DOI: https://doi.org/10.15407/kvt211.01.029

KRYGIN V.M.[1],
Junior Researcher of Pattern Recognition Department,
https://orcid.org/0000-0002-9000-1685, e-mail: valeriy.krygin@gmail.com
KHOMENKO R.O.[2], Programmer
https://orcid.org/0000-0001-7640-4077, e-mail: ruslank3584@gmail.com
MATSELLO V.V.[1], PhD (Engineering), Senior Researcher,
Head of Pattern Recognition Department
https://orcid.org/0000-0002-6969-1554, e-mail: matsello@gmail.com
[1] International Research and Training Center for Information Technologies and Systems of the National Academy of Sciences of Ukraine and Ministry of Education and Science of Ukraine,
40, Acad. Glushkova av., Kyiv, 03187, Ukraine
[2] Postindustria Inc., 1935 Walgrove av., Los Angeles CA 90066, USA.

EXPERIMENTAL VERIFICATION OF THE SELF-DRIVEN ALGORITHMS FOR SOLVING MAX-SUM LABELING PROBLEMS

Introduction. *Max-sum labeling problems play an essential role in modern pattern recognition and can be used with other methods and a stand-alone approach. An essential step in building a pattern recognition system is the choice of an algorithm to solve the problem, which may require experimentation with different algorithms. This fires a need for software that allows solving different problems with the help of different algorithms for further analysis of the results of experiments and the final selection of the algorithm.*

The purpose of the paper *is to demonstrate the capabilities of the developed software for solving max-sum labeling problems.*

Results. *The software containing various algorithms for solving max-sum labeling problems was developed and experimentally tested. The program operation is shown on the example of image processing problems based on labeling: color image restoration, binary image denoising, posterization and binocular stereo vision.*

Conclusions. *The software described in the article verifies in practice the correctness of the self-driven algorithm for solving max-sum labeling problems. The application allows the operator to choose an algorithm for the labeling task and configure its parameters. This program will be helpful for developers of computer vision systems based on labeling problems and undergraduates, graduate students, and researchers studying structural pattern recognition methods.*

Keywords: *labeling problems, pattern recognition, computer vision, software.*

INTRODUCTION

Such applied problems of image recognition as segmentation [1–3], stereo vision [4, 5] and many others [6–10] are reducible to the solution of particular discrete optimization problems. Although they seem very different, they can be formal-

ized in a unified format, known as a labeling problem. These problems belong to the EXP-APX complexity class [11], so the general algorithm for their solution, even an approximate one, is very unlikely to exist. Known algorithms, described in [10], solve different subclasses of these problems or look for an approximate solution whenever possible. An important subclass is supermodular problems, but determining whether a problem is supermodular is a complex problem. To address these difficulties, works [12, 13] propose self-driven algorithms for solving labeling problems. The algorithm is guaranteed either to find one of the optimal labelings or to answer "the problem is not supermodular," ensuring the correctness of the answer. Self-driving means it is up to the algorithm to "decide" which of these two questions to answer.

Even though there are many different methods and algorithms for image recognition, developing applied systems is still time-consuming because each developer should use known recognition methods and create tools for debugging and fine-tuning the algorithms. Available libraries of computer vision provide significant assistance in creating image recognition systems. The most famous is **OpenCV** (Open Source Computer Vision Library) — a library containing algorithms for computer vision, image processing and general-purpose numerical algorithms [14]. Some libraries specialize in random fields, in particular, in solving labeling problems: **Darwin** [15], **DGM** (Direct Graphical Models C++ library) [16, 17], **libDAI** (Library for Discrete Approximate Inference in Graphical Models) [18], **OpenGM** [19, 20], **pgmpy** [21], and others.

The article describes software that implements algorithms for solving max-sum labeling problems, namely dynamic programming [10, 22], diffusion algorithm [10, 12], subgradient descent algorithm [10, 23], as well as self-driven algorithms [12, 13]. The examples of several practical computer vision problems illustrate the program's work.

