

UDC 621.8

DOI: 10.25140/2411-5363-2019-3(17)-96-104

Marek Sukop, Peter Ferenčík

**DESIGN OF INTELLIGENT ROBOTIC CELL WITH CAMERA SYSTEM**

**Urgency of the research.** More complex robotic systems are characterized by a certain degree of intelligent behavior where, based on input, the system is able to adapt its behavior. The implementation of elements that support intelligent behavior in robotic systems, especially those based on the image of devices, is becoming common practice. The reason is simple, such a system is faster and more accurate.

**Target setting.** Creating machine vision, however, is a complex problem, especially when it comes to applications with non-standard requirements. For each task, the vision system needs to be adapted to the conditions and requirements of the monitored objects. Other image adjustments and algorithms need to be applied to static objects rather than moving objects. Two-dimensional image information is sufficient for some manufacturing process, while others require a third dimension to remove a given piece from a disordered pile. Creating an intelligent robotic cell with a camera system therefore requires the creation of a vision system that meets the specified requirements. This is where space is open, because there are many different procedures and principles to deal with, but not all are equally effective and reliable.

**Actual scientific researches and issues analysis.** Many of the image processing methods can be combined with each other, or a new, better way to solve the problem can be developed using the approaches already known. Adding to this fact non-standard requirements profiled in practice, there is an undeniable reason why it is appropriate to deal with image processing for industrial use.

**Uninvestigated parts of general matters defining** are designing and create a robotic cell, whose activity will be controlled on the basis of image perception obtained by digital camera.

**The research objective** of this article is to design and create a robotic cell, whose activity will be controlled on the basis of image perception obtained by digital camera. The obtained image will be subjected to suitable image processing algorithms which will result in the generation of control instructions for controlling the manipulator movement.

**The statement of basic materials.** The work deals with the design of a robotic cell whose task is to manipulate sample objects placed on the conveyor belt by means of a parallel manipulation robot based on image perception. The main part of the design is the creation of control software, which in the first level ensures the proper functioning of the individual components and in the second level their mutual cooperation, which ensures the performance of the required functionality of the robotic system as a whole. Created software runs on Windows 7 operating system, where it offers a simple tool to control the movement of the arms of a parallel robot without using other control means. This means that the robot's movements can be controlled directly from the control program, allowing the robot and object to be manipulated even in manual mode. The image obtained by the camera can be adjusted by software using the implemented tools before the automatic manipulation begins, allowing the user to set the correct input parameters to ensure reliable object identification.

**Conclusions.** In order to design a robotic system whose operation is controlled on the basis of visual perception, it was necessary to acquire theoretical knowledge for the correct selection of individual components of the system as well as their correct placement within the robotic cell. Great emphasis was placed on suitable and economical selection of the sensing device and the way of illuminating the scanned objects.

In order to obtain the camera image it was also necessary to study and understand the principle of working with the image captured by the camera via SDK issued directly by the camera manufacturer. However, obtaining an image was only the first step to start the image processing process. In order to extract the necessary data from the obtained image and then to create control instructions from the data for controlling the robot, it was necessary to study and learn in detail the individual steps and procedures of image processing.

In the part of the work dealing with image processing the acquired knowledge was applied to the processing process itself, but not only known approaches were used. Owing to the reduction of CPU load and consequently shortening of the calculation process, own procedures were also introduced into the image processing process. The actual "economical" approach was applied and tested in the thresholding process where a "shortened thresholding algorithm" was created. The approach was also applied to the object-in-picture search process, creating a "network-based object-in-picture method" that uses the fact that we search and identify known objects in industrial applications as opposed to identifying objects in an unknown environment.

The combination of image acquisition, image processing and robot control with one comprehensive application is also a major benefit. Of course, to ensure this functionality, it was first necessary to create a theoretical base on which to build. The main problem was to create a control part of the robot control in C# and to link it to the basic control program created in C++.

**Keywords:** robotic cell; machine vision; camera system; intelligent system.

**Fig.:** 9. **References:** 15.

**Urgency of the research.** More complex robotic systems are characterized by a certain degree of intelligent behavior where, based on input, the system is able to adapt its behavior. The implementation of elements that support intelligent behavior in robotic systems, especially those based on the image of devices, is becoming common practice. The reason is simple, such a system is faster and more accurate.

