Optical inspection of mounted printed circuit board using National Instruments software

Gyula Korsoveczki

Mechatronics Department

University of Debrecen

Engineering Faculty

Debrecen, Hungary

korsoveczki.gyula@gmail.com

Dr. habil. Géza Husi

Mechatronics Department

University of Debrecen

Engineering Faculty

Debrecen, Hungary

husigeza@eng.unideb.hu

Abstract — **The topic of the given task based on the optical opposition of a mounted printed circuit board using National Instruments software. During the inspection, 7 types of resistors, 3 types of integrated circuits, barcodes and text were detected. The results of the detection have been visualized and a Microsoft Excel scan report has been exported. The optical inspection was carried out using the National Instruments Vision Development Module and LabVIEW development environments.**

Keywords — optical; inspection; application; LabVIEW; image processing; report

I. INTRODUCTION

Mankind is preoccupied with the clock of inception to build machines with human-like skills in both work and mental activities. The most important aspect of this is that the machine needs to have a more direct and complete relationship with its environment. The theme of this article relates to the realization of vision in the case of machines, which has become one of the fundamental tools of the current fourth industrial revolution (Industry 4.0). The inspection of the mounted printed circuit boards is essential, because this is a very important task of the quality controlling. There are many opportunities to accomplish this procedure, but the optical one is the mainly used in industry, because it can be well agorithmised and customized, it is cost-effective and it produces great quality.

II. THE CONCEPTION OF THE MACHINE VISION

Machine vision means the collection of data from an image or from a video and the launch of a control or interpretation mechanism after evaluation. In case of machine vision, there are 2 main components [1]:

- image processing and
- image analysis [2].

The concept of machine image processing includes a toolkit for extracting data from an image or movie. Computer algorithms are mainly used to implement digital image processing. Its basis is digital signal processing (DPS). During the DPS some kind of analogue quantity is transformed into another form that can be processed by computer. While image processing means data extracting, image analysis refers to the evaluation of the data [3]. There are:

- Machine Vision, and
- Computer Vision [3].

Typically, we refer to control processes that start as machine vision processing. Its target area is to replace tasks that exceed human capabilities in hardware-oriented way, such as scaling, positioning, and evaluation tasks. However, computer vision is defined as problems evaluating the results of machine image processing. Operations related to human vision are automated and modelled in a software-oriented way. Computer Vision is responsible for visual-based spatial detection, counting, object detection and tracking, and analysis of various phenomena [3].

III. DIGITAL IMAGE PROCESSING

A. The model of the digital image

A digital image is a set of computer data that can be considered as a source for data collection purposes. To do this, it is necessary to define the digital image using the mathematics toolbar [4].

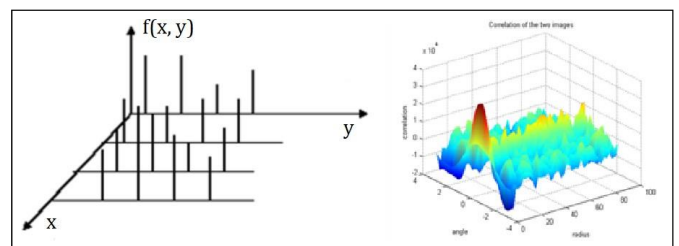The mathematic model of the digital image can be seen in the *Fig. 1.*:

Fig. 1. The model of the digital image [5]

Digital images are made up of pixels, which are the smallest basic units of digital images. A set of pixels can be interpreted as a two-dimensional $f(x, y)$ function with x and y coordinates, and f is the amplitude in the coordinates, which in this case is called intensity or greyness level. And since the

values *x, y* and f are discrete quantities, it means the digital image [5].

B. Sampling and quantization

It is also important to mention the process of creating a digital image, sampling and quantization. During sampling, the location of the continuous image marks is distributed into equal groups. These sites will be categorized as representative marks in the same groups.

Another important process is quantization. During this process, the amplitude values of an analogue signal have been transformed into binary numbers. The bit is the default value of this. As a result of the quantitative formation, the nominal marks categorized during sampling are assigned to a finite set of values, during which the digital image is created [6].

