UDC 004.93'1

Iryna USTENKO
ustenko.irina@gmail.com
ORCID: 0000-0003-1541-2414
Serhii USTENKO
ustenko.s.a@gmail.com
ORCID: 0000-0003-4968-1233
Mykolayiv

GEOMETRIC RECOGNITION METHOD OF RECTANGULAR AREAS ON IMAGES

In this paper, a method of geometric recognition of rectangular areas in images obtained by statically standing cameras is proposed. This method refers to the method when a deep analysis of the characteristics of the image is made, in particular, various geometric characteristics are determined. The method uses the functions of the OpenCV library. This method is implemented as a software product. As a result of his testing, acceptable results were obtained.

Keywords*: pattern recognition, geometric recognition, rectangular areas, cluster analysis, OpenCV.*

Formulation of the problem

The task of pattern recognition is that living systems, including humans, are constantly confronted from the moment they appear. In particular, information coming from the senses is processed by the brain, which in turn sorts the information, provides a decision, and then uses the electrochemical impulses to transmit the necessary signal further, for example, to the organs of movement that implement the necessary actions. Then there is a change in the environment, and the above phenomena occur again. And if you look, each stage is accompanied by recognition.

Pattern recognition (objects, signals, situations, phenomena or processes) is the task of identifying an object or determining some of its properties from its image (optical recognition) or audio recordings (acoustic recognition) and other characteristics [1].

Pattern – classification grouping in the classification system, combining (distinguishing) a certain group of objects by a certain attribute. Images have a characteristic property, manifested in the fact that acquaintance with a finite number of phenomena from the same set makes it possible to recognize an arbitrarily large number of its representatives.

Analysis of recent research and publications

Most machines can receive input only through simple input devices, such as keyboards and mouse, or through wired and wireless communication channels. However, in recent years, cameras and microphones have been added as standard parts of computers and mobile devices (such as phones and tablets). At the same time, the speed of these devices has increased significantly, which made it possible to begin processing this data in a meaningful way. There was a whole area of scientific activity – Computer Vision. It includes the methods used to automate tasks, such as recognizing letters, bar codes, car numbers, persons and other biometric data, speech, etc. Computer Vision systems are becoming ubiquitous [2].

There are a number of high-quality systems in which solutions of computer vision problems can be developed, of which two are distinguished: MATLAB and OpenCV. MATLAB provides an environment that allows you to relatively quickly create prototypes of vision solutions [3]. OpenCV is a high-quality library for C and C++, which provides the implementation of many modern methods of vision [4].

ISSN 2520-2820 (online)
ISSN 2524-0978 (print)

Research Objective

There are three methods of pattern recognition:

1. Brute force method – a comparison with a database is performed, where for every kind of objects various display modifications are presented. For example, for optical pattern recognition, you can apply the method of sorting an object's view from different angles, scales, displacements, deformations, etc.

2. Produced a deeper analysis of the characteristics of the image. In the case of optical recognition, this may be the determination of various geometric characteristics.

3. The use of artificial neural networks. This method requires either a large number of examples of the recognition problem in training, or a special structure of the neural network that takes into account the specifics of this task. However, it is distinguished by higher efficiency and productivity.

The purpose of this work is to develop a method of geometric recognition of rectangular areas in images obtained by statically standing cameras. This method will apply to the second way of pattern recognition.

Basic material

Pattern recognition tasks consist of two stages (fig. 1):

– conversion of the original data to a form that is convenient for recognition;

– proper recognition (indicating that the object belongs to a particular class).

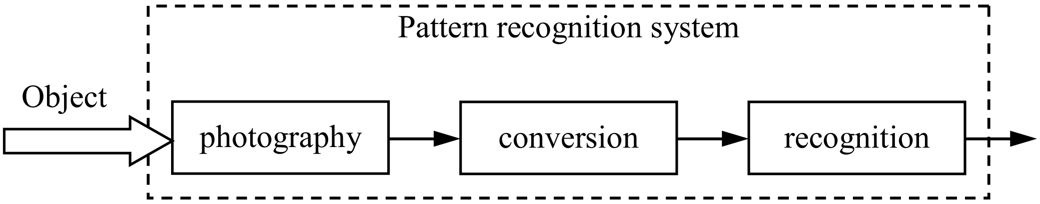

Fig. 1. The structure of the recognition system

For such problems, it is difficult to build formal theories and apply classical mathematical methods (information is often not available for an exact mathematical model or the gain from using a model and mathematical methods is incommensurable with costs).

