

**SILESIAN UNIVERSITY OF TECHNOLOGY**

**FACULTY OF AUTOMATIC CONTROL, ELECTRONICS AND**

**COMPUTER SCIENCE**

Final Project

Integration of vision detection of manipulation object position software with Adept robot controller

Author: Artur Meller

Supervisor: Tadeusz Szkodny, PhD, DSc.

Consultant: Adrian Łęgowski, MSc, Eng.

Gliwice, February 2015

Table of Contents

[1 Introduction 3](#_Toc410811298)

[2 Scope of work 4](#_Toc410811299)

[2.1 Aim of the project 4](#_Toc410811300)

[2.2 Used elements 4](#_Toc410811306)

[3 Robotized stations with vision systems 6](#_Toc410811307)

[3.1 Elements of the station 6](#_Toc410811308)

[3.2 Methods of communication 7](#_Toc410811309)

[3.3 Vision systems 8](#_Toc410811310)

[4 Software description 9](#_Toc410811311)

[4.1 Computer program 9](#_Toc410811312)

[4.1.1 Graphical User Interface 9](#_Toc410811313)

[4.1.2 Image processing 10](#_Toc410811319)

[4.1.2.1 Image acquisition 11](#_Toc410811320)

[4.1.2.2 Thresholding 11](#_Toc410811321)

[4.1.2.3 Erosion/Dilation 14](#_Toc410811322)

[4.1.2.4 Object recognition 16](#_Toc410811323)

[4.1.3 Communication with robot controller 17](#_Toc410811324)

[4.2 Robot controller programming 18](#_Toc410811325)

[5 Communication Tests 20](#_Toc410811326)

[5.1 Test’s scenario 20](#_Toc410811327)

[5.2 Course of tests 20](#_Toc410811328)

[5.3 Results of tests 24](#_Toc410811329)

[6 Summary 25](#_Toc410811330)

[7 References 26](#_Toc410811331)

# Introduction

The development of industry and general automation is raising challenges to designers of robotized systems. It is becoming increasingly popular, to carry out industrial processes with minimal or even zero human participation. Autonomous robotized system equipped with video detection, are capable of executing environment analysis and decision-making on its basis.

Vision systems are used in a growing number of robotic applications. They are used to determine the position and orientation of an object of manipulation to palletizing, packaging etc. Welding robotized systems with vision, can determine trajectory of welding and assess the quality of the weld. Refilling of the holes in semi-finished products is carried out by dispensing the appropriate amount of substance. In this case, the vision system locates defects and calculates volume of substance based on the picture taken. These some of the many applications clearly show how wide the possibilities of robotized systems equipped with vision detection are.

Modern vision sensors are designed to not only take a picture of the environment. Modern camera models such as models of family SENSOPART VISOR Color or OMRON FQ2, besides the image capturing, can carry out its processing, locate specific image elements, even take responsibilities of server in Client-Server communication. Once programmed, the user-friendly graphical interfaces can be disconnected from the computer and work independently in the station.

# ****Scope of work****

This chapter is devoted to the objectives and assumptions that should be implemented in the framework of this project.

## ****Aim of the project****

The aim of this project was to develop software to enable communication between cameras, computer and robot controller. Video detection algorithm was implemented, which with usage of image processing methods, can locate characteristic points on the processed image. The next step was to establish communication between computer, robot controller and industrial camera. The scope of work includes the following:

## Learning to program an industrial robot Adept SIX300 (with help of literature [1,2].

## Learning to handle industrial cameras.

## Gaining knowledge of processing of digital images.

## Getting familiar with the topic of industrial networks and methods of communication.

* Reviewing the literature on C# programming language [3].
* Writing a program, that allows to download images from surveillance camera, process digital photo, locate specific items and send the coordinates of these elements to robot controller.
* Programming of an industrial robot controller so that it is able to generate message and receive answer, concerning object coordinates.

## Design of computer program and tests.

## ****Used elements****

During project realization, following elements were used:

* EDIMAX IC-7100P surveillance camera shown in Fig.1



Fig. 1. EDIMAX IC-7100P Camera

* Adept SIX300 industrial robot with controller shown in Fig. 2.



Fig.2: Adept SIX300 robot arm

* Computer with Windows 7 and Visual Studio Express
* Router

# ****Robotized stations with vision systems****

This section is devoted to the description of robot stations equipped with a vision system. Communication methods within the station are described. Also, the problem of image processing methods for determining the location details in vision systems was raised.

## ****Elements of the station****

The basic elements of the robotized station equipped with vision system [4] are:

* camera,
* computer,
* robot controller,
* industrial robot.

Image is obtained by a camera or a vision sensor, then processed by a processor or a computer with appropriate software. Result of the analysis is next transmitted to the robot controller, which on the basis of it, moves the robot arm.

