

Image Acquisition and Processing on Raspberry Pi in Matlab for 3D-Scanning

Albina Kadyroldina

Department of Instrument Engineering
and Technology Process Automation
D Serikbayev East Kazakhstan State
Technical University
Ust-Kamenogorsk, Kazakhstan
akadyroldina@gmail.com

Assel Kussaiyn-Murat

Department of Instrument Engineering
and Technology Process Automation
D Serikbayev East Kazakhstan State
Technical University
Ust-Kamenogorsk, Kazakhstan
asselkussaiynmurat@gmail.com

Beszedes Bertalan

Alba Regia
Obuda University
Székesfehérvár, Hungary
beszedes.bertalan@amk.uni-obuda.hu

Alexander Krasavin

Department of Instrument Engineering
and Technology Process Automation
D Serikbayev East Kazakhstan State
Technical University
Ust-Kamenogorsk, Kazakhstan
alexanderkrasavin@mail.ru

Darya Alontseva

Department of Instrument Engineering
and Technology Process Automation
D Serikbayev East Kazakhstan State
Technical University
Ust-Kamenogorsk, Kazakhstan
dalontseva@mail.ru

Abstract—The paper presents the results of applying a special image processing method on the Raspberry Pi platform using Matlab. An image processing system has been developed using the necessary element base for quick image processing, including image transmission using a camera. A new algorithm for extracting lines on a recognizable image has been developed, which uses a gradient image processing technique. Images were obtained and this system was successfully tested on the Raspberry Pi 3. It is proposed to use this method for a robotic machine vision system, where the video camera is mounted on a robot manipulator that scans the surface.

Keywords—image processing, machine vision system, 3D-scanning

I. INTRODUCTION

One of the traditional applications of machine vision is robotics. Since robots have been mainly used in industry and their working conditions have been determined, highly specialized solutions using expensive equipment, including optical and computer systems, turned out to be applicable for machine vision problems.

Currently, the most common are machine vision systems that use standard cameras and computers. The term “computer vision” is more suitable for such systems, although there is no clear separation of the concepts of machine vision and computer vision. However, different machine vision systems are equally important. The use of “non-standard” imaging acquisition methods, including the use of spectral ranges other than visible, coherent radiation, structured illumination, hyperspectral devices, time-of-flight, omnidirectional and high-speed cameras, telescopes and microscopes, etc., significantly expands the capabilities of machine vision systems [1].

While the capabilities of the algorithmic support of machine vision systems are significantly inferior to human vision, the capabilities of obtaining information about the observed objects significantly exceed it. Image acquisition issues constitute an independent field of research, and the methods for processing images obtained using various sensors are so diverse that their review is beyond the scope of this paper.

Kawasaki RS10LA robot manipulator has been used for plasma surface treatment in our research, as described in our papers [2–5]. The robot manipulator program was generated according to a given geometric model of the plasma-treated object. However, deviations of the shape of a real object from this model often lead to violation of the processing technological parameters. To solve this problem, it is necessary to obtain a 3D model of a product of complex shape as a result of its scanning and generate a robot program in order to realize the movement of the robot arm along a given 3D path. Thus, at first the 3D model of an object to be processed by plasma should be obtained. Since machine vision devices are one of the most expensive parts of robotic systems, we propose to develop and test an inexpensive and universal machine vision system on the Raspberry Pi platform using Matlab.

The goal of this study is to apply 3 Raspberry Pi platform combined with Matlab for developing machine vision system, followed by its application for robotic scanning objects.

II. METHODS

Various methods of images acquisition and processing, as well as algorithms of objects transformations and recognition most often used in computer vision theory for robotics [6-9] have been considered. The selected methods are briefly described below for implementation on the RPi3 platform.

