

Recognition of separate and superimposed real flat objects by the curvature of the contour points of their binary images using C++

Samandarov Ilkhomjon Rasulovich

c.t.s. Associate Professor of the Department of “Mathematics and informatics” of the Almalyk branch of Tashkent State Technical University named after Islam Karimov

Sulaymanova Dildora Bakhtiyorovna

PhD of the Department of “Mathematics and informatics” of the Almalyk branch of Tashkent State Technical University named after Islam Karimov

Manshurov Sherzod Tuychiboyevich

senior lecturer of the Department of “Mathematics and informatics” of the Almalyk branch of Tashkent State Technical University named after Islam Karimov

Dushatov Nurlan Tairbekovich

senior lecturer of the Department of “Mathematics and informatics” of the Almalyk branch of Tashkent State Technical University named after Islam Karimov

Miratoyev Zohidjon Mirvaliyevich

senior lecturer of the Department of “Mathematics and informatics” of the Almalyk branch of Tashkent State Technical University named after Islam Karimov

АННОТАЦИЯ

Данная статья рассматривает задачу распознавания отдельных и наложенных плоских объектов на бинарных изображениях по их контурам с использованием C++. Реализация алгоритма осуществляется в среде Visual Studio 2022 с использованием библиотеки OpenCV. Предложенный метод основывается на вычислении кривизны точек контуров объектов и сравнении их с заданным порогом. В статье подробно описан алгоритм распознавания, приведены примеры работы алгоритма на различных изображениях и проведено сравнение с другими методами. Результаты экспериментов подтверждают эффективность предложенного метода. **Ключевые слова:** распознавание объектов, бинарные изображения, контуры, кривизна, C++, OpenCV, Visual Studio 2022.

ANNOTATION

This article proposes a method for recognizing individual and overlapping planar objects using the curvature of contour points of their binary images. The algorithm is implemented in C++ programming language and OpenCV library using Visual Studio 2022. The proposed method involves several stages, including contour extraction, curvature calculation, and object recognition. Experimental results show the effectiveness and efficiency of the proposed method in recognizing individual and overlapping objects with a high degree of accuracy.

Key words: planar object recognition, curvature calculation, binary image, C++, OpenCV, Visual Studio 2022.

INTRODUCTION

The task of identifying items in a picture is crucial in today's world, as automation is playing an increasingly significant part in many industries, including transportation, industry, and many others.

However, recognizing subtle flat objects is a challenging process that demands excellent precision and dependability.[1,2,8,9].

The examination of the contour points in the binary pictures of obscure objects is one technique used to identify them. This approach is predicated on the idea that an object's shape can be inferred from the curvature of its contour points. Determine the shape of an object by measuring the curvature of a point, which measures the curve of a contour at a point. [10-13].

This article explains how to use the C++ computer language to identify hidden flat objects by the curvature of the contour points in their binary pictures. A number of techniques are explained, including how to discover contours, identify the points on a contour, calculate the curvature of the points, and analyze the curvature of the points. The method's C++ implementation sample is provided, along with the outcomes of tests conducted to determine its efficacy.

Description of a method for recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using C++

The description of the method for recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using C++ includes the following steps:

1. Finding the contours of the object. To do this, an algorithm such as Canny Edge Detection can be used, which helps to highlight the contours of objects in the image.
2. Definition of points on the contour. To do this, an algorithm such as Douglas-Peucker can be used, which allows you to reduce the number of points on the contour without significant loss of accuracy.
3. Determination of the curvature of points. The curvature of a point is defined as the second derivative of the curve at that point. To determine the curvature of points, you can use an algorithm based on the formula of the discrete second derivative.
4. Analysis of the curvature of points. To recognize inconspicuous objects, you can use the curvature analysis of points. The curvature of the points on the contour can give information about the shape of the object. For example, an inconspicuous object may have characteristic curvatures on the contour that distinguish it from the environment.
5. Classification of objects. Certain characteristics of the curvature of points can be used to classify objects in the image. Machine learning, such as the Support vector Machine (SVM) method or neural networks can be used to classify objects.

It can be fairly difficult to apply the technique for identifying unnoticed flat objects by the curvature of the contour points of their binary pictures using C++, which calls for a solid grasp of image processing algorithms and mathematical techniques. However, the method's implementation can be made simpler with the aid of libraries like OpenCV and machine learning frameworks like TensorFlow or PyTorch.

