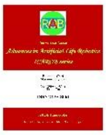
SUGISAKA
MASANORI

Journal of Advances in Artificial Life Robotics
Vol. 3(3); December (2022), pp. 179–184
ON LINE ISSN 2435-8061; ISSN-L 2435-8061
https://alife-robotics.org/jallr.html



Research Article

Using OpenCV for Augmented Reality Applications

Gabdullina Dinara, Zykov Evgeniy, Kugurakova Vlada

Kazan Federal University, Kazan, Russian Federation, 420008, Rep. Tatarstan, Kazan, St. Kremlin, 35, Russia

ARTICLE INFO

Article History

Received 24 November 2022
Accepted 18 April 2023

Keywords

Augmented reality
Object recognition
Computer vision
Optical control
Production automation

ABSTRACT

This article describes the use of OpenCV for use in augmented reality applications. The implementation of the object recognition functionality is described using the basic algorithms: FAST, BRIEF, as well as by applying a mask and defining contours. Attention is also paid to the practical application of this technology on the example of an augmented reality application to assist a specialist in mounting radio-electronic components on a printed circuit board. Experiments were carried out to measure the recognition time depending on the degree of illumination. An analysis of the filters used for image preprocessing has been carried out..

1. Introduction

Augmented reality is a tool for introducing automation into various industries. At the moment, the use of systems with augmented reality in real production is a rare occurrence [1]. The following risk factors are distinguished: temporary risks, financial risks and security risks.

At the moment, each application of augmented reality is a “piece of goods”, sharpened for specific tasks. There is a need to bring augmented reality application development into the mainstream. The use of different models of headsets for the same purposes, as well as the use of classical algorithms that give the correct result when used in changing work conditions. It is necessary to implement an augmented reality library, so for a start it was decided to narrow the range of tasks to be solved, with subsequent possible scaling.

This work is devoted to the development of algorithms that allow the use of object recognition using OpenCV algorithms.

2. Analysis tools for AR

ARKit [2] is a software development kit designed by Apple for building augmented reality apps on IOS devices. It offers a range of features, including the ability to determine the depth of the environment using the Depth API and LiDAR camera, track a person's movements and identify their skeletal structure, and generate topological maps of physical spaces.

ARCore [2] also allows you to develop applications for Android and IOS. It is possible to measure the depth of the surrounding world, evaluate lighting.

The Vuforia augmented reality SDK [2] provides the ability to implement applications for different operating systems: Android, IOS. Vuforia enables the recognition of 3D model object, scanning of physical objects, and identification of groups. Meanwhile, Wikitude [2] offers features such as object recognition from 3D models, marker tracking, location-based anchoring on maps, video overlay, integration with smart glasses, and external plugin integration.

3. Device overview

Before the start of development, an analysis of augmented reality glasses was made. Four AR headsets were selected for technical comparison: Vuzix M400 Starter Kit [3], RealWear HMT-1 & Wall Charger [4], Google Glass Enterprise Edition 2[5] and HoloLens [6].

Corresponding author's E-mail: dinaragabdullina086@gmail.com, Evgeniy.Zykov@kpfu.ru, vlada.kugurakova@gmail.com

The main criteria for choosing a target headset was the performance of the device for computing recognition algorithms. Vuzix M400 Starter Kit: 6 GB RAM, 12 hours battery life, Android 9.0 operating system. The RealWear HMT-1 Wall Charger has 3 GB of RAM, 10 hours of battery life and the Android 10.0 operating system. Google Glass Enterprise Edition - 3 GB of RAM, 8 hours of battery life, Android 8.1 operating system. HoloLens 2 - 4 GB RAM, 3 hours battery life, Windows Holographic operating system. Based on all the characteristics of these headsets, the Vuzix M400 Starter Kit was chosen.

4. Problem statement

As mentioned earlier, at the moment there are augmented reality libraries that provide all the necessary functionality. The problem with their use is the lack of universality that would allow augmented reality applications to be implemented in flow mode.