FORMULATION OF LABELING PROBLEM

Let us introduce a finite set T of objects and a finite set D of labels and define a structure $\Gamma \subset 2^T$ of a neighborhood on the set of objects. Let us state that every element $\{t, t'\} = \gamma \in \Gamma$, $|\gamma| = 2$ of the neighborhood structure contains two objects called neighbors. Labeling is a function d: $T \to D$ that assigns a single label to each object. A pair $(t, \ell) \in T \times D$, $t \in T$, $\ell \in D$ is called a vertex, and a pair $((t, \ell), (t', \ell'))$ of vertices, where $\ell \in D$, $\ell' \in D$ and $\{t, t'\} \in \Gamma$, is called an edge. A function q: $T \times D \to R$ defines the weights of vertices, and a function q: $\Gamma \times D^2 \to R$ defines the weights of edges. The cost function is

$$G(d) = \sum_{t \in T} q_t(d(t)) + \sum_{\gamma \in \Gamma} g_\gamma(d(t), d(t')).$$

The labeling problem is finding such a labeling d that maximizes the cost function.

Dynamic programming [10, 22], diffusion algorithm [10, 12], and subgradient descent algorithm [10, 23], in particular self-driven [12, 13] are implemented in the program to solve the max-sum labeling problem.

The purpose of the paper is to demonstrate the capabilities of the developed software for solving max-sum labeling problems.

WAYS OF OBTAINING LABELING AFTER THE OPERATION OF AN ITERATIVE ALGORITHM

Three methods of obtaining labeling after the operation of the diffusion or subgradient descent algorithm have been implemented.

The naive method chooses such a label $d(t)$ for each object $t \in T$ from which the heaviest edge comes:

$$d(t) \in \arg\max_{\ell \in D} \max_{\substack{\ell' \in D \\ \{t,t'\} \Gamma}} g_{\{t,t'\}}(\ell, \ell').$$

The minimax method selects such labeling that minimizes the maximal gap ε between weights of the heaviest edges and the edges of the chosen labeling in each pair of neighbors:

$$d \in \arg\min_{d:T \to D} \max_{\{t,t'\} \in \Gamma} \varepsilon_{\{t,t'\}}(d(t), d(t'),$$

$$\varepsilon_{\{t,t'\}}(d(t), d(t')) = \max_{\substack{\ell \in D \\ \ell' \in D}} g_{\{t,t'\}}(\ell, \ell') - g_{\{t,t'\}}(d(t), d(t')).$$

The solution of the given minimax problem is implemented in the program as a generalized arc-consistency enforcing algorithm [10, 24, 25], where instead of operations $\vee$ and $\wedge$ we use min and max, respectively.

Self-driven algorithms can find the optimal labeling for a problem with integer weights or determine whether the problem is not supermodular. They are described in [12, 13].

EXAMPLES OF IMAGE PROCESSING PROBLEMS BASED ON LABELING

The program implements the solution to the following problems based on labeling problems: color image restoration, binary image denoising, posterization and binocular stereo vision [9]. The program uses a neighborhood structure with four neighboring pixels. Pixel brightness takes values from 0 to 255. The set of brightnesses is denoted by C.

Binary image denoising. The input is a grayscale image x: $T \to C$. The set of labels $D = \{\text{minC}, \text{maxC}\}$ contains two elements. The weights of vertices are

$$q_t(d) = -|x(t) - d|, \forall d \in D,$$

where $x(t)$ is the brightness of the pixel t.

Color image posterization. The input is a color image. For each color channel, the program solves the labeling problem separately. The output image uses the set of colors that the user indirectly determines by specifying it as an argument of the program the number $n \geq 2$ of desired colors (labels), the value of which is calculated as

$$D = \{(\max C - \min C) \cdot \frac{i}{(n-1)} + \min C : i = 0, n-1$$

The weights of vertices are

$$q_t(d) = -|x(t) - d|, \forall d \in D,$$

where $x(t)$ is the brightness of the pixel t.