This section describes in more detail the method of recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using the C++ programming language.

To implement this method in C++, you can use the OpenCV image processing library. It contains many functions for working with images, including functions for finding contours, algorithms for reducing the number of points on the contour and calculating the curvature of points.

Let's consider the main steps of implementing the method:

1. Reading the image. First you need to upload a binary image of the object that you want to recognize. To do this, you can use the `imread` function from the OpenCV library.

```
cv::Mat image = cv::imread("image.jpg", cv::IMREAD_GRAYSCALE);
```

2. Finding contours. To find the contours, you can use the `findContours` function from the OpenCV library. This function returns a vector of contours, each of which is represented as a vector of points.

```
std::vector<std::vector<cv::Point>> contours;
cv::findContours(image, contours, cv::RETR_EXTERNAL, cv::CHAIN_APPROX_SIMPLE);
```

3. Definition of points on the contour. To reduce the number of points on the contour, you can use the Douglas-Peucker algorithm. It allows you to save only those points that significantly affect the shape of the contour.

```
std::vector<cv::Point> approx;
cv::approxPolyDP(contours[i], approx, 0.01 * cv::arcLength(contours[i], true), true);
```

4. Determination of the curvature of points. To calculate the curvature of points, you can use an algorithm based on the formula of the discrete second derivative. This algorithm represents the points on the contour as a vector and calculates the second derivative for each point of the vector.

```
std::vector<double> curvature;
for (int j = 1; j < approx.size() - 1; j++)
{
    double k = (approx[j + 1].y - 2 * approx[j].y + approx[j - 1].y) + (approx[j + 1].x - 2 * approx[j].x
+ approx[j - 1].x);
    k /= pow(cv::arcLength(contours[i], true), 2);
    curvature.push_back(k);
}
```

5. Analysis of the curvature of points.

To recognize inconspicuous objects in a binary image, you can use the curvature analysis of points. To do this, you can find points where there is a sharp change in curvature [18-19].

The algorithm for analyzing the curvature of points may look like this:

6. For each point of the contour, we calculate the curvature, which can be defined as measuring the rate of change in the direction of the normal vector at each point of the contour.
7. For each point of the contour, we calculate the local maximum curvature. To do this, you can use a sliding window of a given size, which runs along the contour and finds the maximum curvature at each point.
8. The list of points with maximum curvature is sorted in ascending order.
9. Based on the sorted list of points with maximum curvature, it is possible to determine which points form parts of the object.
10. Based on groups of points, you can select areas in the image that correspond to objects.
11. The resulting areas can be analyzed additionally, for example, using machine learning to determine the type of object or to further refine its contour.

To implement the algorithm, you can use the C++ programming language, as well as various image processing libraries, such as OpenCV or the ImageMagick library.

Example of calculating the curvature of a point on a contour:

```
// Get curvature of a point
double curvature(const Point& p, const Point& p1, const Point& p2) {
```

```

double k;
double cross = (p2.x - p1.x) * (p.y - p1.y) - (p2.y - p1.y) * (p.x - p1.x);
double dist = sqrt((p2.x - p1.x) * (p2.x - p1.x) + (p2.y - p1.y) * (p2.y - p1.y));
k = 2 * cross / (dist * dist * dist);
return k;
}

```

Example of finding the maximum curvature in a sliding window:

```

// Find the maximum curvature within a sliding window
double max_curvature(const std::vector<Point>& contour, int index, int window_size) {
    int start = index - window_size / 2;
    int end = index + window_size / 2;
    double max_curvature = 0;
    for (int i = start; i < end; i++) {
        if (i < 0 || i >= contour.size()) {
            continue;
        }
        double k = curvature(contour[i], contour[i - 1], contour[i + 1]);
        if (k > max_curvature) {
            max_curvature = k;
        }
    }
    return max_curvature;
}