The purpose of this work is to implement a library containing proven algorithms for recognizing objects located in various environmental conditions. As part of this work, the following parts were identified that need to be implemented: object recognition using different algorithms, testing different algorithm values, identifying the best algorithm for devices with low performance.

Application functionality:

- Application opening.
- Pointing the camera at an object for recognition.
- Drawing a frame around an object, highlighting additional technical information.
- Closing the application.

4.1 Recognizing an object from an image using OpenCV

OpenCV – Open Source Computer Vision Library. A popular library that is the base for the Khronos computer vision standard. The library is adapted for different platforms. Open, freely distributed code, the ability to modify algorithms is the main advantage of using OpenCV.

To implement all the necessary functions, the OpenCVSharp plus Unity asset [7] and the Unity 2021.2.10f1 environment [8] were used. All development was carried out in the C# programming language.

ORB – oriented FAST and rotated BRIEF, a classic algorithm for determining key points in an image, the algorithm is not patented, so it can be used without restrictions in any projects. This algorithm is the best for devices with low performance according to [9]. The ORB algorithm uses two algorithms internally that are responsible for selecting key points and generating their descriptions. FAST is an algorithm for selecting key points [10]. The pixel is compared with 16 pixels located on a circle around the pixel of interest. Next, the condition is checked for the number of pixels darker or lighter than the one of interest. BRIEF [11] is an algorithm for describing key points. In this case, a binary feature vector is used.

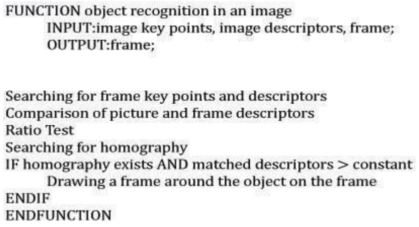
The Flann Matcher [12] algorithm operates more efficiently on large data arrays compared to the Brute-Force Matcher algorithm [13], as it utilizes the k-d-tree algorithm [14]. The algorithm's complexity is O(h) at best, where h represents the tree's height.

In k-d-tree, the search starts from the root of the tree, at each stage the distance from the point of interest to the points on the leaves of the tree is calculated. If the minimum distance is found, a check is made to see if there is a point to which the distance is less. A perpendicular line is constructed from the point of interest to the line passing through the parent node of the found point. If this distance is less than previously found for the point of interest, then the adjacent sheet is checked for the found point.

To leave only the necessary points and filter out all the noise the Lowe's Ratio Test algorithm [15] was applied.

Homography is a 3×3 dimensional transformation matrix for matching points from one image with corresponding points from another image. To display a frame around a recognized object, this method was used

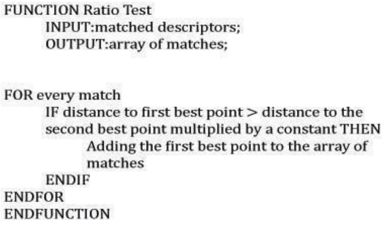
The figures Fig. 1 and Fig. 2 present algorithms for finding key points and getting rid of noise.



```
FUNCTION object recognition in an image
        INPUT:image key points, image descriptors, frame;
        OUTPUT:frame;


Searching for frame key points and descriptors
Comparison of picture and frame descriptors
Ratio Test
Searching for homography
IF homography exists AND matched descriptors > constant
        Drawing a frame around the object on the frame
ENDIF
ENDFUNCTION
```

Fig. 1. Pseudocode to recognize key points on a frame and display the frame



```
FUNCTION Ratio Test
        INPUT:matched descriptors;
        OUTPUT:array of matches;


FOR every match
        IF distance to first best point > distance to the
        second best point multiplied by a constant THEN
                Adding the first best point to the array of
                matches
        ENDIF
ENDFOR
ENDFUNCTION
```

Fig. 2. Pseudocode for deleting keypoints

To reduce the load on the computing power of the device, the calculation of key points on the reference image is performed at the very beginning of the program at startup. The calculation from the frames received from the device's camera is performed continuously. For a more accurate display of the frame around the object, a constant was created, which is a certain limit on the number of points for drawing the frame.