Color image restoration. The input is a color image. For each color channel, the program solves the labeling problem separately. It uses the same set

$$D = \{(\max C - \min C) \cdot \frac{i}{(n-1)} + \min C : i = 0, n-1\}$$

of labels, which are used for the posterization problem. The task is to replace the pixel colors with those corresponding to their neighbors in the image. The user specifies a special brightness $c \in C$ (for example, 0), corresponding to the damaged parts of the image, indicating that their colors must be calculated depending on the colors of the neighboring pixels, regardless of their initial brightness. The weights of the vertices are

$$q_t(d) = -|x(t) - d| \cdot [x(t) \neq c], d \in D.$$

Binocular stereo vision. The task is to find a disparity map of two images. The input consists of two images T_1 and T_2 and the maximal disparity D_{max}. The set of labels is $D = \{0, \ldots, D_{max}\}$. The weights of vertices are

$$q_t(d) = -|x_1(t) - x_2(t + d(t)|, d \in D, t + d(t) \notin T_2,$$
$$q_t(d) = -\infty, d \in D, t + d(t) \notin T_2$$

where $x_1(t)$ is the brightness of the pixel t on the left image, $x_2(t)$ is the brightness of the pixel t on the right image, and $d(t)$ is the disparity of the pixel t on the right image relative to the corresponding pixel of the left image.

PROGRAM EXECUTION

The max-sum-2022.exe program should be executed in the command line with the algorithm's arguments given. Schematically, the program call looks like

```
max-sum-2022.exe
  problem--argument1 value1 --argument2 value2 ...
  penalty--argument1 value1 --argument2 value2 ...
  solver--argument1 value1 --argument2 value2 ...
  rounding--argument1 value1 --argument2 value2 ...
```

First, the name of the problem and its arguments are determined, then the type of penalty function with arguments, then the name of the algorithm for obtaining the solution of the problem and its arguments and for some algorithms, the method of obtaining the labeling must be chosen after the op-

eration of the algorithm for finding the optimum. Arguments can be specified in any order. The problem's name must be entered without additional marks and two minuses (--) must be written before the name of each argument. It can be omitted if an argument is told to have a default value. To use the value $+\infty$, the operator should write "inf". For example, to choose an untruncated penalty function, the operator can write--threshold inf or not use this argument because infinity is the default value.

EXAMPLES OF THE PROGRAM OPERATION

The following are examples of commands that launch the program to solve the specified problems with the specified arguments. Line breaks in the examples are for clarity.

Stereo vision problem with two input images: "left.ppm" and "right.ppm". The maximum shift is 15. Write the result to the file "disparity-map.png" in the current folder (Fig. 1).

Use the penalty function

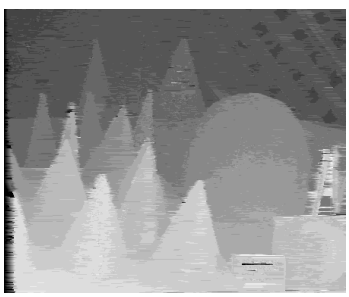
$$10 \cdot \min(9, |d - d'|^3)$$

and dynamic programming (Fig. 1):

```
max-sum-2022.exe
  stereo--left left.ppm--right right.ppm
  output disparity-map.png--max-disparity 15
  Power--smoothness 10--threshold 9--power 3
  dynamic-programming
```

a) Image "left.ppm"

b) Image "right.ppm"



c) Result "disparity-map.png"