A. Edge Detection by Sobel operator

In simple terms, edge detection is the process of identifying points in a digital image where the brightness changes aggressively. Edges in images are areas with strong intensity contrasts a jump in intensity from one pixel to the next. Edge detection is mainly used for feature detection or feature extraction in various fields. In ideal circumstances, the outcome of applying an edge detector to an image may lead to a set of connected curves that indicate the borders of objects, the borders of surface markings as well as curves that correspond to discontinuities in surface orientation. Thus, applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may

be regarded as less relevant, while preserving the important structural properties of an image. Usually gradient operators, Laplacian operators and zero-crossing operators are used for edge detection [6]. The gradient operators are divided into two categories (first and second order) depending on whether they approximate the first derivative (Prewitt, Sobel, etc.) or the second derivative (intersection of the Laplacian zero) of the input image.

The Sobel operator performs a 2-D spatial gradient measurement on an image and emphasizes regions of high spatial gradient that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image. Compared to other edge operators, Sobel has two main advantages:

1. Since the introduction of the average factor, it has some smoothing effect to the random noise of the image.
2. Because it is the differential of two rows or two columns, so the elements of the edge on both sides has been enhanced, so that the edge seems thick and bright [7].

B. Corner Detection by Harris operator

A corner detection is the next qualitative level in an image understanding. All known corner detectors generally differ from edge detectors in stronger response in places where corner is present in comparison with places where only edge appears [8].

The simplest angle detector is the Moravian operator, summing up only eight differences between the center pixel and pixels in an 8-neighborhood. More sophisticated angle detectors are the Harris and Stevens algorithm, the Shea and Tomasi algorithm, and the Rosten and Drummond algorithm. We dismantled the Harris & Stephens corner detector. It is based on the autocorrelation function of a small fragment of the input image. The first step in the Harris algorithm is to compute the Harris matrix, a matrix containing all combinations of gradients in the horizontal and vertical directions. Angles are places in the image where gradients are significant in both horizontal and vertical directions at the same time [8].

C. Line Detection by Hough operator

Each line passing through the image area, Hough Transformation associates the sum of the values of the pixels closest to this line. The Hough transform is used to detect straight-line objects or their various configurations in the image, for example, to detect road markings, search for document boundaries, color segmentation, computational tomography and others. In other words, the Hough transform is a method of detecting straight and curved lines in grayscale or color images.

The method allows you to specify the parameters of a family of curves and provides a search on the image of the set of curves of a given family. The paper [9] is devoted to the study of algorithms for calculating the fast Hough transform for two-dimensional and three-dimensional images. A method for calculating the fast Hough transform (BPH) for straight lines in a three-dimensional image is proposed. The BPH algorithms for approximation in two-dimensional and three-dimensional spaces are considered, the accuracy and completeness properties of the corresponding sets of dyadic patterns are investigated.

III. EXPERIMENTAL RESULTS AND DISCUSSION

The experiment was carried out in the laboratory of the Obuda University, Szekesfehervar (Hungary). As a result of this experiment, the results were obtained, which are analyzed and summarized in this paper.

A. Equipment and Software

Raspberry Pi 3 (Fig. 1) is a single-board computer having a size as small as a credit card. Raspberry Pi uses powerful BCM2837 which offers 1.2 GHz quad-core ARM Cortex-A53 CPU, 1 GB RAM, built-in Wi-Fi 802.11n and Bluetooth 4.1. In addition, the processor has the ARMv53 architecture, which allows using the operating systems: Debian Wheezy, Ubuntu Mate, Fedora Remix, and even MS Windows 10 IoT.

The compact module with infrared camera for Raspberry Pi was used. The second version of the module is equipped with an eight-megapixel Sony IMX219 Exmor sensor. It allows you to capture, record and broadcast video in 1080p, 720p and VGA formats. The maximum resolution for photographs reaches 3280×2464 pixels.

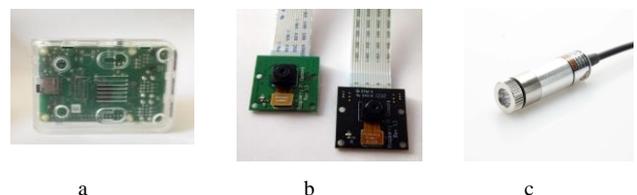


Fig.1. a - Raspberry Pi 3, b- integrated camera, c - Line LASER

A power connector and an Ethernet cable were used for communication between the RPi3 and the PC. All the algorithms developed in Matlab were deployed in the ARM processor on RPi3 via an Ethernet cable.