4.2Program optimization

After the implementation of the recognition algorithm by key points, the problem of the influence of the degree of illumination, which affects the correct operation of the entire algorithm, was identified. Camera shake blurs frames and makes it impossible to recognize key points.

To make the recognition process invisible to the user, the following list of application improvements was compiled: drawing a frame as a rectangle with one object Rectangle() [16].

Applying filters for image processing: converting the image to black and white, filters to get rid of noise and sharpen the image.

To select a specific filter stack it was decided to conduct experiments with varying degrees of illumination of the workplace and measuring time.

Previously, one constant was introduced into the algorithm, which determined the threshold value for displaying the frame, which led to optimization of the calculations. For greater stabilization, a segment was also chosen, when hitting which, the homography is not recalculated. For cases with camera shake and frame blur, the average value over the last 10 frames accumulateWeighted() [17] is taken.

4.3 Application development

After these studies, the functionality was introduced into the application to help a specialist in the output mounting of radio-electronic components on a PCB (Fig. 3) [18].

The use of manual assembly is economically beneficial in the manufacture of products with an output of no more than 15-20 thousand pieces. per year in batches of 100 pcs.

The manual assembly of the boards allows the production to be flexible to accommodate the production of different boards. Also constant visual control by a specialist.

The main limitations in the production of small output of goods is the involvement of specialists with a high level of knowledge to perform technically complex tasks [19].

There are not so many companies producing electronics in large quantities on the Russian market. There is a need to implement a product for low-budget industries with the possibility of expanding the functionality of optical control and reuse.

The main requirements for low cost optical control systems are [20]: identification of the main types of defects – the absence, incorrect orientation, displacement of components, the presence of jumpers and the incorrect soldering, technical information display.

Based on these requirements, a number of tasks can be formulated, the solution of which is necessary when creating low-cost optical inspection systems [20]: loading a reference image and receiving a video stream from the device's camera, selecting individual frames; processing the reference image of frames from the camera using OpenCV filters; object recognition and reference information output.

Regarding the application for optical control when performing output wiring, the following specific functionality can be distinguished: recognition of a printed circuit board and electronic parts, output of technical information. Also, if you look at the examples of instructions [21], then the physical characteristics of electronic parts are indicated in order to prevent incorrect placement of the part and possible damage.

The main points included in the assembly of any printed circuit board: checking the physical characteristics, setting to a specific position on the board, installing the part on the board, trimming the solder pins [22].