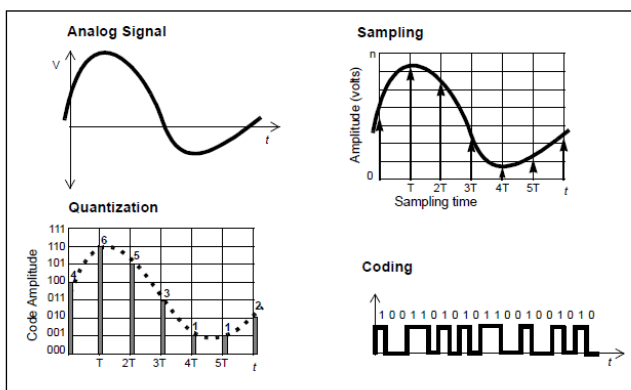The principle of the sampling and quantization can be seen in the *Fig. 2.*:

Fig. 2. The principle of the sampling and quantization [7]

C. The levels of digital image processing

Digital image processing involves the processing of signals using various computer algorithms, resulting in the extraction of targeted data from the digital image. During digital image processing, there are 3 process levels, which are [8]:

- low level processing,
- middle level processing,
- high level processing [8].

At the low level of digital image processing, both input and output are digital images. Targeted pre-processing is carried out at the level, which facilitates the extraction of data from the image at the next level. The output of the level is the modified image that helps the processes of the next level [9].

At the middle level, the input is the qualitatively enhanced image. Here, the image is segmented, which means highlighting certain components of the image. At this level, the computer algorithm used associates data that define segments based on a knowledge base. After the process of representation, segmentation properties are displayed at the middle level output [9].

Processes at the high level of digital image processing make the interpretation of the digital image based on the

attributes specified in the input. Objects have been identified and interpreted.

The concept of the digital image processing can be seen in the *Fig. 3.*:

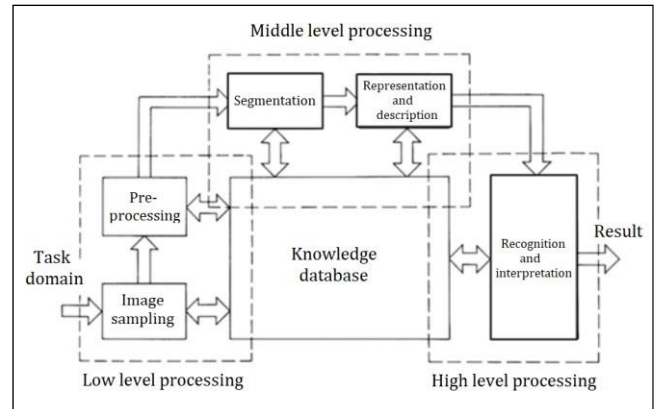Fig. 3. The concept of the digital image processing [9]

IV. THE SOFTWARE COMPONENT

A. Generally about software components

Both machine vision and digital image processing requires a software component. The software component is an algorithm or a set of algorithms that, based on the data collected, make a decision about the results of machine vision or digital image processing [10].

The role of the software component can be decision-making logic based on circuit board or PLC interface, or even in more complex cases, a "deep learning" based neural mesh may be necessary to successfully implement processes. In addition to decision-making processes, the software component can perform other digital image processes, such as segmentation and representation and pre-processing [10].

Software components can be provided by manufacturers, for example, FANUC iRVision, Beckhoff TwinCAT Vision, or NI Vision Development Module [11], [12], [13]. Manufacturer-independent software libraries are OpenCV, VLFlat, or PCL [14], [15], [16].

The optical inspection of the given mounted printed circuit board was implemented with Vision Assistant and LabVIEW software produced by National Instruments.

B. National Instruments Vision Development Module

The Vision Development Module is the National Instrument own product related to the many kinds of image processing applications. It can help develop machine vision, computer vision and image processing applications [17]. The Vision Development Module is connected to the main software product of the National Instruments, which is the LabVIEW. The module can be used with the LabVIEW graphical programming environment, C, C++ or C# programming languages for Windows-based operation systems, and LabVIEW software for real time systems as well.

The Vision Development Module includes hundreds of image processing algorithms with filters, patterns, matchings, classifications and morphologies. The module has own IP and it can use both real time processors and FPGA controllers. The Vision Development Module is also compatible with deep learning algorithms created by TensorFlow [17].

The module has also a special kind of developing environment that is called NI Vision Assistant. In this environment different types of image analyser applications can be managed. The programming happens graphically, the functions are previously defined and the user can customize the properties. Every block means an operation on the picture and they follow each other [17]. In the Vision Assistant, many applications can be managed such as object detection or recognition, text reading and barcode reading. When an application is ready and the tests are successful, the user has the opportunity to generate LabVIEW code to integrate the application in LabvIEW programs or to associate it to a bigger process [17].