**Target setting.** Creating machine vision, however, is a complex problem, especially when it comes to applications with non-standard requirements. For each task, the vision system needs to be adapted to the conditions and requirements of the monitored objects. Other image adjustments and algorithms need to be applied to static objects rather than moving objects. Two-dimensional image information is sufficient for some manufacturing process, while others require a third dimension to remove a given piece from a disordered pile. Creating an in-

TECHNICAL SCIENCES AND TECHNOLOGIES

telligent robotic cell with a camera system therefore requires the creation of a vision system that meets the specified requirements. This is where space is open, because there are many different procedures and principles to deal with, but not all are equally effective and reliable.

**Actual scientific researches and issues analysis.** Many of the image processing methods can be combined with each other, or a new, better way to solve the problem can be developed using the approaches already known. Adding to this fact non-standard requirements profiled in practice, there is an undeniable reason why it is appropriate to deal with image processing for industrial use.

**Design of intelligent robotic cell.** The paper deals with the design of a robotic cell whose task is to manipulate the sample objects placed on the conveyor belt by means of a parallel manipulation robot based on image perception. The main part of the design is the creation of control software, which in the first level ensures the proper functioning of the individual components and in the second level their mutual cooperation, which ensures the performance of the required functionality of the robotic system as a whole. The control program is created in the Visual Studio C ++ development environment, while the software components for developers (SDK) are used to control individual components for easier and reliable implementation of the device into the control program. Image processing is provided by a combination of OpenCV library and our algorithms. The program consists of three basic threads. The first thread provides image acquisition and processing in the parameter setting mode - the so-called FreeRun mode. The second thread provides control of the robot in manual mode by means of sliders - graphical elements for control. The third thread provides automatic mode of operation based on preset parameters.

**Used components.** The chosen camera for the application was the “Basler Scout scA1390-17gc” camera, whose specific parameters (Fig. 1) best suited the chosen purpose. However, this is a compromise in the selection, given that the selection was made from the facilities at the disposal of the TUKE Robotics Department.

Basler Scout scA1390-17gm					
Sensor	Sensor	Sony ICX267	Camera Data	Interface	GigE
	Shutter	Global		Pixel Bit Depth	12 bits
	Image Circle	1/2"		Synchro	SW/HW trigger
	Sensor Type	CCD		Exposure control	via Camera API
	Sensor Size	6,5 x 4,8mm		Digital Input	2
	Resolution	1.4Mpx (1390x1038px)		Digital Output	4
	Pixel Size	4.65 x 4.65 μm		Power Supply	12 - 24 V
	Frame Rate	17 fps		Consumption	3.5W
	Mono/Color	Monochromatic		Lens Mount	C-mount

Fig. 1. Characteristics of the Basler Scout camera

Most camera manufacturers also offer lenses for their equipment in their portfolio. This is also the case with Basler, where it is possible to select the required lens directly on their website in the lens section. After entering the parameters, two relevant results were displayed. It was a Basler C125-218-5M F1.8f12mm and Edmund Optics CFFL F1.8f12mm 2/3 ”lens that fully fit the purpose, but the purchase price was too high. € 400 for the Basler lens and € 300 for the Edmund Optics. Therefore, the cheaper variant of the Tamron 23FM12-L 2/3 fF12 F / 1.8 lens (Fig. 2), Whose purchase price was 140 €, was more than half that of the Edmun lens, defined parameters.



Fig. 2. Selected lens Tamron

ABB IRB 360 FlexiPicker is a modern device designed for fast handling and / or fast removal of products from conveyor belts and their subsequent storage or packaging. The main advantage of these devices is the high handling speed, which means that the handling cycles are very short and the high accuracy and load capacity, whose maximum value is 8 kg. The Flexi-Picker product line is available in a range of 800, 1130 and 1600mm range ranges and a range of 1.3, 6, and 8kg load capacities, making these devices ideal for wide range of applications. The basic design of the Flexipicker consists of a base where the drive units for the individual arms and three parallel arms are located. The robot is placed on the suspension structure, with the handling space being created directly below the robot. Due to different Flexipicker variants, the workspace is not the same for all variants. The ABB IRB 360-6 / 1600, located at the Department of Robotics, has the largest workspace (Fig. 3) in this product line.