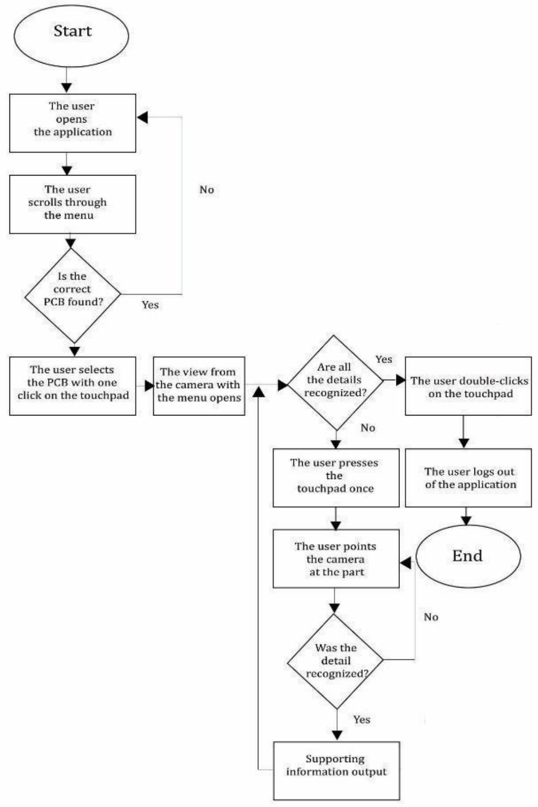

Start
The user opens the application
The user scrolls through the menu
No
Is the correct PCB found?
Yes
The user selects the PCB with one click on the touchpad
The view from the camera with the menu opens
Are all the details recognized?
Yes
The user double-clicks on the touchpad
No
The user presses the touchpad once
The user logs out of the application
The user points the camera at the part
End
No
Was the detail recognized?
Yes
Supporting information output

Fig. 3. Application flowchart

Since radio-electronic parts do not have bright distinguishing features, therefore, the use of recognition by key points is impractical. Most of the details are solid, so use the mask [23] to highlight pixels in a certain range. Algorithm for finding an object on the specialist's table: determination of the required range for the radio-electronic part in the HSV format; filtering pixels according to the selected range InRange() [24]; increasing the white area of the mask to eliminate noise – Dilate() [25]; applying a mask to a frame – BitwiseAnd() [26]; finding contours – FindContours() [27]; finding the corner points of the CHAIN_APPROX_SIMPLE contour.

As mentioned earlier, the application was implemented in Unity. The application interface has been implemented in relation to the display resolution - 640×360.

Following the best practices for user interface development [28], a simple interface with a minimum number of menus was implemented. At the same time, it was necessary to take into account the capabilities of the Vuzix M400 Starter Kit augmented reality headset: a touchpad and three control buttons, excluding the on/off button. To determine the user's actions, the Key Codes function [29] was used, with the help of which a certain function was bound to a click or a swipe.

5. Overview of image preprocessing methods

The image in OpenCV is converted to grayscale format [30] using the formula:

$$RGB\ to\ Gray: Y \leftarrow 0{,}299 * R + 0{,}587 * G + 0{,}114 * B, \tag{1}$$

where Y – is the result of the conversion, R *is* the red channel, G – green channel, B – blue channel.

Let's look at some classic ways to get rid of image noise: mean smoothing, Gaussian blur, median filter, and histogram equalization.

Mean smoothing is one of the simplest smoothing methods. This approach involves selecting a specific area of pixels with a predetermined size and calculating the average value to obtain the desired pixel value [25]. The kernel, which is a matrix with odd dimensions M × N, is used to determine the size of the averaging region [31]. Additionally, increasing the kernel's dimensions results in more pronounced smoothing. As an illustration, consider the following example kernel (2):

$$K = \frac{1}{K_{height} * K_{width}} \begin{bmatrix} 1\ 1\ 1 \\ 1\ 1\ 1 \\ 1\ 1\ 1 \end{bmatrix}, \tag{2}$$

where K – core 3×3.

When using a Gaussian anti-aliasing filter, each pixel is taken and the Gaussian function is applied to it [25]. This filter is designed to get rid of noise.

The median filter, in turn, is used when random black and white pixels occur. A certain area is taken and the median is considered [25].

If the image has strongly overexposed parts on which it is impossible to make out the image, then histogram equalization is applied to correct the display of the image. The technique of histogram equalization involves redefining the pixel values by using the cumulative distribution function [32]. However, this method may result in loss of information due to the wide range of pixel values. To overcome this issue, adaptive histogram equalization is employed where the operation is applied to different parts of the image [33].

6. Implementing image processing

To select the best filters, experiments were carried out with different degrees of illumination and measurement of the object recognition time. The experiments were carried out on a table, without objects that could interfere with recognition. The degree of illumination of the workplace of a specialist was measured using a lux meter – Testo 540. The measurement range of this device is from 0 to 99999 Lux with an error of 3 Lux.

Before starting the image processing, a grayscale conversion was performed to reduce the number of calculations by reducing the color channels to one. Next, the image smoothing and adaptive histogram equalization were performed.

