

**OFFLINE MACHINE VISION
IN THE PRODUCTION CELL CONTROL**

Martin JUHÁS¹, Bohuslava JUHÁSOVÁ¹, Pavol REMÉNYSÉG¹,
Roman DANEL²

¹SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA
INSTITUTE OF APPLIED INFORMATICS, AUTOMATION AND MECHATRONICS
ULICA JÁNA BOTTU Č. 2781/25, 917 24 TRNAVA, SLOVAK REPUBLIC
e-mail: martin_juhas@stuba.sk, bohuslava.juhasova@stuba.sk, xremenyseg@is.stuba.sk

²INSTITUTE OF TECHNOLOGY AND BUSINESS IN ČESKÉ BUDĚJOVICE,
FACULTY OF TECHNOLOGY, DEPARTMENT OF MECHANICAL ENGINEERING
OKRUŽNÍ 517/10, 370 01 ČESKÉ BUDĚJOVICE, CZECH REPUBLIC
e-mail: rdanel@mail.vstecb.cz

Received 12 August 2019, accepted 16 September 2019, published 29 November 2019

Abstract

The paper presents the possibility of using machine vision in the industrial area. The case study is oriented to indirect image processing in a robotic cell using a Matlab tool. Theoretical part of the contribution is devoted to the comparative analysis of various methods of the object detection and recognition. Analysis of the functionality, speed, performance and reliability of selected methods in the object detection and recognition area is processed. In the practical part, a method of implementing an indirect machine vision is designed to control the handling of objects detected and recognized on the basis of an operator requirement. Based on the analysis of the sample robotic workplace and the identified limitations, possibility of using the indirect computer vision is suggested. In such a case, the image of the workspace scene is saved on the storage and then processed by an external element. The processing result is further distributed in a defined form by a selected channel to the control component of the production cell.

Key words

Machine vision, Matlab, production cell, object detection, AppDesigner

INTRODUCTION

The article deals with a possibility of using offline machine vision in control process of a production cell. The article presents several aspects, to which increased attention is paid, since they are likely the elements of the industrial automation future. The current trend of

rationalizing of industrial production will require the deployment of intelligent production cells that can adapt to the process change without wasting time and resources. Machine vision can provide intelligence to these production cells, allowing the process to be changed simply, for example, by changing the control program parameters. In the scientific and partially also in the practical area, Matlab is a well-known efficient tool. Matlab is extremely powerful software which provides sophisticated tools for computer vision, machine learning, and artificial intelligence, with direct connectivity to industrial hardware [14]. These tools are also available in other forms, but, in the Matlab environment, they are integrated into one comprehensive system that is designed for their use in an unlimited range of interconnections and combinations. Despite the wide availability and implementation of Matlab in the academic world, it is not fully rooted in industrial practice yet, and therefore one of the goals of the paper is to bring its benefits closer.

OBJECTS DETECTION IN MATLAB

The Matlab software product is a platform optimized for solving scientific and technical problems as well as for system design and analysis. With strong support, it offers up-to-date methods and features to work in the areas of computer vision and machine learning. The level of sophistication is evidenced by the wide application of Matlab not only in research, but also in industrial practice.

Matlab provides a number of algorithms for object detection and identification. The current key detection methods are described in the paper. The paper covers a test of their functionality, speed, difficulty and reliability. Image processing is provided by the Image Processing Toolbox™ and the Computer Vision System Toolbox™ [14].

Points detection methods

The detection of characteristic points of the image is an essential building block of many computer vision tasks. The Matlab Computer Vision System Toolbox™ includes various functions for detecting image features. These functions return object points that store information specific to each type of features, such as x, y coordinates, in the object properties. In this way, the point object from the detection function can be transferred to various other functions [6], [7].

Point detection is a low level image processing operation. It is usually performed as the first operation in the image processing and examines each pixel, in order to see whether a characteristic feature is presented on a particular pixel.