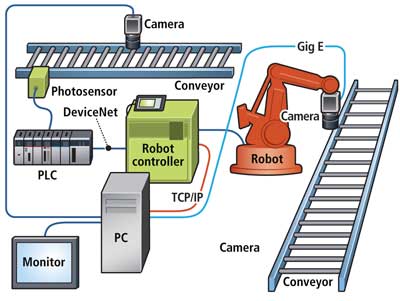


Fig.3: Example of robotized station with vision system

Figure 3 shows the components of the robotic station with vision system. This station also includes a PLC, which communicates with the robot controller using the Device/Net protocol. In practice, the equipment of the station like this can be multiplied in order to enhance its capabilities and consequently obtain almost completely autonomous industrial process.

## ****Methods of communication****

For communication between digital devices, industrial networks are used. High reliability in this case is required. Main result of industrial network work is to have reliable information on the industrial process.

Messages in industrial networks are short (they have small size), but they are often sent, in opposition to general, office networks.

To establish communication between measurement equipment, Fieldbus family of protocols is used, which includes the most common protocols developed by the manufacturers of industrial equipment such as Modbus, Profibus, CC-Link, etc.

Today, however, more and more popular to use in industrial network is Ethernet, that gradually displace another networks. The main advantages of Ethernet networking are compatibility, efficiency and versatility. It also offers constantly increasing transfer speeds, without generating additional costs. An important advantage is the fact, that this network allows the operation with multiple transmission speeds at one time. The baud rate is therefore not limited by the slowest device. Ethernet allows to communicate using different protocols with usage of the same cables. It is not necessary for each of the devices to have the ability to handle all the data formats used. Protocols that are necessary for the transmission of information are TCP and UDP.

One of the proposed architectures for communication is TCP / IP model, whose name is a combination of two protocols of its composition - Transmission Control Protocol and Internet Protocol. This model has a layered structure [5] and is organized in four conceptual layers which are built up on the fifth - device layer:

* The Application layer – user data are generated,
* The Transport Layer – provides host-to-host or end-to-end communication services for applications within a layered architecture,
* The Internet Layer – are used to transport packets from the originating host across network boundaries,
* The Link Layer – operate on the link which has hosts connected to.

Another aspect of designing communication is choice between peer-to-peer or client/server model. The main difference is the lack of one, main server in peer-to-peer communication.

## ****Vision systems****

Advanced vision sensors are equipped with a number of detectors, whose task is to analyze the scene and send the data needed to make a decision about the motion of the robot arm. Detection of the characteristics of the image is the first thing that is done by the vision system. The basis for recognition method is to compare parts of an image to a predefined template.

Image analysis involves the detection of characteristic configurations of pixels that are arranged in a certain structures:

* Edge detector allows to isolate parts of the image, in which the rapid changes in the intensity of the color component occurs.
* Horn detector works in the same way as edge detectors, additionally can detect sudden changes in direction of the edge.
* Blob detector can distinguish parts of image with constant properties.

In addition, image analysis allows to obtain accurate information on the scene such as:

* brightness,
* contrast,
* gray level.

In order to determine the coordinates of the workpiece, a first group of detectors is applied. Contrasting with the background element is detected by one of the above methods. Then, the corresponding algorithms allow to calculate the center of gravity of element and send its coordinates to a computer or robot controller.

An important factor in the robot-camera connection is suitable calibration [6]. The aim of the calibration is to determine the parameters defining the relationship between the base robot system and the system associated with a camera, which are combined with the transformation of perspective and parameters related to optics of the camera. Knowing these relationships, a computer or robot controller is able to calculate the coordinates in the robot base coordinate system on the base of camera coordinates, and pick up workpiece.

# ****Software description****

The chapter is devoted to the description of the video detection software. Topics of image processing, markers location in the image and software solutions to the problems of extracting images from IP cameras as well as robot controller communicating with a computer program were raised.

## ****Computer program****

The main objective of the project was to write a program that would perform the following functions:

* image acquisition from the camera,
* image processing in order to extract markers,
* markers location in the camera coordinate system.

In addition, in a model for client-server communication, according to which the call has been completed, the program will function as a server for the client (the robot controller). The application is written in C # in Visual Studio Express. To run the program, user should double-click Camera.exe file in Release folder that is attached to this work on a CD.

### ****Graphical User Interface****

Figure 4 shows graphical user interface.