To select the best smoothing filter, we measured the time required for object recognition. Moreover, for each of the filters, different parameters of the kernel dimension were used – 3x3 and 5x5. The time was measured inside the code using the StopWatch timer. The timer was stopped if the object was recognized. that is, a homography was found.

Since when conducting this type of experiment, outliers of values are possible, which can give an incorrect result when calculating the average value, the median method was chosen to determine the recognition time [34].

The graph (Fig. 4) shows the recognition results; when filters are applied, the process is faster.

After conducting a series of experiments with various filters and comparing the median results, it was concluded that Blur() with a 3×3 kernel and adaptive histogram equalization is the fastest way to recognize an object.

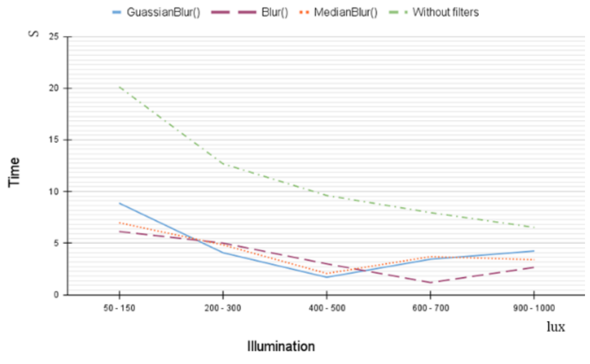


Fig. 4: Graphs without processing and processing by low-pass filters with a kernel

7. Conclusion

OpenCV is a powerful tool for implementing computer vision algorithms, providing the ability to use proven image processing and object recognition algorithms. Together with Unity and C#, it becomes possible to implement complex applications with different specifics.

It should be said that the introduction of image preprocessing and limiting constants optimized the process of object recognition, which made the developed program suitable for implementation in practical tasks.

References

1. K.E. Schein, P.A. Rauschnabel, “Augmented Reality in Manufacturing: Exploring Workers' Perceptions of Barriers”, IEEE Transactions on Engineering Management, pp. 1-14, 2021.
2. Y. Chen, Q. Wang, “An H.C. Overview of augmented reality technology”, Journal of Physics: Conference Series, 2019.
3. Vuzix M400 Starter Kit (2022). (Accessed: 2 February 2022).
4. RealWear HMT-1 & Wall Charger Augmented Reality Module (2022). (Accessed: 2 February 2022).
5. Glass Enterprise Edition 2 – Inputs and sensors (2022). (Accessed: 2 February 2022).
6. HoloLens 2 (2022). (Accessed: 2 February 2022).
7. OpenCV plus Unity (2019). (Accessed: 2 February 2022).
8. Unity 2021.2.10 (2021). (Accessed: 5 February 2022).
9. A. Muryan, Y.Verdi, “Application of oriented fast and rotated brief (orb) and bruteforce hamming in library opencv for classification of plants”, JISAMAR (Journal of Information System, Applied, Management, Accounting and Research), pp. 51-59, 2020.
10. P. Chhabra, N.K. Garg, M. Kumar, “Content-based image retrieval system using ORB and SIFT features”, Neural Computing and Applications, pp. 2725-2733, 2018.
11. F. Siddiqui, S. Zafar, S. Khan, N. Iftekhar, “Computer Vision Analysis of BRIEF and ORB Feature Detection Algorithms”, Applied Computational Technologies, pp. 425-433, 2022.
12. V. Vijayan, P. Kp, “FLANN Based Matching with SIFT Descriptors for Drowsy Features Extraction”, Fifth International Conference on Image Information Processing (ICIIP), pp. 600-605, 2019.
13. N. Antony, B.R. Devassy, “Implementation of Image/Video Copy- Move Forgery Detection Using Brute-Force Matching”, International Conference on Trends in Electronics and Informatics (ICOEI), pp. 1085-1090, 2018.
14. Y. Chen, L. Zhou, Y. Tang, “Fast neighbor search by using revised k-d tree”, Informatics and Computer Science Intelligent Systems Applications, pp. 142-162, 2019.
15. D.G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, International journal of computer vision, Vol. 60. pp. 91-110, 2004.
16. Drawing functions in OpenCV (2022). (Accessed: 20 March 2022).
17. E. Niloy, J. Meghna, M. Shahriar “Hand Gesture-Based Character Recognition Using OpenCV and Deep Learning”, International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), pp. 1-5, 2021.
18. PCB Mounting Output Process (2022). Accessed: 25 March 2022).
19. V.L. Lanin, “Assembly technology, installation and control in the production of electronic equipment”, Bel. State University of Informatics and Radioelectronics. – Minsk: Inpredo. p. 64, 1997. (In Russian).
20. A. Ovchinnikov, “The concept of building budget systems for optical inspection of the quality of printed circuit board assembly”, Technologies in the electronic industry, Vol. 8, No. 36, pp. 41-44, 2009. (In Russian).
21. Soldering kit (2022). (Accessed: 25 March 2022).