Many computer vision algorithms use dot detection as the first step, so there is a high number of detectors. These differ greatly in the types of the features found, computational complexity and repeatability of detectability.

For the purposes of comparative analysis of available detection methods, the methods described in Table 1 were used [1] - [5], [8], [9].

Table 1 Analysed methods of detection		
Method	Advantages	Disadvantages
FAST	much faster than other corner detection methods high level of repeatability even with major changes	low noise resistance depending on the threshold setting
Harris	computationally undemanding repeatability even with changes	low noise resistance low speed

MinEigen	high effectiveness repeatability even with changes	low noise resistance relatively low speed higher computational demands
BRISK points detection	dimensional and angular invariance	inefficiencies in the prediction of lost data
SURF points detection	extremely fast high repeatability resistance to dimensional and rotational variation	high computational demands
KAZE points detection	high repeatability	high computational demands slow and complicated calculations
MSER regions detection	multi-scale detection stability	sensitivity to image blur




Detection functionality testing

Based on the analysis of available Matlab functions for object detection and identification, a list comprising the following steps was composed [10] - [13]:

- initialization,
- loading an image with a mixture of objects,
- selecting a method to detect features,
- selecting a search object and reading its characteristic features,
- extracting the features from an image with a mixture of objects,
- pairing the characteristic features of an object image with a mixture of objects,
- geometric transformation of the searched object contour in the image with a mixture of objects,
- calculating the coordinates of the searched object,
- listing the detection information.

Objects for detection

During the initial tests, various objects with lower texture segmentation were subjected to detection; however, the detectors detected them with almost 100% success. For this reason, computer components were used as test objects (Table 2, Table 3). The high segmentation of their texture is suitable in order to achieve a high number of detected points, and thus a high load on the detector and pairing. On the other hand, these textures are very similar, which makes it extremely difficult to identify pairs and can lead to false positives. Another reason for the use of printed circuit boards is their almost two-dimensional shape, and they are thus suitable for camera scanning directly from the top.

Table 2 List of searched objects		
O_GC_Asus_GS_BB_LR_X graphics card Asus 	O_GC_ATI_GS_BB_LR_X graphics card ATI 	O_GC_GB_GS_BB_LR_X graphics card Gigabyte 

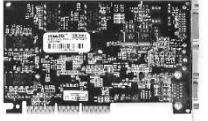



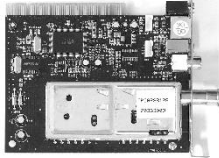
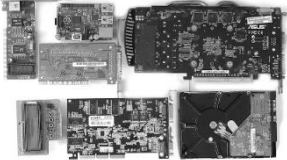
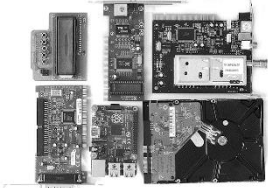
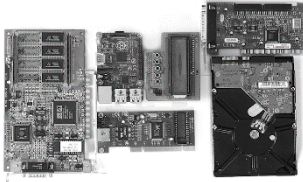
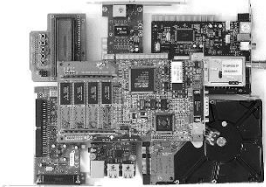
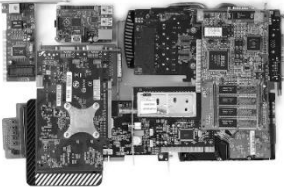
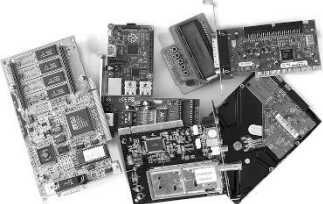
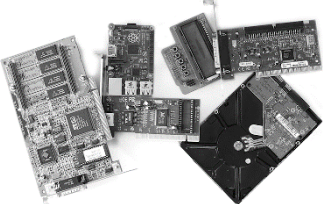
<p>O_GC_Inno_GS_BB_LR_X BB_LR_X graphics card Inno</p> 	<p>O_HDD_GS_BB_LR_X HDD Western Digital</p> 	<p>O_RBP2_GS_BB_LR_X MiniPC RaspBerry Pi 2</p> 
<p>O_SCSI_GS_BB_LR_X SCSI driver Adaptec</p> 	<p>O_TVT_GS_BB_LR_X TV Tuner Philips</p> 	