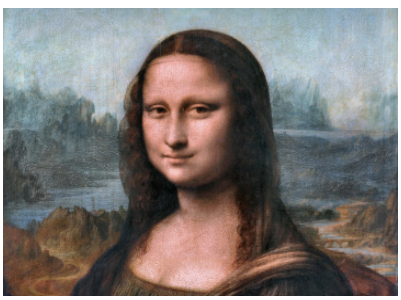
Fig. 1. An example of solving the problem of binocular stereo vision (0.2 seconds, source [26])

The task of restoring the image "damaged.png" by replacing all black pixels (color 0). Save the result in the file "inpainted.png" and use ten shades in each color channel. Use a linear penalty truncated to 2 with a smoothing of 0.2. That is a function $0{,}2 \cdot \min(2, |d - d'|)$. Perform a thousand iterations of the diffusion algorithm and then find the labeling according to the minimax principle (Fig. 2):

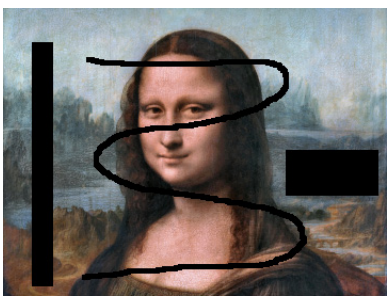
```
max-sum-2022.exe
  inpainting-picture damaged.png --output inpainted.png
  --empty-color 0--colors 10
  Linear--smoothness 0.2--threshold 2
  diffusion--iterations 1000
  minmax
```

The task of posterizing the image "colorful.jpg". Leave only two colors in each channel and save the result in the "posterized.png" file. Use a quadratic penalty function with a smoothing of 100. Perform ten iterations of the diffusion algorithm and apply a naive method of obtaining labels (Fig. 3):

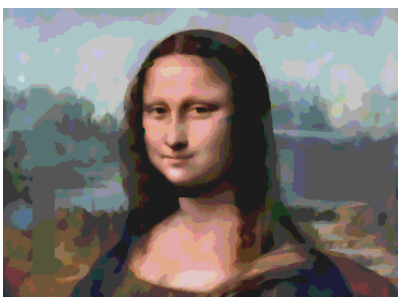
```
max-sum-2022.exe
  posterization  --picture colorful.jpg
  --output posterized.png
  --colors 2
  Quadratic--smoothness 100
  diffusion--iterations 10
  naive
```



a) The input image



b) Damaged image "damaged.png"



c) The output image "inpainted.png"

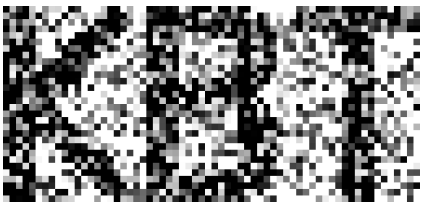
Fig. 2. An example of solving the problem of restoring an image fragment (7 minutes, source: https://commons.wikimedia.org/wiki/File: Da_Vinci%27s_Mona_Lisa_with_original _colors_approximation.jpg)

a) The image “colorful.jpg”

b) The output “posterized.png”

Fig. 3. An example of solving the problem of image posterization (2 seconds, source: https://commons.wikimedia.org/wiki/File:Colourful_Flower_01.JPG)

a) The original image



b) Noisy image “noisy.jpeg”

c) The output “restored.png”

Fig. 4. An example of solving an image recovery problem (16 minutes)

The task of denoising the binary image "noisy.jpeg". Save the result to the "restored.png" file. Use the Potts model [9] with smoothing 80. Check the solution every 100 iterations of subgradient descent. The step length for iteration n is calculated according to the formula $10 \cdot n^{-0,5}$. Use the self-driven approach to find the labeling (Fig. 4):

```
max-sum-2022.exe
  restoration--picturenoisy.jpeg--outputrestored.png
  Potts--smoothness 80
  gradient--iterations 100--initial-step 10
  --attenuation 0.5
  self-driven
```

Thus the software containing various algorithms for solving max-sum labeling problems was developed and experimentally tested. The program operation is shown on the example of image processing problems based on l-abeling: color image restoration, binary image denoising, posterization, and binocular stereo vision.

