale.
3. Compute the difference between the adjacent pixels $y_{i,j} = x_{i,j+1} - x_{i,j}$. Consequently, the 8x8 matrix is obtained.
4. Compute the bits and construct the hash. If pixel less than previous it is replaced by zero and by one otherwise.

gHash (Gradient hash) algorithm differs from dHash by resizing the image to 32x32. Also the gradient is tracked not by pixels, but by columns and rows (each column is replaced by 1 if it is greater than the sum of pixels in comparison with the next column and by 0 otherwise. Then this value is written to the end of the bit chain. Similar actions are performed with the rows).

Fig. 1 shows the result of finding similar images to the given image with a similarity coefficient of 80 %.

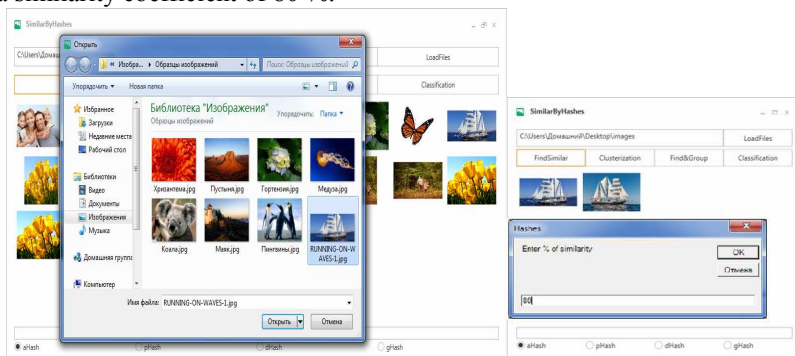


Figure 1 – Searching for similar images

The research of perceptual hash algorithms was carried out according to three criteria: the speed of algorithms, the ability to distinguish images and the resistance of algorithms to different image modifications (scaling and change of brightness).

To research the speed of algorithms a set of random images was used. The results are shown in the Table 1.

Table 1 – Results of researching the speed of algorithms

	aHash	pHash	dHash	gHash
Average time per image, ms	78	632	89	96

It is clear that all algorithms, except for pHash, have roughly the same speed according to the results of this experiment. The pHage algorithm requires more time since more complex calculations are used.

For researching the properties of perceptual hash algorithms to distinguish the images two sets of data were considered: a set of 20 random images with very different motifs and a set of high quality images with one motif (gulls) [7].

According to the results (Table 2), the gHash algorithm demonstrates the best ability to distinguish the images.

Different image manipulation operations were used to verify the robustness of perceptual hash functions.

The images were scaled to a width of 1024 pixels and the height was proportional. After that, the images were compared with the original ones. The results of this research are shown in Table 3.

Table 2 – Results of researching the ability to distinguish images

	pHash	aHash	dHash	gHash
<i>Set of random images:</i>				
Average distance	35,4	24,5	37,5	38,3
Maximum	47	39	48	44
Minimum	21	25	20	26
<i>Set of images with one motif:</i>				
Average distance	25,5	15,7	25,8	28,3
Maximum	29	19	28	32
Minimum	12	2	19	20

Table 3 – Results of researching the resistance to scaling

	aHash	pHash	dHash	gHash
Average distance	0,36	0,09	0,72	0,63
Maximum	1	1	3	3
Minimum	0	0	0	0

According to the results it is clear that all algorithms are stable to image scaling. The best results were shown by the pHash algorithm.

To test the resistance of the resistant to changes of image brightness each image was changed by increasing the brightness by 30 % and then compared with the original image. The results of this research are shown in Table. 4.

Table 4 – Results of researching the hash functions after changing brightness

	aHash	pHash	dHash	gHash
Average distance	2,5	8	2,25	2,75
Maximum	4	13	5	5
Minimum	0	4	0	2

Apparently, all algorithms are very resistant to changes of image brightness. It can be concluded that the gHash algorithm is the most versatile one. If you need to find not only the duplicates (or fuzzy duplicates) of images, but also the similar images, you should apply the aHash algorithm.

Cataloging and identifying similar image groups. Hierarchical clustering algorithms have been developed to solve this problem. The important feature of them is the possibility of constructing a dendrogram which demonstrates the hierarchy of images in directory according to similarity (Fig. 2).