Table 3 List of objects mixtures

<p>Clutter_01_GS_BB_LR Without objects rotation and overlapping</p> 	<p>Clutter_02_GS_BB_LR Without objects rotation and overlapping</p> 	<p>Clutter_03_GS_BB_LR Without objects rotation and overlapping</p> 
<p>Clutter_OL_01_GS_BB_LR Without objects rotation, with overlapping</p> 	<p>Clutter_OL_02_GS_BB_LR Without objects rotation, with overlapping</p> 	
<p>Clutter_OL_DA_01_GS_BB_LR With objects rotation and overlapping</p> 	<p>Clutter_OL_DA_02_GS_BB_LR With objects rotation and overlapping</p> 	

Test of detection success and reliability

Detection in the test was set equally for all methods. Searched objects and objects in the mixture of objects, taken from the same distance, were equally rotated, focused, and none of them were overlapping. The method enables to repeatedly detect an object tested, and therefore the test was repeated 20 times.

Since not all images with a mixture of objects contain all the searched objects, the images combinations are listed in the first row in Table 4.

	O_GC_Asus_GS_BB_LR_X Clutter_01_GS_BB_LR	O_GC_ATI_GS_BB_LR_X Clutter_01_GS_BB_LR	O_GC_GB_GS_BB_LR_X Clutter_01_GS_BB_LR	O_GC_Inno_GS_BB_LR_X Clutter_01_GS_BB_LR	O_HDD_GS_BB_LR_X Clutter_01_GS_BB_LR	O_RBP2_GS_BB_LR_X Clutter_01_GS_BB_LR	O_SCSI_GS_BB_LR_X Clutter_01_GS_BB_LR	O_TVT_GS_BB_LR_X Clutter_02_GS_BB_LR
BRISK	✓	✓	✓	✓	✓	✓	✓	✓
	56	19	21	76	21	3	24	36
FAST	✓	✓	✓	✓	✓	✓	✓	✓
	101	45	33	177	23	18	55	53
KAZE	✓	✓	✓	✓	✓	✓	✓	✓
	909	625	332	835	270	208	289	490
MSER	✓	✓	✓	✓	✓	✓	✓	✓
	54	54	44	71	45	35	31	51
SURF	✓	✓	✓	✓	✓	✓	✓	✓
	148	108	96	223	54	22	59	126
Harris	✓	✓	✓	✓	✓	x	✓	✓
	56	23	24	82	21	0	25	26
MinEigen	✓	✓	✓	✓	✓	✓	✓	✓
	57	43	26	92	45	5	31	26

Test evaluation

The test results show that, except for Harris, all methods were able to detect all objects. The test showed the absolute reliability of the repetitions; every subsequent detection showed the same number of pairs as the first one.

Test of detection speed and computational complexity

Standard detection, as in the previous test, was used in this test, but without a graphical representation of the result. The test was repeated 100 times for each method with automatic duration measurement. Image "O_GC_Asus_GS_BB_LR_X" as a search object and image "Clutter_01_GS_BB_LR" as a mixture of objects were used.

System (processor) load was monitored (Figure 1) using application MSI © Afterburner™ version 4.4.2, while Difficulty coefficient represents the product of processor time and load.