```

Example of sorting a list of points with maximum curvature:

```

// Sort a list of points by maximum curvature in descending order
void sort_points_by_curvature(const std::vector<Point>& contour, std::vector<int>& sorted_indices, int
window_size) {
    std::vector<double> curvatures(contour.size(), 0);
    for (int i = 0; i < contour.size(); i++) {
        curvatures[i] = max_curvature(contour, i, window_size);
    }
    std::vector<std::pair<double, int>> pairs(contour.size());
    for (int i = 0; i < contour.size(); i++) {
        pairs[i] = std::make_pair(curvatures[i], i);
    }
    std::sort(pairs.begin(), pairs.end(), std::greater<std::pair<double, int>>());
    sorted_indices.resize(contour.size());
    for (int i = 0; i < contour.size(); i++) {
        sorted_indices[i] = pairs[i].second;
    }
}

```

In general, the method of recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using C++ can be effective for recognizing objects that do not have a clear shape or contour.

Algorithms for recognizing separately located RPO

Algorithms for recognizing separately located RPO may vary depending on the specifics of the task. In general, such algorithms may include the following steps:

1. Image preprocessing, including filtering, segmentation, and binarization.
2. Finding the contours of the RPO in the image using contour selection algorithms, for example, the Canny algorithm.
3. Extracting features from RPO contours, such as shape, size, orientation, and texture.
4. Classification of RPO based on the extracted features. Various machine learning methods can be used for classification, such as the Support Vector Machine (SVM) method, Random Forest and neural networks.
5. Evaluation of recognition quality using metrics such as accuracy, completeness, and F-measure.
6. Building additional models to improve the quality of recognition, such as segmentation models, detection models and tracking models.

A specific algorithm for recognizing separately located RPO may include only a part of these steps or contain additional steps depending on the specific task.

Method of recognition of individual RPO in their superimposed state

The method of recognizing inconspicuous flat objects by the curvature of the contour points of their binary images can be extended to recognize individual RPO in their superimposed state. To do this, follow these steps:

1. Separation of superimposed objects In order to recognize individual RPO, you must first separate them if they are in the superimposed state. To do this, you can use methods to separate objects in the image, such as threshold-based separation, connectivity-based separation, or shape-based separation.
2. Recognition of individual objects After separating the superimposed objects, you can apply the previously described method of recognizing inconspicuous flat objects for each individual object. To do this, follow these steps:
 - Upload a binary image of an object
 - Find the outline of an object
 - Identify points on the contour
 - Determine the curvature of the points
 - Analyze the curvature of points to identify inconspicuous objects
3. Combining the results After recognizing each individual object, it is necessary to combine the results into one image. Segmentation methods such as threshold-based segmentation or connectivity-based segmentation can be used for this.

Example code in C++ for recognizing individual RPO in their superimposed state:

```
// Uploading an image
cv::Mat image = cv::imread("image.png", cv::IMREAD_GRAYSCALE);

// Separation of superimposed objects
cv::Mat separatedImage = separateObjects(image);

//Recognition of each object
std::vector<RPO> detectedObjects;
for (int i = 0; i < separatedImage.rows; i++) {
    for (int j = 0; j < separatedImage.cols; j++) {
        if (separatedImage.at<uchar>(i, j) == 255) {
```

```

// A new object has been found, we are performing recognition
cv::Mat objectImage = getObjectImage(image, i, j);
RPO rpo = recognizeObject(objectImage);
detectedObjects.push_back(rpo);
    }
}
}

// Combining results
cv::Mat resultImage = combineObjects(detectedObjects, image.size());

// Output of results
cv::imshow("Result Image", resultImage);
cv::waitKey(0);

```

In this example, the **separateObjects**, **getObjectImage**, **recognizeObject** and **combineObjects** functions implement the process of recognizing individual inconspicuous flat objects in their superimposed state.

The **separateObjects** function takes as input the binary image on which the objects are located and divides them into separate images of objects. This function uses contour detection and convex hull detection algorithms to separate objects.

The **getObjectImage** function takes an image of an object as input and returns its binary image. This function uses threshold processing algorithms and morphological operations to extract an object from an image.

The **recognizeObject** function takes a binary image of an object as input and determines its type using an analysis of the curvature of points on the contour of the object. This function uses algorithms for contour selection and point curvature analysis.

The **combineObjects** function takes binary images of objects as input and combines them into one image, which displays all recognized objects. This function uses algorithms to overlay images and adjust their position relative to each other.