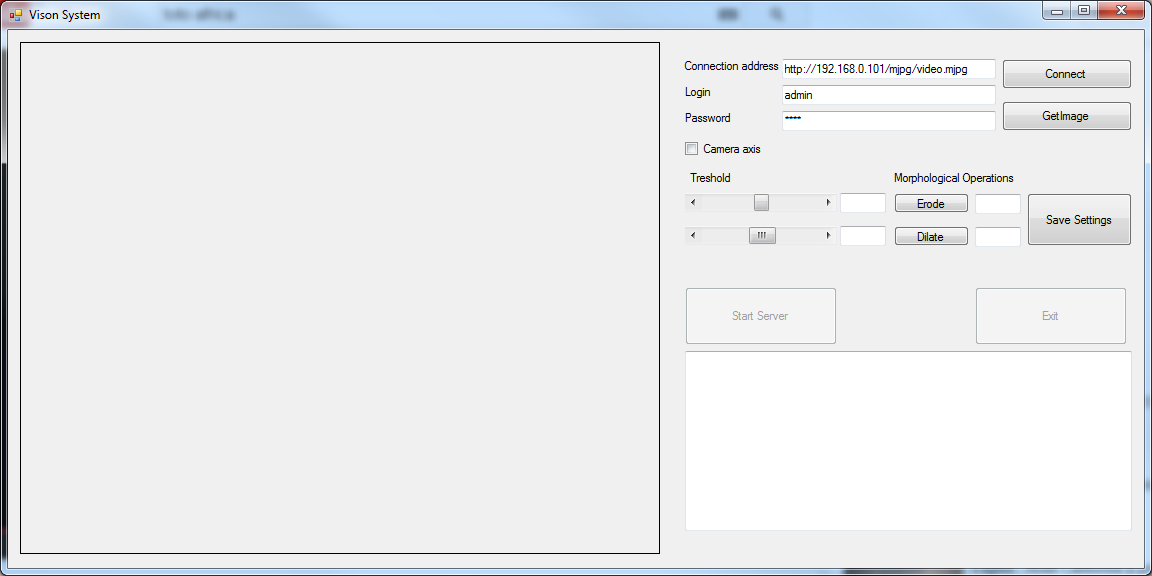


Fig.4: GUI

### The interface can be divided into four segments:

### section devoted to displaying an image from the camera and processed image,

### section that is responsible for communication with the camera and image acquisition,

### the image processing parameters,

### the area, that is responsible for communication with the robot controller.

### ****Image processing****

This section describes the method of digital image processing to obtain marker coordinate. The diagram shown in figure 5 illustrates the process[7]:

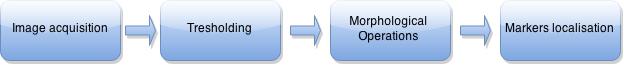


Fig. 5: State diagram of the image processing process

After starting the software, user should manually perform the process. This is a kind of software teaching for object recognition. After the test, the settings must be saved. Then, the subsequent processing and recognition of details will run automatically for each client inquiry. The following paragraphs are a description of the steps of image processing on the example of a color bitmap, presented in the figure, loaded to the program from computer. Application examples of the use of the image captured from the camera can be found later in this work.

#### ****Image acquisition****

Figure 6 shows a section of the interface responsible for connection to the IP camera. 3 text boxes are designed for address the camera image, username and password. Adress of the camera can be obtained by right-clicking on the camera view in the net browser and choosing “Save link as”. Login and password for the camera, if were not changed, can be read from the documentation of the camera. After filling the data, user should press the "Connect" button to view the image from the camera. The current frame is captured and stored for processing by pushing the button "Get Image".

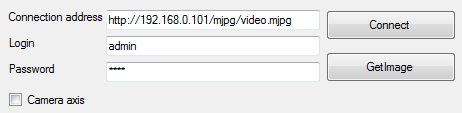


Fig. 6: Data of IP Camera

Class C # MJPGStream was used, through which the image from the specified URL is downloaded to MJPG stream.

#### ****Thresholding****

The first step in image processing is to convert a color image to grayscale. For each pixel illumination rate is calculated according to the formula:

*Y’ = 0.2126R’ + 0.7152G’ + 0.0722B’* [8]

As the weighted sum of the components Red, Green, and Blue. These coefficients are chosen, to well represent the working nature of the human eye, which is most sensitive to the color green, while the lowest blue.

This is followed by thresholding. The objective is to obtain a binary image in which pixels values are 'true' or 'false' depending on their affiliation to the detail or background. This is done in accordance with the thresholding model:

*0 dla f (i, j) < T - t*

*g (i, j) = 1 dla T - t < f (i, j) < T + t*

*0 dla f (i, j) > T + t*

Where:

g(i,j) are pixels in color image

f(i,j) are pixels in greyscale

T – middle value of the threshold

t – half of the threshold

Figure 7 shows a section of the interface for configuration of binarization parameters:

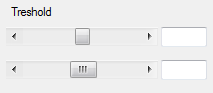


Fig.7: Sliders for T and t parameters determination

The top slider is used to select the T parameter value from 0 to 255. The lower the slider is used to specify the t parameter from 0 to 40. The parameter values are displayed in the text boxes.

Figure 8 shows the color image subjected to processing operations. The image consists of 4 circles on a white background. Each circle has a unique color according to Table 1.



Fig. 8: Image before processing

|  |  |  |  |
| --- | --- | --- | --- |
| Circle | R component | G component | B component |
| Blue | 26 | 181 | 162 |
| Green | 82 | 193 | 58 |
| Pink | 255 | 82 | 95 |
| Orange | 255 | 124 | 6 |

Table 1: Components of circle colors

Figure 9 shows an image converted to grayscale. Table 2 is connected to Table 1 but extended by the gray level values calculated according to the formula.