We will develop a method of geometric recognition of rectangular areas on images obtained by statically standing cameras. An example of such images may be, for example, a photo of a table for weighing parcels (fig. 2). The method will be implemented in the C++ programming language using the OpenCV library.

The first step in solving the information recognition problem will be to pre-process the image, which consists of the following stages:

1. Image reduction in 2 times – to reduce noise in the image. To do this, we use the Image Pyramid, which is a collection of images, all arising from a single original image – that are successively downsampled until some desired stopping point is reached.

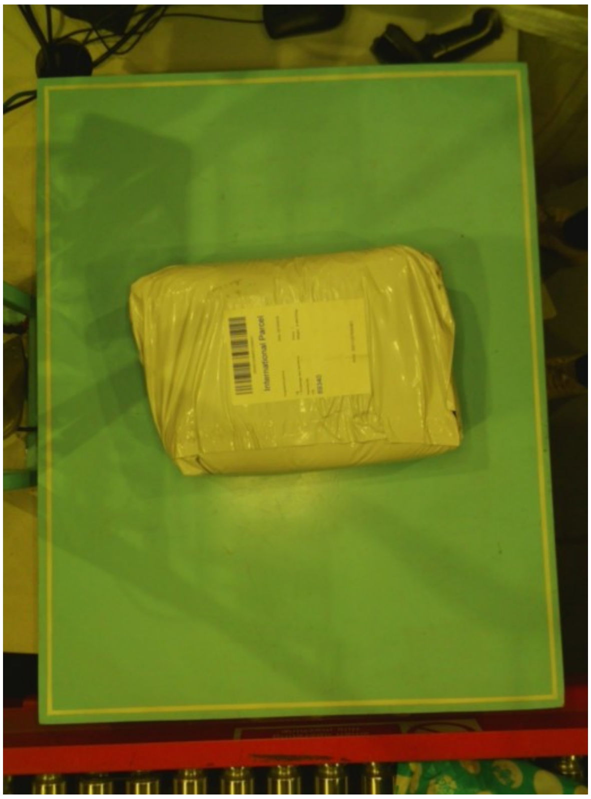

Fig. 2. Image for processing

There are two common types of image pyramids: Gaussian pyramid – used to reduce the image and Laplacian pyramid – used to restore the high-resolution image from the image below in the pyramid (with lower resolution).

Use for this function, which has the following signature:

```
void pyrUp(InputArray src,
           OutputArray dst,
           const Size& dstsize=Size(),
           int borderType=BORDER_DEFAULT)
```

2. Transform an image using an LUT transform to reduce the time required to apply an image transform (they usually have more pixels than the number of intensity values). The values of the LUT-table will be chosen experimentally – for the processed set of images of the tables was selected 0,72. The OpenCV function to perform the conversion is:

```
void LUT(InputArray src,
         InputArray lut,
         OutputArray dst)
```

3. For the final smoothing of noise in the image, we apply blurring – this is a simple and frequently used image processing operation, which consists in applying a filter to the image.

There are many types of filters, but the most useful (although not the fastest) is a Gaussian filter. Gaussian filtering is performed by folding each point of the input array with a Gaussian core and then summing them to create an output array. Call the filter by using the function of this type:

```
void GaussianBlur(InputArray src,
          OutputArray dst,
          Size ksize,
          double sigmaX,
          double sigmaY=0,
          int borderType=BORDER_DEFAULT)
```

4. The final stage of image preprocessing is the use of the Canny Edge Detector, which was developed by John F. Canny in 1986. Known as the optimal detector and aims to satisfy three main criteria:

– low error rate – meaning a good detection of only existent edges;

– good localization – the distance between edge pixels detected and real edge pixels have to be minimized;

– minimal response – only one detector response per edge.