CONCLUSION

The software described in the article verifies in practice the correctness of the self-driven algorithm for solving max-sum labeling problems. The application allows the operator to choose an algorithm for the labeling task and configure its parameters. This program will be helpful for developers of computer vision systems based on labeling problems and undergraduates, graduate students and researchers studying structural pattern recognition methods.

REFERENCES

1. Ishikawa H., Geiger D. Segmentation by grouping junctions. *Proceedings of IEEE computer society conference on computer vision and pattern recognition* (cat. no.98CB36231), 1998, pp. 125–131.
2. Kovtun I.V. Technology of image texture segmentation on the basis of Markov random fields and solution of (max,+) problem. *Control Systems and Computers*. 2004, №2, pp. 61–66. (in Russian)
3. Held K. Markov random field segmentation of brain MR images. *Transactions on Medical Imaging. IEEE*, 1997, Vol. 16, № 6, pp. 878–886.
4. Schlesinger M.I., Flach B. Analysis of optimal labeling problems and their applications to image segmentation and binocular stereovision. East-west-vision 2002 (EWV'02). *International workshop & project festival computer vision, computer graphics, new media*. 2002, pp. 55–60.
5. Schlesinger D., Flach B., Shekhovtsov A. A higher order MRF-model for stereo-reconstruction. *Pattern recognition*. 2004, pp. 440–446.
6. Boykov Y., Veksler O., Zabih R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2001, Vol. 23, № 11, pp. 1222–1239.
7. Boykov Y., Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell*. USA: IEEE Computer Society. 2004, Vol. 26, № 9, pp. 1124–1137.
8. Schlesinger M.I., Gygynyak V.V. Solution of Structural Recognition (MAX,+)-problems by their Equivalent Transformations. Part 2. *Control Systems and Computers*. 2007, № 2, pp. 3–18. (in Russian)

9. Szeliski R. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2008, Vol. 30, № 6, pp. 1068–1080.
10. Savchynskyy B. Discrete graphical models — an optimization perspective. *Foundations and Trends® in Computer Graphics and Vision*. 2019, Vol. 11, № 3–4, pp. 160–429.
11. Li M., Shekhovtsov A., Huber D. Complexity of discrete energy minimization problems *Computer vision – ECCV 2016*, 2016, pp. 834–852.
12. Schlesinger M.I., Antoniuk K.V. Diffusion algorithms and structural recognition optimization problems. *Cybernetics and System Analysis*. 2011, № 2, pp. 3–12. (in Russian)
13. Krygin V., Khomenko R. Self-driven algorithm for solving supermodular (max,+) labeling problems based on subgradient descent. *Cybernetics and Sys. Anal.* 2022, Vol. 58, № 4. pp. 510–517.
14. Bradski G. The OpenCV Library. Dr. Dobb's Journal of Software Tools. 2000.
15. Gould S. DARWIN: A framework for machine learning and computer vision research and development. *The Journal of Machine Learning Research*. 2012, Vol. 13, № 1, pp. 3533–3537.
16. Kosov S. Direct graphical models C++ library. URL: http://research.project-10.de/dgm/, 2013.
17. Kosov S. Multi-layer conditional random fields for revealing unobserved entities: PhD thesis. Siegen University, 2018.
18. Mooij J.M. LibDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*. 2010, Vol. 11, pp. 2169–2173.
19. Andres B., Beier T., Kappes J.H. OpenGM: A C++ library for discrete graphical models. *CoRR*. 2012. Vol. abs/1206.0111.
20. Kappes J.H. A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision*. Springer US, 2015,Vol. 115, № 2, pp. 155–184.
21. Ankan A., Panda A. Pgmpy: Probabilistic graphical models using Python. *Proceedings of the 14th python in science conference* (SCIPY 2015). Citeseer, 2015.
22. Schlesinger M.I., Hlavac V. Ten Lectures on Statistical and Structural Pattern Recognition. Kyiv: Naukova dumka, 2004. (in Russian)
23. Shor N.Z. Minimization methods for non-differentiable functions. *Springer Series in Computational Mathematics*. 1985, Vol. 3,pp. 22–48.
24. Koval V.K., Schlesinger M.I. Two-dimensional programming in image analysis problems. *Automatics and Telemechanics*. 1976, V. 37, № 8. pp. 149–168. (in Russian)
25. Rossi F., Beek P. van, Walsh T. Handbook of constraint programming. Elsevier Science, 2006.
26. Scharstein, Daniel. High-accuracy stereo depth maps using structured light [Text] / Daniel Scharstein, Richard Szeliski. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. *Proceedings. IEEE*. 2003, Vol. 1 —2003, pp. 195–202.