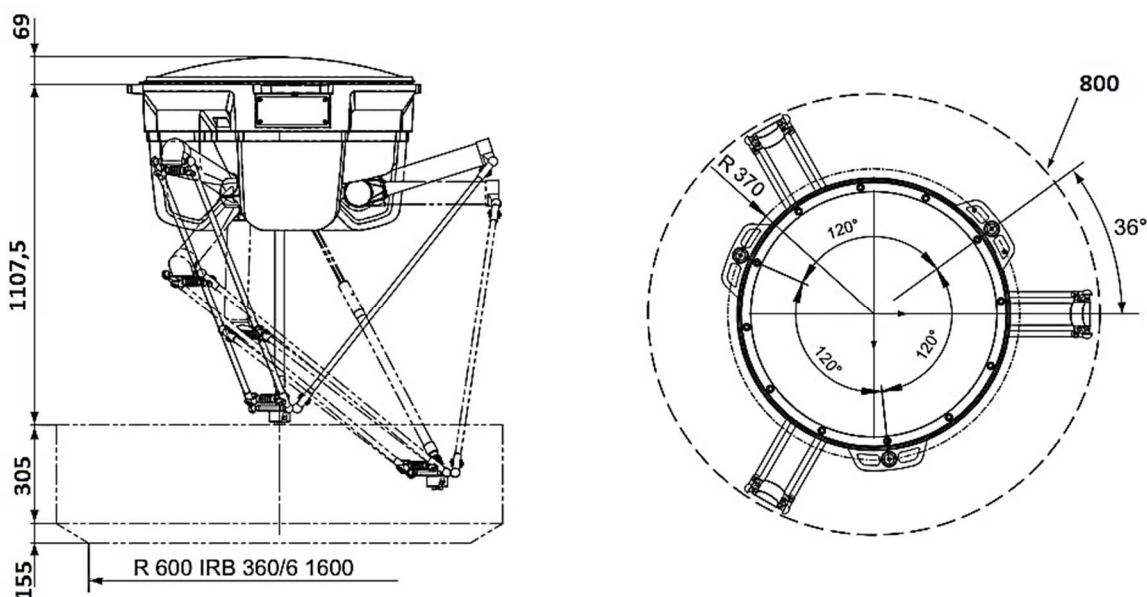


Fig. 3. Workspace of IRB 360-6 / 1600

As this project belongs to the small group, the hardware requirements for it were not extremely extreme. The image provided by the selected camera achieves a resolution of only 1.4MPx and the images will be processed sequentially with a period of 140ms, representing approximately 7 frames per second. The computing power of the control PC is ensured by the AMD FX-6300 processor, which contains 6 physical cores with a clock speed of 3.5GHz (boost up to 4.1 GHz) and a 12MB Chash pmäte. Despite its high performance, this processor is "only" the equivalent of the Intel i5 series processors, but this is enough for this project. The memory is created by two 4GB modules reaching the frequency of 2100Mhz connected in DualChanel mode for faster communication with the CPU. The set is complemented by a graphics card with nVidia GT630 chip supporting DirectX 11, whose clock frequency reaches 810MHz. For storage of data was chosen SSD size of 120GB, which allows writing and reading data at 450MB / s.

**Main program and its structure.** The control program ensures the cooperation and functioning of individual parts of the robotic cell with the camera system in order to ensure the desired result. Since the basic components of the robotic cell are the camera and the robot itself, the control program is divided into three basic parts - the part providing communication with the camera and acquiring the captured image, the part providing connection to the robot control system and its control and the part performing image processing and generation data for robot movement control.

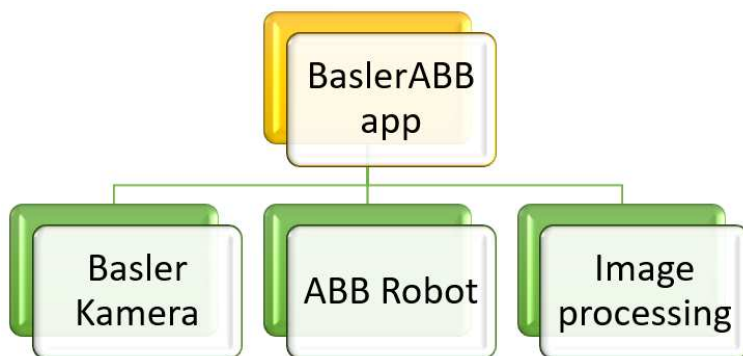


Fig. 4. Basic program blocks

The first block of the program, as already indicated, provides a communication channel between the Basler camera and the control PC. This program block provides software interconnection of the camera and the PC, thereby obtaining the acquired image into the operating memory. Physically, however, the Basler Scout sca 1390 is connected to the control PC via the GigE interface, which means it uses an Ethernet link. Since the primary task is not to create software for camera management and communication between the camera and PC, we will use the official SDK for this purpose, which reliably provides this functionality. The implementation of the Pylon 5 SDK should be performed according to the instructions published by the publisher.