Method	Time [s]	System load	Difficulty coefficient
BRISK	76.039	38%	29
FAST	19.994	50%	10
KAZE	91.752	60%	55
MSER	101.915	38%	39

SURF	14.144	85%	12
Harris	23.475	42%	10
MinEigen	26.618	45%	12

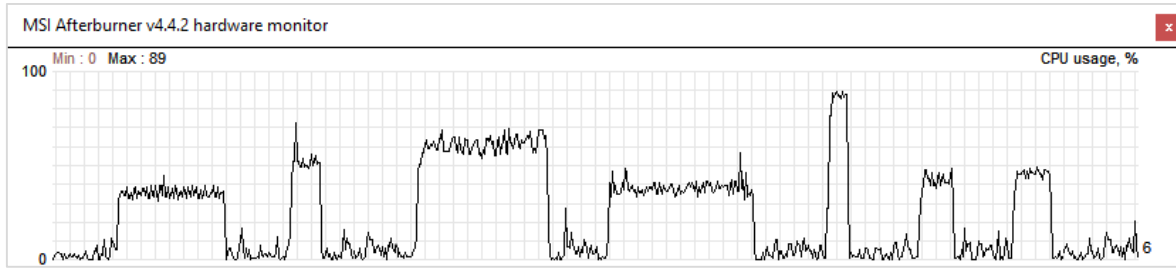


Figure 1 System load during detection process

Test evaluation

Test results (Table 5) show that SURF and FAST are the fastest methods, while FAST and Harris are the least demanding methods; on the contrary, MSER and KAZE are the slowest and also the most demanding as for hardware.

Test of dimensional change resistance

The test monitored the detectability of the method by gradually decreasing the image of the searched object. A standard algorithm was used for detection; the objects in the images were rotated and not overlapping. Owing to the low resolution of the Cognex camera (0.48 MP) and the complexity of object textures, the lower limit of the reduction was set to 50% of the original size, when details were already completely lost.

The "Clutter_01_GS_BB_LR" image was used as a mixture of objects, the "O_GC_Asus_GS_BB_LR_X", "O_SCSI_GS_BB_LR_X", and "O_RBP2_GS_BB_LR_X" images were used as the searched object, which was due to their different size and possibility of testing the effect of the available number of detected points while changing dimension.

	100%	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%
BRISK	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
FAST	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
KAZE	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
MSER	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
SURF	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
Harris	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
MinEigen	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗

	100%	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%
BRISK	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
FAST	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
KAZE	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
MSER	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
SURF	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
Harris	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
MinEigen	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗

	100%	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%
BRISK	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
FAST	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
KAZE	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
MSER	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
SURF	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Harris	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
MinEigen	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗

Test evaluation

The test results (Table 6, Table 7 and Table 8) show a strong dependence on resizing the searched image. The KAZE, MSER and SURF methods best handled the change. A low linear dependence on the original ratio of the searched object size to the mixture of objects image was also identified.

Test of rotation resistance

In the test, the searched object was rotated to different angles, while the success of detection at the angles was observed. The standard detection algorithm was used in the test; the images were tested without resizing or objects covering. The same images were used as in the dimensional change resistance test.

	0°	3°	6°	9°	12°	15°	30°	45°	60°	75°	90°	180°	270°
BRISK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FAST	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KAZE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSER	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SURF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Harris	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MinEigen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

	0°	3°	6°	9°	12°	15°	30°	45°	60°	75°	90°	180°	270°
BRISK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FAST	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KAZE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSER	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
SURF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Harris	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓
MinEigen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

	0°	3°	6°	9°	12°	15°	30°	45°	60°	75°	90°	180°	270°
BRISK	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓
FAST	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓
KAZE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSER	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓
SURF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Harris	x	x	x	x	x	x	x	x	x	x	x	✓	x
MinEigen	✓	x	x	x	x	x	x	x	x	x	✓	✓	✓

Test evaluation

Only KAZE and SURF methods successfully detected the searched object each time at any rotation. Larger objects, however, were detected by all methods (Table 9, Table 10 and Table 11).

Test of overlap resistance

The test determined the dependence of the detectability on the ratio of the object overlapping on the objects mixture image to the searched object. This was performed by gradually cropping an image with a mixture of objects in the area of the object being searched, up to 10% of the original width of the object.