OpenCV function:

```
void Canny(InputArray image,
           OutputArray edges,
           double threshold1,
           double threshold2,
           int apertureSize=3,
           bool L2gradient=false)
```

The second stage of the solution is the recognition of a rectangular area in the image. As a result of the solution, it is necessary to indicate the coordinates of the area inside the image of the rectangular area. We will assume that a region is rectangular if its sides deviate from the horizontal and vertical within the specified error Δ, expressed in degrees, for example, 5°. The stage includes the following stages:

1. Definition of straight lines on the processed image. We use the class BinaryDescriptor to detect lines and calculate their binary descriptor. The class is mainly based on classical detectors and extractors, such as the FeatureDetector and DescriptorExtractor. The obtained information about the lines is stored in KeyLine objects.

2. The choice of the obtained line segments, the angle of inclination of which corresponds to the specified, taking into account the specified error Δ, as well as perpendicular to them.

The selected lines are divided into four groups relative to the center of the image: horizontal, lying above (HA); horizontal, lying below (HB); vertical, lying to the left (VL); vertical, lying on the right (VR).

3. Removal of lines whose length is less than the specified one (percentage of the maximum length – separately for horizontal Δ_H and vertical Δ_L segments).

4. On the basis of cluster analysis, those lines are selected that constitute the required rectangular area. For the example in question, this is easier to do, because the frame on the table has a border bounding a rectangular area

(see fig. 2). Since cluster analysis is a multidimensional statistical procedure that collects data containing information about a sample of objects and then organizes objects into relatively homogeneous groups, it is necessary to determine the characteristics of the lines that allow them to be grouped. As such characteristics, we take the coefficients of the equation of a straight line given by an angular coefficient:

$y = k_1 x + b_1$ – for horizontal lines;

$x = k_2 y + b_2$ – for vertical lines.

After calculating the characteristics, they must be normalized and start the clustering procedure. Let's use the OpenCV method:

```
double kmeans(InputArray data,
              int K,
              InputOutputArray bestLabels,
              TermCriteria criteria,
              int attempts,
              int flags,
              OutputArray centers=noArray())
```

When calling the function, *K* it should be taken equal to 4 – the number of groups of lines corresponding to the number of sides of a rectangular area.

Next, the minimum distance $R_{\min}$ between the neighbors in the sample is found and lines are ejected, which are located at a distance greater than $k_{\max}R_{\min}$, where $k_{\max}$ is selected experimentally (10 for the problem being solved). In this case, the distance is determined for each group separately.

5. Based on the obtained set of lines of four types, we define a rectangular area of the table. To do this, we intersect all the lines of the group with the lines of the neighboring group (HA×VL, HA×VR, HB×VL, HB×VR) and select points at the minimum distance from the center of the image. Thus, we obtain 4 points that form a rectangular area.

6. The last step is to reduce the rectangular area, relative to its center, by a given relative value Δ_S (for the considered task by 1,5%).

On the basis of the proposed method, a software product was developed that allows determining a rectangular area on the images. As a programming language was chosen in C++ using the OpenCV library. To test the software, about 5,000 photographs of a parcel weighing table (with and without parcels) were processed. In 95,4% of the rectangular (working) area of the table was recognized. An example of a recognized area is shown in fig. 3. In other cases, the rectangular area was not determined due to the fact that the parcel was not located in the working area and partially overlapped its bounding frame.