The first step was to combine Matlab with raspberry pi. To do this, Matlab support installation was selected for the Raspberry Pi Home-Add-Ons-Get hardware support packages in the Raspberry Pi list, and the Matlab component was installed. After loading, the process started and the name was set for raspberry pi:

```
Myipi = raspi // The raspberry pi characteristic were received
```

```
mycam = cameraboard(myipi, 'Resolution', '1280x720') // The command was called to create a camera connection board.
```

The shooting modes: brightness, illumination and size were chosen. Two shots were taken - one shot with the Line LASER diode, which is also connected and controlled by the Raspberry Pi and one shot without a laser (see Fig. 2), while the pictures were saved manually.

Since everything else in the camera's field of view remains unchanged, the difference between the two images should have given us all the points that lie on the Line LASER projected onto the object.

The received images (as shown in Fig. 2) have been saved for future use.

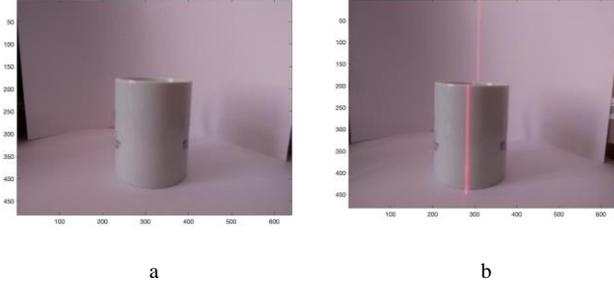


Fig. 2. Images captured using the Raspberry Pi 3
a - without Line LASER diode, b - with Line LASER diode

B. Equations

Using the images taken on the Raspberry Pi3 platform through Matlab, the authors developed a new method for selecting lines. The algorithm uses a gradient technique of image processing. Therefore, below we give the used designations and the necessary explanations. The concept of a continuous image intensity function allows you to introduce into consideration an image function gradient $\vec{g}(x, y)$ (1), often referred to simply as an image gradient:

$$\vec{g}(x, y) = (g_x, g_y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \quad (1)$$

In practice, the image is discrete and the image derivatives $\left(\frac{\partial I}{\partial x} \text{ and } \frac{\partial I}{\partial y} \right)$ functions) in the classical sense (as partial derivatives of the analytically given function) do not exist, therefore either their difference analogs are used or the image derivatives are calculated using some smooth approximation of discrete intensity function. There are many algorithms for calculating derived images and their software implementations. As well known, the gradient image processing technique is widely used for feature extraction [5-8]. Figuratively speaking, regions of an image containing features are characterized by high values of the gradient magnitude function $G(x, y)$ (1) (see Fig. 3):

$$G(x, y) = |\vec{g}| = \sqrt{g_x^2 + g_y^2} \quad (2)$$

Image gradient is a vector field. The two-dimensional vector field is defined by two vector fields (scalar functions of two variables) - $g_x(x, y)$ and $g_y(x, y)$. In software processing of the image, we calculated for each image pixel the e_x and e_y components of the gradient direction vector $\vec{e} = (e_x, e_y)$ (3) and calculated for each grid point (i, j) the magnitude of the image gradient $G_{i,j} = G(i, j)$.

$$\vec{e}(x, y) = \frac{1}{G(x, y)} \cdot g(x, y) \quad (3)$$

Thus, the gradient field was defined by three 2 Darrays. In the description below of the algorithm for converting an image into a graph, we will call \vec{k} a unit vector perpendicular to the gradient vector:

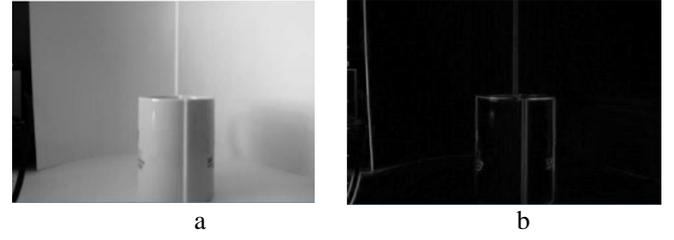


Fig. 3. Processed images
a - black and white digital image; b- magnitude of image gradient.
Maximum value of gradient magnitude corresponds to maximum value of gray scale intensity.