By installing the Pylon SDK, we get an API (Application Programming Interface) in C++ for Basler cameras that use the IEEE1394, GigE, USB and Camera Link interfaces. Thus, the Pylon API offers a broad-spectrum programming interface that frees the programmer from the many differences that exist between individual interfaces, speeding up work and eliminating errors in establishing a communication channel. The new generation of Pylon API version 5 even reduces the hardware components of the control PC. Compared to older version 4, the memory usage is half the load and communication takes three times less time. These two significant benefits are advantageous for older systems, where the introduction of new Basler products does not need to change the hardware configuration of the control PC.

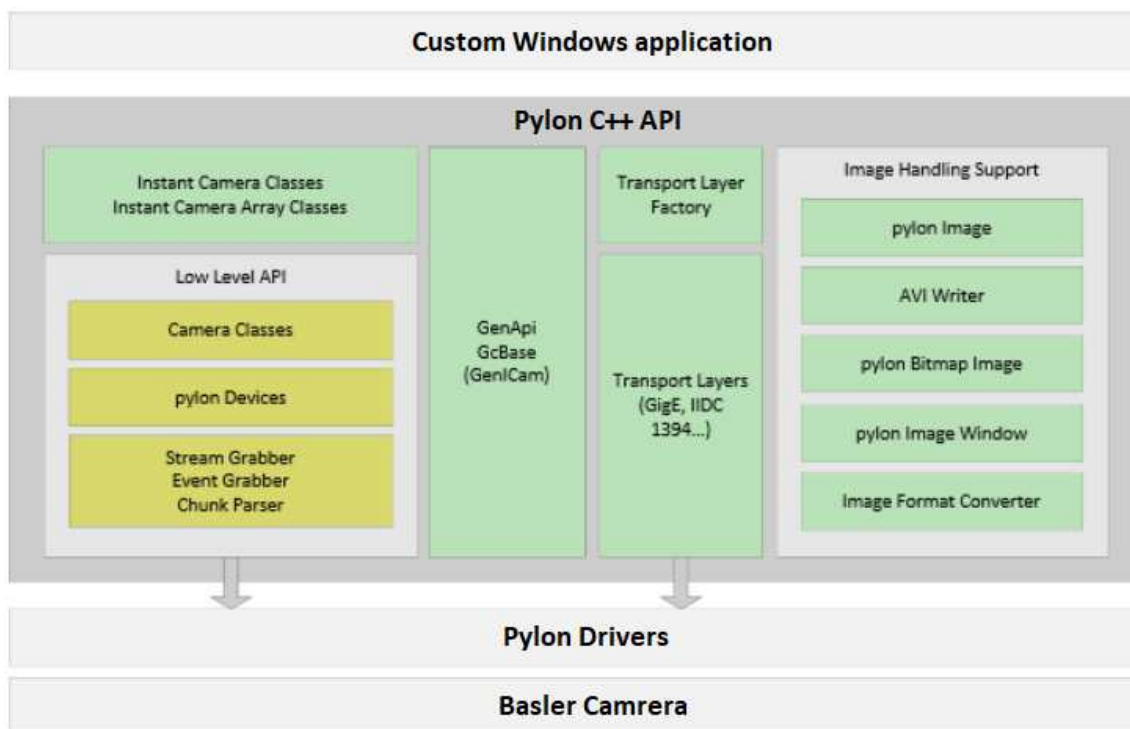


Fig. 5. Pylon 5 API structure

The code being created is almost ready to get the first image from the connected camera, but you still need to set up an image acquisition strategy. By default, the “One-By-One” acquisition strategy is set, which means that the acquired images are processed in the order they arrived. In our case, this method is exactly right, because the image acquisition will be controlled by a software trigger. However, it is possible to set up continuous loop image acquisition and process the image from any buffer found in the "Grab Engine". The maximum number of these buffers is 10, but for our One-By-One acquisition needs, we will limit the number of these buffers to 1. This means that we will still process only the image we capture at the moment.