C. National Instruments LabVIEW environment

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a platform that is available for system-design, and a development environment to use visual programming language that is National Instruments own product. It has a special kind of dataflow programming, the structure of the graphical block diagram determines the flow of the execution. On this the user connects different function-nodes by defining wires. The wires propagate variables and the processes can be executed when the necessary data become available [18].

The software includes user interface creation possibility during the programming. The subroutines are termed virtual instruments that are called VIs. Each VI possesses 3 different components. These components are the front panel, the block diagram and the connector pane. The Front panel contains the elements of the user interface, such as controls or indicators. In this case controls are inputs and the can receive the commands from the user. The indicators represent the outputs and they can display the results of the given inputs. The block diagram panel contains the structure of the program. The structure includes the elements and the connections between that are represented by cables for the right variables [18].

The great advantage of the LabVIEW is that it has a large library, which contains functions for data acquisition, signal generation and processing, mathematic models, statistics, signal conditioning and analysis. In LabVIEW it is very easy to program multiple tasks by using parallel programming method such as connecting two or more while loops to two separate nodes. It offers a great opportunity to test system automation and to run processes, sequencing, data recording and hardware interfacing parallelly [18].

V. THE PROCESS OF THE OPTICAL INSPECTION

Today, as the complexity of electronic devices increases and the functionality of the devices expands, the printed circuit boards in them become more complex. By drastically reducing the size of the circuit board, more components can be placed on a particular printed circuit board.

However, as the complexity of the circuit boards increases, the number of fault options also increases, so checking them is a higher priority. The mass production and market trends, require that the quality of the finished products be unobjectionable and that their production time be reduced. However, the use of an appropriate control method also affects the duration of production.

One of the most advanced methods is optical inspection. In doing so, based on the recording of the circuit, a computer algorithm performs the verification process according to the set control conditions. Aspects during optical inspection are:

- identification of components,
- geometric measurement/control of components,
- checking the number of components,
- Checking texts and signs.

After the inspection, the result should be documented, which is also important to be well algorithmised, otherwise it would cause congestion in the rapid control processes. The project is designed to be mounted on a printed circuit board for optical inspection using National Instruments Vision Assistant and LabVIEW software. During the project, the example the following tasks was completed:

- detection of 7 different types of SMD (Surface Mounted Device) resistors with label and position data,
- detection of 3 different types of SMD integrated circuit with position data,
- detection barcode and text with spelling display and position data.

Optical inspection reports were automatically generated on Microsoft Excel platform.

VI. THE IMAGE PROCESSING APPLICATION

A. Training of resistors and integrated circuits

There are 3 aspects of the optical inspection:

- implementation of the image processing application using NI Vision Assistant software,
- integration of the image processing application into program and the creation of a user interface using NI LabVIEW software,
- automatically generating a report using Microsoft Excel software.

The image processing application was developed using NI Vision Assistant software produced by National Instruments. As the first step in the application, 7 SMD resistor detecting and testing were carried out. For this purpose, resistors with

the following inscriptions were selected: 1000, 4321, 1002, 472, 472, 104 and 0. For this purpose, a 4400x2933 resolution colour image was taken from the printed circuit board. In the Vision Assistant interface, the picture can be displayed in original resolution during the developing processes as well.

The resistors were trained on the OCR/OCV (Optical Character Recognition/ Optical Character Verification) interface of the Processing Functions menu Identification. By opening the user interface, the inspection operation can be named on the Main menu. The object can be trained on the Train tab, where it is necessary to create a sample file about the object. This can be accomplished from a predefined file or by searching and selecting the object in the image. Since the sample file was detected using Vision Assistant's own search algorithm, it was important to select the highest possible image about the sample object, which contains as few disruptive elements as possible that can be masked if it is needed.

After training, the inspection parameters were defined. This was accomplished on the NI OCR Training Interface, where the search process can be widely customized. Adjustable required color value ratio, recognition mode, minimum recognition accuracy, recognition orientation, recognition dimensional tolerance, etc. After setting the appropriate inspection parameters, the procedure for that object was tested. This can also be accomplished on this interface, and the result of the inspection can be seen as well. After successful test results, the parameters were saved into OCR/OCV procedure for the detection of the object.