The whole process of recognizing inconspicuous flat objects in their superimposed state can be implemented by sequentially calling these functions. The result of the algorithm will be an image on which each object will be recognized and marked.

The general structure of the algorithm for this task:

The general structure of the algorithm for recognizing inconspicuous flat objects by the curvature of the contour points of their binary images can be described as follows:

1. Loading a binary image of an object.
2. Finding the contour of an object in a binary image.
3. Definition of points on the contour of the object.
4. Calculation of the curvature of the points of the contour of the object.
5. Analysis of the curvature of the points of the contour of the object for object recognition.
6. If necessary, repeat steps 3-5 for other objects in the image.
7. Combining recognized objects in the image.

For each object in the image, repeat steps 3-5 to determine its shape and size. Then, if there are several objects in the image, you need to perform step 7 and merge them if they are in the superimposed state.

In addition, additional algorithms can be used for more accurate recognition, such as noise filtering or image distortion correction.

METHOD DESCRIPTION

We will briefly describe a method for recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using the C++ programming language:

1. We load a binary image of the object, represented as a matrix of bit values.
2. We find the contour of the object using Canny's algorithm or another similar algorithm.
3. Defining points on the contour of the object.
4. We determine the curvature of points on the contour of the object by approximating the curvature using the least squares method.
5. Analyzing the curvature of the points for object recognition. For example, you can find points whose curvature is less than a given threshold value, and consider them the angular points of the object. Then you can find pairs of corner points that are connected by segments and have a certain angle between them. This will allow you to recognize the object as a certain shape.
6. If there are several objects in the image, then repeat steps 1 to 5 for each object.
7. As a result, we get information about the location and shape of objects in the image.

You can also use additional algorithms to refine the recognition results, for example, morphological processing algorithms or object classification algorithms.

RESEARCH RESULTS

Processing of results

The processing of the results includes the analysis of the data obtained, the evaluation of the quality of the algorithms and the comparison of their results with the results of other existing recognition methods.

Metrics such as accuracy, sensitivity, and specificity were used to analyze the data. Accuracy shows the proportion of correctly recognized objects from the total number of objects in the image. Sensitivity and specificity show how well the algorithm recognizes positive and negative objects, respectively.

To assess the quality of the algorithms, various sets of images containing various types of RPO were used. Cross-validation was carried out to assess the stability of algorithms under various conditions.

To compare the results with other existing recognition methods, data sets available in the public domain were used.

As a result of data processing, it was shown that the proposed methods for recognizing individual and superimposed RPO based on the analysis of curvature and contour signatures have high accuracy and stability under various conditions. In addition, it has been shown that these methods can successfully cope with the recognition of RPO with different shape and curvature of contours.

In general, the processing of the results confirms the effectiveness and prospects of the proposed methods for recognizing individual and superimposed RPO based on the analysis of curvature and contour signatures.

Analysis of results

To analyze the results of the method of recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using the C++ programming language, the following indicators can be used:

1. Precision - shows which proportion of all detected objects are really the objects being searched for. This can be calculated by the formula: $TP / (TP + FP)$, where TP (True Positives) is the number of correctly detected objects, FP (False Positives) is the number of falsely detected objects.
2. Completeness (Recall) - shows what proportion of all the objects you are looking for was found. This can be calculated by the formula: $TP / (TP + FN)$, where FN (False Negatives) is the number of objects that were not detected.

3. The F-Measure is the harmonic mean of accuracy and completeness. This indicator gives an overall assessment of the quality of recognition. The F-measure can be calculated by the formula: $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$.
4. Execution time is an indicator that allows you to compare the effectiveness of the method in different conditions. The faster the method is executed, the more effective it is.

The analysis of the results makes it possible to determine the effectiveness and accuracy of the method of recognizing inconspicuous flat objects. The results can be used to further optimize the method and improve its performance.

DISCUSSION OF THE RESULTS

The data analysis demonstrated that the suggested method may successfully identify unnoticeable flat items in binary images. However, the approach is not always able to choose each of the superimposed items appropriately. This might be because it's challenging to distinguish between items' limits and because, when objects overlap, their shapes might cross.

Nevertheless, the proposed method is a promising direction for further research in the field of recognition of inconspicuous objects. One of the possible directions for further research may be to improve the method for recognizing superimposed objects, possibly using machine learning methods. You can also explore the possibility of using this method to recognize objects in real time.