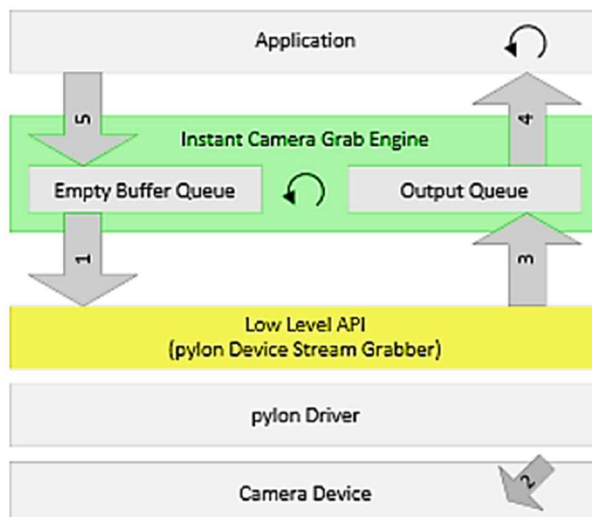


Fig. 6. Picture acquire by strategy One-By-One

The basis of communication and control of the robot is therefore a separate part of the program written in C #, whose functions are called from the main program. However, the call itself does not perform direct management actions in the main program. The call only determines what action to take and provides input parameters. Some calls may result in a return value that is important to the main program, such as querying the current position on the X axis. However, the actual processing of the request ensures part of the managed code.

However, the motion control of the robot is not controlled directly. Because the robot has its own control system on which the Windows application actually being created "parasitizes" it is not possible to control movements by direct command. For the simple but reliable control of the robot's movements it was necessary to "fool" the control system. In the robot control system, I created a simple program that contained a repeating loop whose execution was controlled based on reaching the end position.

```

MODULE MainModule
  VAR robotarget RobotTarget:=[[-0.00,0.00,-870.00],[0,1,0,0],[0,0,0,0]];
  PROC main()
    WaitDI Start, 1;
    Reset InPosition;

    IF (Linear)
    {
      MoveL RobotTarget, v1000, z50, tool0;
    }
    ELSE
    {
      MoveJ RobotTarget, v1000, z50, tool0;
    }

    WaitRob\InPos;
    Set InPosition;

  ENDPROC
ENDMODULE

```

TECHNICAL SCIENCES AND TECHNOLOGIES

The user interface of the application besides the mentioned sliders for controlling the movement of the arm also contains the possibility to change the movement speed in the range of 0 - 100%, which can be adjusted also during automatic operation. There is no such option when controlling a robot by a pendant. Manual motion control is useful, for example, to adjust the position from which the image will be obtained, since the camera can be positioned on the gripping effector. You can also set the location where objects should be uploaded. The UI also displays the name of the currently used robot, its current location and the currently set target.

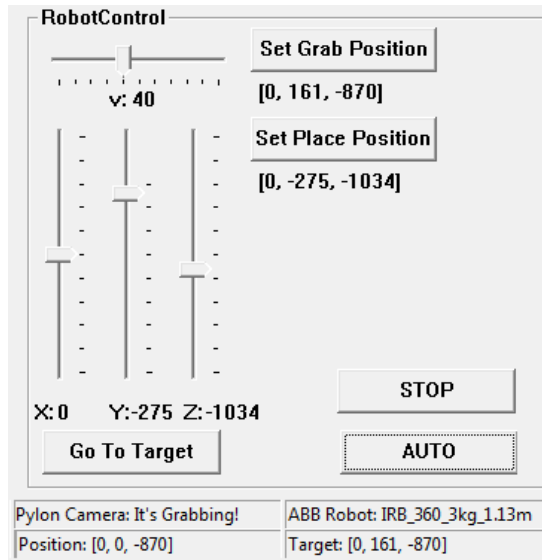


Fig. 7. User interface for robot controlling

A separate part of the application consists of a part that links a part of the code created in the C # programming language, which provides communication with the IRC5 robot control system and its actual control. This part of the program creates an essentially standalone control program that can be used to create another application to control ABB robots whose control system is an ABB IRC5.