A standard detection algorithm was used, the images used were the same as in the previous tests, i.e. the images of the searched object and the mixture of objects were used without changing the size or rotation.

Table 12 Detection success rate in overlapping for O_GC_Asus_GS_BB_LR_X

	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
BRISK	✓	✓	✓	✓	✓	✓	✓	✓	✓	x
FAST	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KAZE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSER	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SURF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Harris	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MinEigen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 13 Detection success rate in overlapping for O_SCSI_GS_BB_LR_X

	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
BRISK	✓	✓	✓	✓	✓	✓	✓	✓	x	x
FAST	✓	✓	✓	✓	✓	✓	✓	✓	x	x
KAZE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSER	✓	✓	✓	✓	✓	✓	✓	x	x	x
SURF	✓	✓	✓	✓	✓	✓	✓	✓	x	x
Harris	✓	✓	✓	✓	✓	✓	✓	x	x	x
MinEigen	✓	✓	✓	✓	✓	✓	✓	✓	x	x

Table 14 Detection success rate in overlapping for O_RBP2_GS_BB_LR_X

	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
BRISK	✓	✓	✓	✓	✓	x	x	x	x	x
FAST	✓	✓	✓	✓	✓	✓	✓	x	x	x
KAZE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSER	✓	✓	✓	✓	✓	✓	✓	✓	x	x
SURF	✓	✓	✓	✓	✓	✓	✓	✓	x	x
Harris	✓	✓	✓	✓	✓	x	x	x	x	x
MinEigen	✓	✓	✓	✓	✓	✓	x	x	x	x

Test evaluation

The test results (Table 12, Table 13 and Table 14) show the ability of the tested methods to detect objects despite a high overlap ratio. The KAZE method was the only one able to detect an object at 10% visibility.

Overall evaluation of the tests

With the current tested configuration, the sensor and object distance and focal length are not changed. Cognex camera has a quality lens with good aperture and geometry. This means that the resilience of the detection to resizing is not as important as the resistance to rotation and overlap.

Based on the set of tests, the KAZE detection method is the most robust in all qualitative parameters, especially in rotation and overlap, but it is relatively slow and poses high hardware requirements, although the SURF method ranked second in success: it was 7.2 times faster than the KAZE method.

Thus, the tests conclude that the KAZE or SURF methods are the most suitable detection method to control a production cell.

IMPLEMENTATION OF OFFLINE MACHINE VISION IN THE PRODUCTION CELL

The production cell of the STU MTF Laboratory of Control Systems at the Institute of Applied Informatics, Automation and Mechatronics (UIAM) served as a case study for this paper. The production cell hardware consists of the ABB IRB 140 robotic manipulator, the Cognex camera system, the Alutec K&K belt conveyor, and the hardware control unit. System intelligence is provided by the Cognex Industrial Vision System programmable with Cognex In-Sight Explorer. This system is technically highly sophisticated and is an industry standard. One of the main functions of the machine vision system is to determine the position and orientation of components on the conveyor belt, eliminating the need for special transport units. The position sensing and processing must be fast enough not to stop the conveyor belt, and thus not to slow down the process. The structure of a model production cell is shown in Figure 2.