Received 02.01.2023

ЛІТЕРАТУРА

1. Ishikawa H., Geiger D. Segmentation by grouping junctions. *Proceedings. 1998 IEEE computer society conference on computer vision and pattern recognition* (cat. no.98CB36231). 1998. P. 125–131.
2. Ковтун И.В. Технология текстурной сегментации изображений на основании марковских случайных полей и решения (max,+)-задач. *Управляющие системы и машины*. 2004. №2. С. 61–66.
3. Held K. et al. Markov random field segmentation of brain MR images. *Transactions on Medical Imaging. IEEE,* 1997. Vol. 16. № 6. P. 878–886.
4. Schlesinger M.I., Flach B. Analysis of optimal labeling problems and their applications to image segmentation and binocular stereovision. East-west-vision 2002 (EWV'02). *International workshop & project festival computer vision, computer graphics, new media.* 2002. P. 55–60.

5. Schlesinger D., Flach B., Shekhovtsov A. A higher order MRF-model for stereo-reconstruction. *Pattern recognition*. Berlin, Heidelberg, 2004. P. 440–446.
6. Boykov Y., Veksler O., Zabih R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2001. Vol. 23. № 11. P. 1222–1239.
7. Boykov Y., Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell. USA: IEEE Computer Society*. 2004. Vol. 26, № 9. P. 1124–1137.
8. Шлезингер М.И., Гигиняк В.В. Решение (MAX,+)-задач структурного распознавания с помощью их эквивалентных преобразований. Часть 2. *Управляющие системы и машины*. 2007. № 2. С. 3–18.
9. Szeliski R. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2008. Vol. 30. № 6. P. 1068–1080.
10. Savchynskyy B. Discrete graphical models — an optimization perspective. *Foundations and Trends® in Computer Graphics and Vision*. 2019. Vol. 11. № 3-4. P. 160–429.
11. Li M., Shekhovtsov A., Huber D. Complexity of discrete energy minimization problems. *Computer vision – ECCV 2016.* 2016. P. 834–852.
12. Шлезингер М.И., Антонюк К.В. Анализ алгоритмов диффузии для решения оптимизационных задач структурного распознавания. *Кибернетика и системный анализ*. 2011. № 2. С. 3–12.
13. Krygin V., Khomenko R. Self-driven algorithm for solving supermodular (max,+) labeling problems based on subgradient descent. *Cybernetics and Sys. Anal. USA: Kluwer Academic Publishers*. 2022. Vol. 58. № 4. P. 510–517.
14. Bradski G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. 2000.
15. Gould S. DARWIN: A framework for machine learning and computer vision research and development. *The Journal of Machine Learning Research*. 2012. Vol. 13. № 1. P. 3533–3537.
16. Kosov S. Direct graphical models C++ library. URL: http://research.project-10.de/dgm/, 2013.
17. Kosov S. Multi-layer conditional random fields for revealing unobserved entities: PhD thesis. Siegen University, 2018.
18. Mooij J.M. LibDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*. 2010. Vol. 11. P. 2169–2173.
19. Andres B., Beier T., Kappes J.H. OpenGM: A C++ library for discrete graphical models. *CoRR.* 2012. Vol. abs/1206.0111.
20. Kappes J.H. A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision*. Springer US, 2015. Vol. 115. № 2. P. 155–184.
21. Ankan A., Panda A. Pgmpy: Probabilistic graphical models using Python. *Proceedings of the 14th python in science conference* (SCIPY 2015). Citeseer, 2015.
22. Шлезингер М.И., Главач В. Десять лекций по статистическому и структурному распознаванию. Киев: Наукова думка, 2004.
23. Shor N.Z. Minimization methods for non-differentiable functions. *Springer Series in Computational Mathematics.* 1985. Vol. 3. P. 22–48.
24. Коваль В.К., Шлезингер М.И. Двумерное программирование в задачах анализа изображений. *Автоматика и телемеханика*. 1976. Т. 37. № 8. С. 149–168.
25. Rossi F., Beek P. van, Walsh T. Handbook of constraint programming. Elsevier Science, 2006.
26. Scharstein, Daniel. High-accuracy stereo depth maps using structured light [Text]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. *Proceedings IEEE*. Vol. 1. 2003. P. 195–202.