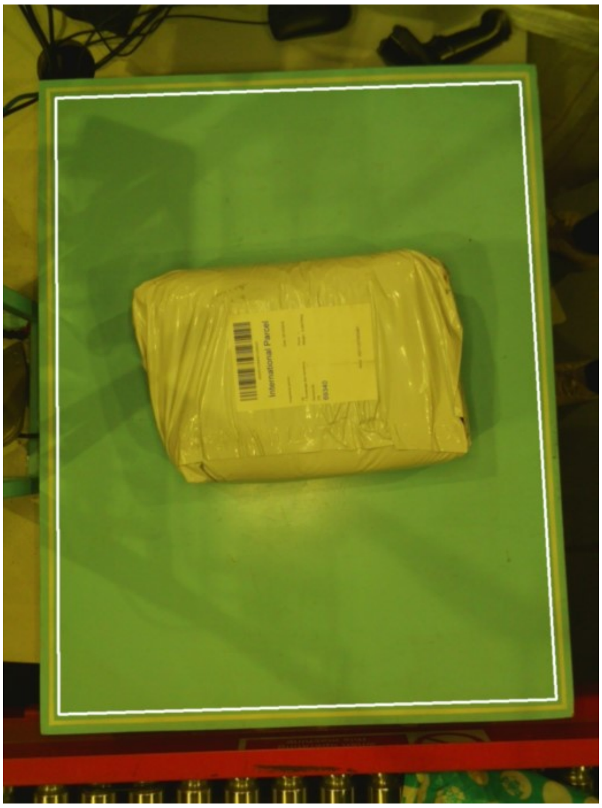

Fig. 3. Processed image

Conclusions

As a result of the work, a method of geometric recognition of rectangular areas in images obtained by statically standing cameras was developed. This method is implemented as a software product. As a result of his testing, acceptable results were obtained.

In the future we plan to develop a method of geometric recognition of parcels and their sizes, located in a rectangular area of the desktop. It is also planned to implement the method of geometric recognition of parcels and their sizes as an application.

REFERENCES

1. Колесников, С. Распознавание образов. Общие сведения [Електронний ресурс] / С. Колесников // Компьютер-Информ. – Режим доступу до ресурсу: http://old.ci.ru/inform03_06/p_24.htm
2. Dawson-Howe, K. A practical introduction to computer vision with OpenCV [Text] / K. Dawson-Howe. – Wiley, 2014. – 217 p.
3. Siogkas, G. Visual Media Processing Using MATLAB Beginner's Guide [Text] / G. Siogkas. – Packt Publishing, 2013. – 308 p.
4. Гарсия, Г.Б. Обработка изображений с помощью OpenCV [Текст] / Г.Б. Гарсия, О.Д. Суарес, Х.Л. Эспиноса Аранда, Х.С. Терсеро, И.С. Грасиа, Н.В.Энано. – М.: ДМК Пресс, 2016. – 210 с.

Ірина УСТЕНКО, Сергій УСТЕНКО
Миколаїв

МЕТОД ГЕОМЕТРИЧНОГО РОЗПІЗНАВАННЯ ПРЯМОКУТНИХ ОБЛАСТЕЙ НА ЗОБРАЖЕННЯХ

У даній роботі запропоновано метод геометричного розпізнавання прямокутних областей на зображеннях, отриманих камерами, що статично стоять. Даний метод відноситься до способу, коли проводиться глибокий аналіз характеристик образу, зокрема, визначаються різні геометричні характеристики. У методі використовуються функції бібліотеки OpenCV. Даний метод реалізований у вигляді програмного продукту. В результаті його тестування були отримані прийнятні результати роботи.

Ключові слова: *розпізнавання образів, геометричне розпізнавання, прямокутні області, кластерний аналіз, OpenCV.*

Ирина УСТЕНКО, Сергей УСТЕНКО
Николаев

МЕТОД ГЕОМЕТРИЧЕСКОГО РАСПОЗНАВАНИЯ ПРЯМОУГОЛЬНЫХ ОБЛАСТЕЙ НА ИЗОБРАЖЕНИЯХ

В данной работе предложен метод геометрического распознавания прямоугольных областей на изображениях, полученных статически стоящими камерами. Данный метод относится к способу, когда производится глубокий анализ характеристик образа, в частности, определяются различные геометрические характеристики. В методе используются функции библиотеки OpenCV. Данный метод реализован в виде программного продукта. В результате его тестирования были получены приемлемые результаты работы.

***Ключевые слова**: распознавание образов, геометрическое распознавание, прямоугольные области, кластерный анализ, OpenCV.*

Стаття надійшла до редколегії 29.10.2018