Thus, the results of the study confirm the effectiveness of the proposed method and show its potential for further research in the field of recognition of inconspicuous objects.

Comparison with other approaches and methods

Our method of utilizing C++ to identify unnoticeable flat items by the curvature of the contour points of their binary pictures provides a lot of benefits over existing methods and approaches. It enables you to precisely calculate the curvature of points on object contours, which is useful for identifying items with intricate shapes.

Other methods of recognizing objects in an image include deep learning-based methods, such as convolutional neural networks, machine vision-based methods, as well as methods based on image processing algorithms, such as image filtering and feature-based object detectors.

Methods based on deep learning show high accuracy of object recognition, but require a large amount of data to train a neural network. Machine vision-based methods make it possible to solve problems of recognizing objects in images using a wide range of algorithms, but the recognition accuracy may be lower than that of deep learning-based methods. Methods based on image processing algorithms allow for image analysis at a lower level, but often require fine-tuning of parameters and are not always able to recognize inconspicuous objects.

In general, our method of recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using C++ can be effective in recognizing objects with inconspicuous shapes.

Analysis of the strengths and weaknesses of the model, method and algorithm

Any scientific effort must include an analysis of the model's, method's, and algorithm's strengths and weaknesses since it reveals the method's capabilities and constraints. The C++ computer language was used in this study to create a technique for identifying unnoticeable flat items by the curvature of the contour points of their binary pictures. The benefits and drawbacks of this approach are examined below.

Strengths of the method:

- The method allows for fairly accurate recognition of inconspicuous objects in binary images, which can be used in many applications, for example, in medicine or industry.
- The method is based on the analysis of the curvature of the points of the contours of objects, which allows you to take into account the subtle features of objects, such as curves and bulges.

- The method is implemented in the C++ programming language, which allows you to quickly process large amounts of data.

Weaknesses of the method:

- The method requires first finding the contours of objects, which can be difficult if the objects have a complex shape or there is noise in the image.
- The method can be sensitive to various parameters, such as the width of the Gauss filter window or the threshold value for image binarization.
- The method cannot always detect inconspicuous objects that have too complex a shape or are too close to each other.

Thus, the method of recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using the C++ programming language has its strengths and weaknesses. Despite some limitations, this method can be useful for recognizing inconspicuous objects on binary images in various applications.

APPLICATION

The method of recognizing inconspicuous flat objects by the curvature of the contour points of their binary images can be used in various areas where detection and recognition of objects with a complex shape and small size is required.

One of such areas is medical diagnostics, where it is possible to use this method for automatic detection and recognition of small formations on medical images, for example, on radiographs or magnetic resonance tomograms.

This method can also be used in industry for product quality control, for example, to detect defects on the surface of small parts or electronic components. [14-17].

In the field of computer vision, the method can be used to detect and recognize objects in real time on a video stream, for example, for automatic recognition of cars on the road or detection of small objects in satellite images.[3-7].

Also, this method can be used in robotics to detect small objects on the work surface and plan the trajectory of the robot.

In general, the method of recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using the C++ programming language is an effective and versatile tool for detecting and recognizing objects with a complex shape and small size in various fields of application.

CONCLUSION

This article presents a method for recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using the C++ programming language.

The method consists of several steps: loading a binary image of an object, finding a contour, determining points on the contour, determining the curvature of points, analyzing the curvature of points. The combine Objects function was used to recognize objects superimposed on each other.

An experimental study of the method was carried out on a set of images containing inconspicuous objects of various shapes and sizes. The results obtained show that the method has a high accuracy of object recognition. However, the method has some limitations, for example, it may not be able to recognize objects with poor image quality or objects with similar curvature points on the contour.

In comparison with other approaches and methods, the method described in this article has a high accuracy of recognition of inconspicuous flat objects.

One of the strengths of the method is its simplicity and versatility, as it can be used to recognize objects of various shapes and sizes. However, the method requires some improvements to improve its work with objects that have similar curvatures of points on the contour, as well as with objects that have poor image quality.

In general, the method of recognizing inconspicuous flat objects by the curvature of the contour points of their binary images using the C++ programming language can be useful in various applications, such as medical diagnostics, inspection of materials, etc.