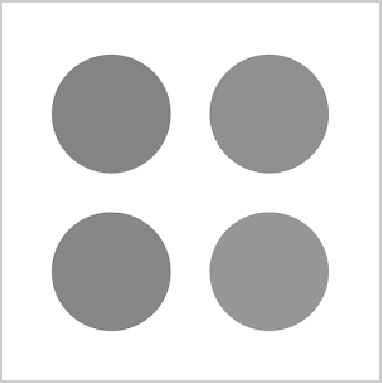


Fig.9: Gray level image

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Circle | R component | G component | B component | Grey level |
| Blue | 26 | 181 | 162 | 146.6752 |
| Green | 82 | 193 | 58 | 159.6544 |
| Pink | 255 | 82 | 95 | 119.7184 |
| Orange | 255 | 124 | 6 | 143.331 |

Table 2: Grey level value for each circle

Figure 10. Shows image after binarization with T = 135, t = 10 parameters.

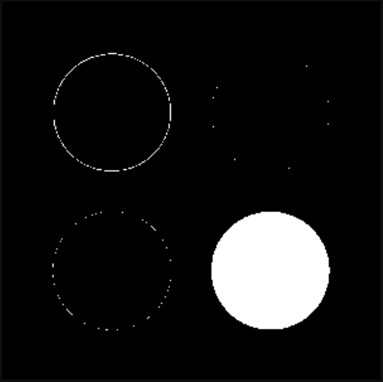


Fig.10: Binary Image

These parameters define the threshold of the minimum value equal to T-t=135-10=125 whereas the maximum equal to T+t=135+10=145. Of the four wheels, only the one with the grayscale value of color is equal to 143,331 fits the threshold. For this reason, on the thresholded image, pixels under that circle were filled with a value of 1. The colors of parts covered by the other circles do not fit the threshold so pixels connected to them ultimately take the value of 0.

#### ****Erosion/Dilation****

Despite binarization process, the image should be subjected to further processing. To remove unwanted pixels from image, morphological operations should be used[9].

One of the terms associated with morphological operations is the neighborhood. It can be used to determine, whether two image points are located next to each other. Frequently 4-connected or 8-conneced neighborhoods are used. In the vicinity of the central pixel is defined so-called structural element - mask, which is used to morphological operations.

The basic morphological operations are dilation (expansion) and erosion (narrowing). These are dual operations that apply a structural element for each pixel in the image. In case of dilation, if even one pixel in the neighborhood of the central point has value of 1 as pixel that belongs to its structural element, then the value of the central pixel is changed to 1. In erosion, if at least one point in a given neighborhood is set to 0, the central pixel is set to 0. The operation dilatation aims to fill the "holes" in an inconsistent picture, and it's consequence is the increase of the object. Erosion, the opposite operation, is used to remove small objects from an image or narrow branches, consequently reducing the object. Performing dilation right after erosion is called opening. The closure is the opposite process. The design of the project uses a 3x3 mask filled with '1' as a structural element.

Erosion of image A by structural element B is described by:

A  \ominus B = \bigcap_{b\in B} A_{-b}

Dilation of image A by structural element B is described by:

A  \oplus B = \bigcup_{b\in B} A_b

Figure 11 shows a part of the user interface used for performing morphological operations on the binary image.

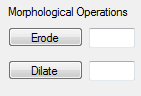


Fig. 11: Erosion and dilation buttons

In the text fields, the amount of operations carried out is displayed. Figure 12 shows the binary image after a single erosion:.

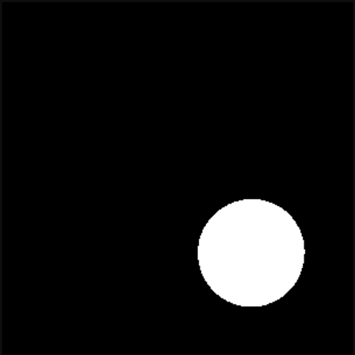


Fig. 12: Image after single erosion

#### ****Object recognition****

In this work it was decided to use as a method of detecting clusters to determine the coordinates of the characteristic points of the image. BlobCounter class was used in this purpose, which counts the individual objects in the image and stored in the memory their parameters, for example coordinates of the center of gravity. Clicking the "Test" activates the detection of clusters. Figure 14 illustrates a portion of the text area in which are listed the coordinates of the center of gravity found element.



Fig. 13: Coordinates of object found

The image area , the circle is used to mark center of gravity of the found object (Fig.14.).