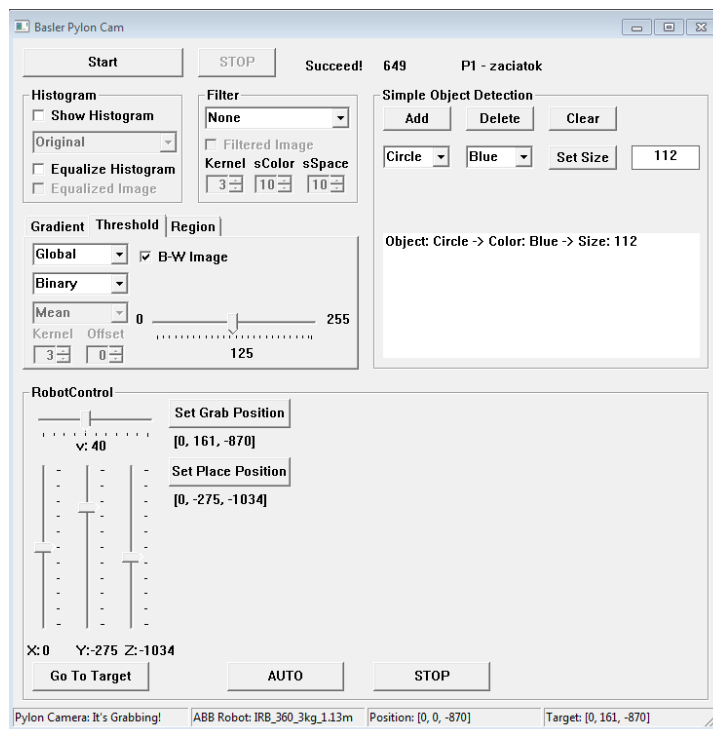


Fig. 8. The main application control and information window



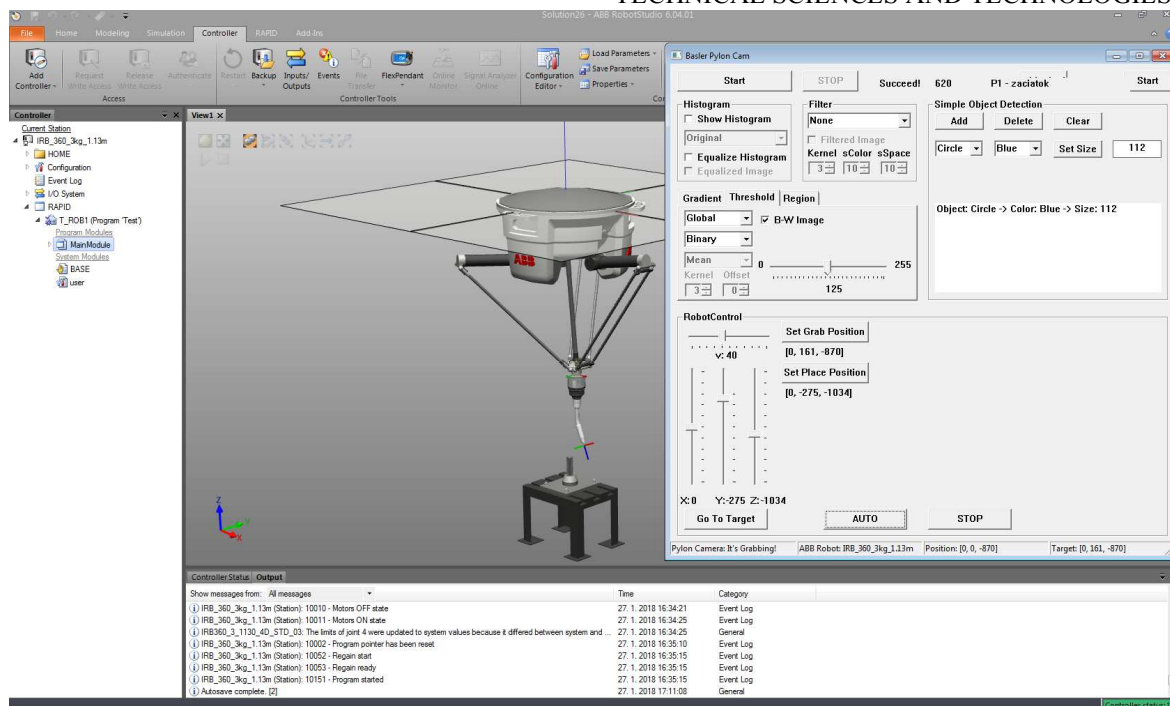


Fig. 9. Simulating robot control using ABB RobotStudio software

**Conclusions.** In order to design a robotic system whose operation is controlled on the basis of visual perception, it was necessary to acquire theoretical knowledge for the correct selection of individual components of the system as well as their correct placement within the robotic cell. Great emphasis was placed on suitable and economical selection of the sensing device and the way of illuminating the scanned objects.

In order to obtain the camera image it was also necessary to study and understand the principle of working with the image captured by the camera via SDK issued directly by the camera manufacturer. However, obtaining an image was only the first step to start the image processing process. In order to extract the necessary data from the obtained image and then to create control instructions from the data for controlling the robot, it was necessary to study and learn in detail the individual steps and procedures of image processing.

In the part of the work dealing with image processing the acquired knowledge was applied to the processing process itself, but not only known approaches were used. Owing to the reduction of CPU load and consequently shortening of the calculation process, own procedures were also introduced into the image processing process. The actual "economical" approach was applied and tested in the thresholding process where a "shortened thresholding algorithm" was created. The approach was also applied to the object-in-picture search process, creating a "network-based object-in-picture method" that uses the fact that we search and identify known objects in industrial applications as opposed to identifying objects in an unknown environment.