22. A.I. Tsukanov, O.V Kuchevasov, “Technologies for mounting and dismantling components and elements of radio-electronic and radio-television equipment: a teaching aid”, St. Petersburg State Educational Institution ", College of Electronics and Instrument Engineering" St. Petersburg, p. 105, 2017.
23. OpenCV – Invert mask (2021). (Accessed: 22nd April 2022).
24. A. Qashlim, B. Basri, H. Haeruddin, A. Ardan, “Smartphone Technology Applications for Milkfish Image Segmentation Using OpenCV Library”, International Journal of Interactive Mobile Technologies (iJIM), pp. 150-163, 2020.
25. H. Singh, “Advanced Image Processing Using OpenCV”, Practical Machine Learning and Image Processing, pp. 63-88, 2019.
26. M.A. Mir, F. Qazi, M. Naseem “Invisibility Cloak using Color Extraction and Image Segmentation with OpenCV”, Global Conference on Wireless and Optical Technologies (GCWOT), pp. 1-6, 2022.
27. V. Harini, V. Prahelika, I. Sneka, “Hand Gesture Recognition Using OpenCv and Python”, New Trends in Computational Vision and Bio-inspired Computing, pp. 1711-1719, 2018.
28. UI Best Design Practices (2022). (Accessed: 23 March 2022).
29. How to map Android keycode to Unity keycode (2021). (Accessed: 23 March 2022).
30. W. Hou, D. Xia, H. Jung “Video road vehicle detection and tracking based on OpenCV”, International Conference on Information Science and Education (ICISE-IE), pp. 315-318, 2020.
31. Convolutions with OpenCV and Python (2016). (Accessed: 24 March 2022).
32. Y. Wei, G. Xu, H. Liu “OpenCV-based 3D printing physical surface defect detection”, IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)”, pp. 1924-1927, 2020.
33. F. Alkhalid, A.M. Hasan, A. Alhamady, “Improving radiographic image contrast using multi layers of histogram equalization technique”, IAES International Journal of Artificial Intelligence (IJ-AI), pp. 151-156, 2021.
34. Median in statistics (2022). (Accessed: 1 May 2022).

Authors Introduction

Ms. Gabdullina Dinara

Laboratory assistant, Research Laboratory of Augmented Reality Technologies in Industrial Processes, Kazan Federal University. Research area: augmented reality, virtual reality, computer vision

Mr. Zykov Evgeniy

Candidate of Science in Physics and Mathematics, Associate Professor, Department of Radio Astronomy, Kazan Federal University. Research area: image processing, artificial intelligence, neural networks, robotics, augmented reality, bionics.

Ms. Kugurakova Vlada

Candidate of Technical Sciences, Associate Professor of the Department of Software Engineering, Kazan Federal University. Research area: virtual reality, augmented reality, 3D models, game engines, artificial intelligence.