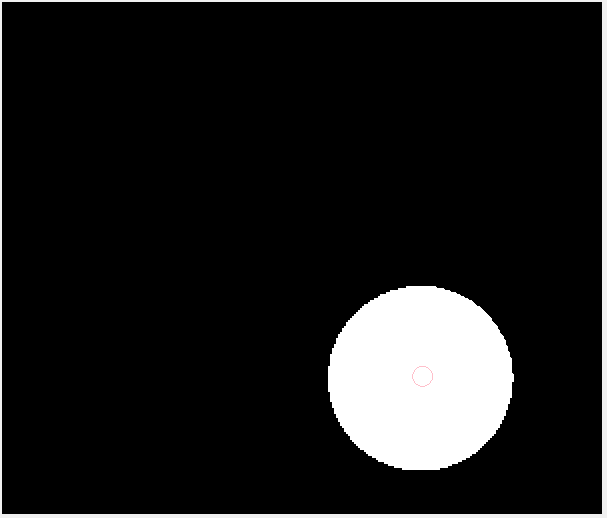


Fig. 14: Center of gravity of found object marked

In case of finding the correct object, the image processing settings have to be saved by clicking on the button "Save settings". Run server button will become active. From this point on, all image processing operations will proceed automatically. In the case of an incorrect finding of the object, entire process should be perform again.

### ****Communication with robot controller****

Computer program that is the subject of this work, in Client-Server architecture, in which was realized the connection between the robot controller and the computer acts as server to the robot controller. TcpListener class was used, that listens for connections from TCP clients. If connection is established, the program waits for the message, which begins the process of image processing as described in Section 4.1.2. In response, the answer containing the coordinates of found objects is sent. The process has been presented in the block diagram illustrated in Figure 15.

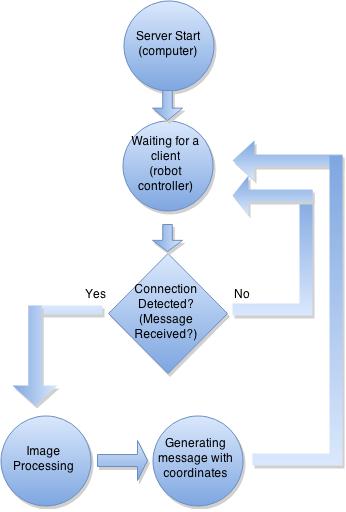


Fig. 15: State diagram of computer program

## ****Robot controller programming****

Robot controller, in architecture Client-Server, takes role of a client to the computer program. In communications, the task is to connect to the server, querying and waiting for a response. Then the answer is analyzed in order to extract object coordinates. A piece of code in the language of V +, that is responsible for establishing communication is presented.

|  |
| --- |
| ATTACH (lun, 4) "TCP"  FSET (lun) "/NODE 'SERVER2' /ADDRESS 172 16 150 1"  FOPEN (lun, 0) "SERVER2/REMOTE\_PORT 5004/BUFFER\_SIZE 1024"  status = IOSTAT(lun)  IF status < 0 THEN  TYPE "Error", $ERROR(status)  END |

An object called "Server2" with the IP address for the computer's network card on which was launched video detection software was created. Then, port 5004, within which communication takes place was open. The process of robot controller program has been presented in the block diagram illustrated in figure 16.

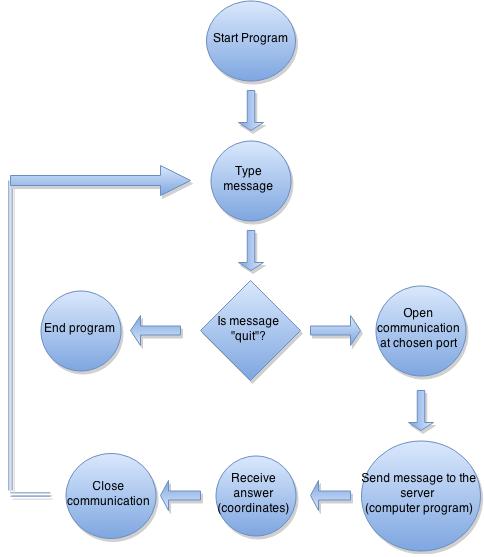


Fig. 16: Block diagram of adept robot program

# Communication Tests

In this section, tests of the applications written during this project are described.

## Test’s scenario

Test station is presented in the figure 17. Robot controller was connected to the computer. Camera was connected via router to another network card in this computer.

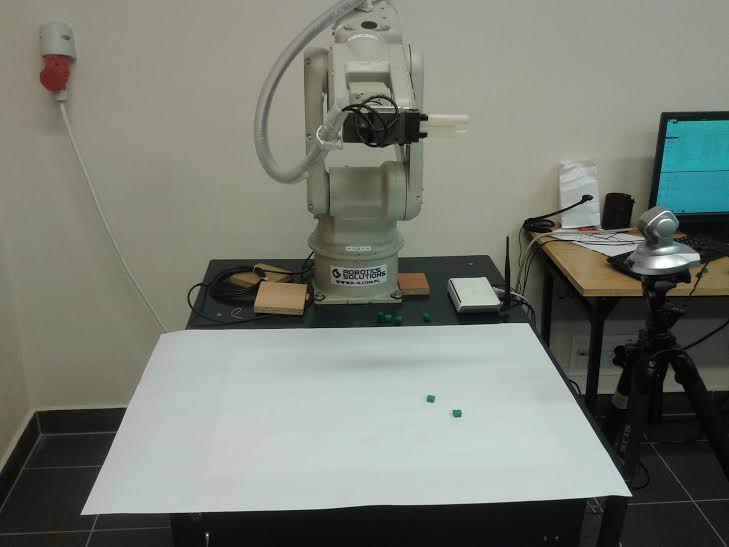


Fig.17: Robot station with camera

The course of the tests assumes execution of a teaching procedure, then start of the vision system server and several coordinates inquiries. Inquiries can be send on trigger or in defined time intervals. In the robot area of motion, two characteristic elements were put. Later on, the number of characteristic points will be increased.