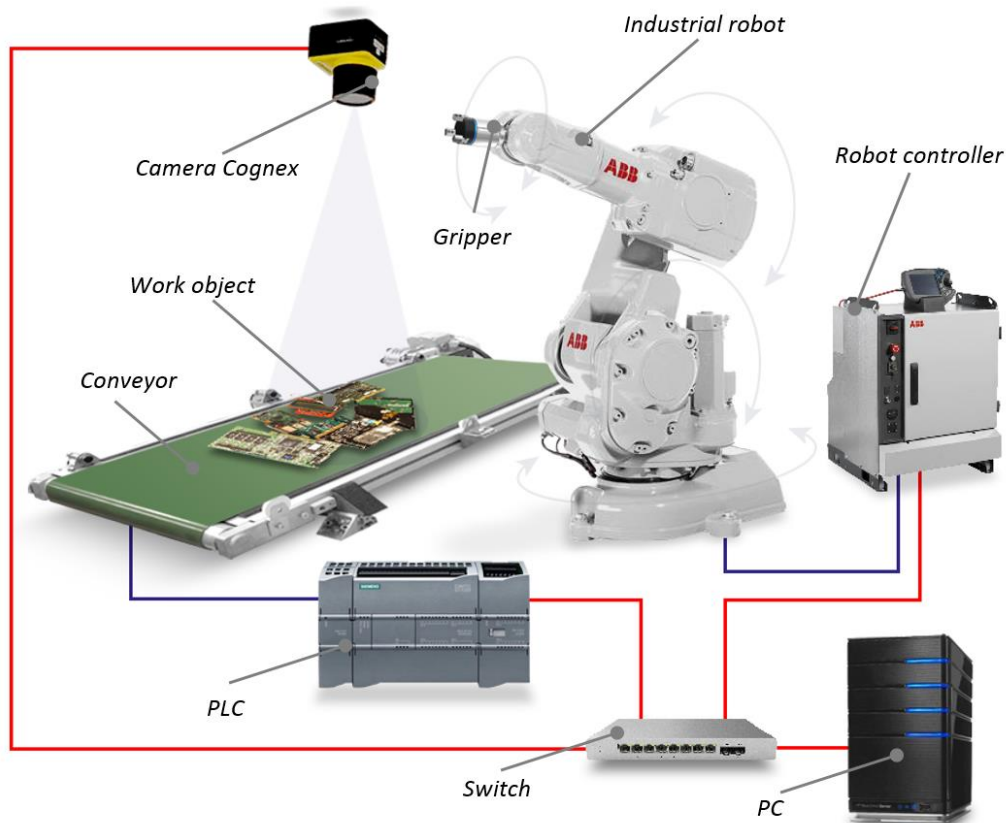


Figure 2 Connection scheme of production cell components

The deployment of machine vision to the production cell control in this case is in the form of detecting and identifying the searched object in the image with a mixture of other objects. The Cognex camera images stored on an FTP server serve as an input for the application. These images are then processed by any local computer with the MATLAB software installed. The output is numerical information about the location of the detected object in Cartesian coordinates, which is subsequently distributed to the master control system of the production cell.

The input-based application (image of object mixture, search object, and preferred detection method) performs the search object detection, which consists of:

- detecting points on the objects mixture image,
- detecting points on the searched object image,
- characteristic features extraction,
- features pairing,
- sufficient number of pairs verifying,
- geometric transformation of the searched object in the image with a mixture of objects,
- calculation of the object coordinates.

After detection, the application displays the object coordinates for the production cell, or displays an error message window if the object has not been detected.

The resulting application, implemented in the Matlab AppDesigner R2017b development environment (Figure 3), has a single 4:3 aspect ratio window of fixed XGA size (1024 x 768 pixels) with respect to the target implementation area - industrial production cell. This ratio and window size are sufficient to display images from the Cognex camera system (fixed resolution 800 x 600 pixels) and for a given number and dynamics of the control elements [14].