The detection of the first type of resistors in NI Vision Assistant environment can be seen in the *Fig. 4.*:

Fig. 4. The detection of Resistor 1 in Vision Assistant [Own figure]

Based on the previous method, the selected resistors and the selected 3 different types of integrated circuits on the printed circuit board have been trained and tested. Every detection means one operation part of the image processing application. They run in a row following each other, and this process is visualized by the Vision Assistant.

First of all the detection of 7 resistors is carried out from 1 to 7, followed by the detection of 3 integrated circuits from type 1 to type 3. The results of the inspection are tabled in the Vision Assistant, and the results are displayed in the sample image.

B. Training of barcode

The next step was to implement barcode reading. For this purpose, the sample printed circuit board also contained 1 barcode. This was also trained on the OCR/OCV interface of the Processing Functions menu Identifications. Vision Assistant has a surface that is specifically designed to train and detect barcodes.

The detection of barcode in NI Vision Assistant environment can be seen in the *Fig. 5.*:
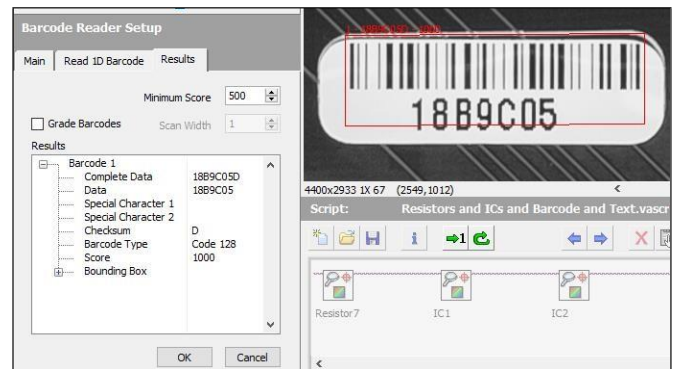

Fig. 5. The detection of barcode in Vision Assistant [Own figure]

Before the barcode was detected, it was necessary to make changes to the image that was used as a sample. Because the barcode is a set of dark characters on a light background, it is important to convert the color image to grayscale to improve the likelihood of correctly detecting the barcode. For this purpose, the conversion was placed before the barcode recognition operation, which is the HSL - Luminance Plane conversion that is available on the Color Plane Extraction Setup. The color image is turned into a grayscale image.

Because the barcode is similar to text, and the detection method is largely the same as the text, a rotating operation has been set before the barcode detection to correct the original rotation of the original image relative to 0°. This was implemented on the Geometry Setup interface and the value of the angle was 180.5°. After the changes have been integrated into the image processing application, the barcode detection has been trained. The training and testing process was carried out in the Barcode Reader Setup interface.

C. Training of text

The next step was the detection of consistent text on the mounted printed circuit board, and to display characters. Text detection and the parameters were defined and implemented in the OCR/OCV interface. The text is trained on the text snippet of the sample image. The Vision Assistant detects the characters based on the same method as in case of shapes, but an important difference that the software needs to detect the characters one by one. Detection succeeds when each character is recognized as a custom shape by software. This allows the user to identify each recognized character. The Vision Assistant associates the detection characteristics of a recognized character with the user-defined typed equivalent,

so that the algorithm displays the printed version of the character at the next detection time, reproducing the text. Because there are many characters (both letter and number) that are very similar to each other, the sample text should be trained in as high-quality and contrasting as possible, to store the software a more accurate sample image of the character, reducing the likelihood that the software will detect a character as a different character.

The image processing application that was implemented in NI Vision Assistant environment consists of a total of 15 consecutive steps. The first step is to verify that an image of the correct type and format is uploaded to run image processing. From the second step to the 8th step, the detection of resistors is realized, from 1 to 7. This is followed by the detection of 3 integrated circuits, also from 1 to 3. Next steps are the grayscale conversion and the rotation conversion. These are followed by the barcode detection and scan, as well as the detection of text and the display of characters.

VII. IMPLEMENTATION IN LABVIEW ENVIRONMENT

A. Creation of LabVIEW code

When the image processing application was complete in Vision Assistant, LabVIEW VI was generated to embed it. This was done by command Create LabVIEW VI in the Tools menu. As a result, all parameters and operations in the image processing application were generated by a LabVIEW VI. To make the optical inspection process easier to use, a user interface (Front Panel) was designed for the LabVIEW implementation. The sample image of the printed circuit board is located on the Front Panel. The results of the detection processes are displayed in this image.