## Course of tests

At first, communication with the camera was established, and then teaching procedure was performed. Figure 18 shows graphical using interface after teaching procedure.

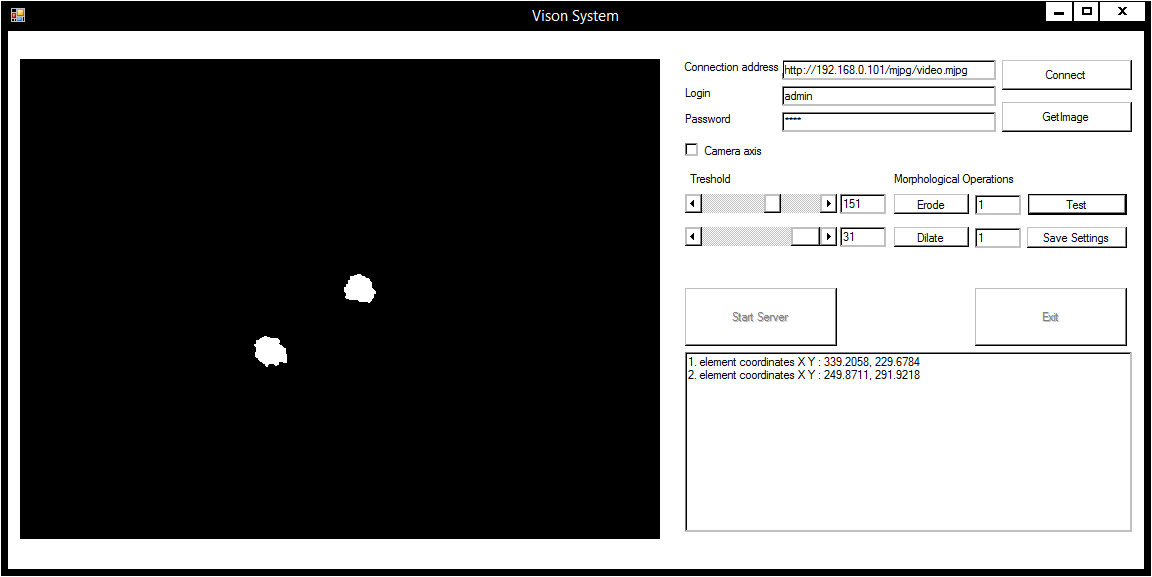


Fig. 18: GUI after teaching procedure

As it can be seen in the figure 18, image processing parameters were chosen as follows:

* T = 151,
* t = 31,
* Number of erosions - 1,
* Number of dilations – 1.

These parameters result obtainment of binary image with one pair of characteristic points of following coordinates:

* 1st element coordinates XY: 339.2085, 229.6784,
* 2nd element coordinates XY: 249.8711, 291.9218.

which were written in the text area after pushing "Test" button. That concludes teaching process. After that, vision system server could be started by pressing “Start Server” button. Figure 19 shows screenshot of a vision system server interface after initialization.