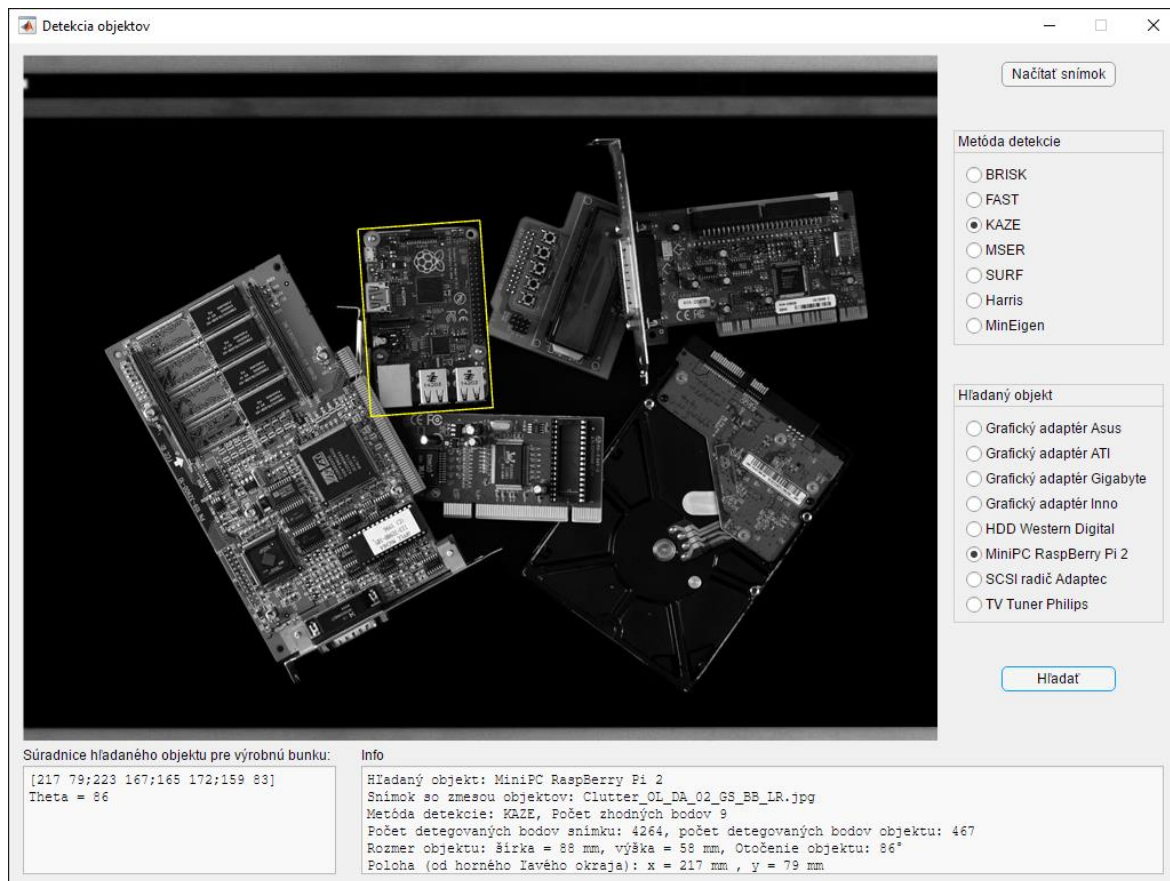


Figure 3 Application of machine vision in production cell

CONCLUSION

Since the functionality of the entire machine vision system depends on the quality of detection, the first step was to analyse the available methods of characteristic feature detection. The most commonly used detection methods, which are also included in software Matlab, were included in the analysis. After analysing the principle of their operation, identifying their advantages and disadvantages, each of the selected methods was applied in Matlab. Using the service script, a number of tests were performed using the methods of the characteristic features detection. The test was focused on detectability, speed and resistance to resize, rotation or overlap. The principles used in the analytical part were applied in the design of the final application, including the extraction of characteristic features and their pairing, transformation algorithm, calculation of coordinates or their conversion to real units. This information was used in the final to build the core of the Matlab application consisting of the initialization, image acquisition with mixture of objects and image of the searched object, method selection, feature detection, feature extraction, feature pairing, geometric transformation, calculation and the final output of obtained information about searched object. This information, distributed in the desired form by the executive element, serves as the data, for example, for the manipulator movement instructions in the controlling process of the production cell operation.

Acknowledgement

This publication was written with financial support of the KEGA agency in the frame of the Project 009STU-4/2018 The innovation of the subject Intelligent Control Methods at the Faculty of Materials Science and Technology of the Slovak University of Technology.