$$\vec{k} = (-e_x, e_y) \quad (4)$$

As is easy to see, vector \vec{k} is just a result of 90° degree counterclockwise rotation of corresponding direction vector \vec{e} . Note also that the vector $\vec{k}(x_0, y_0)$ is a tangent vector to the isoline of the intensity function $I(x, y) = I_0$, where $I_0 = I(x_0, y_0)$.

The next important step of the algorithm is the shift of the vertices (and with them the segments - edges of the graph) in the direction of the gradient vector. Figuratively speaking, the purpose of the procedure is to move the vertex into the "middle point" of the laser strip. It is clear that the choice of the magnitude of the vertex displacement significantly affects the quality of the transformation procedure, so we give a detailed description below. Let us first define the restriction of the intensity function to the line: For a given point $M(x_0, y_0)$ and a normalized vector $\vec{e} = (e_x, e_y)$ we define the function $\varphi(S)$ by equation (5):

$$\varphi(S) = I(x_0 + e_x \cdot S, y_0 + e_y \cdot S) \quad (5)$$

Geometrically, the parameter S can be interpreted as the length of the path traversed from the point M in a straight line with the direction vector. We will assume that the function $\varphi(S)$ is defined on the ray $[0, \infty)$ and will call it the restriction of the function I to the ray with the vertex at the point M and the direction vector \vec{e} . For a given S and the number N , we can build a sequence of values $\{S_1, S_2, \dots, S_N\}$ evenly spaced on an interval $[0, S]$. Using a smooth approximation of the intensity function I , we can calculate the sequence $\{\varphi_0, \varphi_1, \dots, \varphi_{N-1}\}$ of the values of the function φ $\{\varphi_k, \varphi_{k+1}\}$ and construct a piecewise linear approximation $\varphi^*(S)$ of the function φ , defined by(6).

$$\varphi^*(S) = \varphi_{\downarrow k} + ((\varphi_{\downarrow k+1} - \varphi_{\downarrow 0 k}) / \tau \cdot (S - S_{\downarrow k})) \text{ if } S \in [S_{\downarrow k}, S_{\downarrow (k+1)}) \quad (6),$$

where $\tau = S / (N - 1)$.

With a suitable choice of the value of S , the equation $\varphi^*(S) = \varphi_0$ has at least one solution (except trivial solution $S=0$). Let's denote it by $S=L$, so $L > 0$ and $\varphi^*(L) = \varphi_0$

If there are K solutions L_1, L_2, \dots, L_K , then $L = \min\{L_1, L_2, \dots, L_K\}$.

We will call the restriction of φ to the interval $[0, L]$ the profile of the laser strip (see Fig. 4).

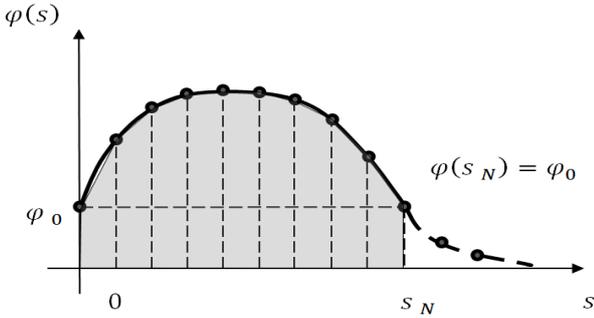


Fig.4. Schematic representation of the use of piecewise linear approximation to determine the displacement value

The main idea of calculating the amount of displacement in our proposed method is to calculate the center of gravity of the laser strip profile. Those, we calculate the value of displacement d according to formula (7).

$$d = \frac{1}{L} \int_0^L \varphi(S) dS \quad (7)$$

The integral $\int_0^L \varphi(S) dS$ is calculated numerically, to

find it we construct a new discretization of the function φ on the segment $[0, L]$. The procedure for building up the broken line representing the laser strip in the image is shown schematically in Fig. 5.