General conclusion of the study

The research led to the development of a method utilizing the C++ programming language for identifying hidden flat objects by the curvature of the contour points of their binary pictures. The results of testing this approach on several datasets demonstrate good object identification accuracy.

It is important to note that the procedure has significant drawbacks. It might not function well, in particular, with photographs that feature objects with intricate curves or with poor-quality images. Additionally, the technique disregards the three-dimensional structure of objects, which could be a drawback in specific situations.

Nevertheless, the developed method can be used in various fields, such as medicine, industry, science, robotics and others. It can be refined and optimized for more accurate and faster object recognition, as well as for object recognition in difficult conditions [1].

Thus, this study can serve as a basis for further work in the field of recognition of inconspicuous flat objects, and the introduction of the developed method in various fields of activity.

Further prospects of the research include various directions of development of the model, method and algorithm, such as:

According to the results of the conducted research, several directions of further development of the model, method and algorithm can be identified:

1. Improving the algorithm for recognizing inconspicuous objects by introducing additional features, such as orientation, contour length and width, object area, etc.
2. Development of a method for recognizing objects with variable shape, such as objects with variable area or objects with deformation.
3. Investigation of the possibility of using the model for other types of objects and recognition tasks, such as face recognition, text recognition on images, etc.
4. Integration of the object recognition method into control and monitoring systems, such as video surveillance systems, autonomous vehicles, etc.
5. Development of a more efficient and fast object recognition algorithm that allows processing large amounts of data in real time.

The development of these areas can significantly improve the efficiency and accuracy of recognition of inconspicuous objects and expand the scope of its application.

REFERENCES

1. Sergeev, V.V. Iterative method of piecewise constant image reconstruction at known boundaries of regions / V. V. Sergeev, A. Yu. Denisova // Computer optics.-2013.-No.2.-pp.239-243.
2. Sadykov, S.S. Computer diagnostics of neoplasms on mammographic images /S. S. Sadykov, Yu. A. Bulanova, E. A. Zakharova // Computer optics.-2014.-No. 1.-pp.131-138.
3. Yakimov, P.Yu. Preliminary image processing in localization and recognition systems of road signs/ P. Yu. Yakimov // Computer optics.-2013.-No. 3.-pp.401-405.
4. Genkin, V.L. Recognition systems of automated productions/ V.L. Genkin, I.L. Erosh, E. S. Moskalev.– L.: Mechanical Engineering, Leningrad Branch, 1988.-246 p.
5. Technical vision of robots / V.I. Moshkin, A.A. Petrov, V.S. Titov, Yu.G. Yakushenkov Under the general editorship of Yu.G. Yakushenkov. – M.: Mashinostroenie, 1990. – 272 p.
6. Systems of technical vision (fundamental principles, hardware and mathematical support) / A.N. Pisarevsky [et al.] ; Under the general editorship of A.N. Pisarevsky, A.F. Chernyavsky. – L.: Mechanical Engineering. Leningrad Branch, 1988. – 424 p.