Отримано 02.01.2023

Кригін В.М.[1],
молодш. наук. співроб., відд. розпізнавання образів
https://orcid.org/0000-0002-9000-1685,
e-mail: valeriy.krygin@gmail.com
Хоменко Р.О.[2],
програміст,
https://orcid.org/0000-0001-7640-4077,
e-mail: ruslank3584@gmail.com
Мацелло В.В.[1], канд. техн. наук, старш. наук. співроб.,
завідувач відділу розпізнавання образів
https://orcid.org/0000-0002-6969-1554, e-mail: matsello@gmail.com

[1] Міжнародний науково-навчальний центр
інформаційних технологій та систем
НАН України та МОН України,
40, пр. Акад. Глушкова, м. Київ, 03187, Україна,
[2] Postindustria Inc.,
1935 Walgrove av., Los Angeles CA 90066, USA.

ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА АЛГОРИТМІВ РОЗВ'ЯЗАННЯ (MAX,+)-ЗАДАЧ РОЗМІТКИ ІЗ САМОКОНТРОЛЕМ

Вступ. *В сучасному розпізнаванні образів важливу роль відіграють (max, +)-задачі розмітки, які використовуються як самостійно, так і у сукупності з іншими методами. Важливим кроком побудови системи розпізнавання образів є вибір алгоритму розв'язання задач, що може потребувати експериментування з різними алгоритмами. Через це виникає необхідність у програмному забезпеченні, яке даватиме змогу розв'язувати різні задачі за допомогою різних алгоритмів з метою подальшого аналізу результатів експериментів та остаточного вибору алгоритму.*

Мета. *Продемонструвати можливості розробленого програмного забезпечення для розв'язання (max, +)-задач розмітки.*

Результати. *Розроблено програмне забезпечення, яке містить різні алгоритми розв'язання (max, +)-задач розмітки. Роботу програмного застосунку продемонстровано на прикладах реставрації кольорового зображення, знешумлення бінарного зображення, постеризації та бінокулярного стереозору.*

Висновки. *Розроблений програмний застосунок забезпечує перевірку правильності самокерованого алгоритму розв'язування задач маркування максимальної суми для виконання практичних завдань. Застосунок дає змогу оператору вибрати відповідний алгоритм для розв'язання конкретного завдання маркування та налаштувати його параметри. Використання цього застосунку буде корисним для розробників систем комп'ютерного зору у завданнях, основаних на проблемах маркування, а також для студентів, аспірантів і дослідників, які вивчають методи структурного розпізнавання образів.*

Ключові слова: *задачі розмітки, розпізнавання образів, комп'ютерний зір, програмне забезпечення.*