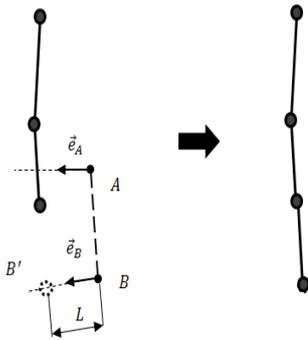


Fig. 5. Scheme of the procedure for constructing a broken line on the laser strip. \overline{AB} is the region of gradient homogeneity transformed into a line segment. \vec{e}_A and \vec{e}_B are the direction vectors of the image gradient at the points A and B , respectively

This procedure is performed for each line segment obtained by converting the corresponding region of the gradient uniformity.

IV. CONCLUSION

The snapshots of images obtained on the Raspberry Pi platform were processed in Matlab using a new algorithm. The main feature of the proposed image processing algorithm is the detection of fragments of the curve, which can be approximated with sufficient accuracy by a straight line segment.

The proposed algorithm is adaptive and its application does not require a priori knowledge of the profile properties of the detected curves.

It has been established that the removal of the contour of the laser strip is the most important part of the three-dimensional measurement based on linear laser scanning, and its accuracy and reliability directly affect the performance of the system.

The results of the research are of significance for a wide range of researchers developing the image acquisition and processing algorithms.

ACKNOWLEDGMENT

This research was supported by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan for the project AP05130525 "The intelligent robotic system for plasma processing and cutting of large-size products of complex shape" and by Erasmus+ mobility PROGRAMME 2018/2019 (OE-RH/79/3/2019)

REFERENCES

- [1] <https://controlengrussia.com/innovatsii/sistemy-komp-yuternogo-zreniya-sovremennye-e-zadachi-metody/>
- [2] D. Alontseva, A. Krasavin, A. Kadyroldina, A. Kussaiyn-Murat, D. Nurekenov, Ye. Zhanuzakov, N. Prokhorenkova, "Development of the Robotic Microplasma Spraying Technology for Applying Biocompatible Coatings in the Manufacture of Medical Products," AIS 2017 - 12 th International Symposium on Applied Informatics and Related Areas, Székesfehérvár, Hungary, pp. 45-48, 2017.
- [3] D. Alontseva, A. Krasavin, D. Nurekenov, O. Ospanov, A. Kussaiyn-Murat, Ye. Zhanuzakov, "Software Development for a New Robotic Technology of Microplasma Spraying of Powder Coatings", Przeglad Elektrotechniczny, vol. 94, No. 7, pp. 26-29, 2018.
- [4] D. Alontseva, A. Krasavin, A. Kadyroldina, A. Kussaiyn-Murat, "Segmentation Algorithm for Surface Reconstruction According to Data Provided by Laser-Based Scan", Springer, Computational and Information Technologies in Science, Engineering and Education, Vol. 998, pp. 1-10, 2019.
- [5] D. Alontseva, A. Krasavin, A. Russakova, A. Kadyroldina "Automation of industrial sites with mechatronic systems", International Journal of Electrical and Electronic Engineering and Telecommunications, vol. 7, No. 4, pp. 146-151, 2018.
- [6] Sh. Savant, "A Review on Edge Detection Techniques for Image Segmentation" in International Journal of Computer Science and Information Technologies, Vol. 5 (4) , pp. 5898-5900, 2014.
- [7] S. Gupta, S. Ghosh Mazumdar, "Sobel Edge Detection Algorithm" in International Journal of Computer Science and Management Research, Vol. 2, Issue 2, February 2013, ISSN 2278-733X.
- [8] M. Sonka, V. Hlavac, R. Boyle, "Image Processing", in Analysis and Machine Vision. Toronto: Thomson, 2008, p. 829, ISBN 978-0-495-08252-1.
- [9] K. Horak, L. Zalud, "Image Processing on Raspberry Pi for Mobile Robotics", 2015, 10.13140/RG.2.1.1440.7281