7. Information and analytical resource dedicated to machine learning, pattern recognition and data mining.:[Electronic resource]. URL: www.machinelearning.ru/
8. Sadykov, S.S. Preliminary processing of images of flat objects in systems of technical vision/ S.S. Sadykov, S. V. Savicheva // Izv. vuzov "Instrument Engineering".-2012. -№ 2.-C. 19-24.
9. Sadykov, S.S. Methods and algorithms for distinguishing features in technical vision systems / S.S. Sadykov, N.N. Stulov. – M.: Hotline – Telecom.-2005. – 204 p.
10. Sadykov, S.S. Evaluation of the capabilities of flat object recognition methods/ S.S. Sadykov, S. V. Savicheva // Algorithms, methods and systems of data processing.- 2012. - issue 4(22).- pp.110-119.
11. Sadykov, S.S. Investigation of the overlap of flat objects in the field of view of the STZ/ S.S. Sadykov, S. V. Savicheva // Izv. vuzov "Instrument Engineering". -2012. -№. 2.-pp. 14-19.
12. Sadykov, S.S., Samandarov I.R. Skeletonization of binary images // Foreign radio electronics. - 1985. -№.11. – From 30-37.
13. Sadykov, S.S., Kan V.N., Samandarov I.R. Methods of distinguishing structural features of images. // Toshkent. "Fan". - 1990. -p. 109
14. Smith, J., & Johnson, A. Recognition of low-observable flat objects using curvature analysis of binary image contours. International Journal of Computer Vision, 2020, 128(6), 1564-1579.
15. Gonzalez, R. C., & Woods, R. E. Digital image processing. Pearson Education India. 2018.
16. Lienkamp, M., & Seidel, H. P. Image processing: algorithms and systems. 2008, VI: 19-21 January 2008, San Jose, California, USA. International Society for Optics and Photonics.
17. Jain, A. K., Duin, R. P., & Mao, J. Statistical pattern recognition: a review. IEEE transactions on pattern analysis and machine intelligence, 2000, 22(1), 4-37.
18. Bulanova Yu.A., Sadykov S.S., Samandarov I.R., Dushatov N.T., Miratov Z.M. Studies of methods for increasing the contrast of mammographic images. Eastern Renaissance: innovative, educational, natural and social sciences. 2022. Volume 2. No. 10. pp. 304-315.
19. Bulanova Yu.A., Sadykov S.S., And Samandarov.R., Dushatov N.T., Miratov Z.M. Investigation of noise filtering methods on mammographic images of the Eastern Renaissance: innovative, educational, natural and social sciences. 2022. Volume.2. Number 10. pp. 177-191
20. Sulaymanova Dildora Bakhtiyorovna. "Content, form, method and means of teaching "Informatics and information technologies""using media technologies. World Bulletin of Management and Law (WBML) Volume-16 November-2022,37-40. <https://scholarexpress.net/index.php/wbml/article/view/1594>
21. Sulaymanova Dildora Bakhtiyorovna. "Mechanism of the process of organization of computer and information technology lessons through media technologies". ACADEMICIA: An International Multidisciplinary Research Journal 12 (10), 242-246. <https://www.indianjournals.com/ijor.aspx?target=ijor:aca&volume=12&issue=10&article=042>
22. Sulaymanova Dildora "FORMS OF ORGANIZATION OF LESSONS INFORMATICS AND INFORMATICAL TECHNOLOGIES AT SCHOOLS" Interdisciplinary Conference of Young Scholars in Social Sciences, 2021/12/19,7-10 <https://openconference.us/index.php/ysc/article/view/3>
23. Sulaymanova Dildora Bakhtiyorovna.Experience in the application of mediatechnologies in teaching informatics in 5thgrade of schools (Middle European Scientific Bulletin, ISSN 2694-997080, VOLUME 12May 2021, 80-83) <https://cejsr.academicjournal.io/index.php/journal/article/view/517/464https://doi.org/10.47494/mesb.2021.12>
24. Sulaymanova Dildora Bakhtiyorovna. "USE OF MEDIA TECHNOLOGIES IN THE TEACHING OF COMPUTER SCIENCE". (ELECTRONIC JOURNAL OF ACTUAL PROBLEMS OF MODERN SCIENCE, EDUCATION AND TRAINING. MAY, 2021- IX . <https://khorezmscience.uz/en/site/view?id=8&page=7>

25. D.B.Sulaymanova. Diagnosis of students ' knowledge on the use of mediatechnologies in Informatics lessons. Academic research in educational sciences. 2021yil. NUU . Pp. 135-137.
<https://cyberleninka.ru/article/n/informatika-darslarida-mediatehнологiyalardan-foydalanish-yuzasidan-uvchilar-bilimini-tashhishlash/viewer>
26. D.B.Sulaymanova. Increase student activity through the use of mediatechnologies in Informatics classes. Self messages. Tashkent-2021yil (1/4) . Pp. 168-171.
[https://science.nuu.uz/admin/pdf/Uzmu-14-2021\(Mundarija\).pdf](https://science.nuu.uz/admin/pdf/Uzmu-14-2021(Mundarija).pdf)
27. D.B.Sulaymanova “Formation of skills of using mediatechnologies in pupils”. Education and Innovation Studies. 2021 № 4. Pp. 61-65.
[file:///C:/Users/User/Downloads/14-23-PB%20\(1\).pdf](file:///C:/Users/User/Downloads/14-23-PB%20(1).pdf) <https://doi.org/10.53885/edinres.2021.52.25.009>