The combination of image acquisition, image processing and robot control with one comprehensive application is also a major benefit. Of course, to ensure this functionality, it was first necessary to create a theoretical base on which to build. The main problem was to create a control part of the robot control in C # and to link it to the basic control program created in C ++.

## References

1. Gonzalez, R.C. - Woods, R.E.: Digital image processing. Upper Sadle River: Prentice Hall, 2002. ISBN 02-01-18075-8.
2. Šonka, M. – Hlaváč, V.: Počítačové vidění. Praha: Grada, 1992. ISBN 80-85424-67-3.
3. Šonka, M. – Hlaváč, V – Boyle, R.: Image processing, analysys and machine vision. Toronto: Thomson, 2008.

## TECHNICAL SCIENCES AND TECHNOLOGIES

4. Svoboda, T. – Kybic, J. – Hlaváč, V: Image processing, analysys and machine vision. A matlab comparsion. Toronto: Thomson learning, 2007. ISBN 04-95295-95-7.
5. Russ, J.C.: Image processing handbook. New york: CRC press, 2002. ISBN 0-8493-1142-X.
6. Hartley, R. – Zisserman, A.: Multiple view geometry in computer vision. Cambridge: Cambridge university press, 2004. ISBN 05-21540-51-8.
7. Laganière, R.: OpenCV 2 computer vision aplication programming cookbook. Birmingham: Packt Publishing, 2011. ISBN 978-1-849513-24-1.
8. Szeliski, R.: Computer vision: algorithms and applications. London: Springer-Verlag, 2011. ISBN 978-1-84882-934-3.
9. Canny, J.: A compotational approach to edge detection. In: IEEE transaction on pattern analysis and machine inteligenca, Roč. 8, č. 6 (1986), s. 679-698. ISSN 0162-8828.
10. Marr, D. – Hildreth, E.: Theory of edge detection. In: Proceedings of the royal society of London, Roč. 116, č. 796 (1980), s. 187-217. ISSN 1364-5021.
11. Smith, S.M. – Brady, M.J.: A new approach to low level image processing. In: International journal of computer vision , Roč. 23, č. 1 (1997), s. 45-78. ISSN 1573-1405.
12. Ružický, E. – Ferko, A.: Počítačová grafika a spracovanie obrazu. Bratislava: Sapientia, 1995. ISBN 80-967180-2-9.
13. Mercimek, M., Gulez, Z., Mumcu, T.V.: Real objects recognition using moment invariants. In: Sadhana, Roč. 30, č. 6 (2005), s. 765-775. ISSN 0973-7677.
14. Žára, J.: Moderní počítačová grafika. Brno: Computer press, 2004. ISBN 80-251-0454-0.
15. Bay, H. – Ess, A. – Tuytelaars, T. – Van Goo, L.: Speedd-up robust features (SUFR). In: Computer vision and image understanding, Roč. 110, č. 3 (2007), s. 346-359. ISSN 1077-3142.

УДК 621.8

*Сукоп Марек, Петер Ференчик*

## РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО РОБОТИЗОВАНОГО МОДУЛЯ З КАМЕРНОЮ СИСТЕМОЮ

**Актуальність теми дослідження.** Більш складні роботизовані системи характеризуються певним рівнем інтелектуальної поведінки, де на основі входу система здатна адаптувати свою поведінку. Реалізація елементів, що підтримують інтелектуальну поведінку в робототехнічних системах, особливо тих, що базуються на зображенні пристроїв, стає загальною практикою. Причина проста – така система швидша та більш точніша.

**Постановка проблеми.** Однак створення машинного зору є складною проблемою, особливо якщо йдеться про додатки з нестандартними вимогами. Для кожного завдання систему зору необхідно адаптувати до умов та вимог об'єктів, що контролюються. Інші налаштування зображення та алгоритми повинні застосовуватися до статичних об'єктів, а не до рухомих об'єктів. Інформація про довільне зображення є достатньою для певного процесу виготовлення, а для інших потрібен третій вимір для видалення заданої частини з невпорядкованого об'єднання. Створення інтелектуального роботизованого модуля із камерною системою вимагає створення системи зору, яка відповідає специфічним вимогам. Тут відкритий простір, оскільки існує багато різних процедур та принципів, з якими потрібно узгоджуватися, але не всі є однаково ефективними та надійними.