Perceptual hashes, which were discussed previously, were used as a metric of similarity.

Clustering can also be used as an automatic search for all duplicates with subsequent deletion by user request (Fig. 2).

Images classification and search for the most relevant directory. When adding a new image, it may be a problem to determine the directory that will good fit by content already located therein images. Methods of classification are proposed to solve this problem. The «PROIMG» system implements the computational scheme of the STOLP algorithm, the results of which are shown in Fig. 3. Studies have shown that the least recognition error ensures the use of the pHash algorithm.

Content-aware image resizing. The created software implements a seam carving algorithm, which is an image resizing approach that deliberately removes or duplicates pixels of non-important parts of the image. This method uses an energy function that determines the importance of each pixel. A seam is a connected path of low energy pixels, which passes through the image from top to bottom (vertical) or from left to right (horizontal). By sequentially removing or duplicating seams you can both increase or decrease the size of the image in both directions.

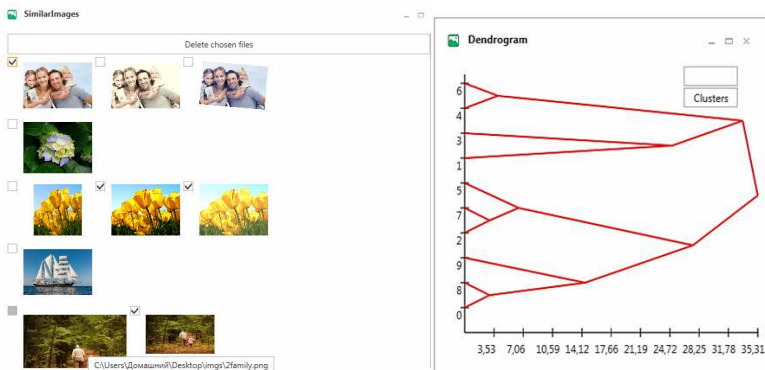


Figure 2 – Results of image clustering

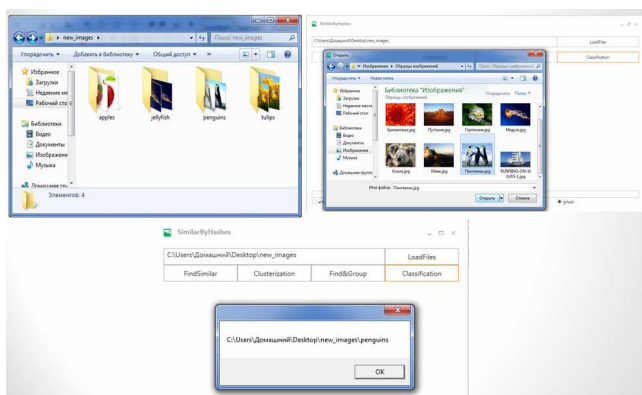


Figure 3 – Classification results

Implementation of seam carving is carried out on the basis of the following scheme:

1. Calculate the energy of each pixel using the Sobel operator [8] and gradient in both directions x and y . The energy function is defined as follows:

$$e(I) = \left| \frac{\partial I}{\partial x} \right| + \left| \frac{\partial I}{\partial y} \right|, \text{ where } I \text{ is an } n \times m \text{ image.}$$

2. Having the energy function $e(I)$, define the cost of a seam as:

$$e(s) = e(I_s) = \sum_{i=1}^n e(I(s_i)). \text{ We need to find the optimal seam}$$

$s^* = \min_s e(s) = \min_s \sum_{i=1}^n e(I(s_i))$. It can be found using dynamic programming.

3. Compute the cumulative minimum energy M for all possible connected seams by the formula:

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1)).$$

4. Detect the optimal seam as follows: first, we find pixel in the last row (for vertical seam) with the lowest cumulative energy. It is used as the starting point. Then from the three adjacent pixels above we choose one with the minimum cumulative energy and so on.

5. Manipulate with the optimal seam (remove or duplicate) depending on the task (reducing or enlarging the image).

An example of the results of finding optimal seams and Sobel energy map is shown in Fig. 4.

The system also allows user to select areas on the image that definitely do not need to be deformed, for more correct operation of the algorithm.