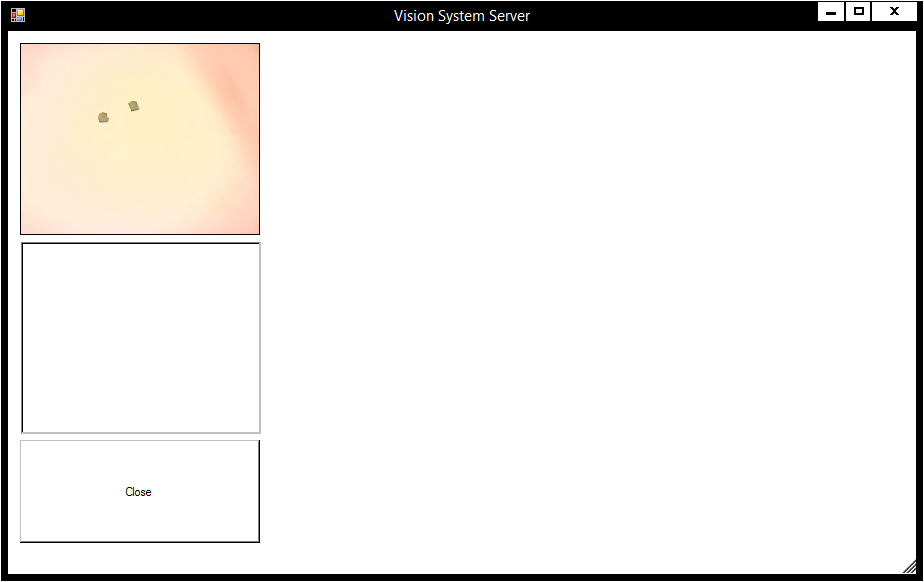


Fig. 19: Vision system server

In the upper-left corner, current view from the camera can be seen. Below is text box for characteristic elements coordinates. Now, since server is running, it is waiting for inquiries from clients. Figure 20 shows Monitor Terminal of Adept Desktop software connected with Adept Robot controller. Communication() function was started. Monitor is asking for an input string and since there is no message concerning error, it means that communication was established properly. “trg” (as trigger) message was sent to the vision system server. Echoed string with coordinates of characteristic points was received. Computer program is adapted to sending coordinates of pairs of characteristic elements. Therefore, message is send in form presented by the formula:

1, x1, y1,z1, x2, y2, z2 … N/2, x(N-1), y(N-1), z(N-1), xN, yN, zN

Where N is the number of located elements.

Now monitor is waiting for another input string from user, as illustrated in the figure 20.

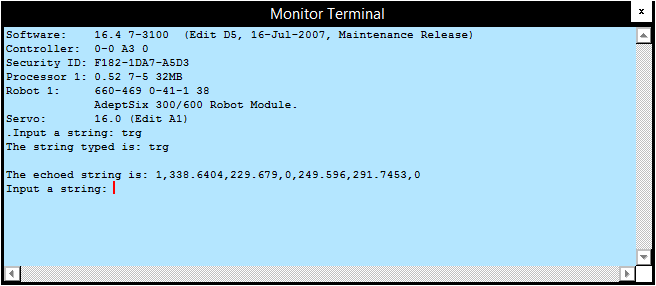


Fig. 20: Monitor Terminal

Vision system server, when inquiry received, displays current frame in form of binary image, in text box coordinates of characteristic points were written as illustrated in figure 21.

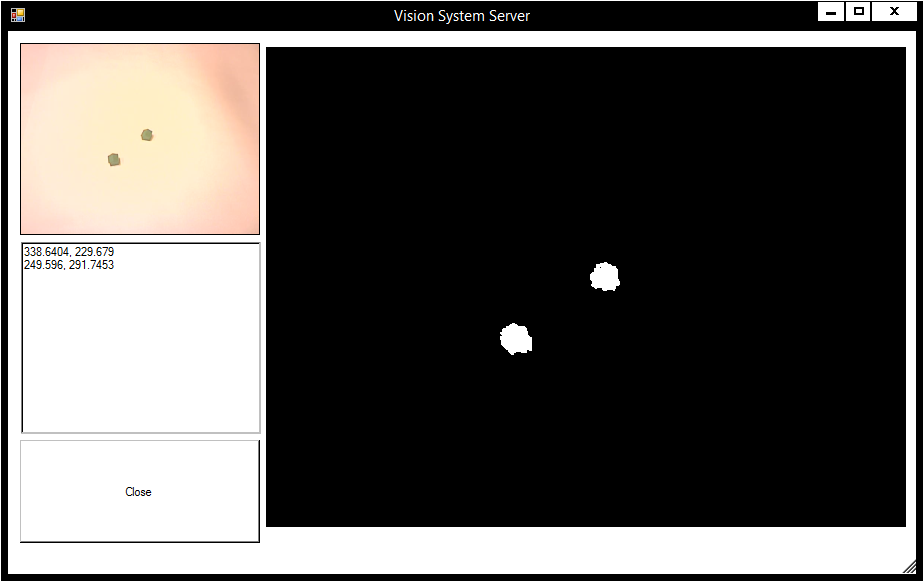


Fig. 21: Screenshot of a vision system server with current frame, coordinates and binary image displayed.

Some inaccuracies were noticed at this point. There is difference in objects position compared to coordinates of objects found during teaching process. The difference between coordinates are from -0.0006 to 0.5681 pixels.Probably it is caused by unnoticeable change of the camera position, however, accuracy of this algorithm should be further investigated.

Another test was performed. Software was not triggered by user, but messages were sent to the server in 5-seconds intervals. Figure 22 shows monitor terminal whereas figure 23 shows vision system server after several exchange of messages.

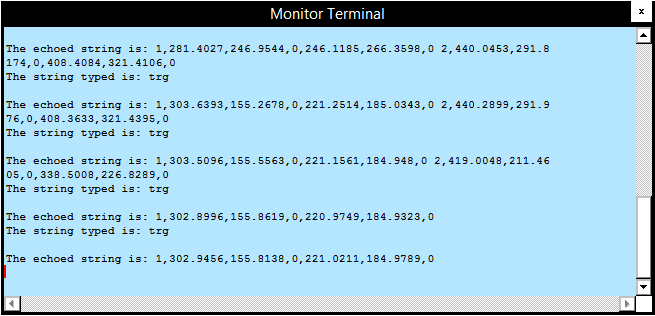


Fig. 22: Monitor Terminal after several exchanges of messages.