This publication has been written thanks to support of the Operational Program Research and Innovations for the project: Research, modelling and simulation of industrial production processes using progressive technologies; ITMS code: NFP313010T589 co-financed by the European Regional Development Fund.

References

- [1] MUTHUGNANAMBIKA, M., PADMAVATHI, S. 2017. Feature detection for color images using SURF. In: *4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, pp. 1-4. doi: 10.1109/ICACCS.2017.8014572
- [2] ZHIHENG ZHOU, XIAOWEN OU and JING XU. 2013. SURF feature detection method used in object tracking. In: *2013 International Conference on Machine Learning and Cybernetics*, Tianjin, pp. 1865-1868. doi: 10.1109/ICMLC.2013.6890899
- [3] HONG, C., CHENG-DONG, W., DONG-YUE, C. and ZI-WEI, L. 2015. A new interest point extraction method. In: *The 27th Chinese Control and Decision Conference (2015 CCDC)*, Qingdao, pp. 5414-5417. doi: 10.1109/CCDC.2015.7161761
- [4] LEUTENEGGER, S., CHLI, M., SIEGWART, R. Y. 2011. BRISK: Binary Robust Invariant Scalable Keypoints. In: *Computer Vision ICCV 2011*. The IEEE International Conference on Computer Vision: Spain, Barcelona, pp. 2548 – 2555. ISBN 978-1-4577-1101-5
- [5] ALCANTARILLA, P. F., BARTOLI, A., DAVISON, A. J. 2012. KAZE Features. In: *12th European Conference on Computer Vision: Italy, Florence, October 7-13, LNCS 7577*, pp. 214–227. ISBN 978-3-642-33718-5
- [6] RUSSAKOVSKY, O., DENG, J. at al. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211–252. ISSN 1573-1405
- [7] MAIR, E., HAGER, G., BURSCHKA, D., SUPPA, M., HIRZINGER, G. (2010). Adaptive and generic corner detection based on the accelerated segment test. *Lecture Notes in Computer Science*, 2 ed., vol. 6312 LNCS, 183-196. ISSN 1611-3349
- [8] ŠIKUDOVÁ, E., at al. 2011. Počítačové videnie (Computer vision). *Detekcia a rozpoznávanie objektov (Objects detection and discrimination)*. Prague: Wikina Publishing House, 2011. 400 p. ISBN: 978-80-87925-06-5.
- [9] YOUSEFI, S., NAJJAR GHABEL, S. 2016. A Survey on Object Detection Methods in Visual Sensor Networks. In: *International Journal of Advanced Smart Sensor Network Systems (IJASSN)*. 10.5121/ijassn.2016.6201.
- [10] <http://www.mathworks.com>, Abhishek, G. - Machine Learning with MATLAB [Online]. [Accessed: 08-2019] Available at <http://www.mathworks.com/videos/machine-learning-with-matlab-81984.html>.
- [11] www.mathworks.com, Avinash, N. - Computer Vision Made Easy [Online]. [Accessed: 08-2019] Available at <http://www.mathworks.com/videos/computer-vision-made-easy-81802.html>.
- [12] www.mathworks.com, Shashank, P., Avinash, N. - Deep Learning for Computer Vision with MATLAB [Online]. [Accessed: 08-2019] Available at <http://www.mathworks.com/videos/deep-learning-for-computer-vision-with-matlab-116080.html>.
- [13] www.mathworks.com, Tannenbaum, B. - Computer Vision with MATLAB for Object Detection and Tracking [Online]. [Accessed: 08-2019] Available at <http://www.mathworks.com/videos/computer-vision-with-matlab-for-object-detection-and-tracking-81866.html>.
- [14] www.mathworks.com, Matlab Documentation [Online]. [Accessed: 08-2019] Available at <https://www.mathworks.com/help/matlab/index.html>.

ORCID

Martin Juhás 0000-0002-8861-1855
Bohuslava Juhásová 0000-0002-5129-8375