There is also a Tab Contol, which has 11 pages surfaces to complete the 11 detection process. On the interface of the resistors and integrated circuits, after the detection has been performed, the number of results, the position data (X and Y), the result orientation, the score threshold, the match value, and the bounding box position data (X and Y) for marking can be seen. On the page of the barcode and text the positions data (X and Y), the number of matches and the detected characters can be seen.

The detection of the first type of resistors in NI LabVIEW environment can be seen in the *Fig. 6.*:

Fig. 6. The detection of Resistor 1 in LabVIEW [Own figure]

There are also Start and Stop buttons on the user interface, which start or stop each detection process and a message box that provides instructions for them. Each operation has its own graphical program part for integration and modifiability. Both the Block Diagram and the Front Panel is generated for the application, which provides the ability to develop a user interface.

The detection of barcode and text in NI LabVIEW environment can be seen in the *Fig. 7.*:
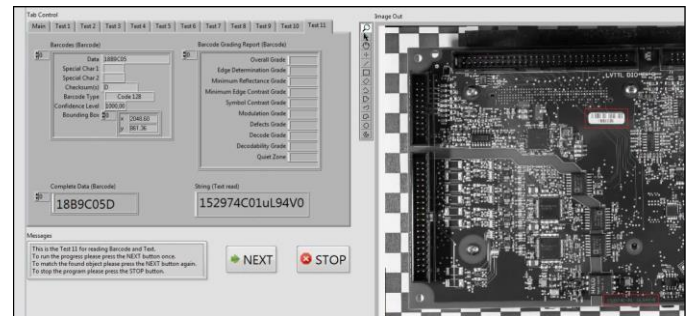

Fig. 7. The detection of barcode and text in LabVIEW [Own figure]

The LabVIEW code is basically divided into 4 parts. The first part read the right image from the given path of data. The second part is responsible for performing the image processing operations. Since 7 resistors, 3 integrated circuits, barcodes and text are trained, each operation is placed in a Case Structure from Test 1 to 11. The third unit contains the operations responsible for marking the detected objects. As a result, LabVIEW draws a red coloured bounding bix around the found objects according to the specified parameters. The fourth part is responsible for reporting the verification process. File Dialog, IMAQ Create, and IMAQ ReadFile blocks paste, name and read the suitable image. To process the specified image file, a temporary memory location is created using the IMAQ Create block. The IMAQ ReadFile block reads the image according to the specified parameters.

All operations of the application placed in a While Loop that provides the cycle running. Within the loop, the object recognition operations are assigned to case numbers in a Case Structure. This allows the user to separate the execution of operations from each other. When a specific case number of the Case Structure is selected, only that operation runs that is associated with the given case number. The case numbers are associated with the Tab Control pages according to the case numbers. This number is 11, which includes the detection processes of 7 resistors, the detection processes of the 3 integrated circuits, and the detection and reading operation of barcode and text. Case Structure case numbers can be selected by the tab tabs in the Tab Control interface of the Font Panel.

The visual display of the detections was implemented with an Event Structure block. Each detection process is considered a separate event, the right one is displayed in the image on the Front Panel by the Image Out indicator. After the results are collected in Index Array, and then converted to integer data format, the IMAQ Overlay Rectangle block displays bounding

boxes around the detected components according to the given parameters. In case of a new detection process, the image is updated by the IMAQ Clear Overlay block. The final step in the optical inspection process is to prepare a report. The purpose of the report is to ensure that the results of the inspection are recorded in tabular form. Generating report is the fourth part of the of the LabVIEW code. An output of the Case Structure contains the position data of the detected objects. The data are collected in an array and then converted to string data format using Number to Decimal string conversion blocks. The transformed data are exported with a Write Delimited Spreadsheet block, according to corresponding definition of the save path.

The report of the first type of resistors in Microsoft Excel can be seen in the *Fig. 8.*:

Index	Label	Value [Ω]	X Position	Y Position
1	1000	100	1538	273
2	1000	100	1616	272
3	1000	100	855	513
4	1000	100	672	515
5	1000	100	592	514
6	1000	100	770	512
7	1000	100	482	515
8	1000	100	392	515

Fig. 8. The report of the first type of resistors [Own figure]

Each type of resistors, integrated circuits, and the barcode and text display have a separate report file. The formats of the files are.xls and they can be opened by Microsoft Excel software. A report of a resistor type contains the number of found objects with indexes, the label on the resistors, the resistance values in [Ω] units, and the *X* and *Y* positions data. In the case of integrated circuits, the inspection reports contain the above-mentioned data, except the values.