**Аналіз останніх досліджень і публікацій.** Більшість методів обробки зображень можуть поєднуватися один з одним, або може бути розроблений новий, більш ефективний спосіб вирішення проблеми з використанням вже відомих підходів. Додавання до цього факту нестандартних вимог, що виникають на практиці, є беззаперечною причиною, чому доцільно займатися обробкою зображень для промислового використання.

**Виділення недосліджених частин загальної проблеми** є розроблення і створення роботизованого модуля, діяльність якого буде контролюватися на основі сприйняття зображення, отриманого з цифрової камери.

**Постановка завдання.** Основне завдання цієї статті полягає у розробці та створенні роботизованого модуля, діяльність якого буде контролюватися на основі сприйняття зображення, отриманого цифровою камерою. Отримане зображення буде оброблено узгодженими алгоритмами обробки зображень, що приведе до створення керівних дій для управління рухом маніпулятора.

**Виклад основного матеріалу.** У роботі розглядається конструкція роботизованого модуля, завдання якого – маніпулювати зразками предметів, розміщених на конвеєрі, за допомогою паралельного робота-маніпулятора на основі сприйняття зображення. Основна частина конструкції - це створення програмного забезпечення управління, яке на першому рівні забезпечує належне функціонування окремих компонентів, а на другому рівні їх взаємну співпрацю, що забезпечує виконання необхідної функціональності роботизованої системи загалом. Створене програмне забезпечення працює в операційній системі Windows 7, де пропонується простий інструмент для управління рухом паралельного робота без використання інших засобів управління. Це означає, що рухом робота можна керувати безпосередньо з програми управління, що дозволяє впливати на робота і об'єкт навіть в ручному режимі. Зображення, отримане камерою, може коригуватися програмним забезпеченням за допомогою реалізованих інструмен-



тів до початку автоматичної маніпуляції, що дозволяє користувачеві встановити правильні вхідні параметри для забезпечення надійної ідентифікації об'єкта.

**Висновки відповідно до статті.** Для розробки роботизованої системи, функціонування якої контролюється на основі візуального сприйняття, необхідно було придбати теоретичні знання для правильного підбору окремих компонентів системи, а також їх правильного розміщення всередині роботизованого модуля. Значну увагу приділено вибору підходящого та економного чутливого елемента пристрою та способу освітлення сканованих об'єктів.

Для отримання зображення з камери необхідно було також вивчити та зрозуміти принцип роботи із зображенням, отриманим камерою через SDK, виготовленим безпосередньо виробником камери. Однак отримання зображення було лише першим кроком для початку процесу обробки зображень. Для того, щоб витягти необхідні дані з отриманого зображення, а потім створити керуючі інструкції з даних для управління роботом, необхідно було детально вивчити окремі кроки та процедури обробки зображень.

У частині роботи, що стосується обробки зображень, отримані знання були застосовані до самого процесу обробки, але використовувалися не лише відомі підходи. Завдяки зменшенню завантаження процесора і, отже, скороченню процесу обчислення, в процес обробки зображень також були введені власні процедури. Фактичний «економний» підхід застосовувався та перевірявся в процесі порогового визначення, де був створений «скорочений алгоритм порогової оцінки». Підхід також застосовано для процесу пошуку об'єкта в зображенні, створюючи «мережевий метод об'єкта в зображенні», який використовує той факт, що шукають та ідентифікують відомі об'єкти в промислових додатках на відміну від ідентифікації об'єктів у невідомому середовищі.

Поєднання зображення, обробки зображень та управління роботом з одним комплексним використанням також є головною перевагою. Звичайно, щоб забезпечити цю функціональність, спочатку потрібно було створити теоретичну базу, на якій проводилася розробка. Основна проблема полягала в тому, щоб створити контрольну частину керування роботом в C# і пов'язати її з базовою програмою управління, створеною в C++.

**Ключові слова:** роботизований модуль; машинне бачення; камерна система; інтелектуальна система.

Fig.: 9. References: 15.

**Sukop Marek** – Associate professor, Ph.D. of Technical Sciences, Department of robotic, Mechanical faculty, Technical University of Kosice (Letna 9, 04200 Kosice, Slovak Republic).

**Сукоп Марек** – доцент, кандидат технічних наук, Технічний університет Кошице (Letna 9, 04200 Kosice, Slovak Republic).

**SCOPUS Author ID:** 36615762200

**E-mail:** marek.sukop@tuke.sk

**Ferenčík Peter** – PhD student in Technical Sciences, Department of robotic, Mechanical faculty, Technical University of Kosice (Letna 9, 04200 Kosice, Slovak Republic).

**Петер Ференчик** – аспірант, Технічний університет Кошице (Letna 9, 04200 Kosice, Slovak Republic).