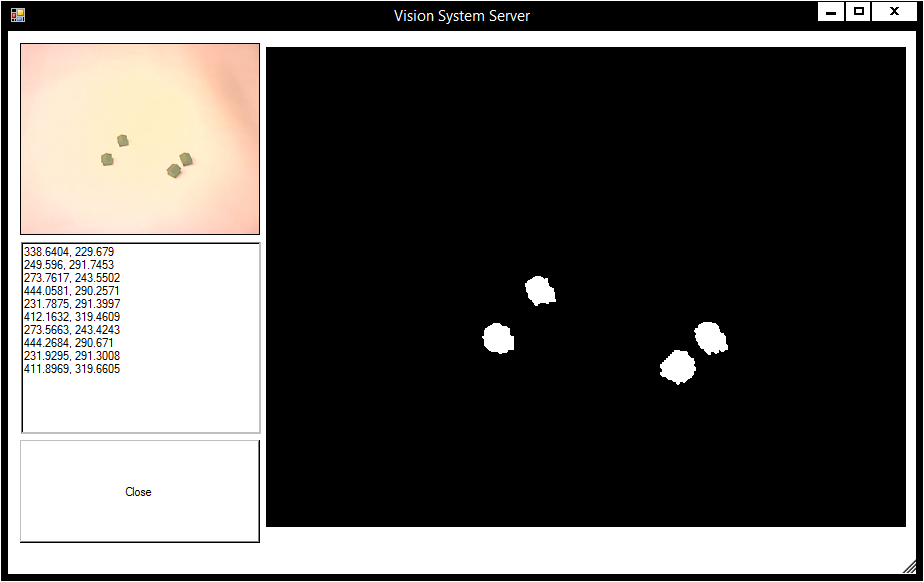


Fig. 23: Vision system server after several exchanges of messages.

## Results of tests

During tests of the connection, after teaching process, several messages were exchanged between robot controller and vision system server. Image processing and element’s location were triggered both by the user via monitor terminal and automatically, within 5-second intervals. Also, different numbers of characteristic elements were put in the robot area of motion. It turned out, that maximal number of pairs of markers is equal to 3. It is limited by the predefined size of message and can be expanded in the code of the program. Messages were exchanged correctly, with no noticeable time loss. There were no “splitted” elements and unwanted pixels which mean that teaching process was performed in a proper way. Accuracy of this algorithm should be further investigated.

# Summary

Vision detection systems are very useful and play increasing role in modern industry processes. However, programmable cameras are, mostly, very advanced systems that have to cost a significant amount of money. Presented solution is customized for every IP camera. Moreover, it can be applied not only to Adept robots, but it is suitable for every robot with Ethernet as a communication option.

The purpose of final project was to design Vision System Software that would allow to acquire of image from industrial camera, perform necessary image processing and to send results to another device. At the same time, Adept robot controller should be also programmed in order to inquire about coordinates and receive them.

These tasks were realized with usage of Ethernet connection in Client-Server model of architecture. Image processing methods as thresholding or morphological operations were used. Predefined functions of C# language were used for markers localization.

However, program is not finished yet. Without proper calibration, there is no connection between coordinates in camera coordinate system and robot’s base coordinate system. Now, calibration should be performed as a next step in robot controller programing and after that, robot arm can reach for the object. Eventually, calibration process could happen inside vision detection software, and sent messages would not only on coordinates of object, but will represent indirect points of robot arm trajectory of motion. Although this is not the subject of this project.

# References

[1] *AdeptSix 300 Robot, Instruction Handbook 2005*

[2] P. Rotyński *Adept Robot Programming. Master Thesis. Institute of Automatic Control. Silesian University of Technology. Gliwice 2011.*

[3]L. O’Brien, B. Eckel *Thinking in C#. Prentice Hall PTR. Upper Saddle River, NJ 2002.*

[4] Zens, Richard *Vision-guided robot system trims parts. Vision Systems Design. PennWell Corporation. Tulsa, OK 2006.*

[5] R. Braden (ed.) *Requirements for Internet Hosts – Communication Layers, October 1989.*

Available on the internet: https://tools.ietf.org/html/rfc1122

[6] T. Szkodny, A.Meller, K.Palka *Accuracy of Determining the Coordinates of Points Observed by Camera. International Conference on Intelligence Robotics and Applications. Guangzhou, China 2014.*

[7] Davies, E.R. *Machine Vision - Theory Algorithms Practicalities (2nd ed.). Harcourt & Company, San Diego, Ca 1996*

[8] M. Stokes, M. Anderson, S. Chandrasekar, R. Motta *A Standard Default Color Space for the Internet – sRGB*

Available on the internet: http://www.w3.org/Graphics/Color/sRGB.html

[9] P. Soille *Morphological Image Analysis; Principles and Application. 2nd edition. Springer Verlag. Heidelberg, Berlin 2002.*