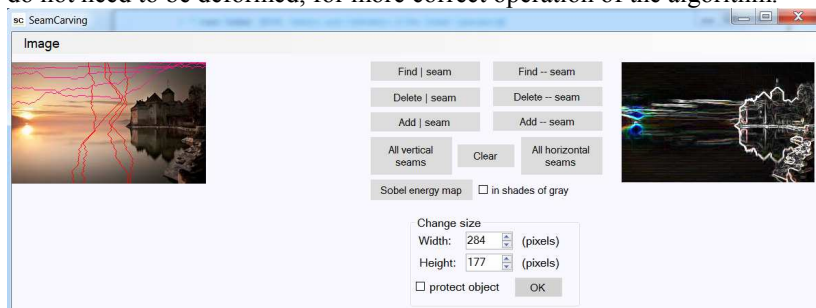


Figure 4 – Result of finding optimal seams and Sobel energy map

Removing objects, detected by user: This seam carving approach is also used to remove objects from an image. The following computational scheme is implemented:

1. generate the image energy map;
2. define the image cumulative energy map;
3. set the minimum energy values for $M(i,j)$, if (i,j) is detected pixel by user;
4. for each iteration:
 - define the optimal seam (in needed direction – horizontal or vertical);

- remove the optimal seam from the image, by that reducing the image size.

The results of the algorithm are shown in Fig. 5.

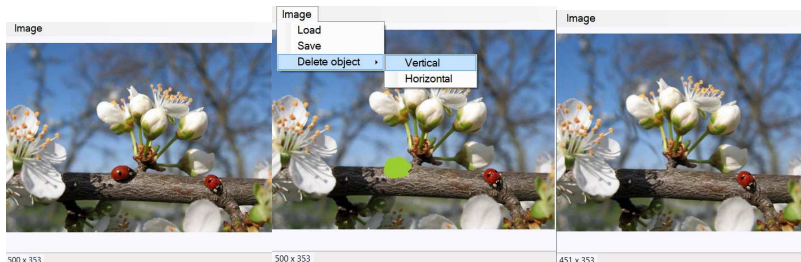


Figure 5 – Results of object removal

The developed program can work with various image formats (.jpg, .bmp, .png), and also read files from subdirectories, if any. If there are other files besides the images in the directory, this will not affect the operation of the program.

Conclusions. The computational schemes and software «PROIMG» were developed for the systematization and processing images considering their content on the basis of clustering, classification, perceptual hash algorithms and seam carving methods. The analysis of the work of perceptual hash algorithms is carried out according to three criteria: speed of algorithms, ability to distinguish images, stability of algorithms to image modifications (scaling and change of brightness). In further research it is planned to expand the functionality of the system and optimize its operation.

References

1. Avidan S., Shamir A. Seam carving for content – aware image resizing // ACM Transactions on Graphics (TOG) Proceedings of ACM SIGGRAPH, Vol. 26. Issue 3. 2007. URL: <http://www.faculty.idc.ac.il/arik/SCWeb/imret/imret.pdf>
2. Alnammi M. Implementing Seam Carving for Image Resizing // CS 766 – Final Repor, 2016. URL: http://pages.cs.wisc.edu/~moayad/cs766/download_files/alnammi_cs_766_final_report.pdf
3. Zauner C. Implementation and Benchmarking of Perceptual Image Hash Functions. Ph.D. thesis. University of Applied Sciences, Hagenberg, 2010. 107 p.
4. Рудаков И. В., Васютович И. М. Исследование перцептивных хеш-функций изображений // Наука и образование: научное издание

МГТУ им. Н.Э. Баумана. Москва: МГТУ им. Н.Э. Баумана, 2015. № 8. 269–280 с.

5. Berkhin P. A survey of clustering data mining techniques // Grouping Multidimensional Data. 2006. P. 25–71.

6. Миркин Б. Г. Методы кластер-анализа для поддержки принятия решений: обзор. Москва, 2011. 88 с.

7. Quality images. URL: https://commons.wikimedia.org/wiki/_Commons:Quality_images

8. Sobel I. History and Definition of the Sobel Operator, 2014. URL: https://www.researchgate.net/publication/239398674_An_Isotropic_3_3_Image_Gradient_Operator

Надійшла до редколегії 3.07.17