The barcode and text are also recorded in a report that includes the spelling of labels, the type of labels (barcode or text), and position data (*X* and *Y*). Completing the optical inspection one time, 11 reports are generated, of which 7 contain the resistors, 3 contain the integrated circuits and 1 contains contains the barcode and text. One full process of the inspection including all resistors, integrated circuits, barcode, tex, report generation runs for approximately 10s, and if the user wants to customize it and accomplish only the desired detections, then it can be shorter duration. The inspection was tested on the computer that uses Windows 10 Pro 64-bit operating system, it has Intel Core i5-3337U CPU, it has 8 GB RAM, and it has NVIDIA GeForce 710M graphic card.

VIII. CONCLUSION

In the course of the project, the optical inspection of mounted printed circuit board was realised using the development environment of National Instruments Vision

Development Module and LabVIEW software. After developing the image processing application onto the Vision Assistant platform, it has been implemented into the LabVIEW development environment. Here a user interface has been created for the optical inspection and the process has also been visualized and documented in the form of Microsoft Excel-based reports. The optical inspection has been tested on different images of the same type of the sample printed circuit board. The tests have also produced good results. Further development opportunities are improving the user interface, expanding the image recognition application by detecting additional components, improving detection results, and developing the barcode and text interpretation.

ACKNOWLEDGEMENT

REFERENCES

[1] V. H. R. B. M. Sonka, Image processing, Analysis, and Machine Vision, United States of America: Cengage Learning, 2015.

[2] E. R. Davies, Computer and Machine Vision: Theory, Algorithms, Practicalities, 4th, Elsevier, 2012.

[3] A. Hornberg, Handbook of Machine and Computer Vision: The Guide for Developers and Users, Wiley, 2017.

[4] R. C. Gonzalez, R. E. Woods, Digital Image Processing, New Jersey, Prentice-Hall, Inc., 2002.

[5] Z. D. V. S. D. S. A. Dr. Rövid, The parallel models and algorithms of the Machine Vision and Image Processing, Budapest: Typotex, 2014.

[6] T. L. Floyd, Digital Fundamentals, Global Edition, Courier Kendallville, United States of America, 2015.

[7] MapYourTech.com, "Pulse Code Modulation", [Online]. Available: http://www.mapyourtech.com/files/resized/320833/652;434;82780895d3 0b355e5296ba8f80de8808a170c0c0.png. [Accessed: 17.01.2020.].

[8] B. H. A. R. J. Beck, Human and Machine Vision, United States of America: Elsevier Inc, Academic Press, 1983.

[9] C. T. M. H. P. T. J. C. I. Brilakis, Infrastructure Computer Vision, Elsevier, 2019.

[10] M. J. T. P. J. Smisek, Consumer Depth Cameras for Computer Vision: Research Topics and Applications, London: Springer-Verlag, 2013.

[11] Machine Vision, „FANUC Corporation," [Online]. Available: https://www.fanuc.co.jp/eindex.html. [Accessed: 17. 01. 2020.].

[12] Beckhoff TwinCAT, „Beckhoff Automation," [Online]. Available: https://beckhoff.com/. [Accessed: 17. 01. 2020.].

[13] N. Vision, „National Instruments," [Online]. Available: https://www.ni.com/hu-hu.html. [Accessed: 17. 01. 2020.].

[14] OpenCV, „OpenCV," [Online]. Available: https://opencv.org/. [Accessed: 17. 01. 2020.].

[15] VLFeat, „VLFeat," [Online]. Available: http://www.vlfeat.org/. [Accessed: 17. 01. 2020.].

[16] PCL, „PointClouds," [Online]. Available: https://pointcloudlibrary.github.io/. [Accessed: 17. 01. 2020.].

[17] National Instruments, "Vision Development Module", [Online]. Available: https://www.ni.com/hu-hu/shop/select/vision-development-module. [Accessed: 17. 01. 2020.].

[18] National Instruments, "LabVIEW", [Online]. Available: https://www.ni.com/hu-hu/shop/labview.html. [Accessed: 17. 01.2020.].